Data Security Implementation using Data Encryption Standard Method for Student Values at the Faculty of Medicine, University of North Sumatra

Ali Ikhwan¹², Rafikha Aliana A. Raof¹, Phaklen Ehkan¹, Yasmin Yacob¹, M. Syaifuddin³
¹Embedded, Networks and Advanced Computing (ENAC) Research Cluster, School of Computer and Communication Engineering, 02600 Arau, Perlis, Malaysia
²Faculty of Science and Technology, Universitas Islam Negeri Sumatera Utara, Jl. Willem Iskandar Pasar V Medan Estate, 20371, Medan, Sumatera Utara, Indonesia
³High School of Informatics Management Triguna Dharma Medan

Corresponding author’s : ali_ikhwan@unisu.ac.id

Abstract. It is undeniable that advances in technology and the rapid spread of information are also accompanied by an increase of crime in the IT field, very valuable information is sought by criminals in the IT field in order to be misused so that they gain enormous profits. One of the information is student value data, so a system is needed that applies a cryptographic algorithm that can secure the information in this case the method used is securing student value data using the Data Encryption Standard (DES) algorithm. DES is asymmetrical cryptographic algorithm and is also classified as a cipher block with a 64-bit key size. DES converts plaintext into a ciphertext of the same size, 64 bits using a 56-bit internal key. By building a system to implement the DES algorithm in securing data values, it is hoped that it can help the Faculty of Medicine, University of North Sumatra in protecting the confidentiality of data values from irresponsible parties

1. Introduction
In today’s society, technology is almost applied in every activity and it plays an important role in society especially the Information Technology (IT) related technology. It helps society and determines how people interact with each other on a daily basis [1-2]. There is information or data involved in these activities to be computerized. As we may know that the exchange of information or data is indeed very beneficial and even crucial to several organizations, whether in the form of commercial organizations, companies, universities, government agencies, individuals or privates where storage and shipping are fast, easy and accurate [3-4]. Information or data security becomes a very important thing in data exchange, but there are also many threats to the data exchange process, especially documents and data that are assumed to be private or confidential. But in some cases, data information still has some weaknesses that can be detrimental to the organization or individual [5-8].

At the University of North Sumatra (USU) the problem of student academic data, especially student value data, often changes and misused of data that can harm the institutions of the University of North Sumatra and the students concerned. Therefore, the problem of data security is an important aspect of a system. For this reason, it is necessary to apply a method of data security with cryptographic techniques [9].
Cryptography is the art and science of maintaining data security which is used as a security method to secure data from the exchange of information [10]. In cryptography, there are several algorithms that can be used to encrypt and description data by using the correct key. Data Encryption Standard (DES) is a cipher block algorithm that is currently quite popular because it has been standardized as a key-symmetry encryption algorithm [11-12]. Using encryption techniques allows the original text to be unreadable until the description process is carried out by the recipient so that the data can be understood. Therefore, DES is used and is able to secure student grade data.

2. Methods

2.1 Cryptography

According to [10] [13] ‘Cryptography is the science and art of maintaining the confidentiality of a message by encoding it in a form that cannot be understood anymore’. Cryptography is connected with the process of converting regular plain text into incomprehensible text and vice-versa. It is a method of storing and conveying data in a particular form so that only those for whom it is intended can read and process it. The purpose of cryptography is to shield data transmitted in the likely occurrence of an adversary.

2.2 DES Algorithm

According to Nugroho in 2018 [14] "DES (Data Encryption Standard) is a popular block cipher algorithm because it is used as a standard for key-symmetry encryption algorithms, although today the standard has been replaced with a new algorithm, such as Advanced Encryption Standard (AES) because DES has been deemed unsafe again [15]".

DES has a global scheme. The flow can be seen in Figure 1.

- The plaintext block is mutated with an initial permutation (IP) matrix.
- The initial permutation results are then encoded 16 times (16 cycles). Each round uses a different or different internal key.
- The enciphering results are then mutated with an inverse initial permutation (IP\(^{-1}\)) matrix into a ciphertext block.

![Figure 1. Global scheme.](image)

i. **Initial Permutation (IP)**

Before conducting the first round of the plaintext block, an initial permutation (IP) is performed. The purpose of this IP is to randomize the plaintext so that the order of the bits changes. Randomization was carried out using the following matrix:
How to read the table or matrix in Table 1 is that the two top left entry entries (58 and 50) means "Move the 58th bit to position bit 1", "move the 50th bit to position bit 2".

