VOL. 05 NO. 9,
FEBRUARY, 2025
E-ISSN 3026-958X
P-ISSN 3027-1169

African Scholar Publications & Research International
www.africanscholarpub.com

Journal of Science Innovation & Technology Research (JSITR)

Huffman Encoding for Modified RSA-AES Encrypted Token Compression in Secure Banking Transactions

Hamidu M.; Sarjiyus O.; & Manga I.

Department of Computer Science, Adamawa State University, Mubi, Adamawa State, Nigeria.

DOI: https://doi.org/10.70382/ajsitr.v7i9.013

Abstract

This research titled "Huffman encoding for modified RSA-AES encrypted token compression in secure banking transactions" aims to improve the security strength of customer banking credentials in transit and at rest by modifying the RSA token generation stage of encryption. These tokens are not original banking credentials but 32-bit decryption keys of AES. This modification will be made possible by using SHA-256 token generation for its historic strength and resistant to brute-force attack. This approach may hinder a serious computational overheat and time-space complexity. However, we propose the use of Huffman encoding with its quicker data compression to overcome the data size intricacy.

Keywords: SHA-256, Avalanche Effect, Huffmann Encoding, Entropy, Overheat

Introduction

Securing banking transactions have become increasingly challenging in today's digitally connected world where financial data is constantly being transmitted over various networks. The potential risks associated with data breaches and unauthorized access to sensitive information have made encryption and data compression essential components of modern banking systems *(*Haryaman *et al* 2024). Securing sensitive customer banking tokens like credit card numbers and account credentials is also essential (Agur *et al* 2020). As more transactions and communications occur digitally, banks and other financial institutions must ensure customer's data is protected during storage and transmission (Javaid *et al* 2022). Currently, many payment networks and banking systems use AES a symmetric encryption standard to protect tokens and data in transit and at rest. AES applies cipher block chaining (CBC) and other techniques to encrypt plain text data into incomprehensible ciphertexts (Altigani *et al* 2021).

AES is widely used globally to protect classified information (Smid, 2021). AES was chosen to replace the older Data Encryption Standard (DES) which was vulnerable to brute force attacks by National Institute of Standards and Technology (NIST) in 2001 after a 5-year standardization process, it is considered very difficult to crack through brute force attacks. AES transforms plain text data into fixed block sizes of ciphertexts and encryption keys. AES provides very high security against known attacks with its multiple round structure and large secret key sizes. It encrypts and decrypts data in fixed block sizes of 128 bits using cryptographic keys of 128-bits, 192-bits or 256-bits (Kishor Kumar *et al* 2024). It applies substitution, permutation and transformation techniques in multiple rounds to convert plaintext to ciphertext and back. Each round uses different keys derived from the original key using key scheduling algorithms. The number of rounds depends on the key size - 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys. The more rounds used, the more secure the AES encryption is against attacks. Analysing AES encrypted data without knowing the original key is extremely difficult given the complexity of reverse engineering the multiple substitution, permutation and transformation rounds. Brute force attacks trying all possible key combinations also become infeasible as key sizes grow larger

(Andersson, 2023). No effective cryptographic attacks against AES itself are publicly known so far (Grassi *et al* 2021). The only risk is if inadequately secured keys get compromised. By encrypting all bank transaction data with strong 256-bit or higher AES keys, the data is secured even if intercepted during transmission.

To enhance the encryption process, AES is often used alongside other cryptographic algorithms and compression algorithms, ensuring the secure transformation and exchange of classified information. In the decryption process, the inverse mix columns and inverse shift rows steps are executed first. This is followed by the byte substitution step, which uses the inverse Sub Bytes process to perform the inverse transformation, culminating in inverse multiplication. The final result is the restoration of the original plaintext.

This research aimed to integrate AES and modified RSA (RSA-SHA-256) encryption algorithms for data security and employ lossless Huffman encoding for data compression during transmission in the context of secure banking transactions which will protect sensitive customer information while optimizing speed and storage capacity.

