Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id

E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei

OPEN ACCESS

https://doi.org/10.63447/jimik.v6i2.1385

Improving DES Robustness for Text Encryption via Blum-Blum-Shub Key Generation

Omega Joel Patria Moata [1*], Andicha Vebiyatama [2], Muhamad Indra [3]

[1*,2,3] Computer Science Study Program, Faculty of Information Technology, Universitas Nusa Mandiri, Central Jakarta City, Special Capital Region of Jakarta, Indonesia.

Corresponding Email: 14230016@nusamandiri.ac.id [1*]

Histori Artikel:
Dikirim 20 Februari 2025; *Diterima dalam bentuk revisi* 20 Maret 2025; *Diterima* 20 April 2025; *Diterbitkan* 10 Mei 2025.

Abstrak

Kerentanan keamanan Data Encryption Standard (DES) memerlukan penyempurnaan inovatif untuk memastikan perlindungan data terhadap ancaman modern. Studi ini mengintegrasikan generator angka pseudorandom Blum-Blum-Shub (BBS) dengan DES untuk meningkatkan ketahanan enkripsi. Dengan memanfaatkan keacakan BBS, metode yang diusulkan memperkuat struktur cipher blok 16 putaran DES, meningkatkan keamanan sekaligus mempertahankan efisiensi komputasi. Aplikasi desktop berbasis Java dikembangkan untuk mengimplementasikan pendekatan ini, memastikan keamanan data tanpa ketergantungan pada cloud. Pengujian kelayakan mengonfirmasi enkripsi yang berhasil dengan waktu pemrosesan rata-rata 2 detik, yang menunjukkan peningkatan keamanan dan penerapan praktis. Analisis komparatif mengungkapkan bahwa integrasi BBS meningkatkan ketidakpastian, membuat dekripsi lebih menantang. Temuan tersebut menunjukkan bahwa menggabungkan DES dengan BBS memberikan solusi yang layak untuk keterbatasan DES. Penelitian di masa mendatang dapat mengeksplorasi skalabilitas, adaptasi terhadap berbagai jenis data, dan pengoptimalan lebih lanjut untuk meningkatkan strategi enkripsi terhadap tantangan keamanan siber yang terus berkembang.

Kata Kunci: Urutan Pseudorandom; Generator Blum Blum Shub; Generator Standar Enkripsi Data; Kompleksitas Komputasi.

Abstract

The security vulnerabilities of the Data Encryption Standard (DES) require innovative enhancements to ensure data protection against modern threats. This study integrates the Blum-Blum-Shub (BBS) pseudorandom number generator with DES to enhance encryption robustness. By leveraging BBS's randomness, the proposed method strengthens the 16-round block cipher structure of DES, improving security while maintaining computational efficiency. A Java-based desktop application is developed to implement this approach, ensuring data security without cloud reliance. Feasibility testing confirms successful encryption with an average processing time of 2 seconds, demonstrating both security improvements and practical applicability. Comparative analysis reveals that BBS integration enhances unpredictability, making decryption more challenging. The findings suggest that combining DES with BBS provides a viable solution to DES's limitations. Future research may explore scalability, adaptation to diverse data types, and further optimizations to enhance encryption strategies against evolving cybersecurity challenges.

Keyword: Pseudorandom Sequence; Blum Blum Shub Generators; Data Encryption Standard Generators; Computational Complexity.

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

1. Introduction

Cryptography, derived from the Greek words "crypto" (hidden) and "graphy" (writing), is a crucial technique used to secure data and communication. Its primary objective is to ensure that only authorized parties can access and interpret the transmitted information. This technique employs mathematical principles and rule-based algorithms to enhance message security, making decryption challenging for unauthorized entities (Srivatsava & Sheeja, 2020). Well-established cryptographic algorithms such as AES, RSA, DES, Blowfish, and Triple DES play a significant role in various security applications, including digital signatures, secure browsing, encrypted email communications, and credit card transactions. In the digital age, ensuring data confidentiality, integrity, and availability has become a pressing concern, necessitating robust encryption mechanisms. This study focuses on analyzing two essential encryption algorithms Data Encryption Standard (DES) and Blum-Blum-Shub (BBS) to enhance text security. Several studies have explored encryption techniques to improve data security. DES, as one of the earliest and most widely used symmetric encryption algorithms, has significantly influenced modern block ciphers (Reyad *et al.*, 2021). However, due to its relatively short key length of 56 bits, DES has been deemed vulnerable to brute-force attacks, prompting the need for enhancements. To address this limitation, research has been conducted to increase the key length, thereby strengthening encryption security (Alhag & Mohamed, 2018). Additionally, recent studies have investigated alternative encryption methods, such as the BBS pseudorandom number generator, which offers cryptographic security by generating unpredictable random sequences (Patil *et al.*, 2016). Despite these efforts, the challenge remains to optimize both security and computational efficiency.

The primary gap in the existing research lies in the limitations of current DES implementations, particularly concerning key length and encryption robustness. While previous studies have proposed increasing the DES key length, a comprehensive evaluation of its security implications and computational feasibility is still required. Furthermore, the integration of pseudorandom number generators, such as BBS, with DES encryption has not been extensively explored. Addressing these gaps is crucial to enhancing encryption mechanisms for secure data transmission. Given the increasing computational power available to adversaries, it is essential to develop encryption techniques that are resistant to evolving threats. This study aims to enhance the security of the DES algorithm by extending its key length to 1024 bits and integrating it with the BBS algorithm. The proposed approach seeks to improve encryption robustness while maintaining computational efficiency. By examining the interplay between DES and BBS, this research contributes to the advancement of secure cryptographic methodologies. The objective of this study is to evaluate the effectiveness of the proposed encryption scheme in enhancing data security, thereby providing a more resilient solution against modern cryptographic threats.