Table 1. Initial Permutation (IP).

| 58 | 50 | 42 | 34 | 26 | 18 | 10 | 2 | 60 | 52 | 44 | 36 | 28 | 20 | 12 | 4 |
| 62 | 54 | 46 | 38 | 30 | 22 | 14 | 6 | 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 |
| 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 63 | 55 | 47 | 39 | 31 | 23 | 15 | 7 |

ii. Internal Key Generation

Sixteen (16) internal keys are needed because there are sixteen (16) rounds, namely K1, K2, ..., K16. These internal keys can be generated before the encryption process or together with the encryption process. The internal key is generated from the external key given by the user, the external key is 64 bits long or 8 characters long. Suppose, an external key composed of 64 bits is K. This external key becomes the input for permutation using the PC-1 compression permutation matrix as revealed in Table 2.

Table 2. PC-1 matrix.

| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 | 58 | 50 | 42 | 34 | 26 | 18 |
| 10 | 2 | 59 | 51 | 43 | 35 | 27 | 19 | 11 | 3 | 60 | 52 | 44 | 36 |
| 63 | 55 | 47 | 39 | 32 | 23 | 15 | 7 | 62 | 54 | 46 | 38 | 30 | 22 |
| 14 | 6 | 61 | 53 | 45 | 37 | 29 | 21 | 13 | 5 | 28 | 20 | 12 | 4 |

In this permutation, every 8 bits (parity bit) of 8 key bytes are ignored. The result of the permutation is 56 bits long, so it can be said that the length of the DES key is 56 bits. Next, these 56 bits are divided into 2 parts, left and right, each of which is 28 bits long, and each of which is stored in C0 and D0:

C0: contains bits from K at position
57,49,41,33,25,17,9,1,58,50,42,34,26,18, 10,2,59,51,43,35,27,19,11,3,60,52,44,36

D0: contains bits from K at position
63,55,47,39,31,23,15,7,62,54,46,38,30,22,14,6,61,53,45,37,29,21,13,5,28,20,12,4

Furthermore, the two halves are shifted to the left (left shift) along one or two bits depending on each spin. Shift operations are wrapping or round-shift. The number of shifts in each round is shown in Table 3 as follows.

Table 3. Number of bit shifts in each round.

| Rounds | Number of Bit Shifts |
|--------|----------------------|
| 1      | 1                    |
| 2      | 1                    |
| 3      | 2                    |
| 4      | 2                    |
| 5      | 2                    |
| 6      | 2                    |
| 7      | 2                    |
| 8      | 2                    |
Suppose \((C_i, D_i)\) states the merging of \(C_i\) and \(D_i\), \((C_i + 1, D + 1)\) is obtained by shifting \(C_i\) and \(D_i\) one or two bits. After the bit shift, \((C_i, D_i)\) undergoes compression permutation using the following PC-2 matrix:

### Table 4. PC-2 matrix.

| 14 | 17 | 11 | 24 | 1 | 5 | 3 | 28 | 15 | 6 | 21 | 10 |
|----|----|----|----|---|---|---|----|----|---|----|----|
| 23 | 19 | 12 | 4  | 26| 8 | 16| 7  | 27 | 20| 13 | 2  |
| 41 | 52 | 31 | 37 | 47| 55| 30| 40 | 51 | 45| 33 | 48 |
| 44 | 49 | 39 | 56 | 34| 53| 46| 42 | 50 | 36| 29 | 32 |

With this permutation, the internal key \(K_i\), derived from \((C_i, D_i)\) in which case \(K_i\) is a merger of \(C_i\) and \(D_i\) in the following positions:

- \(C_i\) contains bits from \(K\) at position:
  - 14, 17, 11, 24, 1, 5, 3, 28, 15, 6, 21, 10, 23, 19, 12, 4, 26, 8, 16, 7, 27, 20, 13, 2

- \(D_i\) contains bits from \(K\) at position:
  - 41, 52, 31, 37, 47, 55, 30, 40, 51, 45, 33, 48, 44, 49, 39, 56, 34, 53, 46, 42, 50, 36, 29, 32

So, each \(K_i\) internal key has a length of 48 bits. If the total number of shifts of the bits in Table 4 is added together, the total number is equal to 28, which is equal to the number of bits in \(C_i\) and \(D_i\). Therefore, after round 16 will be retrieved \(C_{16} = C_0\).

