Analysis of Combination Algorithm Data Encryption Standard (DES) and Blum-Blum-Shub (BBS)

O Laia¹, E M Zamzami² and Sutarman³

¹Student in Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia
²Lecturer in Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Medan, Indonesia
³Lecturer in Faculty of Math and Science, Universitas Sumatera Utara, Medan, Indonesia

Abstract. Data Encryption Standard (DES) has a weakness in the key that is vulnerable to security threats, but it is quite popular to use because of the fast encryption and decryption process. Combining the DES algorithm with the Pseudo-random number generator Blum-Blum-Shub (BBS) in generating external keys in the encryption and decryption process of messages, produces a unique key and a good level of security. The longer the selected key and seed, the higher the security level of the DES encryption and decryption key. The use of BBS as an external key generator in the DES algorithm does not significantly affect the encryption (key and seed generated is around 0.001 – 0.003 seconds for the 2 – 4 digits experiment) processing time, meanwhile, the time in the decryption process has no effect because the BBS key and seed are no longer subject to feasibility testing or number requirements. The combination of DES and BBS algorithms can increase security in terms of encryption and decryption keys because the random number of DES keys generated by the BBS algorithm is unique and does not burden the user in determining the keys used in the DES algorithm because the keys are generated from the programming language.

Keywords: Data Encryption Standard (DES), Blum-Blum-Shub (BBS).

1. Introduction

In the era of increasingly sophisticated globalization of information technology, security is a top priority to ensure that every data exchange process between senders and recipients can be maintained as a whole as the information contained therein [1]. One of the techniques that can be used in information security is encryption [2]. Cryptography has an important role in securing, controlling and identifying true digital data [3, 12].

Cryptography is the study of methods for sending messages secretly in encrypted or disguised form, so that only the intended recipient can remove the disguise and read the message [4]. Cryptography has an important role in securing, controlling and identifying true digital data [3].

There are two types of cryptography based on the keys used, namely the symmetric key and the asymmetric key. Symmetric cryptography, the key used for encryption is the same as the key used in decryption, so key distribution must be done before information transmission [5]. Asymmetric
cryptography is an encryption scheme that uses two keys that are mathematically related, but not identical, namely the public key and the private key [6].

One of the most popular symmetric key algorithms used is the Data Encryption Standard (DES) algorithm in cryptographic encryption blocks. On the other hand, DES is considered insecure because it uses a short key (64 bits), in addition, Brute Force attacks have shown that DES can be practically attacked [7].

Previous research on improving DES key security using a random array of size eight, there are weaknesses where the random numbers in the array are generated through a programming language without using the generator method and the number distance is too short (0-26). Numbers in randomized arrays are most likely to be guessed and encoded messages can be attacked easily [8]. Next research on the combination of the DES algorithm and the LUC algorithm was successfully carried out, but there are weaknesses in the generation of the LUC algorithm keys, namely the determination of prime numbers \( p \) and \( q \) is limited to 1 to 100, so that the generated random numbers are not too unique and the generated random number factorization process is likely to be experiencing repetition in the previous series of random numbers [9].

Random numbers are the most important method to be applied in cryptography, where both are the most important parts of the encryption key [10], [11]. The Blum-Blum-Shub (BBS) algorithm is a random number generator method that functions to generate random numbers in a mathematical process with the resulting output is a series of binary numbers.

Based on the description above, DES has weaknesses in key parts that are vulnerable to security threats. But on the other hand, DES is a cryptographic algorithm that is quite popular because of its fast encryption and decryption process. Meanwhile, the BBS algorithm has advantages related to factorization problems. The BBS algorithm is strong and sensitive to small changes in the key, even with knowledge of the approximate value of the key, there is no possibility for an attacker to crack the password [1].

To prevent attacks on the Standard Data Encryption (DES) cryptographic algorithms, it is necessary to increase the security of the keys. In previous research on the development of the DES algorithm key, there are still weaknesses that allow cryptanalysis to find weaknesses to obtain and view the original message in the encoded message. The author combines the DES algorithm with the BBS pseudo random number generator, with the hope of increasing the security of encrypted messages (ciphertext) by increasing the security of encryption and decryption keys using the BBS algorithm.

