International Journal of Research and Innovation in Multidisciplinary (www.ijrim.com)
|| Volume 2, Issue 1, January 2025 ||

Blowfish Cryptography based Security Technique

[1]Gaurav Singh, [2]Dr. Monika Kapoor
[1]M.Tech Scholar, [2]Professor
Department of Electronics & Communication Engineering,
Lakshmi Narain College of Technology, Bhopal, India

***Abstract*— With the rapid expansion of digital communication and data storage, ensuring data security has become a critical challenge. Cryptographic techniques play a vital role in protecting sensitive information from unauthorized access, cyber threats, and data breaches. Blowfish, a symmetric key block cipher known for its efficiency and strong encryption, is widely used for securing data in various applications, including network security, file encryption, and secure communications. This paper explores the Blowfish cryptographic algorithm, its structure, key expansion process, and encryption-decryption mechanism. By analyzing its strengths, such as fast computation, low memory usage, and resistance to brute-force attacks, as well as its limitations, this study provides insights into its suitability for modern security applications. Additionally, a comparative analysis with other encryption techniques highlights the advantages of Blowfish in terms of speed, security, and resource efficiency. The findings demonstrate that Blowfish remains a robust and reliable cryptographic technique, making it a preferred choice for data security in constrained environments.**

***Keywords*—Internet of Things (IoT), Encryption, Security, Blowfish, Privacy.**

I. INTRODUCTION

The increasing reliance on digital systems for data storage, communication, and transactions has significantly heightened the need for robust security mechanisms. Cryptographic algorithms serve as the foundation for ensuring data confidentiality, integrity, and authenticity in cyberspace [1]. Among various encryption techniques, symmetric key cryptography offers a balance between efficiency and security, making it suitable for real-time applications. Blowfish, a widely recognized symmetric block cipher developed by Bruce Schneier in 1993, has gained popularity due to its strong encryption capabilities and computational efficiency [2].

Blowfish operates on a 64-bit block size and supports variable key lengths ranging from 32 to 448 bits, allowing users to balance security and performance based on specific requirements. Unlike traditional symmetric encryption algorithms such as the Data Encryption Standard (DES), which has been deemed insecure due to its small key size, Blowfish provides stronger resistance against brute-force attacks [3]. Its Feistel network structure, consisting of 16 rounds of encryption, ensures high security while maintaining fast execution speeds. The key expansion phase of Blowfish, which generates subkeys from the user-defined key, is computationally intensive but contributes to its robustness against cryptanalysis [4].

One of the primary reasons for the widespread adoption of Blowfish is its lightweight nature, making it suitable for resource-constrained environments such as embedded systems, IoT devices, and secure cloud computing [5]. Unlike modern encryption standards like the Advanced Encryption Standard (AES), which require higher computational power, Blowfish offers a balance between security and speed, making it an ideal choice for applications where performance is a critical factor. Additionally, its simplicity and public domain availability have led to its integration into various security protocols, including Secure Shell (SSH), Virtual Private Networks (VPNs), and password protection systems [6].

Despite its advantages, Blowfish has certain limitations that restrict its widespread adoption in some scenarios. The 64-bit block size, while sufficient for earlier applications, is now considered small in the context of modern cryptographic security standards, particularly when dealing with large volumes of data [7]. This limitation makes it more susceptible to birthday attacks in long-duration communications. Furthermore, the initial key setup process in Blowfish is computationally expensive, which can introduce overhead in applications requiring frequent key changes. To address these concerns, modified versions such as Twofish and Threefish have been introduced, offering improved security and scalability [8].

Over the years, Blowfish has been extensively analyzed for its security strengths and vulnerabilities. Studies have shown that the algorithm is highly resistant to differential and linear cryptanalysis, making it a reliable encryption

standard for many applications [9]. However, in environments where higher block sizes and stronger cryptographic primitives are required, AES is often preferred. Nevertheless, Blowfish remains a viable option for secure data transmission, authentication protocols, and cryptographic file systems, particularly in scenarios demanding efficient encryption with minimal processing overhead [10].

II. LITERATURE SURVEY

B. M. B. Beron et al.,[1] Authors B. M. B. Beron and colleagues, [1] Concerning data security, software-based solutions may not be enough, and the cost of data breaches is frighteningly high. This research was initiated due to the need for hardware-level cryptography and aims to implement a Blowfish cryptographic core in an ASIC using 0.13 μm CMOS process technology. It will also modify the Blowfish algorithm to reduce propagation delay and improve performance through the use of pipelining.