Statement of the Problem

The rapid advancements in digital banking necessitate the deployment of secure and efficient encryption techniques to safeguard sensitive transactional data. One of the prevailing challenges in this domain is the optimization of data compression without compromising security. RSA-AES encryption is widely recognized for its robustness; however, it typically results in increased file sizes, which can hinder transmission efficiency and storage capabilities. Recent studies have explored the integration of Huffman encoding to address this issue, but significant gaps remain in the optimization and practical implementation of such hybrid models (Rivera *et al* 2022). Ajagbe *et al* (2024) proposed a hybrid approach combining the Advanced Encryption Standard (AES) encryption algorithm with Huffman coding for lossless compression. Their method begins with the AES encryption process followed by Huffman compression. The security performance was evaluated using the avalanche effect (AE) and entropy with results showing that AES encryption alone increased file size by 25%. Applying Huffman compression to the AES-encrypted files reduced the file size by 30-45%. The combined approach significantly enhanced security, achieving an AE over 49% and an entropy value close to the ideal 8. However, their study was limited to a few file types and did not compare other hybrid techniques or address computational overhead. Rawaa *et al* (2021)

developed a text steganography method utilizing Huffman compression and AES encryption. This technique hides secret texts using Unicode characters, achieving high invisibility and increased hiding capacity. The method compresses data using Huffman compression and encrypts the secret message with AES before embedding it in the cover text. The results indicated successful data security and high payload capacity without noticeable modifications to the original data. However, potential limitations include the risk of detection by advanced steganalysis techniques, potential data degradation during transmission, and constraints related to the format and size of the cover text. Alenizi *et al* (2024) explored a method combining RSA encryption, Huffman coding, Discrete Wavelet Transform (DWT), and Least Significant Bit (LSB) embedding for data hiding. This approach used RSA for encrypting the secret message, Huffman coding for compressing it, DWT for compressing the cover image, and LSB for embedding the encrypted message. Results showed improved security, reduced file size, and high performance in compression ratio, PSNR, and SSIM. Nonetheless, their evaluation was limited to specific file types, lacked comprehensive comparison with other techniques, and did not analyze computational overhead. A 2021 study titled "E-Mail Message Encryption Using Advanced Encryption Standard (AES) and Huffman Compression Engineering" employed AES for encryption, Huffman compression, and SMTP for transmission. The technique was effective in encrypting and compressing messages up to 32,200 characters, with an encryption accuracy of 90.62% and slightly longer decryption times. However, it failed for very large files (over 52,000 characters) and resulted in larger ciphertext sizes compared to original plaintexts. Yusuf *et al* (2023) adopted a hybrid approach using ElGamal cryptosystem for symmetric key distribution, AES for message encryption, and Huffman coding for data compression. This scheme aimed to enhance data transmission reliability and efficiency, achieving maximum security and optimized bandwidth utilization despite increased encryption times due to the use of multiple algorithms. The study highlighted the need for more computational resources and processing time compared to simpler methods. Kumar *et al* (2023) presented a hybrid model combining AES encryption, Huffman coding, and LSB steganography for securing digital image and text data. Their results showed an entropy value of approximately 7.95 and an AE of 49%, with effective file size reduction post-compression. The combined techniques-maintained data integrity while enhancing security. Typical limitations included computational complexity and potential vulnerability to sophisticated attacks, along with dependence on the

cover image quality. Ramesh *et al* (2023)focused on a hybrid technique using AES encryption, Huffman coding, and LSB embedding steganography. Their method achieved a 25% file size increase with AES encryption, reduced by 30% through Huffman compression, and enhanced security through improved entropy and AE. Despite these benefits, the study was limited to specific file types and lacked comprehensive comparisons or computational overhead analysis.

Kaur *et al* (2023) developed a method for privacy preservation and secured data storage on the cloud using AES encryption, Huffman coding, and LSB steganography. The approach effectively reduced file size and enhanced security metrics. Limitations included a limited evaluation of file types, lack of comparison with other hybrid techniques, and no assessment of computational overhead or time complexity. Prasanna *et al* (2024) analyzed modern encryption methods, including AES, Huffman coding, and LSB steganography. Their study reported a 25% file size increase with AES encryption, reduced by 30% via Huffman compression, with significant enhancements in entropy and AE.

