International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 14, Issue 3, March 2025)

Cryptography and Blowfish-Based Security Approaches for FPGA Applications: A Comprehensive Review

Ravikant Prasad[1], Prof. Suresh S. Gawande[2]
[1]Research Scholar, Dept. of Electronics & Communication Eng., Bhabha Engineering Reasearch Institute, Bhabha University Bhopal, India
[2]Head of Department, Dept. of Electronics & Communication Eng., Bhabha Engineering Reasearch Institute, Bhabha University Bhopal, India

***Abstract*— Cryptography plays a pivotal role in securing sensitive data, especially in embedded systems such as FPGA (Field-Programmable Gate Array) applications. The increasing demand for high-performance, energy-efficient, and scalable security solutions in various domains, such as IoT, medical devices, and industrial control systems, calls for innovative cryptographic approaches. Among the numerous cryptographic algorithms, Blowfish stands out due to its efficient design, suitability for hardware implementation, and robustness against attacks. This comprehensive review focuses on the application of Blowfish-based cryptographic methods for FPGA platforms, analyzing their advantages, challenges, and optimizations. The paper delves into the implementation details, performance metrics, and security features of Blowfish in FPGA systems, exploring its application in data encryption, authentication, and integrity verification. Furthermore, the review highlights various modifications and improvements to the basic Blowfish algorithm to enhance its efficiency in FPGA-based applications. By examining the synergy between Blowfish and FPGA, this review provides valuable insights into achieving secure, high-performance cryptographic solutions for embedded systems.**

Keywords*— *VLSI, FPGA, Blowfish, Encryption, Security, Privacy.

I. INTRODUCTION

Cryptography is an essential aspect of modern computing, ensuring the confidentiality, integrity, and authenticity of data transmitted over insecure networks or stored in vulnerable devices. As the world becomes increasingly connected through the Internet of Things (IoT), smart devices, and other embedded systems, the demand for robust and efficient cryptographic solutions has surged. Among the different cryptographic techniques available, symmetric key algorithms, such as Blowfish, have garnered significant attention due to their speed, security, and ease of implementation. These features make Blowfish an ideal candidate for resource-constrained systems like FPGA (Field-Programmable Gate Array) applications, where both performance and resource utilization are of paramount importance.

FPGA-based cryptographic systems leverage the parallel processing capabilities of these reconfigurable hardware platforms to achieve high-speed encryption and decryption, making them particularly attractive for real-time applications. The ability to customize the hardware architecture for specific cryptographic operations allows FPGA implementations to outperform traditional software-based solutions in terms of throughput, latency, and energy efficiency. However, despite its potential advantages, implementing cryptographic algorithms such as Blowfish on FPGAs presents certain challenges related to design complexity, hardware resource optimization, and security threats such as side-channel attacks.

Blowfish, originally designed by Bruce Schneier in 1993, is a symmetric block cipher that is widely recognized for its simplicity, speed, and strong security properties. It operates on 64-bit blocks and supports variable-length keys, making it highly adaptable to different security requirements. Due to its low hardware overhead, Blowfish is particularly well-suited for FPGA implementations, where limited resources such as logic gates and memory must be utilized efficiently. Moreover, the algorithm's flexibility in terms of key length and the number of rounds offers a balance between security and performance, which is crucial for embedded systems that require both strong encryption and low resource consumption.

This paper presents a comprehensive review of Blowfish-based cryptographic approaches for FPGA applications. It explores the underlying principles of Blowfish, its suitability for hardware implementation, and the various methods used to optimize its performance on FPGA platforms. The review also examines the trade-offs involved in implementing Blowfish on FPGAs, including the impact on area, speed, and power consumption. Furthermore, it discusses several enhancements and variations of the Blowfish algorithm that have been proposed to address the limitations of its original design and improve its efficiency and security in FPGA environments.

By providing a detailed analysis of the state-of-the-art Blowfish-based FPGA implementations, this review aims to contribute to the development of secure, efficient, and scalable cryptographic solutions for embedded systems. It also identifies potential areas for future research, such as the integration of Blowfish with other cryptographic algorithms and the exploration of novel hardware optimization techniques to further enhance the performance of FPGA-based cryptographic systems.