### iii. Encryption

Each plaintext block experiences 16 times (cycles) of enciphering (encryption). Each enciphering loop is a Feistel network which is mathematically stated as Equation (1).

\[
L_i = R_{i-1}
\]

\[
R_i = L_i - 1 \oplus f(R_{i-1}, K_i)
\]

where,

- \(L = \text{Literacy/Repeat};\)  \(R = \text{Round};\)  \(I = \text{Alternative/Variable};\)  \(K = \text{Key}\)

\(E\) is an expansion function that expands a 32-bit \(R_{i-1}\) block to a 48-bit block. The expansion function is realized with an expansion permutation matrix as shown in Table 5:
Furthermore, the result of the expansion, \( E(R_{i-1}) \), which is 48 bits long is XOR-ed with \( K_i \) whose length is 48 bits produces a vector \( A \) whose length is 48 bits (Equation (2)).

\[
E(R_{i-1}) \oplus K_i = A
\]  

(2)

‘A’ vector is grouped into 8 groups of 6 bits each and becomes input for the substitution process. The substitution process is carried out using eight S-boxes, S1 to S8. Each S-box accepts 6-bit input and produces 4-bit output. The first 6-bit group uses S1, the second 6-bit group uses S2, and so on. The substitution process output is vector B which is 48 bits long. Vector B becomes input for the permutation process. The purpose of permutation is to randomize the results of the S-box substitution process. The permutation is done using the permutation matrix \( P \) (P Box) as follows (Table 6).

### Table 6. Permutation matrix \( P \) (P-box)

|   | 16 | 7  | 20 | 21 | 29 | 12 | 28 | 17 |
|---|----|----|----|----|----|----|----|----|
| 1 | 15 | 23 | 26 | 5  | 8  | 31 | 10 |
| 2 | 8  | 24 | 14 | 32 | 27 | 3  | 9  |
| 19| 13 | 30 | 6  | 22 | 11 | 4  | 25 |

The bits \( P(B) \) is the output of the function \( f \). Finally, the \( P(B) \) bits are XOR-ed with \( L_{i-1} \) to get \( R_i R_{i} = L_{i-1} \oplus P(B) \). So, the output of round 1 is applied Equation (3).

\[
(L_{i}, R_i) = (R_{i-1}, L_{i-1} \oplus P(B))
\]  

(3)

### iv. Last Permutation (Inverse Initial Permutation)

The last permutation is done after 16 turns of the left and right blocks combination. The permutation process uses the inverse permutation matrix (inverse initial or \( IP^{-1} \)). Table 7 shows the values of the inverse initial permutation.

### Table 7. Inverse initial permutation or \( IP^{-1} \)

|   | 40 | 8  | 48 | 16 | 56 | 24 | 64 | 32 |
|---|----|----|----|----|----|----|----|----|
| 39| 7  | 47 | 15 | 55 | 23 | 63 | 31 |
| 38| 6  | 46 | 14 | 54 | 22 | 62 | 30 |
| 37| 5  | 45 | 13 | 53 | 21 | 61 | 29 |
| 36| 4  | 44 | 12 | 52 | 20 | 60 | 28 |
| 35| 3  | 43 | 11 | 51 | 19 | 59 | 27 |
| 34| 2  | 42 | 10 | 50 | 18 | 58 | 26 |
| 33| 1  | 41 | 9  | 49 | 17 | 57 | 25 |
The decryption process is the process of changing ciphertext into plaintext. DES uses the same algorithm for the encryption and decryption processes. If the internal key sequence encryption process used is K1, K2, ..., K16, but in the decryption process the key sequence used is the opposite i.e. K16, K15, ..., K1. For each round of 16, 15, ..., 1, the output for each round of deciphering is as Equation (4).

\[ L_i = R_{i-1} \]
\[ R_i = L_{i-1} \oplus f(R_{i-1}, K_i) \]

which in this case, \((R_{16}, L_{16})\) is the initial input block for deciphering. Blocks \((R_{16}, L_{16})\) are obtained by mutating the ciphertext with an IP permutation matrix.

2.3 Student Value Data
Value transcript data is considered very important to maintain confidentiality so that the data cannot be changed and recognized as belonging to an unauthorized person. The value data used is the data value of students at the Faculty of Medicine, University of North Sumatra (USU), Indonesia.