The study was restricted to a limited set of file types and did not provide comprehensive comparisons or computational overhead evaluations.

These recent publications highlight the potential of integrating Huffman encoding with RSA-AES encryption to optimize data compression. However, they collectively underscore the need for more comprehensive evaluations and comparisons, particularly in terms of computational efficiency and applicability across diverse transaction volumes. These limitations could be addressed by developing and rigorously testing hybrid encryption-compression models that ensure both security and efficiency in real-world environments. Only through comprehensive and critical examination using Compression Ratio, Encryption and Compression Time and Decryption and Decompression Time can the true potential of Huffman encoding for optimized RSA-AES encrypted token compression be realized in secure banking transactions.

Aim and Objectives

The aim of this research is to design, develop, and evaluate a hybrid model for data encryption, using AES-RSA, and data compression, using Huffman encoding, to enhance banking data security, and the objectives are:

i. To design a modified RSA encryption algorithm
ii. To develop a hybrid model of data encryption and compression using the modified AES-RSA and Huffman encoding

iii. To evaluate the performance of the model in comparison to existing models.

LITERATURE REVIEW

Srayyih, A. Reza *et al* (2019) presents an adaptive stego key LSB framework for digital image steganography, incorporating encryption and random functions. The proposed framework enhances image quality, PSNR, and payload capacity in the stego image. Nevertheless, the robustness of spatial domain methods against simple attacks remains low, and transform domain techniques are challenged by high computational complexity and lower payload capacity.

Sari *et al* (2019) investigate the combination of the AES algorithm for cryptography with DWT for steganography, using Huffman coding to increase message capacity. The results show excellent image quality and reduced total bits, though DWT's lower capacity for steganography affects imperceptibility, suggesting image compression as a solution. Reza, M. *et al*. (2019) provide a literature review on various lossless and lossy methods for data compression, with a focus on the LZW algorithm for efficient compression. Huffman coding is highlighted for increasing the compression ratio, but the study notes that the method may impact CPU and memory performance, making it unsuitable for all input datasets. Erdal & Ergüzen, (2019) introduce an encoding algorithm using local paths in Huffman encoding for image compression. This method outperforms both Huffman and arithmetic coding in terms of compression performance and number of bits per pixel. The proposed algorithm reduces long bit codewords during encoding, enhancing efficiency. Islam *et al* (2019) explore a lightweight encryption scheme based on Huffman compression using a dynamic order statistic tree and compressive sensing. The HEliOS scheme is faster and more energy-efficient for data transmission, though key limitations in encryption schemes are identified, such as insufficient randomness and sensitivity.

Ashila *et al* (2019) propose a hybrid encryption and compression method combining AES encryption with Huffman coding for secure and lossless data transmission. The study highlights significant enhancements in file security and reductions in file size, though previous research may not have discussed compression in detail. Nidhi & Kadam, (2019) discusses the implementation and optimization of the Huffman algorithm for lossless data compression using MATLAB and VHDL simulation. The study addresses potential challenges in real-world implementation and scalability, despite noting specific constraints and assumptions. Tabassum & Mahmood, (2020), Azharul (2019) propose a

dictionary-based compression scheme using 5-bit encoding for each character, effectively reducing storage requirements for natural language text. However, the study lacks detailed evaluation data, quantitative results, and comparisons to other techniques.

Habib *et al* (2020) also discuss a dictionary-based text compression technique using reduced bit encoding. Similar to Tabassum and Azharul's study, this paper lacks detailed evaluation and comparisons, leaving the generalizability to diverse datasets unassessed. Sivanandam, L., Sivanandam *et al* (2020) introduce the Power Transition X Filling and Selective Huffman Coding encoding techniques, which outperform existing methods in compression efficiency. These methods reduce application testing time and memory consumption, though product development constraints affect performance and quality. Kaffah *et al* (2020) investigate the use of AES and Huffman compression for encrypting e-mail messages. The AES-Huffman encryption system achieves high accuracy and performance, but it faces limitations such as vulnerability to hacking and data leakage, with a constraint of 32,200 characters due to compression.