Nalawade et al., (S. B.) [2] The authors of this paper suggest using field-programmable gate arrays (FPGAs) to build and execute the Blowfish algorithm. The Virtex-5XC5VLX50T FPGA chip was used as a reconfigurable platform for the development of the Blowfish algorithm, while VHDL was utilised for RTL coding. The system's objective is to assess the Blowfish algorithm's power consumption and throughput performance on a reconfigurable platform. Image and electrocardiogram data has been used in its raw form for the sake of testing.

Setiawan et al., H., [3] Developing an electronic secure disposal application is the focus of this study. It will be built in compliance with Regulation of Electronic Service Manuscripts, number 6 of 2011, which aims to address manual disposition issues in government institutions. The Blowfish algorithm will be used for encryption in the application. in the attached file, along with digital signatures using RSA and SHA-512 hash algorithms. The time needed to decode the ciphertext is used as a metric in the study of M. A. Muin et al. [4]. Decryption times that are longer make it harder to use brute force attacks to recover the original text message, making them more secure. Decryption time for a composite cryptosystem based on AES256-Blowfish was much longer than that of a composite cryptosystem based on Blowfish-AES256, according to the experimental results.

this study compares and contrasts several symmetric key cryptographic algorithms, such as DES, 3DES, AES, and Blowfish, taking into account factors such as encryption and decryption times, entropy, memory use, throughput, avalanche effect, and energy consumption. Instead of focussing on theoretical notions, presented work has emphasised practical algorithm implementation by taking into account tradeoff performance in terms of cost of different parameters.

Based on the work of S. Varshney [6] An architecture for hardware that combines Blowfish and RC6 is suggested in this study; it has inner-loop pipelining and loop unrolling. Two random integers, "a" and "w," are used in the technique to counter the weak key attack and the known plaintext attack on Blowfish. In addition, the method in question avoids Blowfish's collision key attack by using an overlapping process that uses a single S-Box. Comparing the used algorithm to blowfish and RC6, it uses less cycles.
I. A. Landge and colleagues, [7] Embedded devices are equipped with security safeguards to protect sensitive data from threats. Encryption is used before transmission of sensitive data to ensure that no unauthorised person may access it. Secure embedded system design benefits from hardware implementation of encryption algorithms. This article discusses the implementation and analysis of the Blowfish algorithm using VHDL.

According to T. K. Hazra [8] A novel technique for the encryption and decryption of text and picture data is presented in this paper. To put the plan into action, we merge the ideas of the Blowfish algorithm with those of the Diffie Hellman algorithm. At the outset of this novel approach, a user encrypts a file using a secret key produced by the blowfish algorithm. Next, two users attempting to connect across an unsecured channel will have a shared private key produced for them using the Diffie-Hellman protocol.
A. Chauhan et al., [9] In order to improve security, this study suggests a new parallel cryptographic method that combines and modifies the MD5 and Blowfish encryption algorithms. In an effort to circumvent the limitations of both hash function and symmetric block cryptography, a hybrid MD5-Blowfish algorithm was developed.

S. A. [10] The findings are evaluated based on characteristics like storage space and time (both encryption and decryption time), and the study effort employs the hybrid cryptographic method to improve data security via the use of a cloud-based encryption algorithm. This paper demonstrates a comparison with the EDS-AES cryptography technique and combines the Blowfish and MD5 hashing algorithms.

III. CRYPTOGRAPHIC APPROACHES

With the development of security schemes over the years, many new encryption techniques have been devised, and improvements have been done on existing techniques. In general, all the existing techniques can be classified into

Asymmetric and Symmetric encryption techniques. In this paper, we will be concentrating mainly on the widely used Symmetric encryption techniques. A detailed analysis, working and the various attacks on these Symmetric and Asymmetric ciphers. Symmetric encryption techniques are further classified into Block Ciphers and Stream Ciphers. The block and stream ciphers that have been used in this paper are discussed next.