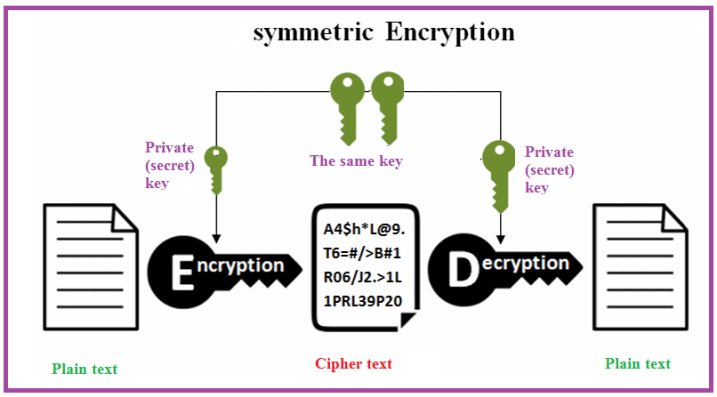

Figure 1. Encryption Decryption Process

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

2. Research Methods

The preliminary study emphasises the significance of choosing cryptographic algorithms carefully, focussing on a study that covers each algorithm's advantages, disadvantages, financial consequences, and performance measures. By developing and thoroughly analysing the cost and performance factors related to popular cryptographic algorithms including DES, 3DES, AES, RSA, and Blowfish, our work expands on this methodology. In contrast to strictly theoretical comparisons, this implementation seeks to present a thorough performance analysis that provides important insights into the real-world applications of these algorithms (Patil *et al.*, 2016). The relevance of the Data Encryption Standard (DES) algorithm, a popular symmetric key block cypher cryptographic approach, is examined in the second research endeavour. By increasing the key length to 1024 bits, this study aims to improve the DES algorithm. After then, this expanded key is split up into 16 keys, each of which has 64 bits. Interestingly, every key is produced separately during every algorithm cycle. When it comes to identifying the encryption key or figuring out how many keys there are overall using techniques like the Blind search method, which tries every potential key, the results of the suggested algorithm demonstrate significant gains over the traditional approach. In order to make the search in the large number space more difficult and deter successful attempts, longer keys, which indicate a higher level of complexity, are included (Alhag & Mohamed, 2018). The third study examines the Peak to Average Power Ratio (PAPR) in a Multicarrier Code Division Multiple Access (MC-CDMA) system by examining a number of sequences, including the Gold Sequence, the Blum Blum Shub (BBS) sequence, and the Walsh Hadamard (WH) sequence. The results show that, in comparison to the Walsh Hadamard (WH) sequence, the Blum Blum Shub (BBS) sequence has a lower Peak to Mean Envelope Power Ratio (PMEPR). In order to reduce PAPR in the MC-CDMA system, the BBS sequence thus shows promise as a substitute for the Walsh Hadamard (WH) sequence (Mohammed *et al.*, 2013). The Super DES Triple Encryption and RSA algorithms are the main topics of the fourth study. The research shows that both approaches are used to encrypt the text-generated data into ciphertext, which is subsequently re-described. This two-pronged strategy helps to guarantee a comparatively high degree of data protection. Super Encryption was added to both algorithms because of its intrinsic complexity, which makes it difficult for cryptanalysts to jeopardise data security.

This innovation greatly strengthens data security, making it less vulnerable to cryptanalyst attacks, even though it might slow down the encryption process. Enhancing encryption security and speeding up the encryption process by combining the strengths of RSA and Triple DES techniques is the main issue this study attempts to solve (Liana *et al.*, 2023). The fifth study describes how to use the Triple DES algorithm to create a secure environment in both local and exportable scenarios. This algorithm's adaptability is increased by the fact that it works especially well for textual content and image encryption. A data hiding key is used in the context of text content to extract more information. Additionally, the picture encryption key can be used to reconstruct the original image, taking use of spatial relationships in the process (Srivatsava & Sheeja, 2020). The sixth research study suggests a novel method for improving the security of textual content called Key-based Enhancement of the Data Encryption Standard (KE-DES). There are two essential steps in the KE-DES approach. First, each key bit in the DES algorithm undergoes an Odd/Even bit change. Secondly, it uses the Key-Distribution (K-D) function to replace the original DES's right-side expansion. In each round, 32 bits from the right side of the data and an extra 8 bits from Permutation Choice-2 (PC-2) are added to the K-D allocation, which starts with 8 bits from the Permutation Choice-1 (PC-1) key outcome. Strong security measures are enhanced by the random generation of the key and data. For text encryption, this special KE-DES model is thought to be more effective (Reyad *et al.*, 2021). The seventh study attempts to give a thorough grasp of the advantages and disadvantages of several random number generating techniques, clarifying their possible areas of use. The intention is to enable readers to evaluate and choose the best strategy for particular applications, possibly combining several different approaches.

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

The process can be started, for instance, by using a physical source of randomness, and then preprocessed to increase the randomness. Following preprocessing, the output can be fed into a variety of generators, including linear congruential generators, cryptographically safe generators, and generators that combine shared key cryptoalgorithms and one-way hash functions in multiple modes of operation. The study also explores the possibility of combining outputs from other generators to produce a final random sequence. To assess the randomness features of the generator outputs, useful randomness tests are provided. The use of cryptographically secure generators in fields where unpredictability is crucial, including eLotteries and cryptographic key generation, highlights the importance of unpredictable random sequences (Bikos *et al.*, 2023). The investigation of altered algorithms for the Blum-Blum-Shub (BBS) generator's operation is the subject of the eighth study, which focusses on improving its statistical properties, especially the sequence repetition period. The study comes to the conclusion that in order to improve the features of the conventional BBS algorithm, the recurrent equation the connection between the sequence's current and previous elements needs to be systematically adjusted. A generalised unified model for the modification of the conventional BBS algorithm is derived in the paper in order to achieve this goal. Both the classical approach and the 80 suggested modifications are examined in terms of computing cost and repetition time. The results show that statistical properties improve when the system's necessary computational power increases slightly. Potential uses in the field of education include teaching students about cryptographic stability in information security systems using the proposed modified BBS pseudorandom sequence generator. Studying this generator combines students' understanding of digital electronics and mathematics (Yu *et al.*, 2022).

The ninth study investigates the creation of external keys for message encryption and decryption by combining the Data Encryption Standard (DES) algorithm with the pseudo-random number generator Blum-Blum-Shub (BBS). This combination guarantees a high level of security and produces a unique key. The length of the chosen key and seed strongly correlates with the security level of the DES encryption and decryption key, highlighting the significance of longer keys for increased security. Notably, the DES algorithm's processing time for encryption is barely affected by using BBS as an external key generator (around 0.001 to 0.003 seconds for the 2–4 digits experiment). Furthermore, since the BBS key and seed are no longer subject to number limitations or feasibility tests, the decryption process takes no longer than expected. The combination of the DES and BBS algorithms provides increased security for encryption and decryption keys. The BBS algorithm generates unique random DES keys, which relieve users of the task of manually figuring out DES algorithm keys because they are generated programmatically (Laia *et al.*, 2021). In the tenth study, the emphasis is on enhancing the Advanced Encryption Standard (AES) to conform to the features and use cases of the Internet of Things (IoT) environment. As a result, the Data Encryption Standard in IoT (DESI) was created. The investigation shows that DESI has better security features as a result of this optimisation, making it a good choice for data encryption in the Internet of Things. The particular security requirements and difficulties related to information protection in the context of the Internet of Things are addressed in part by this research (Su & Zhang, 2019).