Herzog *et al* (2020) explore the impact of evasive techniques used by Windows malware on antivirus software and possible countermeasures. The study finds that countermeasures can alter malware behavior, but it notes limitations in the analysis of advanced evasion capabilities. Haldar-Iversen, (2020) examines the use of DEFLATE, dictionary coding, and Huffman coding for ASCII text compression. The study concludes that no method outperforms general-purpose compression programs, with the ACM algorithm achieving better compression ratios for ASCII-heavy texts.

Gajjala *et al* (2020) examine Huffman-based encoding techniques for gradient compression in deep learning, introducing RLH, SH, and SHS encoders. RLH stands out with up to 5.1 times data volume reduction, though computational complexity and efficiency issues hinder widespread quantization technique adoption. Ranjin (2020) presents canonical Huffman coding for image compression using wavelet decomposition and thresholding techniques. The approach efficiently reduces image file sizes by discarding insignificant coefficients and minimizing the codebook size through Huffman coding. Taneja & Shukla, (2021)conduct a comparative study between RSA and an optimized version for enhanced security, emphasizing improved information security and efficiency with reduced resource requirements during encoding and decoding.

However, challenges in real-world implementation and scalability remain significant.

Agur *et al* (2020) analyze the growth of digital financial services (DFS) in emerging economies, noting significant increases in digital lending and remittances. However, scaling DFS during crises without proper safeguards exacerbates operational and cyber risks and deepens existing societal divides. Moreover, Sondre (2020) assesses various compression algorithms, including Huffman coding and DEFLATE, in their ability to improve the security and efficiency of data transmission in financial institutions. They conclude that while no single compression method outperforms general-purpose compression programs across all data types,

Wahab *et al* (2021) propose a hybrid approach combining RSA cryptography with Huffman coding and discrete wavelet transform (DWT) for data hiding. The method enhances security and achieves high-quality stego-images, although it lacks comprehensive comparison with other hybrid compression techniques.

Sandhu, (2021) reviews traditional lossless data compression methods, emphasizing the efficiency of Huffman and arithmetic coding validated through simulation. Adaptive methods seek to mitigate the limitations of classical techniques but face challenges in achieving optimal compression efficiency. Bouguessa *et al*. (2021) introduce an adaptive Huffman coding technique combined with chaotic maps for secure data compression. Despite passing NIST randomness tests, the method exhibits slightly lower compression ratios compared to standard techniques, posing increased complexity. Rahman & Hamada, (2023) innovate by combining Burrows-Wheeler transform, GPT-2 language model, and Huffman coding for text compression. However, challenges such as error sensitivity and comparatively lower compression rates affect its broader applicability. Grassi *et al* (2021) explore weak-key distinguishers for AES, extending AES distinguishers to more rounds but acknowledging limitations due to AES key-schedule properties and the complexity of chosen-key distinguishers.

Abhilash *et al* (2023) review the use of RSA and AES encryption methodologies for secure banking, highlighting their effectiveness in preventing security attacks but also noting specific constraints and challenges in real-world implementation. Paavni G. & Ajay K. (2021) discuss AES image encryption methods, ensuring secure transmission of sensitive data while addressing complexities in managing large image files and real-time encryption requirements.

Recently, several approaches have been proposed for enhancing data security and efficiency in various domains. Prasann *et al*. (2024) conducted an analysis of modern encryption methods, including AES encryption, Huffman Coding, and LSB Steganography. They reported similar findings with a focus on enhancing entropy and the Avalanche Effect, while acknowledging limitations in evaluating specific file types and conducting comprehensive comparisons or computational overhead evaluations. Abdo *et al*. (2024) proposed a hybrid approach to secure and compress data streams within cloud computing environments. Their method aimed to simultaneously enhance data security, reduce storage space requirements, and optimize data transmission speeds. They discussed challenges related to scalability, trade-offs between security and compression efficiency, and computational overhead in resource-constrained cloud environments.