- **Stream Ciphers**

Stream Cipher algorithms peruse the entire intelligible message and convert each symbol of the plain text directly into a symbol of cipher text. The symbol is generally a bit, and the transformation performed is generally exclusive-OR (XOR). Due to bit by bit encoding, they are lighter and faster schemes relying solely on confusion concepts. They also have statistically random structures and are easier to implement on hardware.

- **Rivest Cipher 4 (RC4)**

Rivest Cipher 4 abbreviated as RC4 was developed by Ronald Rivest in 1987. It relies on a symmetric key algorithm to generate a keystream sequence for encryption and decryption. The data stream is simply XOR-ed with the generated key sequence.

- **Block Ciphers**

Block Cipher cryptographic schemes convert an entire block of plain text into a block of cipher text at a time. These are bulkier and slower ciphers as they involve the division of plain text into blocks and rely on both diffusion and confusion concepts. They have a simpler software implementation and also have different modes of operations.

The following are the advantages of Blowfish algorithm. Blowfish is –

i. One of the unbreakable algorithms available in cryptography.

ii. One of the more flexible encryption methods available.

iii. Comparatively faster algorithm among the available ones. Having high execution speed and throughput.

iv. Consumes less energy for execution as compared to other symmetric algorithms.

v. Need of minimum memory requirement.

- **Advanced Encryption Standard (AES)**

Advanced Encryption Standard or AES is a block encryption technique which was developed by Belgian cryptographers, Vincent Rijmen and Joan Daemen. It is based on the principle of substitution-permutation network, a combination of both substitution and combination. It basically comprises of 3 block ciphers- AES-128, AES-192 AES-256 and each of these ciphers can encrypt and decrypt data in 128-bit blocks using 128, 192 and 256 bit keys respectively. The higher the key size, the stronger the encryption. Since AES is a symmetric cipher, both the sender and the receiver must know the key for encryption and decryption respectively.

- **Data Encryption Standard (DES)**

Data Encryption Standard or DES was developed in 1970 by IBM. It is a block cipher that takes in 64-bit plaintext and after a series of operations, converts it into a 64-bit cipher text. DES is a symmetric cipher and uses a key for these operations of length 64-bits, out of which 56 bits are used for encryption-decryption and the remaining 8 bits are used to check parity. Thus, DES has an effective key length of 56 bits. The algorithm consists of 16 identical rounds.

- **Triple-Data Encryption Standard (3DES)**

Triple Data Encryption Standard or 3DES algorithm basically runs the DES algorithm 3 times on a given plaintext. The original DES's 56-bit key was sufficient to provide security but the availability of additional computational power led to increased brute-force attacks. This led to the development of the 3DES cipher.

- **Blowfish**

Blowfish block cipher was developed 1993 by Bruce Schneier. It uses a fixed block of size 64 bits, with a varying key-length between 32 and 448 bits. It also makes use of large key-dependent S-boxes. Similar to DES, it has a 16-round Feistel cipher structure. It is an open source algorithm which has not yet been broken.

IV. SECURITY CHALLENGES

Despite its robustness and efficiency, Blowfish encryption faces several security challenges that impact its effectiveness in modern cryptographic applications. Some of these challenges include:

1. Limited Block Size and Vulnerability to Birthday Attacks

Blowfish operates with a 64-bit block size, which was considered sufficient at the time of its development. However, with the increasing volume of encrypted data, this block size is now seen as a limitation. When large amounts of data are encrypted using the same key, the risk of birthday attacks increases, making it easier for attackers to find repeating cipher blocks and exploit them to deduce plaintext patterns.

2. Susceptibility to Weak Key Attacks

Although Blowfish supports a variable key length (32 to 448 bits), some weak keys can result in weak subkey generations, making the encryption vulnerable to cryptanalysis. Researchers have identified specific key patterns that reduce security, necessitating careful key selection to avoid predictable encryption structures.

3. High Computational Overhead in Key Expansion

Blowfish requires an extensive key expansion process that involves generating 18 subkeys and several S-box transformations before encryption can begin. This computational overhead can be significant, especially for applications requiring frequent key changes, making it less suitable for environments demanding rapid encryption operations, such as real-time secure communication.

4. Incompatibility with Modern Security Standards

Many contemporary security protocols and applications, such as TLS 1.3, prioritize encryption algorithms with 128-bit or higher block sizes, such as AES. The 64-bit block size of Blowfish makes it less compatible with modern security frameworks, reducing its applicability in high-security applications.

