Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Modified AES-128 with Boundary-Based Splitting and Dynamic Output Encryption

Dhanashree Hadsul[1], Dr. Zahir Aalam[2]
[1]*Research Scholar, Department of Information Technology, Thakur College of Engineering and Technology, Mumbai, India*
dhanashree.hadsul@gmail.com
[2] *Professor and Research Guide, Department of Information Technology, Thakur College of Engineering and Technology, Mumbai, India*
Zahir.aalam@thakureducation.org

ARTICLE INFO

Received: 24 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Encryption is essential for maintaining the confidentiality, integrity, and authenticity of data in today's digital landscape. As cyber threats continue to escalate, the implementation of effective encryption techniques has become crucial for protecting sensitive information from unauthorized access and preventing data breaches. Encryption is utilized across multiple domains, such as securing data transmission, protecting file storage, enhancing email security, enabling secure messaging, safeguarding databases, verifying authentication, ensuring blockchain integrity, complying with regulations, securing IoT devices, and creating digital signatures to protect sensitive information and maintain privacy. The Advanced Encryption Standard (AES) is crucial for ensuring strong security, widespread adoption, efficiency in implementation, scalability with varying key lengths, versatility across applications, resistance to attacks, compliance with regulations, and facilitating secure global communication. AES is used in various applications, including secure data transmission, file encryption, virtual private networks (VPNs), secure web browsing (HTTPS), disk encryption, and protecting sensitive information in cloud storage and databases. This paper introduces a modified version of the Advanced Encryption Standard (AES) aimed at bolstering security beyond that of traditional AES. While AES is well-regarded for its strength, our enhanced approach integrates additional encryption layers and an innovative key management strategy, significantly boosting resilience against various cryptographic threats. We assess the performance and security attributes of the modified AES through Avalanche Effect Encryption Decryption time and throughput. Our findings indicate that this improved algorithm provides greater protection for sensitive data.

Keywords: Advanced Encryption Standard, Base64 encoding, PBKDF2, Initialization Vector, Avalanche Effect.

INTRODUCTION

Encryption serves as a cornerstone of information security, focused on ensuring the confidentiality, integrity, and authenticity of data. As the digital realm expands, the demand for effective encryption techniques has intensified due to the increasing prevalence of cyber threats, data breaches, and privacy issues. Essentially, encryption converts readable data into a format that can only be interpreted by authorized users who have the correct decryption keys. This mechanism not only protects sensitive information but also assists organizations in meeting regulatory standards such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA) [1]. Over the years, numerous encryption algorithms have emerged, primarily categorized into symmetric and asymmetric encryption. Symmetric encryption, notably represented by the Advanced Encryption Standard (AES), employs the same key for both encrypting and decrypting data, making it particularly efficient for large datasets [2]. Conversely, asymmetric encryption utilizes a pair of keys—one public and one private—allowing secure communication without the necessity of key exchange [3]. The significance of encryption extends well beyond simple data protection; it is vital for securing communications over networks, safeguarding intellectual property, and facilitating safe transactions within digital economies [4]. As technology advances and cyber threats

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

grow in complexity, ongoing innovation in encryption techniques is crucial for upholding the integrity and confidentiality of sensitive data [5].

The Advanced Encryption Standard (AES) is essential for several reasons. Primarily, AES is a symmetric key encryption algorithm known for its strong security against various attack methods, making it one of the most secure encryption standards available today [6]. Established as a federal standard by the U.S. National Institute of Standards and Technology (NIST), AES is widely recognized and utilized across different industries, promoting consistency in security measures [7]. Its design prioritizes efficiency in both hardware and software applications, making it suitable for a broad spectrum of devices, from servers to mobile devices [8]. Moreover, AES supports key lengths of 128, 192, and 256 bits, enabling organizations to choose the appropriate level of security based on their needs [9]. The versatility of AES allows it to protect data at rest, such as files and databases, as well as data in transit, including network communications and protocols like SSL/TLS [10]. Additionally, AES is resilient against known cryptographic attacks, including brute force and differential cryptanalysis, ensuring the long-term security of encrypted data [11]. Regulatory frameworks, such as GDPR and HIPAA, often recommend or mandate the use of AES for safeguarding sensitive information, making it crucial for compliance [1]. As a global standard, AES also supports secure communication and data protection across international borders, which is vital for businesses operating in a global market [12].