3. Results and Analysis

3.1 Encryption Process
The encryption process is changing the plaintext data into ciphertext. There are several steps in the encryption process as follows:

\textit{i. Change the plaintext and key to binary numbers}

Change Plaintext into binary form based on ASCII table (Table 8).

| HEXA | Binary |
|------|--------|
| 86   | 01010110 |
| 105  | 011101001 |
| 114  | 01110010 |
| 111  | 01110111 |
| 108  | 01101100 |
| 111  | 01110111 |
| 103  | 01110011 |
| 105  | 01110100 |

Change key into binary form based on ASCII table (Table 9).

| HEXA | Binary |
|------|--------|
| 50   | 01010000 |
| 53   | 01010011 |
| 33   | 00110011 |
| 46   | 01001100 |
| 4B   | 01001011 |
| 55   | 01010101 |
| 53   | 01010011 |
| 55   | 01010101 |
ii. Initial Permutation (IP) in the Plaintext

Perform Initial Permutation (IP) on the plaintext bit using the following IP:

Descriptions in the IP table:
- The sequence of bits in the 58th plaintext table, placed in position 1 in the IP table.
- The sequence of the bits in the plaintext table is the order of 50, placed in position 2 in the IP table.
- The sequence of bits in the plaintext table sequence 42, placed in position 2 in the IP table, and so on (Table 10).

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| L0 |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
| R0 |   |   |   |   |   |   |   |

Table 10. Initial permutation (IP) results.

iii. Performed PC-1 Compression Permutation

A key that has been converted into a binary number is then mutated using the PC-1 compression permutation table. In this step compression occurs 64 bits to 56 bits by making 1 bit (parity bit) on each key block.

Information on PC Table 1 bit:
- The sequence of bits in the 57th Key Table, placed in position 1 in Table PC-1.
- The sequence of bits in the Key Table sequence 49, placed in position 2 in Table PC-1 and so on (Table 11).

|   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |
|   |   |   |   |   |   |   |   |

Table 11. PC-1 Compression permutation results.

The output can be written as

Output: 0000000 0111111 1100000 0000000 0110010 1101000 0001110 0000101

Then the bit in the output is broken up into 2 parts namely C₀ and D₀. Therefore, the results are as follows:
iv. Left Shift Operation

Left Shift on \( C_0 \) and \( D_0 \) using the 16-bit shift table, as for the 16-round table as follows:

Information:
For round 1, shift 1 bit to the left,
For the second round, the shift is done 1 bit to the left, and so on until the 16th round.
Here are the results of the rotation of the Left Shift:
Round 1, shifted 1 bit to the left.
\[
\begin{align*}
C_1 & : 000000 111111 1000000 000000 0 \\
D_1 & : 110010 1101000 0001110 0000101 0
\end{align*}
\]
Round 2, shifted 1 bit to the left.
\[
\begin{align*}
C_2 & : 000000 0111111 1100000 0000000 00 \\
D_2 & : 100100 1101000 0001110 0000101 01
\end{align*}
\]
Round 3, shifted 2 bits to the left.
\[
\begin{align*}
C_3 & : 0000000 0111111 1100000 0000000 0000000 1100101 1010000 0011100 0001010 \\
D_3 & : 101000 001001 001001 000010 000011 100110 000001 100110
\end{align*}
\]
Round 8, shifted 2 bits to the left.
\[
\begin{align*}
C_8 & : 1100000 0000000 0000000 0111111 \\
D_8 & : 0001110 0000101 0110010 1101000
\end{align*}
\]
Round 16, shifted 1 bit to the left.
\[
\begin{align*}
C_{16} & : 0000000 0111111 1100000 0000000 \\
D_{16} & : 0110010 1101000 0001110 0000101
\end{align*}
\]
After shifting the bits, each spin is merged back into \( C_iD_i \) and entered into the Permutation Compression 2 (PC-2) table and 48-bit \( C_iD_i \) data compression occurs. So, it produces information:

The sequence of bits in the 14th \( C_iD_i \), placed in position 1 in table PC-2,
The sequence of bits in the 17th \( C_iD_i \), placed in position 2 in table PC-2,
The sequence of bits in the 11th \( C_iD_i \), placed in position 3 in table PC-2,

