Enhanced Security Protocol for VLSI Systems: Modified AES Algorithm for Robust Data Transmission

1st Jinsha Lawrence
Department of Computer Science and Engineering
Faculty of Engineering
Karpagam Academy of Higher Education (Deemed to be University)
Coimbatore, India
jinshalarence@gmail.com

2nd Dinesh Kumar Budagam
Department of MS in Software Engineering
Security Architect at VISA
California, United States
getbdinesh@gmail.com

3rd P. Mukilan
Department of Electronics and Communication Engineering
Dhanalakshmi Srinivasan College of Engineering
Coimbatore, India
venmukilan@yahoo.co.in

4th Kavitha Veerappan
Department of Electronics and Communication Engineering
Kongu Engineering College
Erode, India
mail2kavieceian@gmail.com

5th E. Chandrasekhar
Department of Electronics and Communication Engineering
Chaitanya Bharathi Institute of Technology (A)
Hyderabad, India
chandrasekhar404@gmail.com

6th K. Barathi
Department of Electrical and Electronics Engineering
Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
Chennai, India
barathi@veltech.edu.in

2025 3rd International Conference on Integrated Circuits and Communication Systems (ICICACS) | 979-8-3315-0845-6/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICICACS65178.2025.10968738

***Abstract*—The development of compact and efficient devices has been made possible by the growth of Very Large Scale Integration (VLSI) technologies, which has transformed modern electronics. However, there are cautions regarding the security of data, especially when it comes to transmission due to the extensive usage of technology. To address these, this study proposes a modified Advanced Encryption Standard (AES) algorithm as a solution for improved security protocol in VLSI systems. The proposed algorithm are intended to support the encryption procedure, increasing stronger production against cryptographic attacks while preserving the system's effectiveness and speed factors that are critical aspects for VLSI applications. Through extensive simulations and testing, the modified AES algorithm demonstrated significant security enhancements the operational efficiency of the VLSI system. The attained outcomes of the proposed work shows that the Modified AES strategy provides a workable technique to protect data in VLSI-based devices while maintaining data communication integrity and confidentiality. The proposed modified AES algorithm demonstrates a significant improvement in performance with a propagation delay of 7 ns, power consumption of 29 mW, and a computational overhead of 62 bits, leading to enhanced efficiency in cryptographic operations. This study highlights that the modified AES algorithm improve security in modern electronic systems and maintain performance.**

Keywords*—VLSI, AES, cryptographic attacks, integrity and confidentiality.

I. INTRODUCTION

In the modern era of digital communication, data transmission security is essential, particularly in VLSI systems where massive amounts of data are processed and transferred [1]. This makes possible of efficient solutions such as electrical appliances, communication networks, and embedded devices [2]. There is rising demand for strong security mechanisms that protection sensitive data from cyber-attacks and illegal access due to the complexity and miniaturization of these systems [3-4]. For data security, conventional encryption techniques using such as AES, Data Encryption Standard (DES), Rivest-Shamir-Adleman (RSA) and Speck cipher.

In reference [5-6] presents DES based on the Feistel structure, making it easy to implement and efficient in hardware, such as VLSI, due to its simple encryption mechanism. However, its key length (56 bits) is deemed insecure by modern standards since it is vulnerable to brute-force attacks, making it unsuitable for securing systems. In reference [7-8] enhanced RSA algorithm increases encryption by utilizing three unique integers, increasing attack resistance and ensuring message security. However, its increases computation time, which have an influence on performance, particularly with larger key sizes. SPECK lightweight encryption algorithm provides security by confusion and diffusion, leading to the technique's high security qualities involved sensitivity, collision and pre-image resistance. Nonetheless, its lightweight nature makes it susceptible to advanced cryptographic attacks, which decreases its security in extremely sensitive environments [9-10]. The strict requirements of VLSI systems, these security methods improved as threats change and the need for more efficiency increases [11-12].

Therefore, in this work implemented the Modified AES algorithm for improved security protocol especially for VLSI devices to overcome these issues.