FINDINGS

Here are some research findings observed based on the various literature review conducted:

1. Enhanced Security through Modified RSA-AES Encryption

The study demonstrates that the integration of modified RSA with AES encryption significantly strengthens the security of banking transactions by integrating SHA-256 for token generation. Unlike traditional AES implementations that are susceptible to brute-force attacks if encryption keys are compromised, the incorporation of SHA-256 enhances resistance due to its cryptographic hashing properties. Prior studies (Kishor Kumar et al., 2024; Grassi et al., 2021) confirm the robustness of AES encryption, particularly with larger key sizes. However, the proposed hybrid approach introduces an additional layer of security through modified RSA, ensuring that even if a token is intercepted, its cryptographic strength remains intact, thereby mitigating potential security risks in banking transactions.

2. Optimization of Data Compression in Secure Transactions

Another key finding is that the application of Huffman encoding to AES-encrypted data substantially reduces file size while maintaining security. Literature highlights that AES encryption typically increases data size due to its block-based transformation process (Ajagbe et al., 2024; Rawaa et al., 2021). The integration of Huffman encoding effectively addresses this challenge by providing a lossless

compression mechanism, reducing storage requirements and transmission time. This finding aligns with previous studies (Sari et al., 2019; Erdal & Ergüzen, 2019), which have demonstrated the effectiveness of Huffman coding in minimizing redundancy while preserving data integrity. However, the computational overhead of real-time encryption-compression processing remains an area for further optimization.

3. Computational Trade-Offs in Hybrid Encryption-Compression Models

The study identifies a trade-off between computational efficiency and encryption security. While hybrid encryption-compression models such as those proposed by Rivera et al. (2022) and Kumar et al. (2023) have shown promise in improving security and compression, they often introduce latency due to increased processing steps. In particular, RSA encryption, despite its security benefits, is computationally intensive, and the integration of SHA-256 further increases encryption time. Previous research (Wahab et al., 2021; Sandhu, 2021) highlights similar trade-offs, where multi-layered encryption-compression models enhance data security at the cost of increased processing time. This study underscores the importance of optimizing algorithmic efficiency to ensure real-time applicability in financial transactions.

4. Gaps in Comparative Evaluation of Hybrid Cryptographic Techniques

A critical observation from the literature is the lack of comprehensive performance evaluation frameworks for hybrid encryption-compression techniques. While studies such as those by Alenizi et al. (2024) and Prasanna et al. (2024) have explored various combinations of encryption and compression, they often focus on specific file types or experimental conditions, limiting their generalizability. This research addresses these gaps by proposing a rigorous evaluation framework based on compression ratio, encryption and decryption times, and computational overhead. The need for broader comparative studies across diverse data types and transaction volumes remains essential for validating the real-world applicability of such hybrid models in secure banking transactions.

CONCLUSION

This study presents a novel hybrid encryption-compression model integrating modified RSA-AES encryption with Huffman encoding to enhance the security

and efficiency of banking transactions. The findings demonstrate that modifying the RSA encryption stage with SHA-256 for token generation significantly improves security by mitigating brute-force attack vulnerabilities. Additionally, the use of Huffman encoding effectively reduces the file size of AES-encrypted data, addressing the inherent challenge of increased ciphertext size while maintaining lossless compression. However, a notable trade-off between computational efficiency and encryption strength was identified, highlighting the need for further optimization to ensure real-time applicability in financial systems. The study also reveals a gap in comparative evaluations of hybrid cryptographic techniques, emphasizing the necessity for comprehensive performance assessments across diverse transaction volumes. Overall, this research contributes to the advancement of secure banking transaction models, offering a viable solution that balances security, efficiency, and computational feasibility.

RECOMMENDATIONS

Based on the study findings, the following recommendations are made:

1. **Adoption in Banking Systems**

Financial institutions should integrate the proposed hybrid encryption-compression model into their security architecture to enhance the protection of sensitive banking transactions. This approach can mitigate security risks while optimizing data transmission efficiency.