Modifying the AES (Advanced Encryption Standard) algorithm is essential in certain scenarios to address specific security, performance, or implementation challenges. While AES is highly secure, modifications can help improve resistance to emerging attacks, such as side-channel attacks or future quantum threats, and optimize the algorithm for resource-constrained environments like IoT devices [13], [14]. Customizations may also enhance performance through techniques like parallelization or hardware acceleration and ensure compatibility with new protocols or regulatory requirements [15]. Additionally, certain use cases, such as secure multi-party computation, real-time applications, and privacy-preserving data sharing, may require AES to be tailored for specific operational needs [16]. Modifying AES allows it to remain flexible, scalable, and secure while meeting the evolving demands of cryptographic applications, but these changes must be thoroughly tested to avoid introducing vulnerabilities.

In this paper we proposed modified Advanced encryption standard which includes salt generation, key derivation, and message padding. This algorithm provides a secure implementation of AES-128 encryption and decryption, employing an innovative technique that splits data based on specific boundaries. Additionally, it uses Base64 encoding to ensure that the encrypted data can be safely transported across systems. This approach ensures robust security of data.

WORKING OF ADVANCED ENCRYPTION STANDARD – 128 BIT

The Advanced Encryption Standard (AES) is a widely utilized symmetric key encryption algorithm designed to protect sensitive information across various applications. Operating on fixed block sizes of 128 bits, AES supports key lengths of 128, 192, or 256 bits. In the case of AES-128, a key length of 128 bits is employed, which enables a theoretical maximum of 2128 possible keys, thus providing a high level of security against brute-force attacks.

STRUCTURE OF AES-128

AES-128 is structured based on a substitution-permutation network (SPN) architecture that comprises multiple rounds of processing. The number of rounds is determined by the key length; specifically, AES-128 performs a total of 10 rounds[17]. Each round involves a series of transformations applied to the data block, which includes the following operations:

- SubBytes
- ShiftRows
- MixColumns
- AddRoundKey

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

KEY EXPANSION

Prior to the encryption process, the 128-bit key undergoes expansion into a key schedule consisting of 44 words, each 32 bits, organized into a 4x4 matrix. The key expansion entails several steps:

1. **Initial Key**: The original 128-bit key is split into four words, forming the first four words of the key schedule.
2. **Round Key Generation**: For each subsequent word, the following operations are performed:
    a. **RotWord**: The word is rotated left by one byte.
    b. **SubWord**: Each byte of the rotated word is replaced using the AES S-Box.
    c. **Round Constant (Rcon)**: A round constant is incorporated into the first byte of the word to enhance non-linearity.
3. **Key Schedule Generation**: Each new word is derived by XORing the previous word's output with the word located four positions earlier in the key schedule. This procedure continues until all 44 words are generated.

ENCRYPTION PROCESS

The AES encryption process consists of the following stages:

1. **Initial Round**:
    a. **AddRoundKey**: The plaintext undergoes an XOR operation with the initial round key from the key schedule.
2. **Main Rounds (1 to 9)**: Each of the nine rounds comprises the following transformations:
    a. **SubBytes**: Each byte in the state array is substituted using the S-Box, introducing non-linearity.
    b. **ShiftRows**: The rows of the state array are cyclically shifted left by varying offsets, promoting diffusion.
    c. **MixColumns**: A linear transformation mixes each column of the state, enhancing diffusion further.
    d. **AddRoundKey**: The state is XORed with the current round key.
3. **Final Round (10th Round)**: The concluding round consists of three transformations:
    a. **SubBytes**
    b. **ShiftRows**
    c. **AddRoundKey**

Note that the **MixColumns** transformation is omitted during this final round. After the last round, the output is the ciphertext, representing a 128-bit encrypted form of the initial plaintext.

DECRYPTION PROCESS

Decryption in AES-128 reverses the encryption steps, employing the round keys in reverse order. The process includes the following stages:

1. **Initial Round**:
    a. **AddRoundKey**: The ciphertext is XORed with the final round key.
2. **Main Rounds (1 to 9)**: Each of these nine rounds applies the following transformations in reverse order:
    a. **AddRoundKey**
    b. **InvMixColumns**: This transformation inverses the MixColumns effect by applying an inverse linear transformation.
    c. **InvShiftRows**: The rows are shifted back to their original positions.
    d. **InvSubBytes**: Each byte is substituted using the inverse S-Box.
3. **Final Round (10th Round)**: This last round consists of three operations:
    a. **AddRoundKey**
    b. **InvShiftRows**
    c. **InvSubBytes**

The outcome of the decryption process is the original plaintext, successfully retrieved from the ciphertext [1].