Thus the contribution of the developed work are listed below:

- The key expansion unit to generate stronger and more unpredictable keys for avoiding cryptographic attacks, increasing encryption adaptability.
- The Modified AES algorithm improves diffusion and confusion qualities by using the Feistel structure, making the system more secure against differential and linear attacks.
- VLSI implement to optimize the encryption process for low power consumption, fast speed, and minimum

hardware difficulty, which leads to efficient real-time data security.

The following portions of the paper are structured as below. Section II debates related work carried out in previous research. Section III discusses modified AES implementation in the proposed work, including a description of proposed work and modeling of the proposed work. Section IV discuss the experimental result and discussion of proposed work effectiveness. Section V conclusion of study by assessing the research's contributions and effects on the study, followed by proposed for future work.

II. Related Works

Salman Ahmed et al. (2024) developed lightweight AES for IoT devices by decreasing intermediate registers, and implementing multiplexed designs. Its provide a result in a large reduction in area usage and having negligible impact on throughput and power consumption. This technology is efficiency in resource use and its maintains the security standards required for IoT applications. However, the number of components designs result in a larger design overhead, causing implementation difficult for devices with extremely limited resources [13]. Osama Fouad Abdel Wahab et al (2021) have implemented RSA, and DWT to encrypt, compress, and hide data in a cover image, resulting in efficient storage and high-quality outputs. The method decreases message size and minimizing distortion, providing high security and compact image size. The system provides highly security, effective compression, and low data loss, which make it perfect for high-quality, compact image storage and transmission. However, use of lossy compression occur in data loss, and affect processing time for large datasets [14]. Talapala Lakshmi Prasanna et al. (2022) have introduced AES-128 cryptography algorithm on FPGA module for processing image using UART protocol. The technique effectively encrypts and decrypts images while utilizing only 141 LUTs and 0.0291W of electricity. This technique preferred great efficiency, power consumption and resource utilization. However, this system is the possible sensitivity to data loss and signal interference in wireless connection, which risk the security of the transmitted data [15].

III. Proposed Method

A. Proposed Method Description

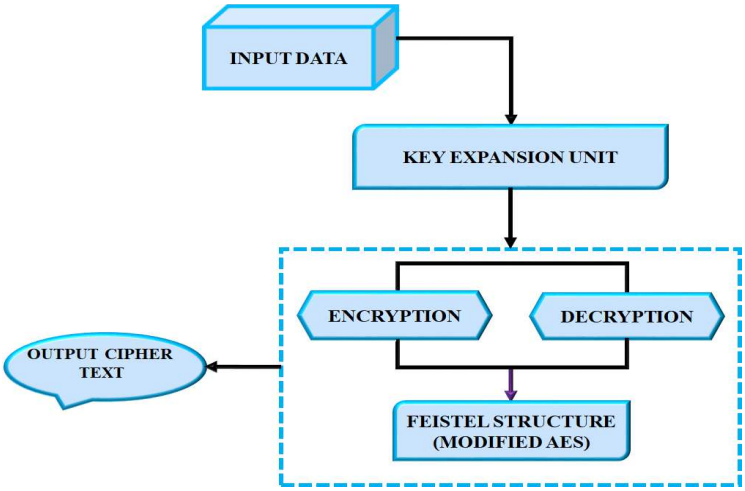

Fig. 1. Proposed Block Diagram

The proposed technique, depicted in Fig. 1, aims to increase the security of VLSI systems by changing the modified AES algorithm for accurate data transfer. The consecutive processes of data encryption and outlines the architecture of this system. By giving each encryption round a discrete key, this stage is essential for fortifying the cryptographic process. A potential vulnerability is minimized by the protected design of the key expansion unit, which agreements the safe generation and distribution of keys.

After key expansion the modified AES algorithm is utilized in the main encryption procedure. By adding further security layers like dynamic S-box transformations and an enhanced round function, structure improves on the conventional AES. These adjustments are especially designed to endure cryptanalytic attacks, guaranteeing that the encryption are continue to be strong against different types of operational threats. This all-inclusive method guarantees that information is securely encrypted during transmission within VLSI systems, protecting the information's integrity and confidentiality. In addition to bolstering VLSI system security, the proposed solution retains excellent performance and efficiency, which are necessary for real-world implementation.

