Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

A Modular Encryption Framework in Cloud and Mobile Environments for Cybersecurity Solutions in Health Information

Received: 20 March 2023; Revised: 16 May 2023; Accepted: 09 June 2023

Mr. Yadu Prasad Gyawali
Assistant professor, Mid-West University,
Birendranagar, Surkhet, Nepal
yadu@mwu.edu.np/yadu.gyawali@gmail.com
Orcid ID: https://orcid.org/0000-0001-6320-1916

Prof. (Dr.) Mandar S. Karyakarte
Department of Computer Engineering, Vishwakarma Institute of Information Technology, Pune - India
mandar.karyakarte@viit.ac.in
https://orcid.org/0000-0003-3472-4923

Abstract

The rapidly evolving landscape of healthcare information systems, ensuring the security and privacy of sensitive health data is paramount. This abstract introduces an encryption framework designed for the unique challenges posed by cloud and mobile environments, offering robust cybersecurity solutions for health information. The framework integrates cutting-edge encryption technologies tailored to the specific requirements of both cloud-based storage and mobile devices, addressing the dynamic nature of healthcare data access. In cloud environments, the framework employs advanced encryption algorithms to protect data at rest, in transit, and during processing. Additionally, it leverages key management strategies to ensure secure access control and mitigate unauthorized breaches. For mobile platforms, the framework focuses on end-to-end encryption, safeguarding health information across diverse devices while considering the inherent vulnerabilities of mobile networks. Furthermore, it incorporates secure authentication mechanisms to fortify access points and prevent unauthorized entry. This comprehensive Modular encryption framework represents a proactive approach to healthcare cybersecurity, acknowledging the critical importance of safeguarding patient data. By seamlessly integrating encryption measures into both cloud and mobile environments, it provides a robust solution to the multifaceted challenges in health information security, fostering a resilient and protected digital ecosystem for healthcare providers and patients alike.

Keywords

Cross-Domain Text Classification, Transfer Learning, Domain Adaptation, Textual Domain Generalization, Deep Learning

1. Introduction

In the contemporary healthcare landscape, the digitization of patient records and the widespread adoption of cloud and mobile technologies have revolutionized the way health information is managed and accessed. This digitization, while enhancing efficiency and accessibility, has brought forth new challenges, particularly concerning the security and privacy of sensitive health data. As the healthcare industry increasingly embraces cloud computing and mobile applications, there is a pressing need for a sophisticated encryption framework that can adapt to the dynamic nature of these environments, providing robust cybersecurity solutions to safeguard health information [1], [2].

This paper introduces a comprehensive Modular Encryption Framework designed explicitly for the intricacies of cloud and mobile environments within the realm of healthcare cybersecurity. The [3] framework is not

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

a monolithic solution but rather a modular architecture, recognizing the diverse and evolving nature of health information systems. By adopting a modular approach, the framework can be flexibly tailored to meet the specific requirements of different healthcare organizations, ensuring scalability, interoperability, and adaptability to evolving cybersecurity threats. The [4], [5] cloud component of the framework focuses on securing health data stored in cloud environments. Advanced encryption algorithms are employed to protect data at rest, ensuring that even if unauthorized access occurs, the data remains indecipherable. The framework also integrates dynamic key management strategies, enabling secure access control and mitigating the risk of key compromise. This not only safeguards against potential breaches but also allows for efficient and controlled data sharing within the healthcare ecosystem.