PROPOSED MODIFIED ADVANCED ENCRYPTION STANDARD

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Modifying the Advanced Encryption Standard (AES) is crucial in response to advancing security threats, particularly with the advent of quantum computing, which poses a potential risk to traditional encryption methods [19]. Improvements are also necessary to enhance performance in high-speed data processing environments, such as IoT devices and real-time systems, ensuring AES remains effective in resource-constrained settings [20]. Additionally, these modifications strengthen defences against side-channel attacks, which target the physical implementation of the algorithm rather than its cryptographic strength [21]. As technologies like blockchain and 5G continue to evolve, AES must be tailored to meet the specific security demands of these applications [18]. Regular updates to AES are vital to maintaining its resilience against both current and future cryptanalytic techniques, ensuring ongoing protection of sensitive information [22].

This paper outlines a method that necessitates a straightforward modification to the implementation of the AES algorithm while preserving its fundamental characteristics. This algorithm offers a secure implementation of AES-128 encryption and decryption by utilizing a technique that segments data according to defined boundaries. Furthermore, it incorporates Base64 encoding to facilitate the safe transmission of encrypted data across various systems. This method guarantees a high level of data security.

STRUCTURE OF MODIFIED AES

Modified algorithm works as follows:

1. Generate Salt and IV: Create a random salt and IV for encryption.
2. Derive Key: Use PBKDF2 with the passphrase and salt to derive the encryption key.
3. Pad the Plaintext: Ensure the plaintext is a multiple of the block size.
4. Encrypt the Plaintext: Use AES in CBC mode to encrypt the padded plaintext.
5. Construct Encrypted Output: Combine encrypted parts, salt, IV, and boundary value.
6. Decrypt Process: Extract components from the encrypted message, derive the key, and decrypt the ciphertext.
7. Unpad and Return Plaintext: Recover and return the original plaintext.

WORKING

The modified encryption algorithm operates by first generating a random salt and Initialization Vector (IV) to enhance security. Using a passphrase and the generated salt, the algorithm then derives an encryption key through PBKDF2. Next, the plaintext is padded to ensure it matches the required block size for encryption. The algorithm then encrypts the padded plaintext using AES in Cipher Block Chaining (CBC) mode. The final encrypted output is constructed by combining the encrypted data, salt, IV, and any necessary boundary values. For decryption, the algorithm extracts each component from the encrypted message, derives the key again, and decrypts the ciphertext. Finally, it removes the padding to recover and return the original plaintext.

RESULTS

The result of the modified AES-128 algorithm is a secure and efficient encryption and decryption process that enhances data confidentiality as shown in the figure 1 and 2. By splitting the encrypted data into two parts and using boundary-based segmentation along with Base64 encoding, the algorithm ensures that the encrypted message can be safely transmitted across different systems[23]. The use of PBKDF2 key derivation, salt, and initialization vector (IV) adds further security by making it resistant to brute-force attacks and other cryptanalytic techniques[24].

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

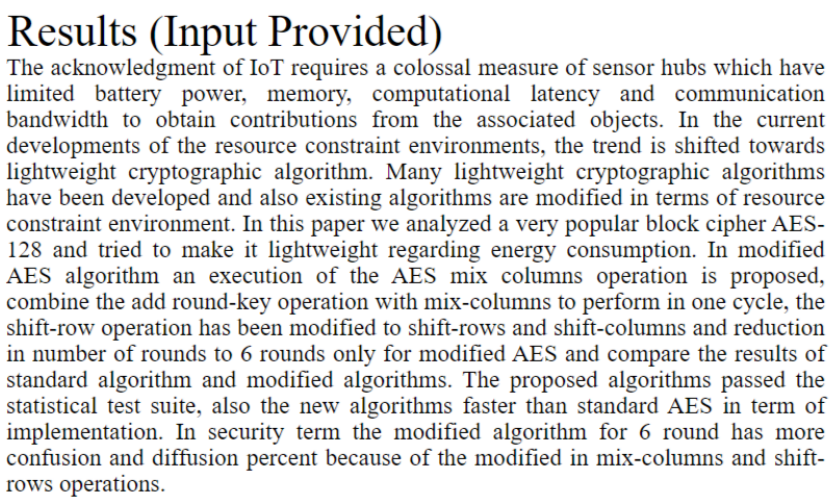

Fig. 1 Input Provided to the Modified AES Algorithm.

In the encryption phase, plaintext is transformed into a base64-encoded string that includes the salt, IV, and encrypted data, ensuring that the message remains secure and transportable. During decryption, the algorithm follows the reverse process to accurately recover the original plaintext, preserving data integrity.

Fig. 2 Output of the Modified AES Algorithm.