IV. Proposed System Modelling

A. Modified AES Algorithm

The secret key is capable of any length, and according to the proposed AES technique, a 320-bit key is uses three several size of key, like 128, 192, and 256 bits. This is critically important to comprehend. Improving the number of rounds enhances the system's security and provides unauthorized users with greater privacy. Hackers find it more difficult to compromise the system with the increased number of rounds. According to popular belief, the AES algorithm cannot be broken by any transformational simplification. As a result, a 320-bit key size was used to achieve better results.

B. Encryption process of modified AES

The transformation of Plain text into Cipher text is one way to characterize it. 16 rounds are used in the AES encryption procedure, as opposed to 10 rounds previously as shown in Fig. 2. This first key is produced using the Polybius square.

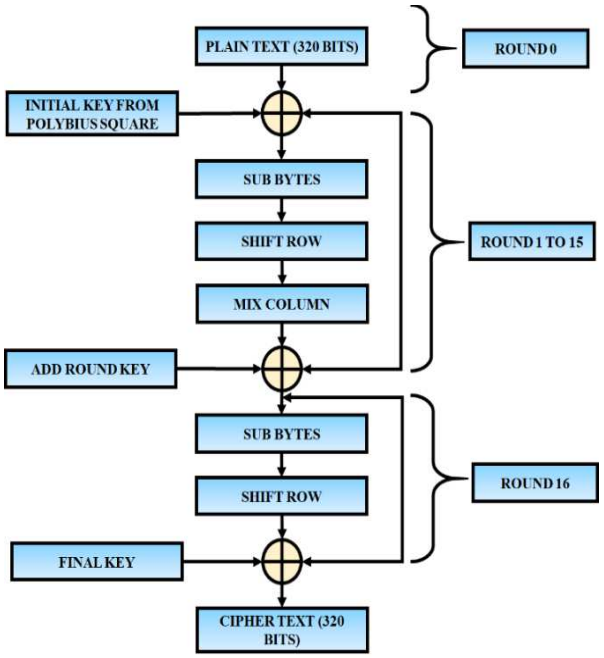

Fig. 2. Modified AES proposed encryption process

C. *Decryption process of modified AES*

Decryption is the method of transforming encrypted text back into its original plain text form. As shown in Fig. 3, the transformations used in the decryption process are AddRoundKey, InvShiftRows, InvMixColumns, and InvSubBytes. These changes correspond to the encryption's modifications.

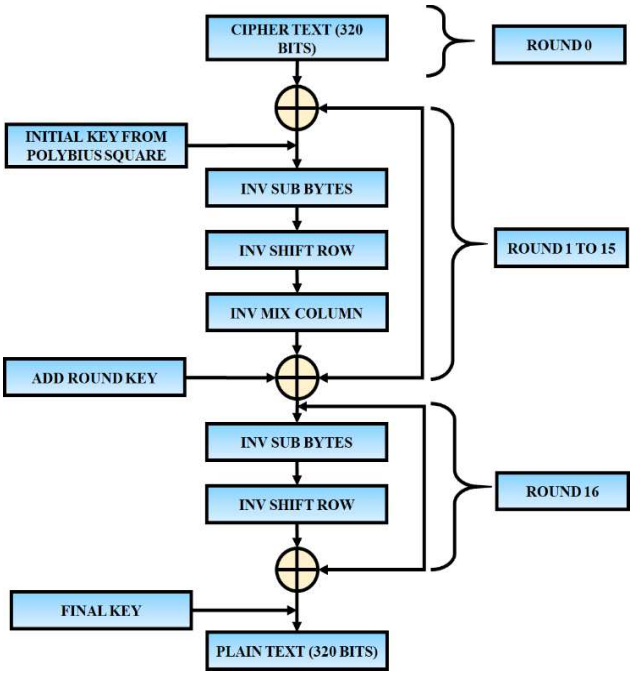

Fig. 3. Structure of decryption process in modified AES

D. *Process of key generation*

Polybius Square is applied to the $6 * 6$ matrix to produce the key. The Polybius square helps to provide secure data by filling in letters and digits devoid of duplication between left to right. As seen in Table I, the numbers are placed in ascending sequence from 0 to 9.

