A NOVEL LIGHTWEIGHT SYMMETRIC ENCRYPTION AND DECRYPTION HARDWARE FOR CLOUD SECURITY

Priyanka Ram
Department of Electronics and Communication
National Institute of Technology Kurukshetra
Haryana, India
Email: priyankaram1702@gmail.com

Dr. Vikas Mittal
Department of Electronics and Communication
National Institute of Technology Kurukshetra
Haryana, India
Email: vikas$_m ittal@nitkkr.ac.in$

2025 International Conference on Electronics, AI and Computing (EAIC) | 979-8-3315-5358-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/EAIC66483.2025.11101291

***Abstract*—The increasing reliance on cloud servers for data storage and processing, with ensured secure data transmission has become a critical challenge. While significant efforts focus on mitigating external threats, internal vulnerabilities remain a substantial risk. This paper proposes a lightweight symmetric encryption framework aimed at enhancing data security in cloud environments. The design integrates three layers of encryption to achieve robust protection: symmetric key-based XOR encryption, salt-based randomness, and a data-reversing operation. The encryption process begins by XORing the plaintext with a dynamically generated 8-bit symmetric key, followed by appending a 4-bit salt value to introduce randomness. A final reversing operation is applied to the encrypted data to add an additional layer of obfuscation, strengthening resistance to potential attacks. The decryption process sequentially reverses these steps: the data is restored to its original order, the salt is removed, and XOR is applied with the symmetric key to retrieve the plain text. The proposed method is implemented in Verilog HDL, with validation conducted through a test bench designed to process text data for encryption and decryption. The test bench demonstrates accurate functionality for a pre-decided limited maximum limit on the number of letters that a word can contain . A comparative evaluation highlights the superior security and performance of the proposed algorithm against vis-a-vis. This study underscores the effectiveness of hardware-based symmetric cryptography in addressing both internal and external threats to cloud server security, offering a reliable solution for secure data exchange.**

***Index Terms*—Hardware encryption , symmetric encryption, verilog, cryptography,clouds security, symmetric algorithm, Encryption, decryption, asymmetric algorithm.**

I. Introduction

The rapid advancement of cloud computing has revolutionized how organizations manage, store, and transfer data. Cloud servers provide scalable and efficient solutions for handling extensive datasets, but they also present significant security risks. While efforts are often directed toward protecting cloud infrastructures from external threats, research indicates that internal vulnerabilities, such as unauthorized insider access or compromised credentials, pose a significant risk. Therefore, there is a growing demand for effective encryption methods to secure data across all stages of cloud communication.

Symmetric cryptography is widely regarded as an efficient solution for secure data transmission due to its simplicity and computational speed as shown figure 1. This paper introduces a hardware-based symmetric encryption method designed to address security challenges in cloud environments. The proposed framework incorporates three key elements: XOR-based encryption using a symmetric key, a salt-generation technique to enhance randomness, and a reversing operation to further obscure the data. Combined, these mechanisms ensure robust protection against advanced cryptographic attacks.

The encryption process begins with XORing plain text characters with a dynamically generated 8-bit symmetric key, followed by appending a 4-bit salt for added randomness. A reversing operation is then applied to the encrypted data to bolster its resistance to brute-force and pattern-based attacks. Decryption involves reversing these steps to restore the original plain text.

The framework is implemented in Verilog HDL and tested using a comprehensive testbench. Simulations confirm its efficiency and reliability for encrypting and decrypting short words of up to six characters. This paper demonstrates the effectiveness of hardware-based cryptographic solutions in enhancing the security of cloud data exchange.

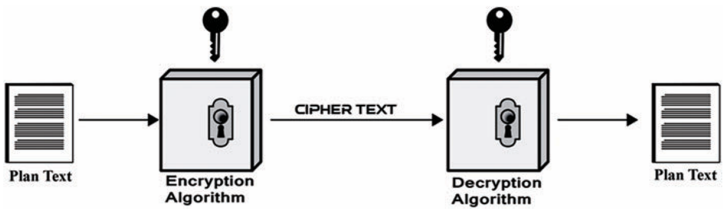

Fig. 1. Encryption.

II. Literature Review

In [1], the authors used an encryption function that is homomorphic to reduce the forward node complexity using a distributed privacy protection based on random linear network coding for the application in smart grids. This approach suggested data privacy and secrecy maintaining characteristics.