2.1 Cryptography

People can safely exchange private messages with each other or the intended recipient by using cryptography. It entails using a coded method to make sure that the communication is incomprehensible to outsiders or those with bad motives, such hackers. When the transmission system is vulnerable to interception by unauthorised parties, this security measure becomes much more important. Cryptography serves as a method for protecting and sending data in a way that only those who possess the right decryption key can understand, guaranteeing the safety of important data whether it is transferred across communication networks or kept on different types of storage media (Liana *et al.*, 2023). Cryptography is a technology and a science used to protect data from unauthorised parties. Encryption and decryption are two essential cryptographic operations. The process of encryption entails utilising a key to change the original text, or plain text, into encrypted text, or

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

cypher text. On the other hand, decryption means using the same key to transform the encrypted text (cypher text) back into the original text (plain text). This procedure is especially important when using symmetric-key cryptography, which encrypts and decrypts data using the same key (Permana & Nurnaningsih, 2020). In almost every cryptography application, pseudorandom numbers are used. These numbers have properties that mimic actual randomness, which makes it difficult for hostile actors to tamper with the application. Pseudorandom number generators (PRNGs) are the techniques that produce these pseudorandom numbers, which are basically bit sequences (Bikos *et al.*, 2023). Deterministic methods known as pseudo-random number generators (PRNGs) are made to produce sequences that pass a variety of randomness tests. Recursion is frequently used by these generators to create numerical sequences that resemble randomness. Numbers are generated recursively in this procedure, meaning that the previous number determines the next number in the sequence. A certain value known as the seed is used to start these recurrences. Crucially, a recurrence yields the same sequence when it begins with the same seed. The period or cycle length is the length of a sequence before it repeats (Aljahdali, 2020).

2.2 Data Encryption Standard (DES)

Block cyphers, which are a subset of symmetric cryptography, include the DES (Data Encryption Standard) algorithm. Although DES was widely used and accepted as a standard symmetric key method, security issues have rendered it obsolete, and newer algorithms have taken its place. In order to create 64 bits of coded data, DES encrypts and processes 64 bits of source text using 64-bit blocks. A 56-bit internal key or subkey is used in this encryption procedure (Thahara & Siregar, 2021). In today's digital world, data security is crucial, and the Data Encryption Standard (DES) Algorithm is one of the most popular cryptographic algorithms for protecting the integrity and confidentiality of data. DES, a key instrument in the world of cryptography, is essential for tackling the problems brought on by the changing digital age (Pratama *et al.*, 2023). The term "plain text" describes the first or original message sent during a communication session. The techniques of encryption and decryption are applied to this plain text. The hidden message that emerges from the encryption of the original message (Plain Text) during the cryptography process is known as Cypher Text. The supplied key can be used to return the Cypher Text to its original form, which is Plain Text. The process of converting plaintext data into ciphertext, or veiled data, is called encryption. One encryption technique used in block cypher systems is the DES (Data Encryption Standard) algorithm. This encryption method uses 64-bit input blocks (original text) to randomise data block by block. It also generates output (ciphertext) in 64-bit blocks. A symmetric key method with a 56-bit key length is employed. The DES cypher system was first developed as the industry standard for protecting data that was kept and sent, but it soon gained international. acceptance for a range of applications that need to be encrypted while in use (Buulolo & Sindar, 2020). The symmetric key block cypher known as the Data Encryption Standard (DES) has a block size of 64 bits and a key length of 56 bits. DES, which was first created by IBM in 1972 as a data encryption method, was later accepted as a standard encryption system by the US government. A 64-bit key was used initially, but the National Security Agency (NSA) eventually enforced a 56-bit key length limit. As a result, DES used the compressed 56-bit key for data encryption in 64-bit blocks, deleting 8 bits from the original 64-bit key. Even with its extensive use, DES has flaws, especially when a weak key is employed. Its versatility is increased by a variety of operating modes, including CBC, ECB, CFB, and OFB. Notably, DES was put to the test in 1998 when the supercomputer DES Cracker broke DES in under 22 hours with the help of several dispersed PCs on the Internet (Patil *et al.*, 2016). In fact, DES (Data Encryption Standard) is a member of the Feistel cypher family, and its structure is consistent with what is expected of Feistel cyphers. Nevertheless, within this framework, DES is distinguished by several details:

1) Block Length: The DES block length is 64 bits. Both the original plaintext and the resulting ciphertext have a size of 64 bits.
2) Key Size: The DES key is 64 bits in size. Each round of the DES algorithm employs a round key of 48 bits.

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

3) Number of Rounds: DES operates through 16 rounds. This means that the encryption or decryption process involves iterating through 16 consecutive rounds, each utilizing a different round key.

The unique structure and operation of DES as a Feistel cypher are facilitated by these specifications (Thahara & Siregar, 2021).

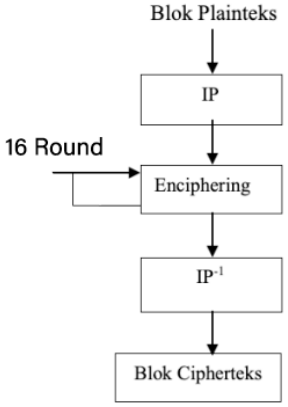

Figure 2. Global Scheme Of The Des Algorithm

The plaintext block is first split into the left (L) and right (R) portions, each of which has 32 bits, in the DES (Data Encryption Standard) enciphering procedure. The DES encryption mechanism then goes through 16 rounds with these two components. The right block (R) is the input to a transformation function (f) in each round, which is represented as round i. The right block (R) is joined using an internal key inside the function f. The primary encryption key in DES is the source of this round-specific internal key. In order to change and mix the bits in the right block and aid in the encryption process as a whole, the transformation function f is essential (Thahara & Siregar, 2021).