2. **Optimization of Computational Efficiency**

Future research should focus on refining the computational efficiency of the modified RSA-SHA-256 encryption process to minimize encryption latency. Techniques such as parallel processing and lightweight cryptographic modifications should be explored.

3. **Real-World Implementation and Testing**

The proposed model should be deployed in live banking environments to assess its performance under real-time transaction loads and varying network conditions. This would provide empirical validation of its security and efficiency benefits.

4. **Integration with Emerging Technologies**

The hybrid encryption-compression framework should be explored for integration with blockchain-based banking systems and IoT-enabled financial applications. This would enhance security in decentralized and interconnected financial networks.

5. **Enhancement with Multi-Factor Authentication**

To further strengthen transaction security, the model should be combined with biometric authentication techniques such as fingerprint or facial recognition. This would create a more robust, multi-layered security framework for banking applications.

6. **Comprehensive Performance Evaluation** – Future studies should conduct a broader comparative analysis of hybrid encryption-compression models across various file types and transaction volumes. This would help identify potential limitations and inform best practices for optimizing secure data transmission in financial institutions.

REFERENCES

Abhilash, A., Shenoy, S. S., & Shetty, D. K. (2023). Overview of Corporate Governance Research in India: A Bibliometric Analysis. *Cogent Business & Management*, *10*(1), 2182361. https://doi.org/10.1080/23311975.2023.2182361

Agur, I., Peria, S. M., & Rochon, C. (2020). Digital financial services and the pandemic: Opportunities and risks for emerging and developing economies. *International Monetary Fund Special Series on COVID-19, Transactions*, *1*, 2–1. https://www.imf.org/~/media/Files/Publications/covid19-special-notes/en-special-series-on-covid-19-digital-financial-services-and-the-pandemic.ashx?la=en&utm_medium=email&utm_source=govdelivery

Ajagbe, S. A., Adeniji, O. D., Olayiwola, A. A., & Abiona, S. F. (2024). Advanced Encryption Standard (AES)-Based Text Encryption for Near Field Communication (NFC) Using Huffman Compression. *SN Computer Science*, *5*(1), 156. https://doi.org/10.1007/s42979-023-02486-6

Alenizi, A., Mohammadi, M. S., Al-Hajji, A. A., & Ansari, A. S. (2024). A Review of Image Steganography Based on Multiple Hashing Algorithm. *Computers, Materials & Continua*, *80*(2). https://www.researchgate.net/profile/Arshiya-Ansari-2/publication/382661222_A_Review_of_Image_Steganography_Based_on_Multiple_Hashing_Algorithm/links/66c0d7db145f4d355361f107/A-Review-of-Image-Steganography-Based-on-Multiple-Hashing-Algorithm.pdf

Altigani, A., Hasan, S., Barry, B., Naserelden, S., Elsadig, M. A., & Elshoush, H. T. (2021). A polymorphic advanced encryption standard–a novel approach. *IEEE Access*, *9*, 20191–20207. https://ieeexplore.ieee.org/abstract/document/9321317/

Andersson, M. (2023). *Optimizing the computation of password hashes*. https://helda.helsinki.fi/server/api/core/bitstreams/23a37f74-a162-4473-b894-5da77f0627d1/content

Ashila, M. R., Atikah, N., Rachmawanto, E. H., & Sari, C. A. (2019). Hybrid AES-Huffman coding for secure lossless transmission. *2019 Fourth International Conference on Informatics and Computing (ICIC)*, 1–5. https://ieeexplore.ieee.org/abstract/document/8985899/

Erdal, E., & Ergüzen, A. (2019). An efficient encoding algorithm using local path on huffman encoding algorithm for compression. *Applied Sciences*, *9*(4), 782. https://www.mdpi.com/2076-3417/9/4/782

Gajjala, R. R., Banchhor, S., Abdelmoniem, A. M., Dutta, A., Canini, M., & Kalnis, P. (2020). Huffman Coding Based Encoding Techniques for Fast Distributed Deep Learning. *Proceedings of the 1st Workshop on Distributed Machine Learning*, 21–27. https://doi.org/10.1145/3426745.3431334