Afterwards they conducted several tests and security analyses and concluded that the suggested approach might successfully protect privacy.

In [2], the authors presented another location data protection approach using a multi-level query tree structure for generating such database. This approach was shown better in accessing data and preserving privacy as compared to other privacy measures.

In [3], the authors explained numerous security methods and algorithms. There they highlighted two major research challenges viz. user authentication and full implementations holes in cryptography as identified by them. However the first gap that they identified was the authentication process and the second gap as lack of full implementations. Alongside, they presented benefits and drawbacks of several cryptography techniques for the consideration in the future studies.

In [4], the authors addressed the difficulties that were encountered in handling multi-view data using hashing using discrete multi-graph hashing (DMGH) technique with reduced distortion errors. This method was shown to outperform other multiview hashing techniques based on extensive trials on huge datasets.

In [5], the authors presented an encryption algorithm that encrypted the file prior to its uploading on the client-side cloud. The file was encrypted by using the same keys once it was downloaded. Initially, a website was developed to post a file, followed by an encrypted file. The keys were then uploaded along with this encrypted file. The encrypted file was then decoded after being downloaded from local storage. This technique works well for encrypting text files.

In [6], the authors presented a number of cloud company encryption techniques. The authors showed an encryption technique to encrypt data and defend against man-in-the-middle attacks. Prior to being transformed to binary, the data is first translated to ASCII values numbers to protect the information. The binary number was translated to ASCII once more after adding 0 for full bits and taking the 1's complement of the final four bits if the binary number lacked eight bits. The data was decrypted by first converting ASCII value to binary, then reversed the final four binary bits, and finally converted back to ASCII.

All of the above algorithms designed to protect the data have been discussed by various researchers. Researchers disscused key gap of the cryptography methods. However, several researchers have compared symmetric and asymmetric cryptography to assess the algorithm's efficiency. To protect data againts attacks, multiple researchers converted the data into bits and reversed the data. To execute the decryption technique, some researchers provided the data encrypted with a special key to the recipient. Basic data security can be achieved with the use of all these measures, however these techniques make it difficult to transmit data securely and efficiently, and they cannot completely protect cloud data. So that this paper is different from any other paper. In this paper a lightweight symmertric encryption and decryption algorithm has been developed in hardware using verilog VHDL for cloude security, which will support data security in two stages. There will be two layer cryptography for encryption and decryption. By this algorithm, it will be difficult for the attecker to decrypt and find the original plain text. By using methods like bit conversion, bit reversal, key generation, bit salting, and bit XORtion for data encryption, a comprehensive encryption security mechanism will be created with pre-decided maximum limit on the number of letters that a word can contain.These mechanisms will enable the data to be encrypted and decrypted simultaneously when all of them have been applied.

III. METHODOLOGY

A. *Encryption process*

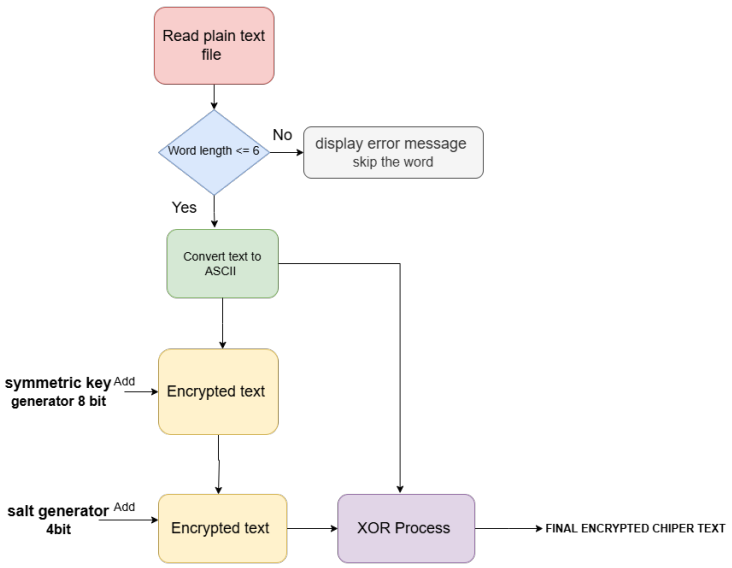

Fig. 2. FlowChart.