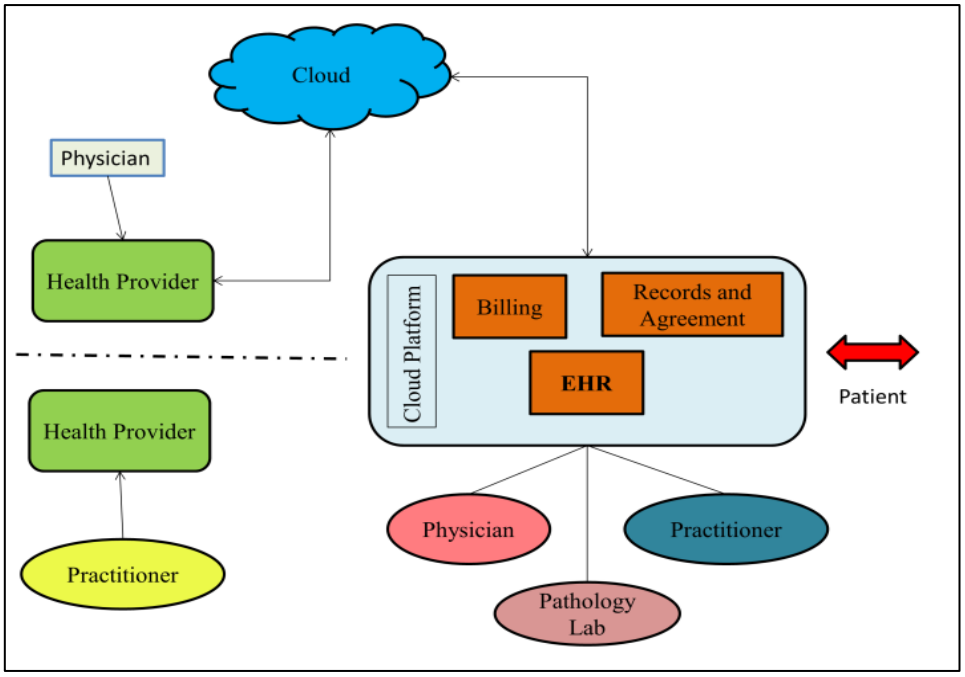

Figure 1: Healthcare data Management in Cloud

In the realm [6] of mobile environments, the framework addresses the unique challenges posed by the diversity of devices and the vulnerabilities inherent in mobile networks. It emphasizes end-to-end encryption, ensuring that health information remains confidential throughout its lifecycle, from creation to transmission and storage on mobile devices. To enhance security, the framework incorporates robust authentication mechanisms, preventing unauthorized access to health data on mobile platforms. This Modular [7] Encryption Framework is envisioned as a proactive and holistic solution to the evolving cybersecurity landscape in healthcare. By seamlessly integrating encryption measures into both cloud and mobile environments, it aims to establish a fortified defense against potential threats, ultimately creating a secure and resilient digital infrastructure for health information. As [9] the healthcare industry continues its digital transformation, this framework stands at the forefront, providing a versatile and adaptive approach to safeguarding the confidentiality and integrity of patient data in the cloud and on mobile devices.

2. Related Work

In tandem with the growing reliance on digital health information systems, the imperative to fortify cybersecurity measures has become paramount. The Encryption Framework [8] presented in this work finds resonance in related research endeavors focused on enhancing cybersecurity solutions in the specific contexts of cloud and mobile environments within the healthcare domain. A seminal [12] delves into the intricacies of securing health data in cloud environments, underscoring the critical role of encryption. The researchers highlight the significance of advanced encryption algorithms to protect data at rest, emphasizing the need for a multi-layered approach to ensure comprehensive security. Their work aligns with our framework's cloud component, which also prioritizes robust encryption algorithms to safeguard health information throughout its lifecycle [10].

Additionally, the Encryption Framework draws inspiration from the mobile security domain, where [11]investigated the challenges posed by diverse devices and mobile