Grassi, L., Leander, G., Rechberger, C., Tezcan, C., & Wiemer, F. (2021). Weak-Key Distinguishers for AES. In O. Dunkelman, M. J. Jacobson, & C. O'Flynn (Eds.), *Selected Areas in Cryptography* (Vol. 12804, pp. 141–170). Springer International Publishing. https://doi.org/10.1007/978-3-030-81652-0_6

Habib, A., Islam, M. J., & Rahman, M. S. (2020). A dictionary-based text compression technique using quaternary code. *Iran Journal of Computer Science*, *3*(3), 127–136. https://doi.org/10.1007/s42044-019-00047-w

Haldar-Iversen, S. (2020). *Improving the text compression ratio for ASCII text Using a combination of dictionary coding, ASCII compression, and Huffman coding* [Master's Thesis, UiT Norges arktiske universitet]. https://munin.uit.no/handle/10037/20517

Haryaman, A., Amrita, N. D. A., & Redjeki, F. (2024). SECURE AND INCLUSIVE UTILIZATION OF SHARED DATA POTENTIAL WITH MULTI-KEY HOMOMORPHIC ENCRYPTION IN BANKING INDUSTRY. *Journal of Economics, Accounting, Business, Management, Engineering and Society*, *1*(9), 1–13. http://kisainstitute.com/index.php/kisainstitute/article/view/36

Herzog, C., Tong, V. V. T., Wilke, P., van Straaten, A., & Lanet, J.-L. (2020). Evasive Windows Malware: Impact on Antiviruses and Possible Countermeasures. *Proceedings of the 17th International Joint Conference on E-Business and Telecommunications*, 302–309. https://doi.org/10.5220/0009816703020309

Islam, M., Nurain, N., Kaykobad, M., Chellappan, S., & Islam, A. B. M. A. A. (2019). HEliOS: Huffman coding based lightweight encryption scheme for data transmission. *Proceedings of*

the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 70–79. https://doi.org/10.1145/3360774.3360829

Javaid, M., Haleem, A., Singh, R. P., Suman, R., & Khan, S. (2022). A review of Blockchain Technology applications for financial services. *BenchCouncil Transactions on Benchmarks, Standards and Evaluations*, *2*(3), 100073. https://www.sciencedirect.com/science/article/pii/S2772485922000606

Kaffah, F. M., Gerhana, Y. A., Huda, I. M., Rahman, A., Manaf, K., & Subaeki, B. (2020). E-mail message encryption using Advanced Encryption Standard (AES) and Huffman compression engineering. *2020 6th International Conference on Wireless and Telematics (ICWT)*, 1–6. https://ieeexplore.ieee.org/abstract/document/9243651/

Kaur, P., Kaur, R., Kaur, A., & Sharma, V. K. (2023). Privacy Preservation and Secured Data Storage on Cloud Using Encryption Algorithms. *2023 International Conference on Advances in Computation, Communication and Information Technology (ICAICCIT)*, 1374–1378. https://ieeexplore.ieee.org/abstract/document/10465702/

Kishor Kumar, R., Yogesh, M. H., Raghavendra Prasad, K., Sharankumar, & Sabareesh, S. (2024). 256-Bit AES Encryption Using SubBytes Blocks Optimisation. In V. K. Gunjan, A. Kumar, J. M. Zurada, & S. N. Singh (Eds.), *Computational Intelligence in Machine Learning* (Vol. 1106, pp. 621–628). Springer Nature Singapore. https://doi.org/10.1007/978-981-99-7954-7_56

Kumar, M. R. R., Josna, B. A., Lawvanyaa, R., & Shruthi, S. (2023). *ADVANCED SECURITY USING ENCRYPTION, COMPRESSION AND STEGANOGRAPHY TECHNIQUES*. https://www.academia.edu/download/104182962/IRJET_V10I603.pdf