The objective is to develop a program that provides user text files by focusing on plain text characters and handling words containing up to six letters as shown in figure 2. Implementation involves creating a test bench that simulates the encryption process. This test bench reads characters from the file, stores them in an array, and excludes words longer than six letters from processing. For each valid word, the program converts it into ASCII format, applies an encryption method, and displays the intermediate steps along with the final encrypted result after converting plain text characters into numeric values for further processing they are mapped to their corresponding ASCII codes. This is achieved through the implementation of a module called plain-ascii, which maps each 8-bit input character directly to its equivalent 8-bit ASCII code. The module streamlines text processing by converting input characters into machine-readable numeric values, facilitating subsequent operations. Using an 8-bit Linear Feedback Shift Register (LFSR) to create a symmetric key for encryption. This is implemented through a module called symmetric-key generator, which produces an 8-bit symmetric key based on the LFSR's state. The LFSR is initialized with a

predefined seed and utilizes a feedback mechanism involving XOR operations on specific bits. With each clock cycle, the key is dynamically updated, providing increased security and unpredictability. The objective is to improve the randomness of encrypted data by incorporating a lightweight and dynamic salt value. This is achieved through a salt-generating module that produces a 4-bit salt using a straightforward feedback mechanism within a shift register. The generated salt is appended to the encrypted data, introducing variability and increasing resistance to potential attacks. In addition, encrypt ASCII-encoded data using a combination of a symmetric key and a salt value. For each character in plain text, the ASCII code is encrypted by performing an XOR operation with the dynamically generated symmetric key. The resulting 8-bit encrypted code is combined with a 4-bit salt, creating a final encrypted value of 12 bits. To ensure that the encryption system functions correctly, its implementation is validated through a structured process. A clock signal is used to drive the key and salt generation modules. The test bench outputs detailed information for each plain text character, including its ASCII value, the generated symmetric key, the intermediate encrypted value, the appended salt, and the final encrypted code. These results is shown in figure 4 are displayed for every character processed, facilitating thorough verification of the encryption process.

B. Decryption Process

First, the encrypted bits were inverted to decrypt the cipher text. Afterwards salt values with 4 bits were eliminated from the reversed bits' ends. Then they were XORed using the generated key during the encryption process. Thus, both the information and key were transmitted together accessible only to the authorized users. The binary values derived from XOR were divided into 8-bits each followed by their equivalent decimal values. ASCII value corresponding to each decimal value is then determined to be combined to produce plain text as shown in figure 4.

IV. RESULTS AND DISCUSSION

None of the above methods has been used as a two-way encryption algorithm for securing the data alongwith making its decryption difficult for attackers had they been used. However, in this paper the two-layered implementation would make it challenging for hackers to attack and decode the data even if they are able to devise a decryption mechanism since different mechanisms have been used.

Verilog HDL simulations were used to implement and evaluate the suggested symmetric cryptography-based encryption methodology for cloud security. The outcomes shows that by combining XOR-based encryption with a dynamic salt generation mechanism, the encryption process successfully secures textual data and simulation result of RTL wave as shown in figure 4 and figure 5. Figure 3 illustrates how a plain text file with a word limit is regarded for encryption. This two-layer encryption approach will use Verilog VHDL code for hardware to encrypt the text "PLAIN TEXT ENCRYPTION" that contained in the user text file. The XOR operation ensures effective plain text data when combined with a dynamically generated symmetric key. The addition of a salt value enhances the randomness of the cipher text, making it resilient to brute-force and pattern-based attacks.

The Linear Feedback Shift Register (LFSR) and salt generation have been used to make it more random and also to make the implementation lightweight. The approach is compatible with a restricted number of hardware configurations because it is optimized for processing words with up to six characters. Figure 4 illustrates how the original plaintext was recovered after decryption, confirming that the encryption and decryption procedures worked as intended.