The difference between this paper and research that has been studied before is that the BBS random number generator algorithm is used as an external DES key generator and uses a larger length number (1-4 digits) as a random number generator key on the BBS so that the random number is not easy to predict.

2. Research methods

This study aims to improve the security of the encryption key and message decryption in the Data Encryption Standard (DES) algorithm. The security enhancement is done by combining the Blum-Blum-Shub (BBS) random number generator algorithm as an external key generator in the DES algorithm. The use of the BBS algorithm as a key generator is based on its reliability, which is able to generate random numbers that are not easy to predict and its sensitivity to small changes in keys.

This study uses text data in the form of a file with .docx extension. The text data in question is in the form of letters, numbers and symbols according to the ASCII 256 table.

The process of encrypting the DES algorithm message with the external key generation of the BBS algorithm is almost the same as the original DES encryption process, what distinguishes it is that the DES external key is not generated manually but is generated using the BBS algorithm so that a series of random numbers is obtained which will be used as an external key in the encryption process and DES algorithm decryption.

The decryption process using the DES algorithm is almost the same as the previous DES encryption flow. For DES decryption, we still use the random number generated by the BBS algorithm as the
external key, but in decryption the initial input block is \( R_{16} = L_{16} \). Flowchart encryption and description of DES algorithm with BBS algorithm external key generator as shown in Figure 1.

As a result of the design that has been made, then implemented in an application built using the Java programming language with the NetBeans IDE application.

![Flowchart](image.png)

**Figure 1.** DES encryption and decryption flowchart with BBS external key generator.

DES and BBS algorithms encryption and decryption steps:

- a. Input data (plaintext or ciphertext), then convert it to binary numbers.
- b. Generate BBS keys \( p, q \) provided they are congruent to the value of \( 3 \mod 4 \).
- c. Do the multiplication process between \( p \) and \( q \) and store it in \( n \), where the number of the product is an integer called Blum \( n \).
- d. Choose a random number \( s \) as bait, with the criteria \( 2 \leq s < n \) and \( s, n \) is relatively prime.
- e. Calculate the value of \( x_0 = s_2 \mod n \).
- f. Generate a 64-bit random number, by: calculate \( x_i = x \times (i-1) \times 2 \mod n \) and produce \( z_i = \) bits taken from \( x_i \). The 64-bit random becomes the external DES key.
- g. Generate DES internal key then perform Initial Permutation (IP) of DES internal key (for encryption) or Initial Permutation (IP) of ciphertext bit (for decryption).
- h. Do expansion against \( R_0 \) for encryption or expansion against \( L_0 \) for decryption.
- i. Permutation P-box and Final Permutation (IP-1), resulting in plaintext or ciphertext.

### 3. Result and discussion

At this stage, the testing process is carried out on the system design that has been made in this study. This test aims to determine the effect of processing time based on the number of characters that are encrypted or decrypted. The following is an image of the encryption and decryption test results.
The following are the results of the encryption and decryption testing using a combination of the DES and BBS algorithms. The test was carried out on four files with different characters and the test was grouped into 3 parts of the BBS key (2-digit, 3-digit and 4-digit key).

Table 1. Encryption Process Experiment Results.

| Number of Characters | Key Length | BBS Keys | Processing Time (Seconds) |
|----------------------|------------|----------|--------------------------|
|                      |            | p | q | s | Encryption | Decryption |
| 1st Trial            |            |   |   |   |            |            |
| 20000                | 2-Digit    | 59| 71| 13| 0.047      | 0.046      |
| 40000                | 2-Digit    | 11| 83| 47| 0.093      | 0.109      |
| 60000                | 2-Digit    | 79| 71| 43| 0.156      | 0.203      |
| 80000                | 2-Digit    | 79| 71| 97| 0.234      | 0.266      |
| 100000               | 2-Digit    | 11| 83| 23| 0.297      | 0.359      |
| 2nd Trial            | 3-Digit    | 239| 947| 797| 0.110      | 0.094      |
| 40000                | 3-Digit    | 163| 163| 809| 0.170      | 0.140      |
| 60000                | 3-Digit    | 167| 599| 197| 0.187      | 0.172      |
| 80000                | 3-Digit    | 487| 811| 257| 0.234      | 0.250      |
| 100000 | 103 | 887 | 727 | 0.297 | 0.328 |
|--------|-----|-----|-----|-------|-------|
| 20000  | 4-Digit | 1667 | 3391 | 8167 | 0.125 | 0.063 |
| 40000  | 8867 | 1063 | 1901 | 0.171 | 0.109 |
| 60000  | 1511 | 3571 | 1201 | 0.219 | 0.187 |
| 80000  | 4967 | 4787 | 5701 | 0.250 | 0.234 |
| 100000 | 3347 | 8287 | 2861 | 0.313 | 0.312 |