TABLE I. GENERATING KEY USED BY POLYBIUS SQUARE

0	1	2	3	4	5	
0	A	B	C	D	Y	4
1	E	F	G	H	Z	5
2	I	J	K	L	0	6
3	M	N	O	P	1	7
4	Q	R	S	T	2	8
5	U	V	W	X	3	9

TABLE II. POLYBIUS SQUARE USED GENERATING KEY

Plain text	**S**	**E**	**C**	**U**	**R**	**I**	**T**	**Y**	**1**	**2**	**3**
Position	1	2	3	4	5	6	7	8	9	10	11
Cipher Test	42	10	02	50	41	20	43	04	34	44	54

The original text, known as the plaintext, is encrypted using a few codes to create a cipher text that prevents hackers from identifying it. The message "SECURITY123" needs encoded and decoded to original message, as demonstrated Table II. In the same way, the decryption procedure is carried out. As a result, a key is generated that is used in the encryption and decryption processes.

V. RESULTS AND DISCUSSION

To demonstrate the significance of the developed method, a comparative study compared to conventional approaches is included in this section along with the proposed research and outcomes.

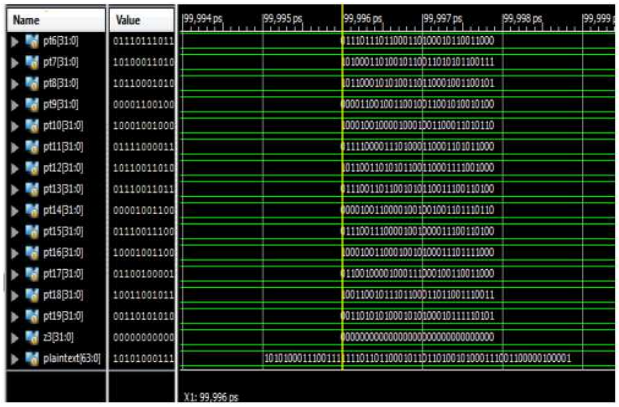

Fig. 4. Encryption of plaintext

The process of encrypting plaintext, which involves converting the initial legible message into cipher text via a sequence of cryptographic procedures, is shown in Fig. 4. This ensures that the original message remains secret whether it is being transmitted or stored because the cipher text that is produced cannot be read without the matching decryption key.

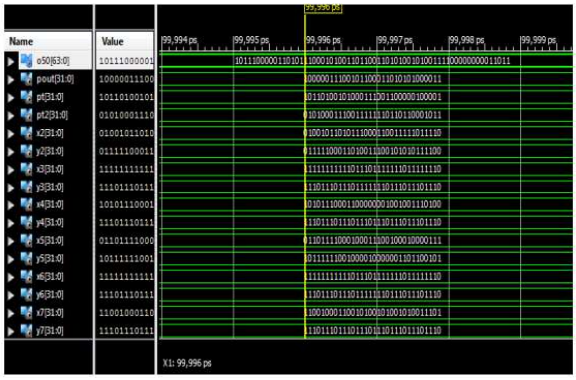

Fig. 5. Encryption of Cipher text

Fig. 5 shows first layer of cipher text, they find it more difficult to decrypt the data using this technique, which is also known as double encryption or layered encryption.

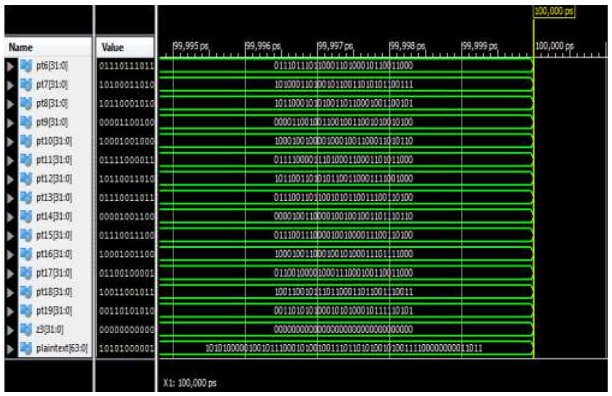

Fig. 6. Decryption of Plaintext

The decryption of plaintext, which returns the encrypted data (cipher text) to its original, readable form as shown in Fig. 6. The decryption sequence is displayed as a waveform, with binary values denoting various stages of the process. This graphic shows the encryption processes are successfully reversed by the decryption algorithm, bringing the encrypted text back to its original form.