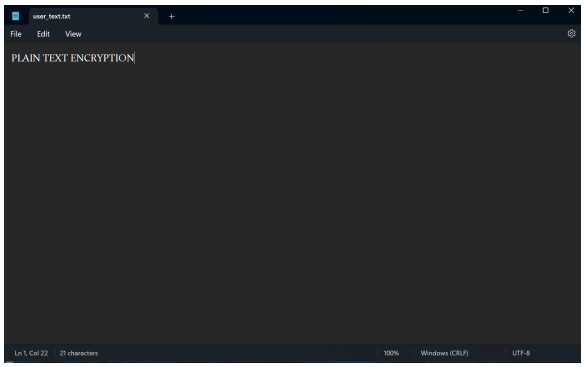

Fig. 3. User Text File

V. CONCLUSION

In this paper, an effective encryption technique is presented to protect cloud server data. The system combines salt generation, XOR-based symmetric encryption, and a lightweight hardware-oriented methodology to achieve a strong harmony between computational efficiency and security. By ensuring secure data exchange, the implementation reduces the possibility of illegal access to cloud servers. Applications requiring hardware-based, lightweight cryptographic solutions can benefit from the additional security offered by the combination of XOR encryption and salt addition. The method shows that it is feasible to replace more resource-intensive cryptographic algorithms with scalable and secure encryption processes for cloud-based applications.

Despite its strengths, the proposed method has some limitations. The encryption system is currently limited to processing words with a maximum length of six characters. This constraint may limit its applicability to certain types of data.The 4-bit salt value provides a relatively small degree of randomness, which could be insufficient for highly sensitive data that require advanced encryption standards.While the design is efficient for small-scale hardware systems, its scalability to larger datasets or real-time streaming data requires further investigation.Symmetric encryption relies on secure key distribution, which is not addressed in this work. Ensuring key exchange and management in real-world applications remains a challenge.

```
Encryption - Character: P, ASCII Code:  80, Symmetric Key: 10101010, Encrypted Code: 11111010, Salt: 1010, Final Encrypted Code: 111110101010
Encryption - Character: L, ASCII Code:  76, Symmetric Key: 10101010, Encrypted Code: 11100110, Salt: 1010, Final Encrypted Code: 111001101010
Encryption - Character: A, ASCII Code:  65, Symmetric Key: 10101010, Encrypted Code: 11101011, Salt: 1010, Final Encrypted Code: 111010111010
Encryption - Character: I, ASCII Code:  73, Symmetric Key: 10101010, Encrypted Code: 11100011, Salt: 1010, Final Encrypted Code: 111000111010
Encryption - Character: N, ASCII Code:  78, Symmetric Key: 10101010, Encrypted Code: 11100100, Salt: 1010, Final Encrypted Code: 111001001010
Decryption - Encrypted Code: 11111010, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  80, Character: P
Decryption - Encrypted Code: 11100110, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  76, Character: L
Decryption - Encrypted Code: 11101011, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  65, Character: A
Decryption - Encrypted Code: 11100011, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  73, Character: I
Decryption - Encrypted Code: 11100100, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  78, Character: N
Encryption - Character: T, ASCII Code:  84, Symmetric Key: 10101010, Encrypted Code: 11111110, Salt: 1010, Final Encrypted Code: 111111101010
Encryption - Character: E, ASCII Code:  69, Symmetric Key: 10101010, Encrypted Code: 11101111, Salt: 1010, Final Encrypted Code: 111011111010
Encryption - Character: X, ASCII Code:  88, Symmetric Key: 10101010, Encrypted Code: 11110010, Salt: 1010, Final Encrypted Code: 111100101010
Encryption - Character: T, ASCII Code:  84, Symmetric Key: 10101010, Encrypted Code: 11111110, Salt: 1010, Final Encrypted Code: 111111101010
Decryption - Encrypted Code: 11111110, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  84, Character: T
Decryption - Encrypted Code: 11101111, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  69, Character: E
Decryption - Encrypted Code: 11110010, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  88, Character: X
Decryption - Encrypted Code: 11111110, Salt: 1010, Symmetric Key: 10101010, Decrypted ASCII Code:  84, Character: T
Error: This word has more than six letters and will not be encrypted.
```