2.3 Blum-Blum-Shub (BBS) Algorithm

The Blum Blum Shub (BBS) algorithm generates binary sequences called Blum Blum Shub sequences by acting as a pseudorandom bit generator (Alhag & Mohamed, 2018). The following steps are part of the generation process:

1) Selection of Primes: Identify two prime numbers, p and q, both of the form 4k+3, where k is any integer. It is crucial that both p and q are congruent to 3 modulo 4.
2) Modulus Calculation: Determine the modulus, n, as the product of p and q ($n = p \times q$).
3) Random Integer Selection: Choose a random integer 'r' that is coprime to n (i.e., it shares no common factors with n).
4) Seed Initialization: For the first iteration, set $x_0 = r^2 \bmod n$. This initial value x_0 is referred to as the seed.
5) Sequence Generation: Generate subsequent elements in the sequence using the recurrence relation $Xn+1 = (xn)^2 \bmod n$.

Each decimal digit is converted to binary, and the Least Significant Bit (LSB) is extracted in order to transform the generated random bits into binary form. According to the binary representation convention, 0 is represented as -1 and 1 as 1. In many applications, the resulting 1s and -1s sequence can be used as spreading sequences (Mohammed *et al.*, 2013).

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

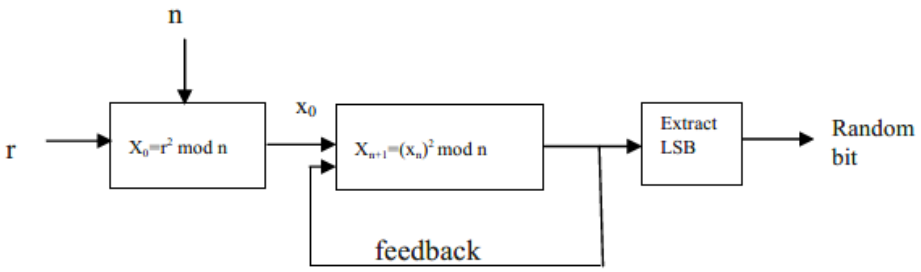

Figure 3. Block Diagram For Blum Blum Shub Sequence Generator

In terms of theoretical complexity, the Blum Blum Shub (BBS) number generator is one of the most straightforward and potent Cryptographically Secure Pseudorandom Number Generators (CSPRNGs). The BBS algorithm, which was created in 1986 by Lenore Blum, Manuel Blum, and Michael Shub (Permana & Nurnaningsih, 2020), generates pseudorandom numbers using the following equation:

$$Xn + 1 = (xn)^2 \, mod \, n \quad (1)$$

The current value in the sequence is denoted by Xn in this equation, and the modulus operation with 'n' is indicated by the operation mod n. Despite having a straightforward formulation, the BBS algorithm has robust cryptographic characteristics that make it appropriate for safe pseudorandom number generation (Lestariningsih *et al.*, 2022).

2.4 Java Implementation

Java is a flexible, object-oriented programming language that can be used to create apps for desktop, web, and mobile platforms. James Gosling developed it in 1995 using Sun Microsystems' Java platform (Gunarto *et al.*, 2018). Although Java offers a more straightforward and inflexible syntax with restricted operating system access, its syntax is influenced by those of C and C++. Because of this design decision, Java is comparatively simple to understand and learn. Files with the.java extension are used in Java programming and then recompiled into.class files. Bytecode found in.class files can execute on any Java Virtual Machine (JVM), regardless of the processor architecture or operating system. Java is an object-oriented, class-based language that emphasises platform freedom, enabling programs to function flawlessly across a variety of processor and operating system contexts (Gunarto *et al.*, 2018).

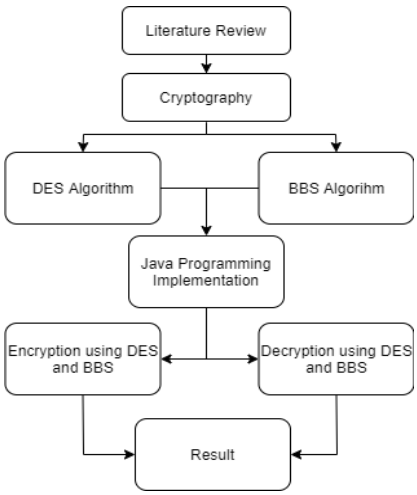

Figure 4. Theoritical Framework

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

3. Result and Discussion

3.1 Results

According to our investigation, because of its age and decreased use, the DES encryption method is currently susceptible to simple decryption. As a result, we have improved it by fusing the Blum Blum Shub (BBS) Algorithm with the 16-block cypher rounds of DES. Our main goal is to develop an application that may be used to solve the security issues with the outdated DES algorithm in the present and the future. In this version, we improve the encryption results of DES by utilising the arithmetic computations of BBS. Our program is intended for desktop use and is written in the Java programming language. Without relying on cloud services, which are thought to have flaws, this method guarantees data security. Encrypting or locking words or images is a notion in cryptography. The DES algorithm, which has been improved by adding the BBS algorithm, is the one used for this encryption. We use a bespoke program tool created with the Java programming language to perform encryption and decryption on data, with a concentration on text. Testing is done using a variety of plaintexts to gauge the precision of the encryption and decryption procedures for data or information in order to determine whether the suggested approach is feasible. The procedures for the encryption process on the data or information that will be subjected to accuracy testing are as follows:

1) Determine the DES Key that will be used to lock or encrypt the data or information

