International Journal of Information System & Technology
Akreditasi No. 158/E/KPT/2021 | Vol. 6, No. 5, (2023), pp. 613-617

Text Data Security Using Stream Cipher Algorithm

Supiyanto[1], Agung Dwi Saputro[2] dan Muhammad Asghar[3]
[1,2,3]Information System Study Program, Universitas Cenderawasih, Indonesia
Email: supi6976@gmail.com[1], dwisaputro321@gmail.com[2],
asghar07111993@gmail.com[3]

Abstract

The Stream Cipher cryptographic algorithm is a cryptographic algorithm that can be used to secure symmetric key-type data. The Stream Cipher algorithm uses the same key when performing encryption and decryption. In carrying out the encryption process, the Stream Cipher algorithm uses the stream cipher method, where the cipher comes from the result of an XOR operation between the plaintext bit and the key bit. This study aims to implement the Stream Cipher algorithm with a Matlab programming language for text security. The text data used in this study is in the form of a file with a txt extension. The stages of research methods included literature studies, algorithm analysis, program design, and testing. The method used in this study is to apply the Stream Cipher algorithm to a programming language to produce an application that can be used for data security. The output of this research is the creation of an application program that can be used to secure text data so that the information contained is not easily stolen or misused by irresponsible people. The test results in the study using a key that was modulo-kan 65 first will produce a random message in the form of clear characters while using a key that is not modulated will produce a random message that is less clear in the form of boxes.

Keywords: *Stream Cipher, cryptographic algorithm, text security, symmetric key*

1. Introduction

Along with the development of the times and the rapid advancement of technology, more and more computers are connected in cyberspace, commonly known as the internet. Companies, governments, financial institutions, and many other institutions are also involved in this virtual world. Many things can be done through the internet to facilitate relations between institutions, including delivering data from one party to another and transmitting data that the public can access. An essential thing to pay attention to in the delivery of data or sending data through public networks is the security of the data. It becomes essential to maintain the confidentiality of data, especially if the data is confidential and can only be known to the appropriate party [1].

One of the ways used for data security is to use a cryptographic system, namely by encoding the contents of the information (*plaintext*) into content that is not understood through the encryption process and to retrieve the original information; the decryption process is carried out, accompanied by using the correct key. One of the encoding algorithms in cryptography is the Stream Cipher. According to the type of key used, Stream Cipher is a symmetric key algorithm, which is an algorithm that uses the same key in the encryption and decryption process [2]. Based on its occurrence, Stream Ciphers are classified into modern cryptography, operating in the form of a single bit which is processed in bits in terms of keys, particles, and ciphertexts [3]. Thus, this algorithm is more complex and more challenging to solve when compared to classical key cryptography.

2. Research Methodology

This research is applied research, namely research that aims to apply a Stream Cipher algorithm to an applied field, namely data security, using the Stream Cipher algorithm. This research begins with a literature study, namely collecting reference materials from

ISSN : 2580-7250

books, articles, papers, journals, papers, and internet sites regarding the Stream Cipher algorithm, the underlying mathematical concepts, and several other references to support the achievement of research objectives.

In the Stream Cipher Algorithm, bits only have two values, so the encryption process only causes two states in the bits, namely changing or not changing. These two states are determined by an encryption key called keystream flow. In simple terms, the process of encrypting and decrypting the Stream Cipher algorithm can be seen in the image below: [4].

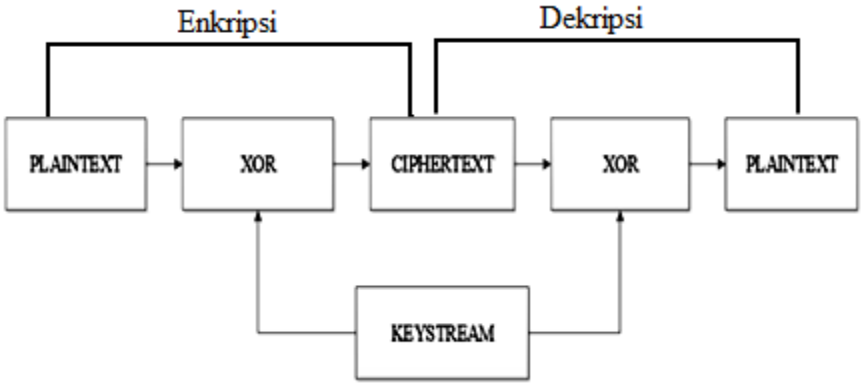

Figure 1. The Process of Encrypting and Decrypting Stream Cipher Cryptographic Algorithms