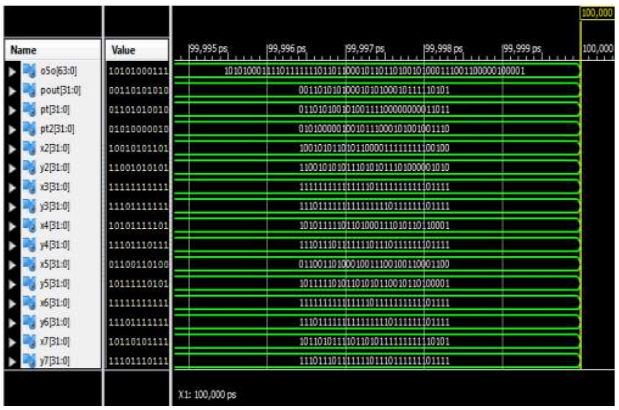

Fig. 7. Decryption of Cipher text

The process of decrypting cipher text, which involves processing encrypted binary data to recover the original plaintext, is illustrated in Fig. 7. The waveform graphic illustrates different signals binary values sequentially change throughout the decryption process.

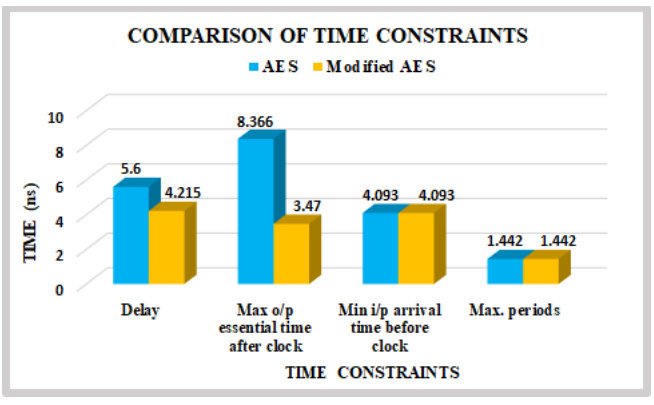

Fig. 8. Comparison of time constraints

Fig. 8 compares the time constraints of AES and Modified AES encryption algorithms. It presents four key metrics: delay, maximum output essential time after the clock, minimum input arrival time before the clock, and maximum periods. Modified AES shows improved performance with reduced delay and essential time compared to AES.

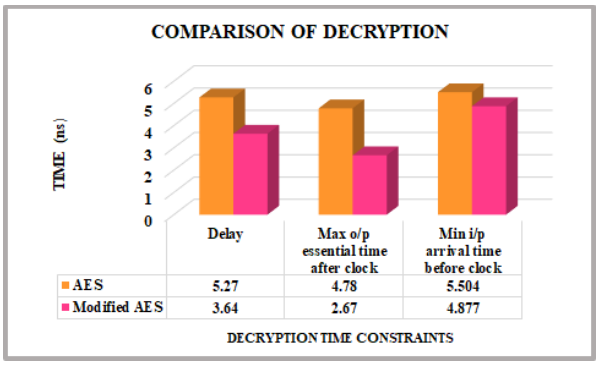

Fig. 9. Comparison of Decryption

A comparison of the decryption performance of the AES and Modified AES algorithms is shown in Fig. 9. It demonstrates that Modified AES has a smaller latency and maximum output critical time after the clock than AES. The minimal input arrival time before the clock is likewise substantially reduced in Modified AES.