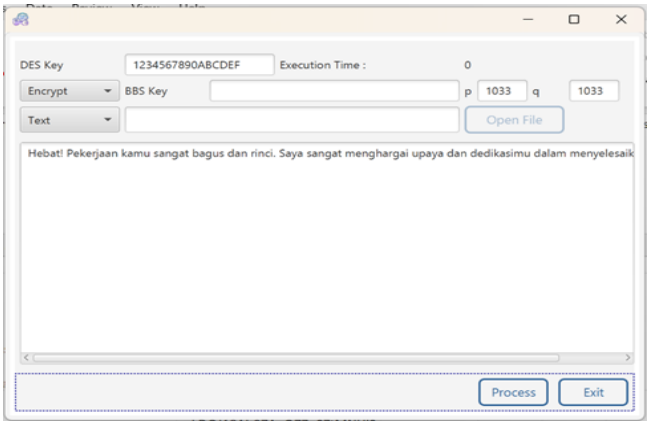

Figure 5. Encryption text

2) Select Encrypt. In the image above, it can be seen that we specify the DES Key, where DES Key refers to Data Encryption Standard (DES) used to determine the block cipher permutation mechanism in DES for 16 rounds. For the selection of the Encrypt menu option, it is a functionality that determines whether the program will encrypt or decrypt. The Process button here represents the transformation from plaintext to ciphertext, producing a sequence of random or pseudo-random binary bits. This algorithm is also associated with modifying or strengthening the algorithm. The algorithm consists of two stages: Xr * c mod n and Xr * c' mod n. The first stage multiplies the input r by a constant c and takes the modulo with a number n. The second stage does the same with a different constant c'. The final output of this algorithm is a random bit, a binary bit that is randomly assigned a value of 0 or 1. This random bit can be used for various purposes, such as data encryption, data compression, or data simulation.

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E - ISSN: 2723-7079 | P - ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

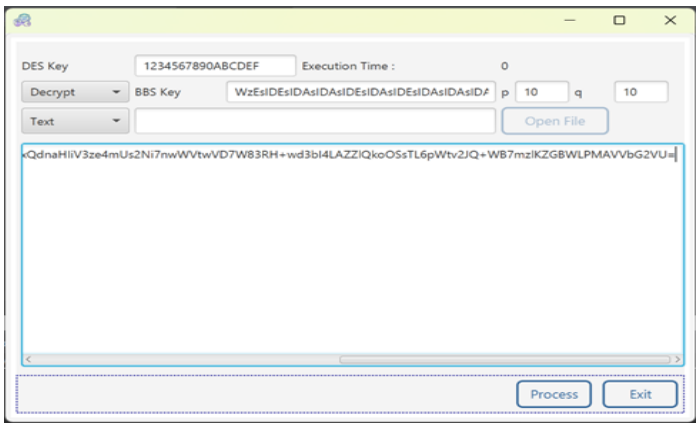

Figure 6. Encryption text for Decrypt

3) Input variables p and q are used as BBS Keys to generate random values that will enhance processing time during data encryption. In the image, there are specific options to determine the success of encryption by specifying the P x Q BBS (Blum - Blum Shub) to obtain random ciphertext results. This serves as the key to decrypt the encrypted message.

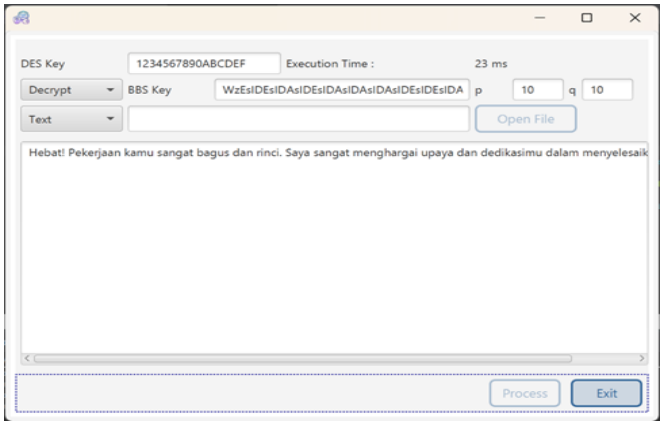

Figure 7. Decryption text result

4) Prepare the text for encryption. In this image, we have used several sample words or sentences that will be scrambled using the program for encryption.

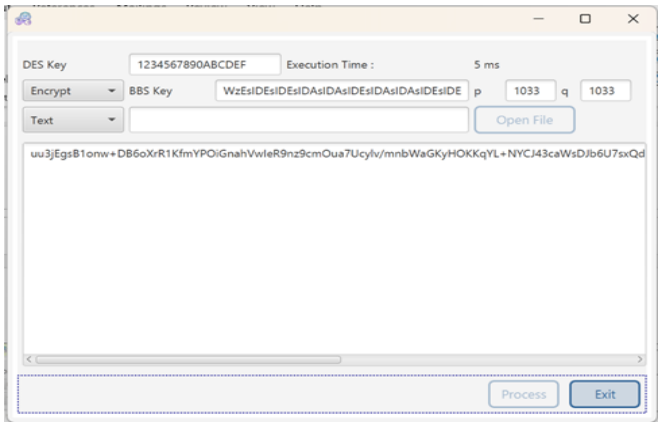

Figure 8. Encryption text result

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

5) Click the process button to initiate the encryption process. After clicking the process button, the encryption results and the measured time can be seen in the image. This shows the result of the ciphertext generated by the program. This ciphertext key will be used later to decrypt the plaintext.

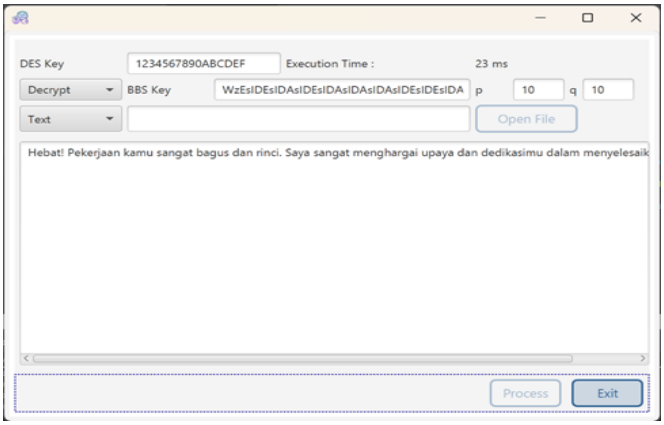

Figure 9. Decryption text result

6) Text Decryption. After clicking the process button for decryption results and measuring the time, as shown in the image below, it reveals the ciphertext that we input to open the plaintext or restore it to its original form. The following is a table of several example sentences that are used as test cases or test data using the program:

Table 1. Encryption text result

No	Plaint text	Chiper text	Time Process
1	Great! Your work is very good and detailed. I really appreciate your effort and dedication in completing this task.	uuzjEgoB14nx+TF6oHvR1KbmYfOjGnegVgIfR9jz9MiPuK7Ucylu/mjaWKGLyXKLKqYK+dcDJ43caWoDJb6U7sxRd3aHIyV3ze4nUsyMi7nwWFtxVT7X8nVH+gd3bI4LAZZkQ0oOS8XL6pStvmJQ+WF6mzlKZWFXLfMAVFfH2FU=	5 ms
2	Perhaps we could consider some minor improvements to the user interface to make it more user-friendly. I believe it would improve the overall user experience.	TtOXPbTGUcM5n4yBDaEbLYki6xnJO9ICCUB3BazApsR4C+MR4y4tUfXFoKl7okFvk73dU3a/1eO3k01+qtoRdsyBHnA47wXEHc28WCw88LJdnopR2fbvftm8i0tfMytlMqHi73s4VXwWxPOLqlPLxFiHfICAX82QTEYTd1fQ//D0gAYk4zUnnoWQh5JA3NEbBkkxXiQO9Cs=	2 ms
3	Your presentation is already good, but try to speak at a slower pace and engage more with the audience. This will make your presentation more memorable and memorable.	XdwJ6l074HyNnlaInfpIEXLuhGQ8RHmKpkNX7QxGvQ1ioJH1NHsnkq3rOUb3bV4ISwcze+Yr9SuSXA1aE8W00ZU9PhLfBVk3wOTRHnaJ2PAFrgXoAbsj68xzlH9u7CCOI43Drk/9s2pZHc3ZC2dWOcZjGhzGr7dY+zUTxtFVYjiYqoLur4+U7sWCfbBLek2jcyYq/TN1n7WbDY7Vkc7OzQ==	1 ms
4	Thank you for your positive contribution to this project. Your	LN49i3vm5XGXsCa1F7Ketw6yBl6CpsqYOUyFCHxk+qkAQtHI1QQD5y2wS4gNTaSr0UpI6mnwWF1VinmXpNBgK3PaGvezV41mzUATHLOxYg	1 ms

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

	opinions during the planning sessions were valuable and had a great impact on the success of the project. Let's continue to work together effectively!	8pCOPr0YBv7T8WsVmvAh3wSKObw0tZHzPqVxtikQAJY76T2NAoKOugcuOG2NZKEJqjQT0tlohLvt16Tftuslptjd6qMA/ET/DruYGtD17lrw==	
5	You already have great skills, but consider attending additional training in technology. This will expand your repertoire and make you more valuable to the team.	beaeCRJ6+RMZ3hPn+gPUUm11nZW9KJtQ2meDblRgxZniKELt08hUwCesEIDm829g2hRfXoAfUnady1j5g/EqWeQBE0wnxWrJrSpKey4x6BDnNRm7NFLMu7ebq8EKLUxgiZSV6oBEgf0AHXxdaoqJLWTSWX+AHq80gQS1xw0E+1GwhccVDl8WZSXCl+RJ79OdsZCX66jXVphU1UnccQ6D0g==	3 ms

Example number 1 the text "Great Your work is very good and detailed I really appreciate your effort and dedication in completing this task" changes to DES KEY u+ziEgsA1onx+TB7oXrQ1afmYPOjGnagVwIeR9ny9MiPua7Vcylu/mnaWKCKyXKLK6cK+NcCJozdaGsCJb+U781RdnaGIiR3ze4mU82NirjxWFpxVT7W8nRH+wd2bY4LAZdkQkoOS8XK6pSsv2JQ+WB7mzhKZGBWLPIBVVfG2FQ=

From this DES key, it occurs solely due to the mechanism of DES itself that utilizes block permutation. BBS (Blum - Blum Shub) is employed as an improvisation to enhance the DES algorithm, thereby resulting in the ciphertext or encrypted output.

uuzjEgoB14nx+TF6oHvR1KbmYfOjGnegVgIfR9jz9MiPuK7Ucylu/mjaWKGLyXKLKqYK+dcDJ43caWoDJb6U7sxRd3aHIyV3ze4nUsyMi7nwWFtxVT7X8nVH+gd3bI4LAZZkQ0oOS8XL6pStvmJQ+WF6mzlKZWFXLfMAVFfH2FU=

The mechanism and outcomes of the DES and BBS concepts themselves, with a processing time of 5 milliseconds, make the encryption and decryption processes of the text using the developed program intriguing for analyzing data or information security systems. The alteration of these bits is crucial for enhancing data and information security, as it adds time for other tools to decrypt the resulting ciphertext. The findings indicate that the proposed process performs better, resulting in a reasonably fast processing time. The encrypted plaintext can be observed in the result table in the form of ciphertext, with an average processing time of 2 seconds. These results make this proposal promising, as the process is more secure and challenging to crack compared to standard DES passwords. Based on our research findings, it can be stated that the use of this encryption provides the following outcomes: Vulnerabilities of the DES Algorithm, the analysis of the Data Encryption Standard (DES) algorithm in this research highlights its susceptibility to straightforward decryption due to age and reduced usage. The identified vulnerabilities underscore the pressing need for enhancement to address evolving security concerns in the rapidly advancing digital landscape. Enhancement through Integration with BBS Algorithm, to fortify DES security, we propose integration with the Blum-Blum-Shub (BBS) algorithm, a cryptographic pseudorandom number generator. This integration combines DES's 16-block cipher rounds with BBS's arithmetic calculations, creating a robust encryption method resilient to modern cryptographic attacks. Implementation Details and Program Development, the Java-based desktop program developed aims at ensuring data security independently of cloud services. Leveraging BBS's pseudorandom number generation enhances DES encryption outcomes, creating a versatile program for present and future use. Feasibility Testing and Results, feasibility testing involving various plaintexts demonstrates the

E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

program's efficacy. The encryption process, encompassing DES key determination, encryption options selection, BBS key initialization, and plaintext preparation, yields successful results with measured processing times. Encryption and Decryption Mechanism, the encryption mechanism transforms plaintext into ciphertext using DES and BBS algorithms. The DES key dictates the block cipher permutation for 16 rounds, while BBS contributes to random ciphertext generation. Decryption involves inputting ciphertext and using the corresponding BBS key for successful recovery. Comparative Analysis and Security Improvements, comparing the proposed method with traditional DES encryption reveals significant security improvements. BBS integration introduces a pseudo-random element, adding complexity to the encrypted data. Bit alteration through BBS enhances overall data security, making decryption more challenging. Processing Time and Efficiency, with an average processing time of 2 seconds for various test cases, the proposed method maintains efficiency crucial for practical implementation. In addition to quantitative tests such as the Avalanche Effect and statistical randomness tests, it is also important to consider the theoretical resistance of the DES-BBS method against classic cryptanalytic attacks on DES, namely Differential Cryptanalysis and Linear Cryptanalysis. These attacks exploit imperfect non-linear properties and statistical correlations in the standard DES structure, particularly in the S-box functions and the key schedule. The integration of pseudorandom output from BBS, particularly when used to modify round keys or perform XOR operations on data before or after each round, helps disrupt patterns commonly leveraged by attackers. The added randomness strengthens resistance by concealing linear correlations and high-probability differential paths typically exploited in While formal mathematical analysis is beyond the scope of this study, the incorporation of a strong pseudorandom generator like BBS is expected to introduce additional complexity in the application of differential and linear attacks compared to standard DES.