RESULT ANALYSIS OF MODIFIED AES

By evaluating the avalanche effect, encryption/decryption time, and throughput, we can comprehensively assess the complexity, security, and performance of the modified AES-128 algorithm. The avalanche effect helps determine the algorithm's sensitivity to small changes in the input data or key by analyzing how much the output (ciphertext) changes when a single bit in the input is altered. A strong avalanche effect indicates that the algorithm offers robust security through significant diffusion [25]. The encryption and decryption times reveal the computational efficiency of the algorithm, as faster times are essential in applications requiring real-time processing, such as IoT devices or streaming platforms [26] [27]. Throughput, which measures the rate at which data is processed, provides insight into the algorithm's capability to handle large volumes of data efficiently[28] [29]. By assessing these metrics together, the modified AES-128 algorithm can be judged for both its security strength and operational efficiency across different environments, ensuring it is well-suited for modern cryptographic needs.

AVALANCHE EFFECT

The avalanche effect in encryption algorithms describes the phenomenon where even a slight change in the input—like altering a single bit of plaintext or modifying a key bit—leads to substantial and unpredictable alterations in the output, or ciphertext. This property is vital for ensuring the security of cryptographic systems, as it complicates an

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

attacker's ability to draw connections between the original plaintext and its encrypted form. Ideally, a robust algorithm should yield about 50% of the output bits changing when just one bit of the input is modified, effectively obscuring any discernible patterns that could be exploited [30].

To assess the avalanche effect, one can choose a specific encryption algorithm and prepare a series of test inputs. By systematically changing bits in either the plaintext or the key and recording the resulting ciphertexts, the differences between the original and modified outputs can be analyzed. This evaluation sheds light on the algorithm's resilience against potential attacks, particularly those involving differential cryptanalysis. A strong avalanche effect is indicative of a secure encryption method, while a weak effect may reveal vulnerabilities, underscoring the importance of this characteristic in maintaining data integrity and confidentiality [31].

Avalanche Effect= (Number of Bits changed in ciphertext / Total Number of bits in ciphertext) * 100

In this example, we compared the conventional Advanced Encryption Standard (AES) with a modified version. In the first scenario, we calculated the avalanche effect for the conventional algorithm by changing a single character in the input. In the second scenario, we evaluated the modified AES without altering the input for execution. Finally, in the third scenario, we changed a single character in the modified algorithm and assessed the avalanche effect. The results indicate that the modified Advanced Encryption Standard produced satisfactory outcomes.

Case 1: Calculation of Avalanche Effect (Conventional AES, modified single character)

Hash 1: eebe2bf12e03924187082170fadff6938c216752167ff307fe17eb06202d24bd

Hash 2: b74c71f069b467e708973a9bba5a575095eb7ed2c0507c70b65946e92b5f6f95

Hamming Distance: 129

Percentage of Different Bits: 50.39%

Case 2: Calculation of Avalanche Effect (Modified AES without modification of input)

Hash 1: 18afb788b88e341607b2f575ccb97b6673603d894ead04656a83711ad06a242e

Hash 2: 06f7f324517ea182af50a4aa5faf6f180856332f39e3d3d056ace561493e4b72

Hamming Distance: 134

Percentage of Different Bits: 52.34%

Case 3: Calculation of Avalanche Effect (Modified AES with Modification of single Character)

Hash 1: 06f7f324517ea182af50a4aa5faf6f180856332f39e3d3d056ace561493e4b72

Hash 2: 7b84c988ecfdcb354d10ae148b9aa6a2d25273aacf81153cae2fa84b7f9715a5

Hamming Distance: 130

Percentage of Different Bits: 50.78%

ENCRYPTION / DECRYPTION TIME

Encryption and decryption times significantly impact the practical suitability of an encryption algorithm, especially in applications requiring low latency and minimal computational load. Real-time security solutions, such as those in streaming services or Internet of Things (IoT) devices, benefit from algorithms with short encryption times, ensuring seamless performance without sacrificing security efficiency [32]. Resource-constrained environments,