Fig. 4. Encryption and Decryption results

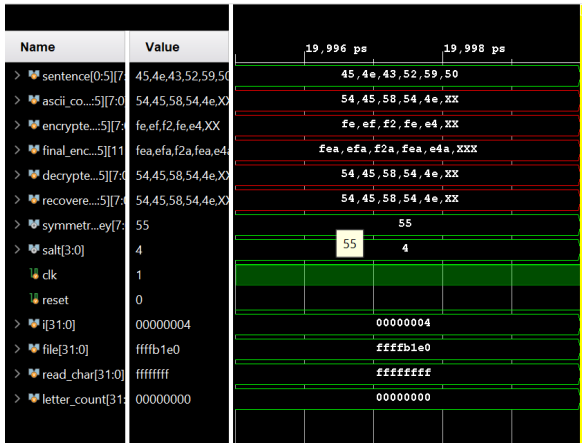

Fig. 5. Simulation result RTL wave

A. *Futurework*

Future research will address these limitations by extending the methodology to handle longer data sequences. Exploring alternative salt generation techniques to enhance randomness.Additionally, an asymmetric encryption mechanism will be employed by improving salt randomness and utilizing larger bit sizes to increase security. Consideration will also be given to using longer word lengths to strengthen the encryption process. Investigating the integration of dynamic key management protocols for secure deployment in real-world cloud environments.

REFERENCES

[1] S.M.He,W.N.Zeng,K.Xie,H.M.Yang,M.Y.Laietal.,"PPNC:Privacy preserving scheme for random linear network coding in smart grid," KSII Transactions on Internet and Information Systems, vol. 11, no. 3, pp. 1510–1532, 2017.

[2] K. Gu, L. H. Yang and B. Yin, "Location data record privacy protection based on differential privacy mechanism,"Information Technology and Control, vol. 47, no. 4, pp. 639–654, 2018.

[3] S. A. Ahmad and A. B. Garko, "Hybrid cryptography algorithms in cloud computing: A review,"in 2019 15th Int. Conf. on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, pp. 1–6, 2019.

[4] L.Y.Xiang,X.B.Shen,J.H.QinandW.Hao,"Discrete multi-graph hashing for large-scale visual search," Neural Processing Letters, vol. 49, no. 3, pp. 1055–1069, 2019.

[5] A.MusaandA.Mahmood,"Client-side cryptography based security for cloud computing system,"in 2021 Int. Conf. on Artificial Intelligence and Smart Systems (ICAIS), India, pp. 594–600, 2021. .

[6] F. Bentil and I. Lartey, "Cloud cryptography A security aspect," International Journal of Engineering Research Technology (IJERT),vol. 10, no. 5, pp. 448–450, 2021.

[7] Z. Xu, C. Xu, J. Xu and X. Meng, "A computationally efficient authentication and key agreement scheme for multi-server switching in WBAN,"International Journal of Sensor Networks, vol. 35, no. 3, pp. 143–160, 2021

[8] Nadeem, Muhammad, et al. "Two Layer Symmetric Cryptography Algorithm for Protecting Data from Attacks." Computers, Materials Continua 74.2 (2023).

[9] Swayamprakash, B., Shreshta, P. J., Reddy, C. M., Jamal, K., Mannem, K., Suneetha, M. (2023, April). Design of Advanced Encryption Standard using Verilog HDL. In 2023 7th International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 676-682). IEEE.

[10] Srinivasan, K., et al. "A VLSI Perspective on Encryption Algorithm Analysis." 2024 International Conference on Communication, Computing and Internet of Things (IC3IoT). IEEE, 2024.

[11] Reddy, Mahesh, and Kameswara Rao. "A Lightweight Symmetric Cryptography based User Authentication Protocol for IoT based Applications." Scalable Computing: Practice and Experience 25.3 (2024): 1647-1657.

[12] Cheng, Hao, et al. "RISC-V instruction set extensions for lightweight symmetric cryptography." IACR Transactions on Cryptographic Hardware and Embedded Systems (2023): 193-237.

[13] Gunathilake, Nilupulee A., Ahmed Al-Dubai, and William J. Buchana. "Recent advances and trends in lightweight cryptography for IoT security." 2020 16th International Conference on Network and Service Management (CNSM). IEEE, 2020.

[14] Tan, Quan Quan. "Cryptanalysis of lightweight symmetric-key cryptographic algorithms." (2023).

[15] Teoh, Yuan Ju, et al. "A Survey to Lightweight Cryptography World." 2024 International Conference on Green Energy, Computing and Sustainable Technology (GECOST). IEEE, 2024.