However, further cryptanalytic investigation is necessary to fully determine the extent of this resistance enhancement in future research. Although formal mathematical analysis is beyond the scope of this study, conceptually, the use of a strong. Swift encryption and decryption processes ensure applicability in real-world scenarios without introducing significant delays. Implications and Future Work, the successful integration of DES with BBS opens avenues for further research. The outcomes of this approach include enhanced data security, flexible key selection, efficient encryption and decryption processes, a user-friendly interface, and significant improvements over standard DES. From a practical standpoint, this method presents a viable alternative for strengthening communication security in legacy industrial control systems or other devices that remain reliant on DES-based protocols and are either costly or technically challenging to upgrade entirely to modern standards such as AES. In such scenarios, where additional protection against basic eavesdropping or replay attacks is required without drastic infrastructure modifications, the DES-BBS method offers a potential compromise solution. Beyond improving resistance to common exploitation patterns found in standard DES, this approach also maintains the computational efficiency necessary for real-world applications. With a combination of robust encryption and rapid processing, the method can be deployed across various resource-constrained systems, such as IoT devices or embedded systems that continue to rely on DES-based cryptographic algorithms. Future research may explore scalability, optimal parameter tuning, and further cryptanalytic evaluations to refine its effectiveness and adaptability in diverse security environments. Flexible key selection, fast and efficient encryption processes, a user-friendly interface, and notable improvements over standard DES. Future studies may explore scalability, applicability to diverse data types, and optimizations for enhanced performance, emphasizing the need for adaptive encryption strategies to meet evolving security challenges. Consolidated Discussion, our research not only identifies vulnerabilities in the aging DES algorithm but also proposes a comprehensive solution through the Enhanced DES-BBS algorithm. This amalgamation showcases improved data security, flexibility in key selection, efficient processing times, and a user-friendly interface. The comparative analysis highlights its superiority over traditional DES, emphasizing the importance of adaptive encryption strategies in addressing contemporary security challenges.

Jurnal Indonesia : Manajemen Informatika dan Komunikasi
https://journal.stmiki.ac.id
E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

3.2 Discussion

This research aims to address the vulnerabilities inherent in the Data Encryption Standard (DES), which, despite its extensive use, remains susceptible to various types of attacks. DES, with its relatively short 56-bit key, is easily compromised by brute-force attacks, highlighting the need for improvements to maintain its relevance in the face of evolving cyber threats (Alhag & Mohamed, 2018). To address this limitation, a proposed enhancement involves extending DES's key length to 1024 bits and integrating the Blum-Blum-Shub (BBS) algorithm, a pseudorandom number generator known for its ability to generate highly unpredictable sequences. This modification seeks to increase the complexity of the encryption process by introducing greater randomness, making it more resistant to cryptanalysis (Laia *et al.*, 2021). The key enhancement of combining BBS with DES lies in adding an additional layer of security by increasing the unpredictability of the keys used in the encryption process. BBS, based on number theory and modulus operations with prime numbers, generates random sequences that are extremely difficult to predict, even by advanced computational systems (Bikos *et al.*, 2023). This added randomness is crucial in counteracting differential and linear cryptanalysis attacks, which exploit patterns in the DES key schedule and S-box functions. By incorporating BBS, the encryption becomes less predictable and harder to break, as the key generation is now influenced by a more complex pseudorandom sequence (Patil *et al.*, 2016). In addition to enhancing security, the integration of BBS with DES does not significantly impact computational efficiency. Testing has shown that the combined method maintains an average processing time of just 2 seconds, demonstrating that the added complexity of BBS does not excessively slow down the encryption and decryption processes. This performance is important as it ensures the proposed method can be implemented in real-world applications without compromising system speed (Laia *et al.*, 2021). Despite the increased complexity of the encryption process, the method remains efficient enough for practical use in systems that require both high security and fast processing. When compared to traditional DES, the DES-BBS method provides a significant improvement in terms of security. The combination of BBS's unpredictability with DES's 16 rounds of encryption strengthens the overall encryption process, making the ciphertext more difficult to decipher. The increased complexity introduced by BBS means that even if an attacker attempts to break the encryption, the added randomness will make decryption much more challenging. This improvement has been demonstrated through testing, where various plaintexts showed that the proposed method significantly outperforms DES in terms of resistance to cryptanalysis (Buulolo & Sindar, 2020). The use of BBS also prevents the formation of predictable patterns that could be exploited in attacks, offering stronger protection against potential breaches.

One practical application of this enhancement is in systems that still rely on DES, such as certain industrial control systems that have not yet transitioned to more advanced algorithms like AES due to cost or infrastructure limitations. Using DES-BBS as a more secure alternative offers increased protection for data without requiring a complete overhaul of existing systems (Lestariningsih *et al.*, 2022). This is especially useful for devices with limited resources, such as IoT or embedded systems, where maintaining security is critical but computational resources are constrained. However, while this research demonstrates the potential of the DES-BBS method to improve security, further exploration is needed. Future studies could focus on testing the method with other types of data, such as images or audio, to determine how well it performs with various data formats (Yu *et al.*, 2022). Additionally, more in-depth analysis of its resistance to advanced cryptanalytic attacks is necessary to assess its robustness in the face of newer, more sophisticated attack methods. Overall, this research successfully demonstrates that integrating the BBS algorithm with DES can address some of the major weaknesses of DES, enhancing its security without compromising efficiency. This method provides a more secure solution for protecting data in modern cryptographic applications and opens the door for further advancements in cryptography. The findings of this study suggest that DES-BBS can offer a reliable and practical encryption solution in scenarios where DES is still in use but requires stronger protection against modern threats.