2.1. Cipher Stream Encryption Process

The Stream Cipher encryption algorithm in pseudocode form below [5]:

Step 1: Input plaintext
Step 2: Key input
Step 3: Convert each plaintext character into ASCII Code

```
I = 0
Jum = LEN(Plaintext)
For i = 1 to jum
P(i) = Asc(substr(Plaintext, 1,i))
Next i
```

Step 4: Convert each character in the key into ASCII *Code* form

```
I = 0
Jum = LEN(Kunci)
For i = 1 to jum
K(i) = Asc(substr(Kunci, 1,i))
Next i
```

Step 5: Encrypt with formulas

```
I = 0
Jum = LEN(Plaintext)
For i = 1 to jum
C(i) = P(i) XOR (K(i)
Next i
```

Step 6: Convert ASCII *ciphertext* into character form

```
I = 0
Jum = LEN(Ciphertext)
For i = 1 to jum
Karakter_C(i) = chr(substr(C(i), 1,i))
Next i
```

2.2. Cipher Stream Decryption Process

The Stream Cipher decryption algorithm in pseudocode form is below:

Step 1: Input ciphertext
Step 2: Key input
Step 3: Convert each ciphertext frog into ASCII Code

```
I = 0
Jum = LEN(Ciphertext)
For i = 1 to jum
C(i) = Asc(substr(Ciphertext, 1,i))
Next i
```

Step 4: Convert each character in the key into a shape ASCII Code

```
I = 0
Jum = LEN(Kunci)
For i = 1 to jum
K(i) = Asc(substr(Kunci, 1,i))
Next i
```

Step 5: Encrypt with formulas

```
I = 0
Jum = LEN(Ciphertext)
For i = 1 to jum
P(i) = C(i) XOR (K(i)
Next i
```

Step 6: Convert ASCII plaintext into character form

```
I = 0
Jum = LEN(Plaintext)
For i = 1 to jum
Karakter_P(i) = chr(substr(Plaintext, 1,i))
Next i
```

3. Results And Discussion

The process that will be carried out for testing in this application is to simulate sending messages in the form of files in .txt format between sender and recipient with a single key / predetermined key so that in the end, the original message sent by the sender must be ama with the message received by the recipient.

Table 1. System test results

No	Isi Pesan	Kunci	Enkripsi	Dekripsi	Hasil
1.	K	5	N	K	Berhasil
	1001011	0000101	1001110	1001011	
2.	MAKAN	25	TXRXW	MAKAN	Berhasil
	1001101 1000001 1001011 1000001 1001110	0011001 0011001 0011001 0011001 0011001	1010100 1011000 1010010 1011000 1010111	10011011000001 1001011 1000001 1001110	
3.	Belajar Pemograman	supi123	PQCI{sa2DJE~ uaSYNF	BELAJAR PEMOGRAMAN	Berhasil
	1000010 1000101 1001100 1000001 1001010 1000001 1010010 0100000 1010000 1000101 1001101 1001111 1000111 1010010 1000001 1001101 1000001 1001110	0010010 0010100 0001111 0001000 0110001 0110010 0110011 0010010 0010100 0001111 0001000 0110001 0110010 0110011 0010010 0010100 0001111 0001000	1010000 1010001 1000011 1001001 1111011 1110011 1100001 0110010 1000100 1001010 1000101 1111110 1110101 1100001 1010011 1011001 1001110 1000110	1000010 1000101 1001100 1000001 1001010 1000001 1010010 0100000 1010000 1000101 1001101 1001111 1000111 1010010 1000001 1001101 1000001 1001110	