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

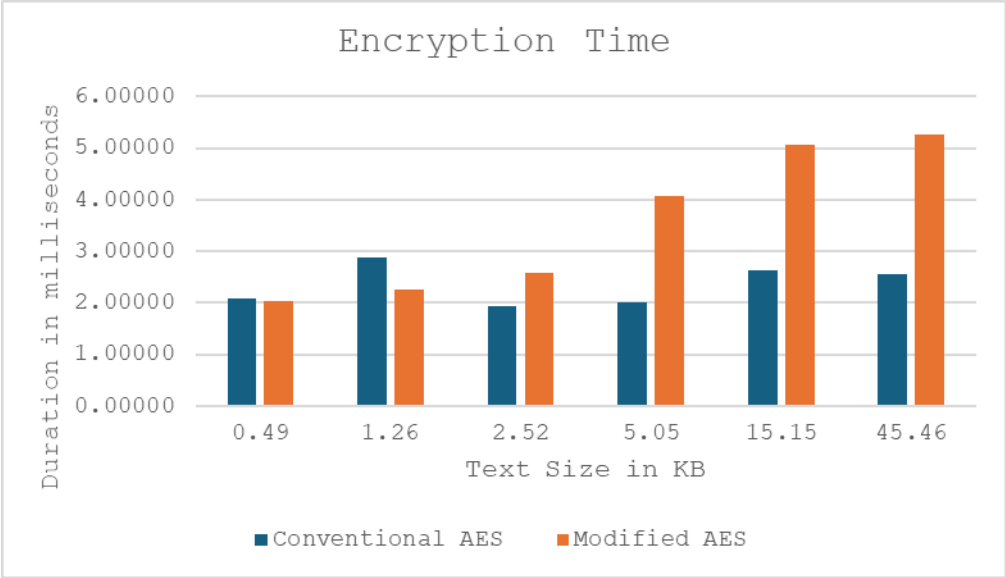

Fig. 3 comparison of encryption time for conventional Advanced Encryption Standard Algorithm and modified Advanced Encryption Standard.

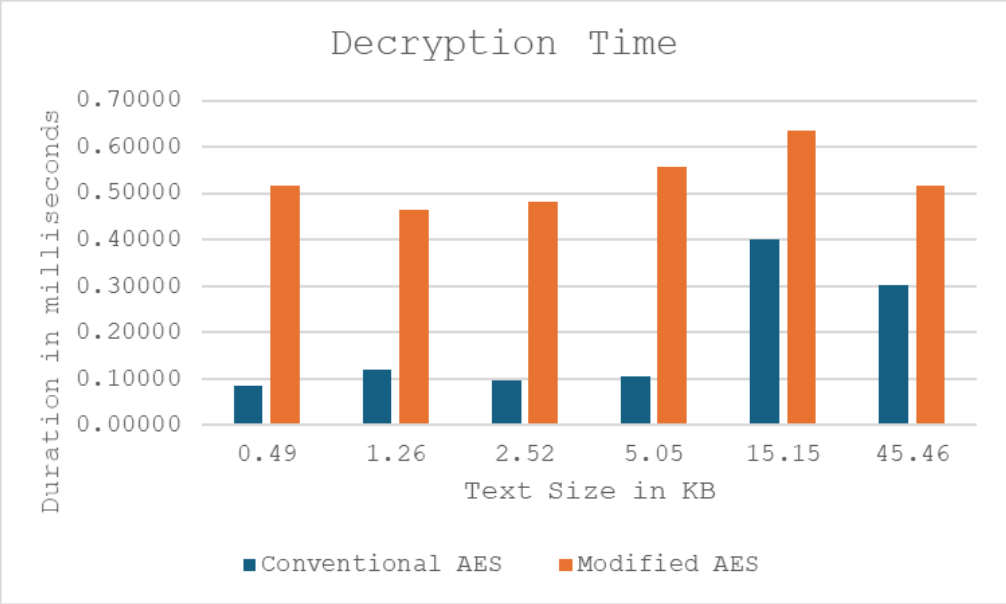

Fig. 4 comparison of decryption time for conventional Advanced Encryption Standard Algorithm and modified Advanced Encryption Standard.

including mobile platforms, also require encryption methods that use minimal CPU and memory, enhancing their suitability for widespread use without imposing heavy processing requirements [33].

These timing factors further influence the resilience of algorithms against timing attacks. If encryption time varies with inputs or key characteristics, it could lead to timing attacks, where adversaries exploit time discrepancies to derive information about encryption keys or plaintext, thus compromising security [34]. Consistent encryption and decryption times across inputs help reduce these risks, making algorithms more resistant to timing-based vulnerabilities[35].

Moreover, encryption time is indirectly linked to an algorithm's complexity and security level. Adding rounds or transformations, as seen in Advanced Encryption Standard (AES), enhances cryptographic security but also increases processing time[36]. Balancing additional rounds for improved security with the increased encryption time is crucial to ensure overall efficiency [37].

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

The graphs above illustrate the encryption and decryption times for the standard Advanced Encryption Standard (AES) algorithm compared to the modified AES algorithm. It has been observed that with smaller data sizes, encryption takes less time, but as data size increases, the encryption time for Modified AES exceeds that of conventional AES. Additionally, the decryption time for Modified AES is consistently longer than that of conventional AES, regardless of data size.

THROUGHPUT

Another parameter that we have considered is throughput. Throughput is a key measure of an encryption algorithm's efficiency, representing the rate at which data is processed, often in megabits per second (Mbps) [38]. Calculating throughput involves dividing the total data size (in bits) by the encryption time taken (in seconds) [39].

$$Troughput = \frac{Total\ data\ size\ in\ bits}{Encryption\ time\ in\ sec.} X\ 100$$

This metric helps determine how effectively an algorithm can handle large datasets, with higher throughput indicating faster data processing and better performance for applications that demand quick encryption [40].