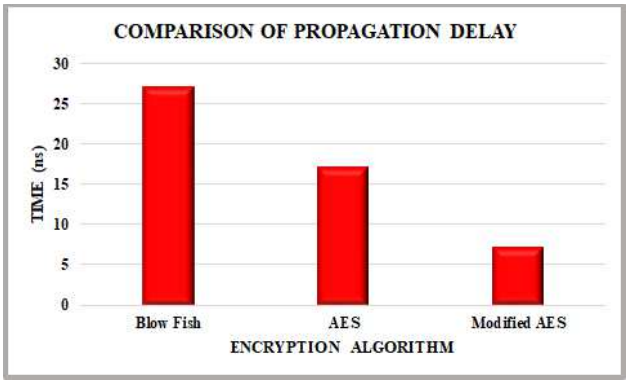

Fig. 10. Comparison of propagation delay

The comparison of propagation latency between the blowfish, AES and Modified AES algorithms is shown in Fig. 10. The Modified AES performs better and processes data faster than the traditional AES due to a much smaller propagation latency. The Modified AES is now a more efficient option for applications that need faster data transfer, as evidenced by the reduction in propagation delay.

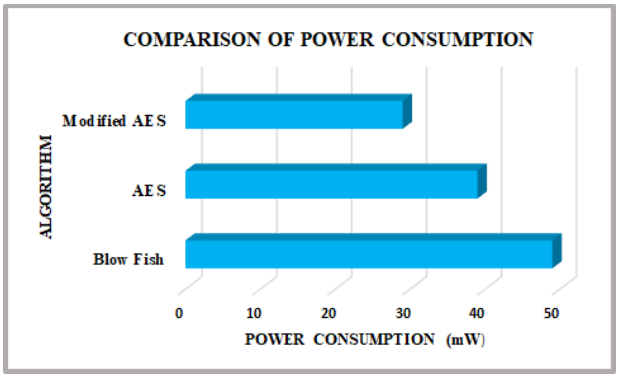

Fig. 11. Comparison of power consumption

A comparison of the power usage of the AES, Blowfish, and Modified AES algorithms is presented in Fig. 11.

Comparing the Modified AES to the other conventional methods, the results show that a significant reduction in power usage is achieved. The Modified AES is a more energy-efficient choice because of its efficiency, making it perfect for applications where energy conservation is essential.

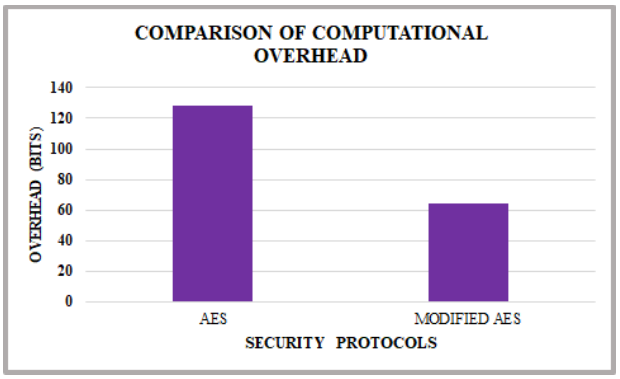

Fig. 12. Comparison of Computational Overhead for Security Protocols

Fig. 12 shows compares the computational overhead (in bits) of the AES [16] and Modified AES security algorithms, demonstrating that Modified AES has a lower overhead than regular AES.

VI. Conclusion

The modified AES algorithm described in this paper to provide a secure communication for VLSI implementation. Although the AES approach manages to balance security and performance, the efficient form prioritizes encoding speed, power consumption, and hardware footprint reduction are critical factors for VLSI applications. The algorithm's is implemented in hardware to verify the throughput and efficiency of proposed algorithm. Robust simulations synthesis results demonstrate that the updated technique surpasses current encryption benchmarks of speed and energy efficiency, making it ideal secure real-time communication in VLSI systems with limited resources. As the propagation latency has decreased, the Modified AES is now a more effective choice for applications that require speedier data transport. The Xilinx simulation implemented in this work successfully provided results for optimizing system performance, validating the proposed design, and ensuring accuracy in the final output. The modified AES algorithm significantly improves security and efficiency for VLSI systems by balancing encryption strength and performance. However, future research should concentrate on optimizing the method for a broader range of VLSI architectures and addressing upcoming cryptographic risks, such as caused by quantum computing.

References

[1] D. Reis, H. Geng, M. Niemier, and X. S. Hu, "IMCRYPTO: an in-memory computing fabric for AES encryption and decryption," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 5, pp. 553–565, Mar 2022.