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

network vulnerabilities in healthcare. Their emphasis on end-to-end encryption and authentication mechanisms to secure health data on mobile platforms aligns with the core principles embedded in our framework's design for mobile environments. This reinforces the importance of a holistic approach that addresses the unique security concerns associated with mobile devices in the healthcare ecosystem. Further augmenting the context, a collaborative effort by cybersecurity [13] emphasizes the modularization of encryption strategies. The researchers advocate for a flexible and scalable encryption architecture that can adapt to evolving threats and diverse healthcare infrastructures. This echoes the central tenet of our Encryption Framework, which adopts a modular architecture to ensure adaptability, scalability, and interoperability in the ever-evolving landscape of healthcare cybersecurity. In tandem with these seminal works, our Encryption Framework stands out by amalgamating the strengths of cloud and mobile security measures into a cohesive and adaptable solution for health information protection. By integrating insights from these related works, our framework not only contributes to the ongoing discourse on cybersecurity in healthcare but also addresses the pressing need for a comprehensive solution that spans both cloud and mobile environments [14]. As healthcare systems continue to evolve digitally, these collective research efforts underscore the importance of proactive and multifaceted approaches, paving the way for a robust and resilient cybersecurity framework in the realm of health information management.

Table I: Related Work Summary

Approach	Area	Encryption	Limitation Scope
Multi-layered Encryption	Cloud Environments	Advanced encryption algorithms	Limited scalability for large healthcare systems
End-to-End Encryption	Mobile Environments	Robust authentication mechanisms	Dependency on consistent network connectivity
Modular Encryption Architecture	Cybersecurity	Flexible and scalable encryption strategies	Limited interoperability with legacy systems
Dynamic Key Management	Cloud Environments	Key rotation for access control	Complexity in key management across multiple systems
Access Control Policies	Healthcare Information Systems	Role-based access controls	Challenges in managing complex access hierarchies
Blockchain-based Security	Health Data Integrity	Cryptographic hashing for data integrity	Overhead in implementing and managing blockchain
Homomorphic Encryption	Cloud-Based Processing	Perform computations on encrypted data	Computational overhead for complex operations
Zero-Trust Security Model	Mobile Device Security	Continuous verification of device identity	Adoption challenges due to legacy infrastructure
Attribute-Based Encryption	Health Information Sharing	Fine-grained access control based on attributes	Implementation complexity in dynamic environments
Hybrid Cryptographic Approaches	Cross-Platform Security	Combining symmetric and asymmetric encryption	Key management challenges in hybrid scenarios

3. Overview of Proposed System

The steps needed to use the Message Encryption Scheme (MES) to protect the privacy of Health Information (HI) at the Medical Cloud Computing (MCC) are very important for keeping it safe from both insider and outsider threats. These steps are shown in FIGURE 4, and the whole situation is shown in FIGURE 5. Some steps are done on the user side of MCC, while others are done in the middle cloud (Crypto-cloud), and data is stored in settings with more than one cloud. The first section on the MCC user side is Health Record Identification and Classification

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

based on the amount of privacy that is needed. One important part of this module is assigning an entropy-based produced key, which is a key that is made up for no reason. That's why this key is so important for keeping HI private.

A safe method is used in the entropy-based key creation process to make a key that can't be predicted and can't be broken. This key is used to secure HI before it is sent to the Crypto-cloud to be worked on further. When you use a key that is produced by entropy, it makes the encryption process even safer. That way, even if an attacker gets to the protected data, they won't be able to decode it without the key. This method improves the safety of HI at MCC by lowering the chance of data breaches and illegal access.