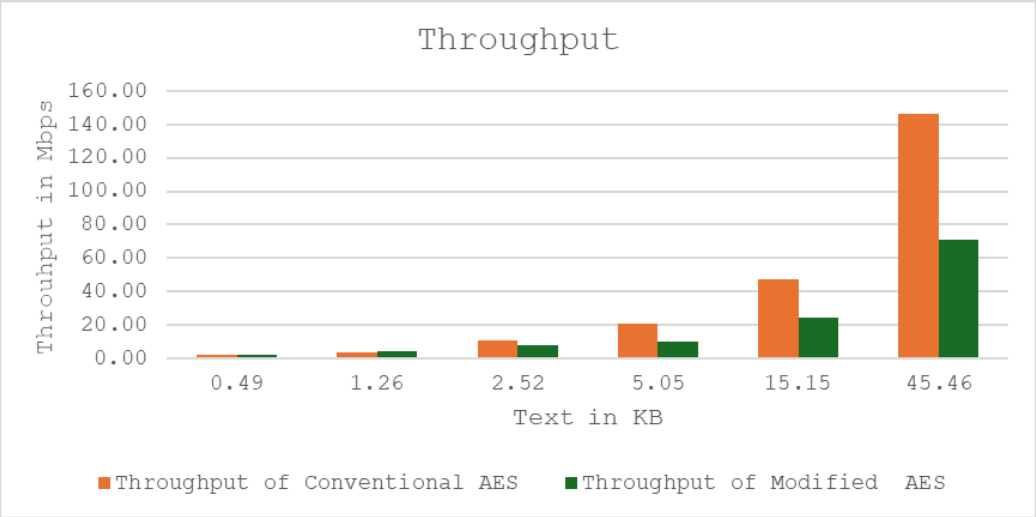

Fig 5 : Throughput of Conventional AES and Modified AES

Observations indicate that for smaller text sizes, the Modified AES algorithm achieves a higher throughput than conventional AES. However, as text size grows, the throughput of the Modified AES algorithm decreases to nearly half that of conventional AES.

In summary, the proposed algorithm requires only a simple modification to the AES implementation, maintaining its core attributes. This approach provides a secure AES-128 encryption and decryption process, utilizing a unique boundary-based splitting technique and base64 encoding to ensure transportability.

Each encryption of the same input produces a distinct encrypted output. The algorithm demonstrated satisfactory results concerning the Avalanche Effect for identical and varied inputs. Additionally, the complexity of the encryption algorithm can influence performance, with larger key sizes generally demanding more computational resources and potentially reducing throughput.

CONCLUSION

As a result, we adopted the Modified Advanced Encryption Standard (AES), an encryption algorithm that is both safer and more effective than the Conventional AES. A modified algorithm that uses a unique boundary-based splitting technique and base64 encoding for portability implements a safe AES-128 encryption and decryption process. The outcomes offer distinct encrypted outputs for identical inputs and yield superior Avalanche effects, surpassing 50%. Because of this, the encryption and decryption times are marginally longer than with traditional

Research Article

AES. In comparison to a traditional technique, throughput is good for smaller data sizes; however, as text size increases, throughput decreases to half that of a normal AES algorithm.