4.	Ternyata belajar pemograman memang susah !!	M4kanNas!	XqXNTLTSNqFAGLR2qIyEG_LMSo,yOMLCG2rYgKH-,!	Ternyata belajar pemograman memang susah !!	Berhasil
	1010100 1000101 1010010 1001110 1011001 1000001 1010100 1000001 0100000 1000010 1000101 1001100 1000001 1001010 1000001 1010010 0100000 1010000 1000101 1001101 1001111 1000111 1010010 1000001 1001101 1000001 1001110 0100000 1001101 1000101 1001101 1000001 1001110 1000111 0100000 1010011 1010101 1010011 1000001 1001000 0100000 0100001 0100001	0001100 0110100 0001010 0000000 0001101 0001101 0000000 0010010 0100001 0001100 0110100 0001010 0000000 0001101 0001101 0000000 0010010 0100001 0001100 0110100 0001010 0000000 0001101 0001101 0000000 0010010 0100001 0001100 0110100 0001010 0000000 0001101 0001101 0000000 0010010 0100001 0001100 0110100 0001010 0000000 0001101 0001101 0000000	1011000 1110001 1011000 1001110 1010100 1001100 1010100 1010011 0000001 1001110 1110001 1000110 1000001 1000111 1001100 1010010 0110010 1110001 1001001 1111001 1000101 1000111 1011111 1001100 1001101 1010011 1101111 0101100 1111001 1001111 1001101 1001100 1000011 1000111 0110010 1110010 1011001 1100111 1001011 1001000 0101101 0101100 0100001	1010100 1000101 1010010 1001110 1011001 1000001 1010100 1000001 0100000 1000010 1000101 1001100 1000001 1001010 1000001 1010010 0100000 1010000 1000101 1001101 1001111 1000111 1010010 1000001 1001101 1000001 1001110 0100000 1001101 1000101 1001101 1000001 1001110 1000111 0100000 1010011 1010101 1010011 1000001 1001000 0100000 0100001 0100001	
5.	Belajar Pemograman	supi123	{sas~ua{sas ~ua	BELAJAR PEMOGRAMAN	Berhasil
	1000010 1000101 1001100 1000001 1001010 1000001 1010010 0100000 1010000 1000101 1001101 1001111 1000111 1010010 1000001 1001101 1000001 1001110	1010011 1010101 1010000 1001001 0110001 0110010 0110011 1010011 1010101 1010000 1001001 0110001 0110010 0110011 1010011 1010101 1010000 1001001	0010001 0010000 0011100 0001000 1111011 1110011 1100001 1110011 0000101 0010101 0000100 1111110 1110101 1100001 0010010 0011000 0010001 0000111	1000010 1000101 1001100 1000001 1001010 1100001 1110010 0000000 1010000 1000101 1001101 1001111 1100111 1110010 1000001 1001101 1000001 1001110	

4. Conclusion

Based on the results of the discussion on testing the Stream Cipher algorithm in securing text data, The application is designed for the simulation of encrypting and decrypting messages with *.txt extension using the Stream Cipher algorithm, runs very well, and is used for the data security process. Testing using a key modulated 65 first will produce a random message in clear characters as test result No. 3 in the test data. Meanwhile, using unmodulated keys will result in a random message that needs clarification, as per test data No.5. The use of keys is difficult to guess because it uses text to binary; the possibility of leaking keys while exchanging single key information can be avoided.

References

[1] Supiyanto and T. Suparwati, “Penerapan Matriks Invers Tergeneralisasi (MIT) Untuk Keamanan Data Pada Sandi Hill,” Indonesia, 2020.

[2] N. Rajender Reddy, C. Aravind Kumar, P. Rajkumar, and V. Velde, “Public key authentication schemes in asymmetric cryptography,” *Mater. Today Proc.*, no. xxxx, pp. 1–5, 2021, doi: 10.1016/j.matpr.2021.02.172.

[3] R. Munir, "Kriptografi," *Inform. Bandung*, 2006.
[4] M. Jumeidi, D. Triyanto, and Y. Brianorman, "Implementasi Algoritma Kriptografi Vernam Cipher Berbasis Fpga," *Coding J. Komput. dan Apl.*, vol. 4, no. 1, 2016, doi: http://dx.doi.org/10.26418/coding.v4i1.13329.
[5] Y. Irnanda, "Enkripsi dan Dekripsi Dengan Menggunakan Metode Kriptografi Vernam Cipher (XOR)," *Kumpul. Karya Ilm. Mhs. Fak. sains dan Tekhnologi*, vol. 1, no. 1, p. 47, 2019.