5. Lack of Built-in Authentication and Integrity Protection

Blowfish is purely an encryption algorithm and does not inherently provide message authentication or integrity verification. This means that additional cryptographic mechanisms, such as Message Authentication Codes (MACs) or hashing functions, must be used alongside Blowfish to ensure data integrity, increasing complexity in implementation.

6. Vulnerabilities in Certain Implementation Scenarios

While the Blowfish algorithm itself is considered secure against many traditional cryptanalysis techniques, its security depends heavily on proper implementation. Poorly implemented key management, insecure padding schemes, and incorrect cryptographic configurations can introduce vulnerabilities, making systems susceptible to attacks like side-channel attacks and ciphertext manipulation.

V. CONCLUSION

Blowfish remains a widely used cryptographic algorithm due to its efficiency, flexibility, and strong encryption capabilities. Its fast execution speed and adaptability to various applications make it suitable for securing data in resource-constrained environments. However, challenges such as its 64-bit block size, vulnerability to birthday attacks, high key expansion overhead, and lack of authentication mechanisms limit its applicability in modern security frameworks. While Blowfish continues to be a reliable choice for many encryptions needs, evolving cybersecurity threats and advancements in cryptographic techniques have led to the preference for newer algorithms like AES and Twofish. Therefore, while Blowfish is still relevant in certain scenarios, organizations must carefully evaluate its suitability based on their security requirements and the evolving landscape of data encryption technologies.

REFERENCES

1. B. M. B. Beron, V. T. Duhaylungsod, K. G. Jimenez, J. Hora, R. C. O. Calimpusan and O. Joy Gerasta, "ASIC Implementation of Pipelined Blowfish Cryptographic Core in 0.13 µm CMOS Process Technology," 2019 IEEE 11th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Laoag, Philippines, 2019, pp. 1-6, doi: 10.1109/HNICEM48295.2019.9073385.
2. S. B. Nalawade and D. H. Gawali, "Design and implementation of blowfish algorithm using reconfigurable platform," 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), Bhopal, 2017, pp. 479-484, doi: 10.1109/RISE.2017.8378204.
3. H. Setiawan and K. Rey Citra, "Design of Secure Electronic Disposition Applications by Applying Blowfish, SHA-512, and RSA Digital Signature Algorithms to Government Institution," 2018 International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), Yogyakarta, Indonesia, 2018, pp. 168-173, doi: 10.1109/ISRITI.2018.8864280.

4. M. A. Muin, M. A. Muin, A. Setyanto, Sudarmawan and K. I. Santoso, "Performance Comparison Between AES256-Blowfish and Blowfish-AES256 Combinations," 2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, 2018, pp. 137-141, doi: 10.1109/ICITACEE.2018.8576929.
5. S. Vyakaranal and S. Kengond, "Performance Analysis of Symmetric Key Cryptographic Algorithms," 2018 International Conference on Communication and Signal Processing (ICCSP), Chennai, 2018, pp. 0411-0415, doi: 10.1109/ICCSP.2018.8524373.
6. S. Varshney, T. Sudarshan and S. Khare, "Efficient Hardware Architecture for Amalgam of Blowfish and Rc6," 2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, 2017, pp. 1126-1130, doi: 10.1109/CTCEEC.2017.8455189.
7. I. A. Landge and B. K. Mishra, "VHDL based BLOWFISH implementation for secured Embedded System design," 2017 Third International Conference on Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), Chennai, 2017, pp. 497-501, doi: 10.1109/AEEICB.2017.7972363.
8. T. K. Hazra, A. Mahato, A. Mandal and A. K. Chakraborty, "A hybrid cryptosystem of image and text files using blowfish and Diffie-Hellman techniques," 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), Bangkok, 2017, pp. 137-141, doi: 10.1109/IEMECON.2017.8079577.
9. A. Chauhan and J. Gupta, "A novel technique of cloud security based on hybrid encryption by Blowfish and MD5," 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, 2017, pp. 349-355, doi: 10.1109/ISPCC.2017.8269702.
10. A. Gaur, A. Jain and A. Verma, "Analyzing storage and time delay by hybrid Blowfish-Md5 technique," 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai, 2017, pp. 2985-2990, doi: 10.1109/ICECDS.2017.8390003.