REFRENCES

[1] W. Stallings, "Cryptography and Network Security: Principles and Practice," 7th ed. Pearson, 2016.
[2] N. Koblitz, "A Course in Number Theory and Cryptography," 2nd ed. Springer, 1994.
[3] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems," Communications of the ACM, vol. 21, no. 2, pp. 120-126, Feb. 2001.
[4] S. Garfinkel and B. Spafford, "Web Security and Commerce," O'Reilly Media, 1997.
[5] A. Menezes, P. van Oorschot, and S. Vanstone, "Handbook of Applied Cryptography," CRC Press, Jan 2018.
[6] M. Chakole and S. S. Dorle, "Design of Advanced Encryption Standard Algorithm," 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, Pune, India, 2022, pp. 1-5, doi: 10.1109/ICCUBEA54992.2022.10011065.
[7] National Institute of Standards and Technology, "Announcing the Advanced Encryption Standard (AES)," NIST FIPS 197, Nov. 2001. [Online]. Available: https://doi.org/10.6028/NIST.FIPS.197
[8] K. I. Masud, M. R. Hasan, M. M. Hoque, U. D. Nath and M. O. Rahman, "A New Approach of Cryptography for Data Encryption and Decryption," 2022 5th International Conference on Computing and Informatics (ICCI), New Cairo, Cairo, Egypt, 2022, pp. 234-239, doi: 10.1109/ICCI54321.2022.9756078.
[9] A. Gupta and A. Agrawal, "Advanced Encryption Standard Algorithm with Optimal S-box and Automated Key Generation," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 2112-2115, doi: 10.1109/ICACITE53722.2022.9823662.
[10] Tariq U, Ahmed I, Bashir AK, Shaukat K. A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review. Sensors. 2023; 23(8):4117. https://doi.org/10.3390/s23084117
[11] A. Author, "Comparative Analysis of AES and RSA with Other Encryption Techniques for Secure Communication," International Journal of Scientific Research in Computer Science Engineering and Information Technology, vol. 10, no. 2, pp. 565-574, Apr. 2024. DOI: 10.32628/CSEIT2410263.
[12] B. Sarkar, A. Saha, D. Dutta, G. D. Sarkar, and K. Karmakar, "A Survey on the Advanced Encryption Standard (AES): A Pillar of Modern Cryptography," International Journal of Computer Science and Mobile Computing, vol. 13, no. 4, pp. 68-87, Apr. 2024. ISSN 2320–088X.
[13] K. Mahanta and H. B. Maringanti, "An Enhanced Advanced Encryption Standard Algorithm," Int. J. Adv. Trends Comput. Sci. Eng., vol. 4, no. 4, pp. 28-33, Special Issue of ICEEC 2015, Aug. 2015. [Online]. Available: http://warse.org/IJATCSE/static/pdf/Issue/iceec2015sp06.pdf.
[14] P. Satyanarayana, N. Sriramdas, B. Madhavi, and M. N. Arun, "Enhancement of Security in IoT Using Modified AES Algorithm for IoT Applications," in Proc. 2023 Int. Conf. Sustainable Communication Networks and Application (ICSCNA), Nov. 2023, doi: 10.1109/ICSCNA58489.2023.10370606.
[15] M. Narayanan and S. Subha, "Parallel AES Algorithm for Performance Improvement in Data Analytics Security for IoT," Int. J. Netw. Virtual Organ., vol. 18, no. 2, pp. 112-123, Jan. 2018, doi: 10.1504/IJNVO.2018.10012669.
[16] J. I. Choi and K. R. B. Butler, "Secure Multiparty Computation and Trusted Hardware: Examining Adoption Challenges and Opportunities," Security and Privacy in Wireless Networks, vol. 2019, Article ID 1368905, 2019. doi: 10.1155/2019/1368905.
[17] D. R. Stinson, Cryptography: Theory and Practice, 3rd ed. Boca Raton, FL, USA: CRC Press, 2006.
[18] J. Daemen and V. Rijmen, The Design of Rijndael: AES - The Advanced Encryption Standard, Berlin, Germany: Springer, 2002.
[19] O. A. Ajala, C. A. Arinze, and O. C. Ofodile, "Exploring and reviewing the potential of quantum computing in enhancing cybersecurity encryption methods," Magna Scientia Advanced Research and Reviews, vol. 10, no. 1, pp. 321-329, Feb. 2024. DOI: 10.30574/msarr.2024.10.1.0038.
[20] A. H. Ali, E. K. Gbashi, H. Alaskar, and A. J. Hussain, "A Lightweight Image Encryption Algorithm Based on Secure Key Generation," IEEE Access, vol. 12, pp. 1-12, July 2024. DOI: 10.1109/ACCESS.2024.3414334.