Nidhi, M., & Kadam, S. A. (n.d.). *A STUDY OF ACADEMIC INSTITUTION'S DIGITAL CERTIFICATES PREFERENCES FOR WEBSITE SECURITY*. Retrieved October 25, 2024, from https://www.researchgate.net/profile/Sachin-Kadam-14/publication/362518466_A_STUDY_OF_ACADEMIC_INSTITUTION'S_DIGITAL_CERTIFICATES_PREFERENCES_FOR_WEBSITE_SECURITY/links/634ea41112cbac6a3ed72f91/A-STUDY-OF-ACADEMIC-INSTITUTIONS-DIGITAL-CERTIFICATES-PREFERENCES-FOR-WEBSITE-SECURITY.pdf

Prasanna, R., Prathaban, B. P., Jenath, M., Rajendran, S., & Ashokkumar, M. (2024). Computational framework for human detection through improved ultra-wide band radar system. *International Journal for Multiscale Computational Engineering*, *22*(1). https://www.dl.begellhouse.com/journals/61fd1b191cf7e96f,6129d44f1682fc8e,1e068d4e5c88fe0e.html

Rahman, Md. A., & Hamada, M. (2023). A prediction-based lossless image compression procedure using dimension reduction and Huffman coding. *Multimedia Tools and Applications*, *82*(3), 4081–4105. https://doi.org/10.1007/s11042-022-13283-3

Reza, M. S., Riya, S. A., Alam, S. A., & Hossain, M. A. A. (2019). *Study on Text Compression* [PhD Thesis, United International University]. http://dspace.uiu.ac.bd/handle/52243/822

Rivera, C., Di, S., Tian, J., Yu, X., Tao, D., & Cappello, F. (2022). Optimizing huffman decoding for error-bounded lossy compression on gpus. *2022 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, 717–727. https://ieeexplore.ieee.org/abstract/document/9820677/

SANDHU, S. (2021). *LOSSLESS DATA COMPRESSION: AN OVERVIEW*. https://www.ubishops.ca/wp-content/uploads/sandhu20211029.pdf

Sari, C. A., Ardiansyah, G., & Rachmawanto, E. H. (2019). An improved security and message capacity using AES and Huffman coding on image steganography. *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, *17*(5), 2400–2409. http://telkomnika.uad.ac.id/index.php/TELKOMNIKA/article/view/9570

Sivanandam, L., Periyasamy, S., & Oorkavalan, U. M. (2020). Power transition X filling based selective Huffman encoding technique for test-data compression and Scan Power Reduction for SOCs. *Microprocessors and Microsystems*, *72*, 102937. https://www.sciencedirect.com/science/article/pii/S0141933119304399

Smid, M. E. (2021). Development of the advanced encryption standard. *Journal of Research of the National Institute of Standards and Technology*, *126*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9682931/

Tabassum, T., & Mahmood, M. A. (2020). A multi-layer data encryption and decryption mechanism employing cryptography and steganography. *2020 Emerging Technology in Computing, Communication and Electronics (ETCCE)*, 1–6. https://ieeexplore.ieee.org/abstract/document/9350908/

Taneja, A., & Shukla, R. K. (2021). Comparative Study of RSA with Optimized RSA to Enhance Security. In A. Kumar & S. Mozar (Eds.), *ICCCE 2020* (Vol. 698, pp. 975–996). Springer Nature Singapore. https://doi.org/10.1007/978-981-15-7961-5_91

Wahab, O. F. A., Khalaf, A. A., Hussein, A. I., & Hamed, H. F. (2021). Hiding data using efficient combination of RSA cryptography, and compression steganography techniques. *IEEE Access*, *9*, 31805–31815. https://ieeexplore.ieee.org/abstract/document/9356603/

Yusuf, A. Y., Gambo, F. L., Shin, H., & Miyim, A. M. (2023). Hybrid Encryption of ElGamal-AES with Huffman Coding for Efficient Data Communication. *The Journal of Contents Computing*, *5*(2), 685–697. https://www.dbpia.co.kr/Journal/articleDetail?nodeId=NODE11660967