E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

4. Conclusion

This study successfully addresses the security vulnerabilities of the Data Encryption Standard (DES) by integrating it with the Blum-Blum-Shub (BBS) algorithm, enhancing encryption robustness through the combination of DES's 16-block cipher rounds with the pseudo-random properties of BBS. The implementation of a Java-based encryption program demonstrates the feasibility of this approach, ensuring data security without reliance on cloud services. Feasibility testing confirms that the enhanced DES-BBS method significantly improves encryption security while maintaining efficiency, with an average processing time of 2 seconds. The comparative analysis reveals that BBS integration introduces an additional layer of randomness, strengthening data confidentiality and making decryption more challenging. This method ensures a balance between security enhancement and computational feasibility, making it suitable for real-world applications. Future research can explore scalability, adaptation to diverse data types, and further optimization of processing time while integrating other cryptographic techniques to enhance security against evolving cyber threats.

5. Acknowledgements

We want to express our sincere gratitude to everyone who helped us finish and make this research project a success. This research would not have been feasible without the assistance, direction, and motivation of numerous people and institutions. First and foremost, we would like to sincerely thank Dr. Dewi Yanti Liliana, S.Kom., M.Kom., our advisor, whose knowledge, guidance, and unshakeable dedication were crucial in determining the course of this study. Your insightful comments and helpful criticism greatly improved the calibre of our work. We are appreciative of Nusa Mandiri University's facilities and resources, which made it possible to perform our research in a favourable setting. Additionally, the collaborative environment has proved crucial in promoting academic development. We would like to express our sincere gratitude to everyone who so kindly offered their time and assistance during the Enhanced DES-BBS algorithm testing phase. Their participation was essential to confirming the effectiveness and functionality of the suggested encryption method.

6. References

Alhag, N. M. M., & Mohamed, Y. A. (2018, August). An enhancement of data encryption standards algorithm (DES). In *2018 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)* (pp. 1-6). IEEE. https://doi.org/10.1109/ICCCEEE.2018.8515843.

Aljahdal, A. O. (2020). Random Number Generators Survey. *International Journal of Computer Science and Information Security (IJCSIS)*, *18*(10), 14-22.

Bikos, A., Nastou, P. E., Petroudis, G., & Stamatiou, Y. C. (2023). Random number generators: Principles and applications. *Cryptography*, *7*(4), 54.

Buulolo, N., & Sindar, A. (2020). Analisis dan Perancangan Keamanan Data Teks Menggunakan Algoritma Kriptografi DES (Data Encryption Standard). *Respati*, *15*(3), 61-65.

Gunarto, G., Abdullah, A., & Irawan, D. (2018). Model Matematis Turbin Pelton Dengan Menggunakan Bahasa Pemrograman Java. *Machine: Jurnal Teknik Mesin*, *4*(2), 9-14. https://doi.org/10.33019/jm.v4i2.481.

E-ISSN: 2723-7079 | P-ISSN: 2776-8074
Vol. 6 No. 2 (2025) | Mei
OPEN ACCESS
https://doi.org/10.63447/jimik.v6i2.1385

Laia, O., & Zamzami, E. M. (2021, June). Analysis of combination algorithm data encryption standard (DES) and Blum-Blum-Shub (BBS). In *Journal of Physics: Conference Series* (Vol. 1898, No. 1, p. 012017). IOP Publishing. https://doi.org/10.1088/1742-6596/1898/1/012017.

Lestariningsih, E., Ardhianto, E., & Handoko, W. T. (2022). ADOPSI PEMBANGKIT KUNCI EXTENDED VIGENERE MENGGUNAKAN FUNGSI RANDOM DAN BLUM BLUM SHUB. *Jurnal Informatika dan Rekayasa Elektronik*, *5*(2), 263-271. https://doi.org/10.36595/jire.v5i2.706.

Liana, L., Zarlis, M., & Tulus, T. (2023). Hybrid Cryptosystem Analysis RSA Algorithm And Triple DES Algorithm. *Sinkron: jurnal dan penelitian teknik informatika*, *7*(3), 1461-1473. https://doi.org/10.33395/sinkron.v8i3.12467.

Mohammed, V. N., Mallick, P. S., Nithyanandan, L., Asrani, M., & Saxena, M. (2013). Blum BlumShub Pseudorandom Sequence based Peak Power Control in MC-CDMA System. *IERI Procedia*, *4*, 309-315.

Patil, P., Narayankar, P., & Narayan D G, M. S. M. (2016). A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish. *Elsevier (ICISP 2015 Proceedings)*. https://doi.org/10.1016/j.ieri.2013.11.044.

Permana, A. A., & Nurnaningsih, D. (2020). Application of cryptography with data encryption standard (des) algorithm in picture. *JIKA (Jurnal Informatika)*, *4*(2), 82-87.

Pratama, A., Arif, M. N., Nazir, M., & Dannaun, Z. (2023). Algoritma DES (Data Encryption Standard) Untuk Keamanan Digital. *JURNAL SITEBA*, *2*(1), 15-18.

Reyad, O., Mansour, H. M., Heshmat, M., & Zanaty, E. A. (2021, March). Key-based enhancement of data encryption standard for text security. In *2021 National Computing Colleges Conference (NCCC)* (pp. 1-6). IEEE. https://doi.org/10.1109/NCCC49330.2021.9428818.

Srivatsava, J., & Sheeja, R. (2020). Implementation of triple des algorithm in data hiding and image encryption techniques. *Int J Adv Sci Technol*, *29*(3), 10549-10559.

Su, N., Zhang, Y., & Li, M. (2019, March). Research on data encryption standard based on AES algorithm in internet of things environment. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (pp. 2071-2075). IEEE. https://doi.org/10.1109/ITNEC.2019.8729488.

Thahara, A., & Siregar, I. T. (2021). Implementasi Kriptografi untuk keamanan Data dan Jaringan menggunakan Algoritma DES. *Jurnal Rekayasa Teknologi Informasi (JURTI)*, *5*(1), 31.

Yu, S., Krzysztof, P., Yan, L., Maksymovych, V., Stakhiv, R., Malohlovets, A., & Kochan, O. (2022). Development of modified blum-blum-shub pseudorandom sequence generator and its use in education. *Measurement Science Review*, *22*(3), 143-151. https://doi.org/10.2478/msr-2022-0018.