II. LITERATURE SURVEY

P. Palka et al., [1] Cryptography is a fundamental subject for contemporary digital systems. Over the last twenty years, FPGA-based encryptions have shown efficacy. This research investigates a cryptographic use of FPGAs by implementing the Blowfish symmetric-key block cypher on a Xilinx ZedBoard. The objective is to investigate the use of FPGAs for enhancing the performance of a symmetric-key block cypher. We want to evaluate the performance of a hardware implementation of Blowfish against a software counterpart. The Blowfish block cypher is executed in VHDL and has been functionally validated against a Python reference model.

B. M. B. Beron et al. [2] The cost of a data breach is very concerning, and software-based solutions for tackling information security may be insufficient. This research was initiated due to the necessity for hardware-level cryptography, focussing on the ASIC implementation of a Blowfish cryptographic core using 0.13 µm CMOS process technology, alongside the application of pipelining and datapath modifications to the Blowfish algorithm to reduce propagation delay and enhance performance.

H. Setiawan et al. [3] This study will develop an electronic secure disposition application in compliance with the Regulation of the Minister of State for Administrative Reform and Bureaucratic Reform number 6 of 2011, addressing manual disposition issues in Government Institutions. The program will use the Blowfish technique for encryption. and digital signatures using SHA-512 hash algorithms and RSA digital signatures in the accompanying document.

M. A. Muin et al. [4] evaluate the study based on the time necessary to decipher the ciphertext. Extended decryption duration results in an increased time required to execute a brute force attack to get the original text message, hence enhancing security. The experimental results indicate that a composite cryptosystem using AES256 followed by Blowfish necessitates a higher decryption duration compared to the composite cryptosystem arranged in reverse order (Blowfish followed by AES256).

S. Vyakaranal et al. [5] The proposed study examines several symmetric key cryptographic algorithms, including DES, 3DES, AES, and Blowfish, by evaluating encryption time, decryption time, entropy, memory utilisation, throughput, avalanche effect, and energy consumption via practical implementation in Java. The suggested study emphasises the practical implementation of algorithms, focussing on performance trade-offs related to the costs of different parameters rather than just on theoretical principles.

S. Varshney et al. [6] This paper proposes a hardware architecture with inner-loop pipelining and loop unrolling for the integration of Blowfish and RC6. The used approach utilises two random values, "a" and "w," which assist in mitigating weak key attacks and known plaintext attacks on Blowfish. The used approach utilises a single S-Box via an overlapping procedure that mitigates the collision key attack associated with Blowfish. The used method needs less cycles than Blowfish and RC6.

I. A. Landge et al. [7] Embedded devices are equipped with integrated security mechanisms to safeguard sensitive data from threats. The sensitive data is encrypted prior to transmission, ensuring that only authorised users may access this information. The hardware implementation of encryption algorithms is beneficial for building safe embedded systems. This article discusses the implementation and analysis of the Blowfish algorithm based on VHDL.

T.K. Hazra and colleagues, [8] This study presents a novel technique for the encryption and decryption of pictures and text data. The suggested solution integrates the principles of the Diffie-Hellman algorithm with the Blowfish algorithm. Initially, a computer user will encrypt a file with a secret key

created by the Blowfish method. Subsequently, the Diffie-Hellman protocol will facilitate the generation of a shared private key for two computer users attempting to communicate via an unsecured channel.

A. Chauhan et al. [9] This study introduces an innovative parallel cryptographic method that integrates and modifies the MD5 and Blowfish encryption algorithms to enhance security. A hybrid MD5-Blowfish cryptographic algorithm is developed to address the vulnerabilities of symmetric block cryptography and hash function methodologies.

A. Gaur et al. [10] This study employs a hybrid cryptographic algorithm to augment data security using a cloud-based encryption method, with outcomes evaluated based on characteristics such as storage capacity and time (including both encryption and decryption durations). This study presents the integration of the Blowfish algorithm with the MD5 hashing technique, with a comparative analysis with the EDS-AES cryptographic algorithm.

R. Ahmad et al. [11] The suggested memory-centric approach aims to enhance the efficacy of Blowfish. The performance is evaluated based on its architecture, throughput, and power consumption. The findings indicate that the suggested Blowfish decreases slice utilisation by 63% and improves throughput by 29% while maintaining low power consumption.