[2] R. Rajashree, V. Peroumal, L. Kishore, K. V. D. Reddy, S. Reddy, and M. Jagannath, "Implementation of high speed and lightweight symmetric key encryption algorithm-based authentication protocol for resource constrained devices," International Journal of Electronic Security and Digital Forensics, vol. 14, no. 3, pp. 238–263, 2022.

[3] J. S. Ng, J. Chen, N. A. Kyaw, K. S. Chong, and B. H. Gwee, "Securing Against Side-Channel Attacks with Wide-Range in Situ Random Voltage Dithering on Async-Logic AES Engine," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Jun 2024.

[4] K. P. More and R. A. Patil, "A modified advanced encryption standard-based model for secured data transmission in cognitive radio with multi-channels," International Journal of Wireless and Mobile Computing, vol. 26, no. 3, pp. 238–250, 2024.

[5] N. Dhandapani, M. Z. Mohamed Ashik, K. Reddy Bhargav, N. Achyuth, and D. Jose, "VLSI Implementation of BCH Encoder with Triple DES Encryption for Baseband Transceiver," In Mobile Radio Communications and 5G Networks: Proceedings of Third MRCN 2022, pp. 329–341, 2023.

[6] S. Hou, Y. Ma, D. Deng, Z. Wang, and G. Ren, "Modeling and physical attack resistant authentication protocol with double PUFs," Journal of Information Security and Applications, vol. 76, pp. 103543, Aug 2023.

[7] V. Balani, C. Kharya, S. N. Shivhare, and T. P. Singh, "An Enhanced RSA Algorithm to Counter Repetitive Ciphertext Threats Empowering User-centric Security," Scalable Computing: Practice and Experience, vol. 25, no. 6, pp. 4669–4682, 2024.

[8] T. K. Goyal, V. Sahula, and D. Kumawat, "Energy efficient lightweight cryptography algorithms for IoT devices," IETE Journal of Research, vol. 68, no. 3, pp. 1722–1735, May 2022.

[9] A. Sevin, and Ü. Çavuşoğlu, "Design and Performance Analysis of a SPECK-Based Lightweight Hash Function," Electronics, vol. 13, no. 23, pp. 4767, 2024.

[10] K. I. Jones, and R. Suchithra, "Information Security: A Coordinated Strategy to Guarantee Data Security in Cloud Computing," International Journal of Data Informatics and Intelligent Computing, vol. 2, no. 1, pp. 11–31, Mar 2023.

[11] K. Assa-Agyei, F. Olajide, and T. Alade, "Optimizing the performance of the advanced encryption standard techniques for secured data transmission," International Journal of Computer Applications, vol. 185, no. 21, pp. 31–36, 2023.

[12] S. Pattanavichai, "Program for Simulation and Testing of Apply Cryptography of Advance Encryption Standard (AES) Algorithm with Rivest-Shamir-Adleman (RSA) Algorithm for Good Performance," International Journal of Electronics and Telecommunications, pp. 475–481, 2022.

[13] S. Ahmed, N. Ahmad, N. A. Shah, G. E. Mustafa Abro, A. Wijayanto, A. Hirsi, and A. R. Altaf, "Lightweight AES Design for IoT Applications: Optimizations in FPGA and ASIC with DFA Countermeasure Strategies," IEEE Access, 2025.

[14] O. F. A. Wahab, A. A. M. Khalaf, A. I. Hussein, H. F. A. Hamed, "Hiding data using efficient combination of RSA cryptography, and compression steganography techniques," IEEE access, vol. 9, pp. 31805–31815, 2021.

[15] T. L. Prasanna, N. Siddaiah, B. M. Krishna, and M. R. Valluri, "Implementation of the advanced encryption standard algorithm on an FPGA for image processing through the universal asynchronous receiver-transmitter protocol," International Journal of Electrical and Computer Engineering, vol. 12, no. 6, pp. 6114–6122, 2022.

[16] P. Thaenkaew, B. Quoitin, and A. Meddahi, "Leveraging Larger AES Keys in LoRaWAN: A Practical Evaluation of Energy and Time Costs," Sensors, vol. 23, no. 22, pp. 9172, 2023.