Here are the outputs:
\[
\begin{align*}
C_1D_1 & : 0000000 111111 1110000 0000000 1100101 1010000 0001100 0001010 \\
K_1 & : 101000 001001 001001 000010 000011 100110 000001 100110 \\
C_2D_2 & : 0000001 1111111 0000000 0000000 0000000 1001011 0100000 0110000 0001010 \\
K_2 & : 101000 000001 001001 010010 010100 000100 100110 100011
\end{align*}
\]
\[
\begin{align*}
C_{15}D_{15} & : 0000000 0011111 1110000 0000000 1011001 0110100 0000111 0000010 \\
K_{15} & : 111000 001001 001000 100010 001111 010001 010000 000011
\end{align*}
\]
v. Data Expansion
In this step, we will expand the 32-bit $R_{i-1}$ data to $R_i$ 48 bits 16 times with a rotation value of $1 \leq i \leq 16$ using the expansion table $(E)$. The results $E(R_{i-1})$ are then XORed with $K_i$ and produces an $A_i$ matrix vector.

Here are the results of the output:

**Iteration 1**

$E(R_{1-1})$: \[100000 \ 000000 \ 000000 \ 000001 \ 010100 \ 001001 \ 010010 \ 100010\]

$K_1$: \[101000 \ 001001 \ 001001 \ 000010 \ 000011 \ 100110 \ 000001 \ 100110\]

\[\begin{array}{c}
\hline
\text{XOR} \\
\hline
A_1: \ 001000 \ 001001 \ 001001 \ 000011 \ 010111 \ 101111 \ 010011 \ 000100
\end{array}\]

In iteration one (1) obtained $A_1$ from the results of XOR $E(R_{1-1})$ and $K_1$, after that the next process goes directly to the 6th step first, where $A_1$ will be entered into S-BOX and produce $PB_1$ then XOR right with $L_0$ and produce $R_1$ value. This $R_1$ value is used to continue the 2nd iteration.

Decription Process is carried out to find out the actual message content, ciphertext conversion will be done into binary form to get the ciphertext bit.

vi. Permutation of the Cipher Bit
Cipher in binary = \[01001111 \ 11000111 \ 01000101 \ 00011011 \ 10100111 \ 11010101 \ 11110110 \ 01101010\] or Cipher in HEXA = 4FC7451B87D5F66A. The bits in the IP (cipher) are then broken into 2 parts, $L_0$ and $R_0$, therefore, the results are as follows:

\[L_0 = 11100111 \ 01101000 \ 01110111 \ 00111111\]

\[R_0 = 01110010 \ 11010000 \ 10001001 \ 11011011\]

**Iteration – 16**

$P(B_{16})$ = \[00001100 \ 00001101 \ 01001001 \ 11010010\]

$L_{15}$ = \[11100111 \ 01101000 \ 01110111 \ 00111111\]

\[\begin{array}{c}
\hline
\text{XOR} \\
\hline
R_{16} = 11101011 \ 01100101 \ 00111110 \ 11101101
\end{array}\]

\[L_1 = 01110010 \ 11010000 \ 10001001 \ 11011011\]

\[\begin{array}{c}
\hline
\text{XOR} \\
\hline
R_1 = 11111111 \ 0010011 \ 10101010 \ 01101011
\end{array}\]

**Iteration -1**

$P(B_{1})$ = \[01011111 \ 00101000 \ 01000110 \ 11010100\]

$L_0 = 10100000 \ 00111011 \ 11101100 \ 10111111\]

\[\begin{array}{c}
\hline
\text{XOR} \\
\hline
R_1 = 11111111 \ 0010011 \ 10101010 \ 01101011
\end{array}\]

\[L_1 = 00000000 \ 00000000 \ 10100100 \ 10001001\]

vii. Permutation $R_1$ and $L_1$ with Table IP$^{−1}$
Then $R_1$ and $L_1$ are mutated with the inverse initial permutation (IP$^{−1}$) table to produce output:

Cipher in binary = \[01010011 \ 01010101 \ 01001000 \ 01000101 \ 01010010 \ 01001101 \ 01000001 \ 01001110\]
Or cipher in HEXA = 53 55 48 45 52 4D 41 4E
and in character form = Virology

A value data form is a form that is used to enter or save student value data into a database. In this form the data encryption process is also carried out and the process of decrypting value data if one day the data is needed, and all the data is stored in a database (refer to Figures 2 and 3).