V. C. Dongre and colleagues,[12] This study presents a random network coding strategy combined with the Blowfish encryption algorithm to ensure source anonymity and hide message contents. This technique improves the characteristics of homomorphic encryption, namely Paillier encryption, to successfully prevent traffic analysis attacks. The suggested system has the capability of random coding. The characteristic of inverting GEVs with a high probability is used by each sink to retrieve the source packets. The efficacy and robustness of the proposed system are shown via theoretical analysis and simulation data evaluation.

III. CRYPTOGRAPHIC APPROACHES

Cryptographic techniques are fundamental in safeguarding sensitive data against unauthorized access, manipulation, or tampering, especially in embedded systems and resource-constrained environments. With the advent of the Internet of Things (IoT), medical devices, and other IoT-based applications, the need for lightweight yet secure cryptographic solutions has increased dramatically. In this section, we explore various cryptographic approaches that are applicable to FPGA (Field-Programmable Gate Array) applications, with a specific focus on the symmetric-key cryptographic algorithms, and how they can be implemented efficiently in hardware for high-performance security solutions.

Symmetric vs Asymmetric Cryptography

Cryptographic algorithms can generally be classified into two categories: symmetric-key and asymmetric-key cryptography.

1. **Symmetric-key cryptography** involves the use of the same key for both encryption and decryption. Since the key is shared between the sender and the receiver, the primary challenge lies in securely distributing and managing this key. Symmetric-key algorithms, such as AES (Advanced Encryption Standard), DES (Data Encryption Standard), and Blowfish, are generally faster and less resource-intensive compared to their asymmetric counterparts. This makes symmetric-key cryptography ideal for use in embedded systems, where both speed and efficiency are paramount.
2. **Asymmetric-key cryptography** uses a pair of keys: a public key for encryption and a private key for decryption. Asymmetric algorithms, such as RSA (Rivest-Shamir-Adleman) and ECC (Elliptic Curve Cryptography), offer higher security levels and are often used for key exchange, digital signatures, and authentication. However, these algorithms are computationally expensive and may not be suitable for resource-constrained FPGA applications due to their relatively high latency and memory requirements.

Overview of Blowfish Algorithm

Blowfish is a symmetric-key block cipher designed by Bruce Schneier in 1993. It operates on 64-bit data blocks and supports variable-length keys, ranging from 32 bits to 448 bits. The algorithm's design is highly efficient, with the key expansion and encryption functions optimized for speed. Blowfish is composed of two main components: the key scheduling algorithm (which generates a set of subkeys from

the main encryption key) and the encryption process (which applies multiple rounds of operations to the data block).

Blowfish's design makes it particularly suitable for hardware implementations on FPGAs. The algorithm requires fewer hardware resources, such as logic gates and memory, compared to other block ciphers like AES, and its performance can be enhanced by exploiting FPGA's parallel processing capabilities. Moreover, Blowfish's versatility in key length allows for scalability, ensuring it can be adapted to various security needs.

Key Design Features of Blowfish

Several key design features contribute to the efficiency and security of Blowfish:

1. **Key Expansion**: Blowfish performs a key expansion process that generates 4168 bytes of subkey data from the original key. This process is computationally intensive but can be parallelized on FPGAs, leading to enhanced performance.
2. **Feistel Network**: Blowfish employs a Feistel network structure, where the data block is divided into two halves, and each half undergoes a series of transformations involving bitwise operations, substitution, and permutation. This structure is highly suitable for parallel hardware implementation, as each round of transformation can be implemented in separate processing units.
3. **Variable Key Length**: The ability to use a variable-length key (from 32 to 448 bits) allows Blowfish to offer flexibility in terms of security. Longer keys provide a higher level of security, while shorter keys can improve performance for less security-critical applications.
4. **Rounds**: Blowfish operates with 16 rounds of encryption, and the number of rounds can be adjusted to balance security and performance. FPGA implementations can optimize the number of rounds to match the specific performance requirements of the application.