Research Article

[21] M. Ş. Açikkapi, F. Özkaynak, and A. B. Özer, "Side-Channel Analysis of Chaos-Based Substitution Box Structures," IEEE Access, vol. 7, pp. 76140-76149, June 2019. DOI: 10.1109/ACCESS.2019.2921708.
[22] U. Tariq, I. Ahmed, A. K. Bashir, and K. Shaukat, "A Critical Cybersecurity Analysis and Future Research Directions for the Internet of Things: A Comprehensive Review," Sensors, vol. 23, no. 8, p. 4117, Apr. 2023. DOI: 10.3390/s23084117.
[23] A. B. Mahmood and R. D. Dony, "Segmentation based encryption method for medical images," 2011 International Conference for Internet Technology and Secured Transactions, Abu Dhabi, United Arab Emirates, 2011, pp. 596-601.
[24] G. Kodwani, S. Arora and P. K. Atrey, "On Security of Key Derivation Functions in Password-based Cryptography," 2021 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 2021, pp. 109-114, doi: 10.1109/CSR51186.2021.9527961.
[25] S. D. Sanap and V. More, "Performance Analysis of Encryption Techniques Based on Avalanche effect and Strict Avalanche Criterion," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 2021, pp. 676-679, doi: 10.1109/ICSPC51351.2021.9451784.
[26] R. Chandrashekhar, J. Visumathi and A. P. Anandaraj, "Advanced Lightweight Encryption Algorithm for Android (IoT) Devices," 2022 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 2022, pp. 1-5, doi: 10.1109/ACCAI53970.2022.9752555.
[27] C. Liu, Y. Zhang, J. Xu, J. Zhao and S. Xiang, "Ensuring the Security and Performance of IoT Communication by Improving Encryption and Decryption With the Lightweight Cipher uBlock," in IEEE Systems Journal, vol. 16, no. 4, pp. 5489-5500, Dec. 2022, doi: 10.1109/JSYST.2022.3140850.
[28] J. Kala, J. Panda and L. Tanwar, "FPGA Implementation Of a High Throughput Low Power Advanced Encryption Standard (AES-128) Cipher," 2023 10th International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 2023, pp. 367-372, doi: 10.1109/SPIN57001.2023.10116674.
[29] Y. M. Khattabi, M. M. Matalgah and M. M. Olama, "Revisiting Lightweight Encryption for IoT Applications: Error Performance and Throughput in Wireless Fading Channels With and Without Coding," in IEEE Access, vol. 8, pp. 13429-13443, 2020, doi: 10.1109/ACCESS.2020.2966596.Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
[30] J. Kaur and K. R. R. Kumar, "Analysis of Avalanche effect in Cryptographic Algorithms," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-4, doi: 10.1109/ICRITO56286.2022.9965127.
[31] H. T. Assafli and I. A. Hashim, "Security Enhancement of AES-CBC and its Performance Evaluation Using the Avalanche Effect," 2020 3rd International Conference on Engineering Technology and its Applications (IICETA), Najaf, Iraq, 2020, pp. 7-11, doi: 10.1109/IICETA50496.2020.9318803.
[32] M. Rasori, M. La Manna, P. Perazzo, and G. Dini, "A Survey on Attribute-Based Encryption Schemes Suitable for the Internet of Things," IEEE Internet of Things Journal, vol. 9, no. 11, pp. 8269–8287, Jun. 2022.
[33] S. M. Florence, S. Alban and H. Mogalipuvvu, "A Hybrid Cryptographic Algorithm for Resource-Constrained IoT Devices," 2024 IEEE International Conference on Computing, Power and Communication Technologies (IC2PCT), Greater Noida, India, 2024, pp. 1914-1918, doi: 10.1109/IC2PCT60090.2024.10486560.
[34] M. k. Rameshbhai, D. Mishra and S. Verma, "Advanced Encryption for Enhanced Data Protection for Heterogeneous Sensor Networks," 2024 11th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2024, pp. 1-6, doi: 10.1109/ICRITO61523.2024.10522322.
[35] D. Jayasinghe, A. Ignjatovic and S. Parameswaran, "RFTC: Runtime Frequency Tuning Countermeasure Using FPGA Dynamic Reconfiguration to Mitigate Power Analysis Attacks," 2019 56th ACM/IEEE Design Automation Conference (DAC), Las Vegas, NV, USA, 2019, pp. 1-6.
[36] R. Podder and R. K. Barai, "Hybrid Encryption Algorithm for the Data Security of ESP32 based IoT-enabled Robots," 2021 Innovations in Energy Management and Renewable Resources(52042), Kolkata, India, 2021, pp. 1-5, doi: 10.1109/IEMRE52042.2021.9386824.
[37] S. Mahaboob Basha, V. Rishik, V. J. Naga Krishna and S. Kavitha, "Data Security in Cloud using Advanced Encryption Standard," 2023 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 2023, pp. 1108-1112, doi: 10.1109/ICICT57646.2023.10134339.

Journal of Information Systems Engineering and Management
2025, 10(46s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

[38] K. Meera and N. Selvaganesan, "Study on Various Encryption/Decryption Algorithms for Secure Communication using Chaotic based Hashed Key," 2023 Ninth Indian Control Conference (ICC), Visakhapatnam, India, 2023, pp. 79-84, doi: 10.1109/ICC61519.2023.10442209.

[39] M. Ahirrao, B. Pathak and M. Dixit, "AES and its Multiple Modifications for Various Applications," 2023 6th International Conference on Advances in Science and Technology (ICAST), Mumbai, India, 2023, pp. 476-479, doi: 10.1109/ICAST59062.2023.10454951.

[40] S. Souror, N. El-Fishawy and M. Badawy, "SCKHA: A New Stream Cipher Algorithm Based on Key Hashing and Splitting Technique," 2021 International Conference on Electronic Engineering (ICEEM), Menouf, Egypt, 2021, pp. 1-7, doi: 10.1109/ICEEM52022.2021.9480652.