**Figure 2.** Form data value after encryption.

**Figure 3.** Form data value after decryption.

### 4. Conclusion

As the conclusion, the standard data encryption algorithm is used to secure student grade data at the Faculty of Medicine, University of Northern Sumatera because of its difficulties to break the passwords and ciphertexts. This had undergone through several transformation processes so that it is very much appropriate to be used to encode or secure the data. The internal audit data security system design uses the data encryption standard (DES) algorithm with a key length of 56 bits and a web programming-
based system, and the steps to implement the 64-bit DES algorithm in securing student data value is to enter DES 64-bit calculations into the system so that it can provide a solution to secure student data values. Internal audit data security applications can run according to their functions and have been tested using the DES 64-bit algorithm successfully. With these implementations, all the student values at the Faculty of Medicine, University of North Sumatra will be safe and secure to be used.

References

[1] Todd Gebski, “How Does Technology Impact Your Daily Life?”, 2015, https://www.motus.com/how-does-technology-impact-your-daily-life/ (Published on 8th. October 2015)

[2] Mickeel Allen, “Technological Influence on Society”, 2019, https://www.bctv.org/2019/11/07/technological-influence-on-society/ (published on 7th. November 2019)

[3] Pranoto, G., Nyoto, R. D., Safriadi, N., “Rancangan Bangun Aplikasi Terpadu Sistem Pengadaan Barang dan Sistem Jurnal Perusahaan Berbasis Web”, Jurnal Sistem dan Teknologi Informasi, 2016, 4(1), pp. 1-5.

[4] Salahuddin, N. S., Iramadhan, N., Sari, S. P., Himawati, D., “Perancangan E-Kuliner Rumah Jagung”, Seminar Nasional Sistem Informasi, 2018, pp.1064-1072.

[5] Josi, A., “Implementasi Framwork Bootstrap Pada Website STMIK Prabumulih”, Jurnal Manajemen dan Informatika Komputer Pelita Nusantara (MANTIK), 2016, 20(1), pp. 1-4.

[6] Lestanti, S., Susana, A. D., “Sistem Penggarsipan Dokumen Guru dan Pegawai Menggunakan Metode Mixture Modelling Berbasis Web”, Jurnal Antivirus, 2016, 10(2), pp 69-77.

[7] Muslihuddin, M., Larasati, A., “Perancangan Sistem Aplikasi Penerimaan Mahasiswa Baru di STMIK Pringsewu Menggunakan PHP dan MySQL”, Jurnal TAM (Technology Acceptance Model), 2014, 3, pp. 32-39.

[8] Santoso, H., “Problem of Letter Architecture Completion using a Search Method”, Zero - Jurnal Sains, Matematika dan Terapan, 2018, 4(2), pp. 1-13.

[9] Ridho, M. F., “Perancangan Aplikasi Keamanan Data Dengan Algoritma Serpent”, Jurnal Pelita Informatika, 2017, 6(1), pp. 116-120.

[10] Fithri, D. L., “Analisa Kriptografi Untuk Keamanan Dalam Manajemen Sistem Terdistribusi Perusahaan”, Jurnal Disprotek, 2016, 7(1), pp.1-7.

[11] Sulaiman, O. K., Ilhwan, M., Rizki, S. F., “Model Keamanan Informasi Berbasis Tanda Tangan Digital dengan Data Encryption Standard (DES) Algorithm”, Info’TekJar (Jurnal Nasional Informatika dan Teknologi Jaringan), 2016, 1(1), pp.14-19.

[12] Priatmoko, A, Harahap, E., “Implementasi Algoritma DES Menggunakan MATLAB”, Jurnal Teori dan Terapan Matematika, 2017, 16(1), pp. 11-19.

[13] Harahap, M. K., “Analisis Perbandingan Algoritma Kriptografi Klasik Vigenere Cipher dan One Time Pad”, Jurnal Nasional Informatika dan Teknologi Jaringan, 2016, 1(1), pp. 61–64.

[14] Nugroho, A. R., Pramusinto, W., “Implementasi Kriptografi Dengan Algoritma Caesar Cipher, AES 192 dan DES untuk Aplikasi Pesan Instan Berbasis Android”, Jurnal Sistem Komputer dan Teknik Informatika (SKANIKA), 2018, 1(1), pp 14-19.

[15] Pabokory, F. N., Astuti, I. F., Kridalaksana, A. H., “Implementasi Kriptografi Pengamanan Data Pada Pesan Teks, Isi File Dokumen, dan File Dokumen Menggunakan Algoritma Advanced Encryption Standard,” Jurnal Informatika Mulawarman, 2016, 10(1), pp 20-31.