Blowfish Optimizations for FPGA

Implementing Blowfish on FPGA can lead to significant performance improvements, but certain optimizations must be considered to enhance its efficiency and scalability. These optimizations include:

1. **Pipelining**: By dividing the Blowfish encryption process into smaller stages and implementing pipelining, multiple encryption operations can be processed simultaneously. This reduces the overall processing time and increases throughput.
2. **Parallelization of Key Expansion**: Blowfish’s key expansion process, which generates the subkeys, can be parallelized to speed up the overall encryption process. This is particularly beneficial in FPGA implementations, where parallel execution is a key strength.
3. **Resource Optimization**: FPGAs have limited resources, so optimizing the use of logic gates, memory, and interconnects is essential. Blowfish’s relatively simple design allows it to be implemented with minimal hardware resources, but optimizing memory usage and ensuring efficient interconnection between components can further improve performance.
4. **Custom Hardware Units**: Custom hardware units, such as specialized S-box units, can be designed to accelerate the substitution and permutation operations in the Blowfish algorithm. These hardware accelerators can be optimized to match the FPGA's architecture, providing significant performance gains.

Blowfish-based cryptographic approaches are a promising solution for FPGA applications due to their simplicity, efficiency, and adaptability. Optimizing Blowfish for FPGA, through techniques such as pipelining, parallelization, and custom hardware accelerators, can result in highly efficient cryptographic implementations suitable for a wide range of applications. In the following sections, we will explore the various FPGA-based Blowfish implementations, their performance metrics, and potential enhancements for further optimization.

IV. CHALLENGES

Implementing the Blowfish cryptographic algorithm on FPGA platforms presents several challenges that need to be addressed to achieve optimal performance and security. Below are eight key challenges encountered in the process:

1. **Resource Constraints**: FPGAs have limited resources such as logic gates, memory, and interconnects. Efficiently utilizing these resources while maintaining high encryption speeds and low latency is a major challenge. Blowfish's key expansion and multiple rounds of encryption can consume significant FPGA resources, requiring careful optimization.
2. **Power Consumption**: While FPGAs offer the advantage of low-latency encryption, power consumption can become a concern, especially in resource-constrained embedded systems or battery-powered devices. Optimizing power efficiency without compromising performance remains a key challenge in FPGA-based Blowfish implementations.
3. **Design Complexity**: Implementing Blowfish on an FPGA requires designing efficient hardware architectures for key expansion, encryption rounds, and S-box operations. The complexity of the design increases as optimizations like pipelining and parallelization are introduced to improve performance, demanding high expertise in hardware design.
4. **Side-Channel Attacks**: FPGAs are vulnerable to side-channel attacks, such as timing, power, and electromagnetic analysis. Protecting the Blowfish implementation from these types of attacks, while maintaining its performance and efficiency, is a critical challenge in hardware cryptographic implementations.
5. **Latency Issues**: Even though FPGAs are known for their low-latency operations, Blowfish's multiple rounds of encryption can still introduce delays, particularly in real-time applications. Reducing latency without sacrificing security is a complex optimization problem, especially when balancing the number of rounds and parallelism.
6. **Scalability**: Blowfish allows for variable key sizes, which offers flexibility in encryption strength. However, scaling the algorithm to support larger key sizes and adapting it for higher security levels in FPGA-based systems can lead to increased resource consumption and design complexity.

V. CONCLUSION

Blowfish-based cryptographic approaches offer an efficient and flexible solution for FPGA applications, particularly in terms of speed, security, and low resource consumption, there are significant challenges to overcome. These include managing limited FPGA resources, optimizing power consumption, preventing side-channel attacks, minimizing latency, ensuring scalability, and improving the key expansion process. Despite these challenges, with careful design and optimization techniques such as pipelining, parallelization, and hardware-specific enhancements, FPGA-based Blowfish implementations can provide a robust and high-performance cryptographic solution suitable for a wide range of embedded and real-time security applications.

REFERENCES

1. P. Palka, R. A. Perez, T. Fang and J. Saniie, "Design Flow of Blowfish Symmetric-Key Block Cipher on FPGA," 2022 IEEE International Conference on Electro Information Technology (eIT), Mankato, MN, USA, 2022, pp. 193-197, doi: 10.1109/eIT53891.2022.9814070.
2. 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.
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.
11. R. Ahmad[1], A. A. Manaf and W. Ismail, "Development of an improved power-throughput Blowfish algorithm on FPGA," 2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA), Malacca City, 2016, pp. 237-241, doi: 10.1109/CSPA.2016.7515838.
12. V. C. Dongre and S. G. Shikalpure, "Ensuring privacy preservation in wireless networks against traffic analysis by employing network coding and Blowfish encryption," 2016 International Conference on Signal and Information Processing (IConSIP), Vishnupuri, 2016, pp. 1-5, doi: 10.1109/ICONSIP.2016.7857442.