Based on the data in Table 1, the processing time required is influenced by 2 indicators, namely the number of characters that are encrypted or decrypted and the length of the BBS key that is used as an external key generator in the DES algorithm as illustrated in the following graph.

**Figure 4.** Encryption Process Experiment Graph.
4. Conclusions

According to the results and testing the combination of Standard Data Encryption algorithm (DES) and Blum-Blum-Shub (BBS) was successfully executed for message encryption and decryption. The use of BBS as an external key generator in the DES algorithm does not really affect the encryption and decryption processing time. The effect of time on the encryption process on the length of the BBS key and seed generated is around 0.001 – 0.003 seconds for the 2 – 4 digits experiment. Meanwhile, the time in the decryption process has no effect because the BBS key and seed are no longer subject to feasibility testing or number requirements.

For next research, it can be increased by increasing the length of the key digit and BBS seed to further increase security and adapt to a programming language that can accommodate and run programs at large scale keys and data.

References

[1] B. Assa, M. and Khaled, and G. Lakhdar, “Implementation of blum blum shub generator for message encryption,” Int. Conf. Control. Eng. Inf. Technol., pp. 118–123, 2014.
[2] P. Mahajan and A. Sachdeva, “A Study of Encryption Algorithms AES, DES and RSA for Security,” undefined, 2013.
[3] S. Bhattacharya, “CRYPTOLOGY AND INFORMATION SECURITY-PAST, PRESENT, AND FUTURE ROLE IN SOCIETY,” Int. J. Cryptogr. Inf. Secur., vol. 9, no. 1, 2019, doi: 10.5121/ijcis.2019.9202.
[4] R. A. Mollin, “An Introduction to Cryptography, Second Edition,” 2007.
[5] M. Agrawal and P. Mishra, “A Comparative Survey on Symmetric Key Encryption Techniques.”
[6] M. E. Hellman, “AN OVERVIEW OF PUBLIC KEY CRYPTOGRAPHY.”
[7] N. M. M. Alhag and Y. A. Mohamed, “An Enhancement of Data Encryption Standards Algorithm (DES),” Oct. 2018, doi: 10.1109/CCCEEE.2018.8515843.
[8] D. Bansal, “Improved Key Generation Algorithm In Data Encryption Standard (DES ),” 2016.
[9] F. Ruziq, P. Sihombing, and C. Author, “Combination Analysis of Data Encryption Standard (DES) Algorithm and LUC Algorithm on File Security,” Int. J. Res. Rev., vol. 7, no. 2, p. 140, 2020.
[10] Priyanka, I. Hussain, and A. Khalique, “Random Number Generators and their Applications: A Review A Novel Trust Establishment Model in SIoT Network based on Sociological Aspects of Users in Social Networking Services View project Elliptic Curve Cryptography Using
Smart Cards View project Random,” vol. 7, Accessed: Oct. 30, 2020. [Online]. Available: https://www.researchgate.net/publication/335920952.

[11] O. Laia, E. Muisa Zamzami, F. Gratianus Nafiri Larosa, and A. Gea, “Application of Linear Congruent Generator in Affine Cipher Algorithm to Produce Dynamic Encryption,” p. 12001, 2019, doi: 10.1088/1742-6596/1361/1/012001.

[12] K. Agung, Fatmawati, H. Suprajitno, (2018, April). Image encryption based on pixel bit modification. In Journal of Physics: Conference Series (Vol. 1008, No. 1, p. 012016). IOP Publishing.