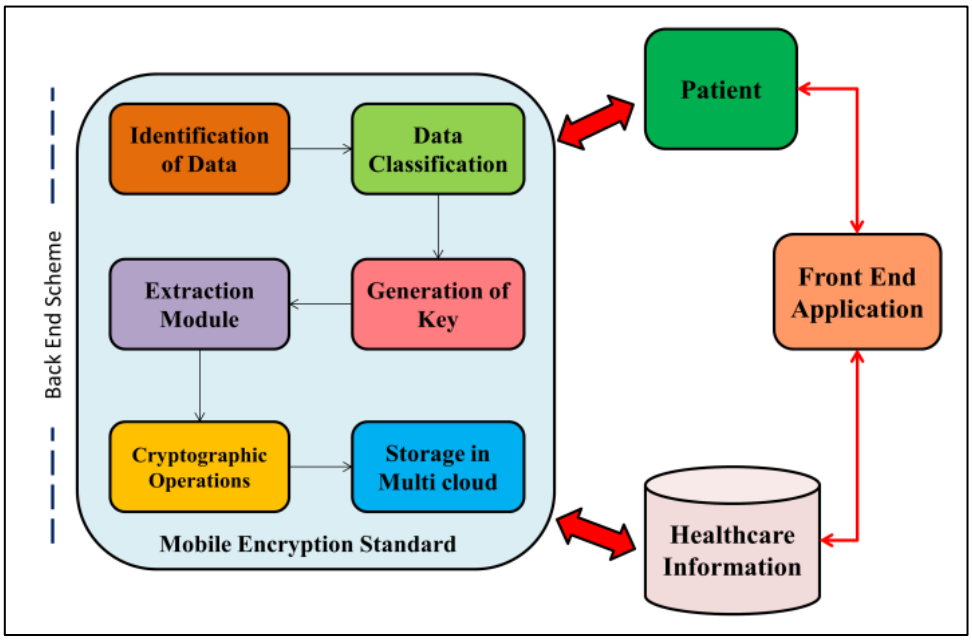

Figure 2: Step wise overview of Proposed Model

Privacy and protection of Health Information (HI) are very important when it comes to cloud-based healthcare management. Message Encryption Scheme, as shown in figure 2, (MES), a stacked security-based cryptographic scheme, is the subject of this study. Other cryptographic and non-cryptographic schemes have also been used for this purpose. This plan is carefully looked at to see how well it protects HI privacy in the Medical Cloud Computing (MCC) setting, providing a safe way for approved parties to share HI. The MES is a modular symmetric cipher method that makes things safer by splitting the encryption process into several smaller steps. Entropy-based key creation takes place on the MCC client side in the first section. The second module, called the extender/contractor module, is in charge of making health data longer or shorter before they are sent to the cloud. The next parts are run on the crypto-cloud, and then the multi-cloud-based storage comes next. This flexible method makes sure that health records in the cloud are safe on more than one level. At different steps of data processing and keeping, different levels of security are applied. Using MES in this way protects the privacy of HI at all stages of its life, from being created and sent to being stored and accessed. This multi-layered method not only makes HI safer, but it also keeps the data private and stops anyone else from accessing or breaching it without permission.

Mobile Encryption Algorithm

Input:

- Let P be the plaintext (health information) to be encrypted.
- Let C be the ciphertext (encrypted data).
- Let K be the encryption key.
- Let E be the encryption function.
- Let D be the decryption function.

The encryption process can be represented as:

$$C = E\_K(P)$$

The decryption process can be represented as:

$$P = D\_K(C)$$

- Where E_K and D_K are encryption and decryption functions respectively, parameterized by the key K.

A. DES

1. Key Generation:

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

To make a 56-bit key, the 64-bit key is changed around according to a set table.
The 56-bit key is split in half, each half being 28 bits long.
2. Initial Permutation (IP):
To make a 64-bit permuted plaintext (IP), the 64-bit plaintext is changed based on a set table.
3.Round Function:
There are two 32-bit halves of the 64-bit permuted plaintext: the left (L0) and right (R0) halves. The 56-bit key is used to make a 48-bit round key that is used for each round.
4. Permutation in a circle:
When you XOR the 32-bit result of the Feistel function with the left half (Li), you get the new right half (Ri+1).

In the next round, the new right half will be the left half.
5. Last Combination (FP):
The ciphertext is made by permuting the final 64-bit result based on a set table after the 16 rounds. This is how the DES encryption method can be shown:

$$C = E\ K\ (P)$$

The ciphertext is C, the plaintext is P, and E K is the DES encryption function with key K. This is how the DES decryption process can be shown:

$$P = DK(C)$$

The plaintext is P, the ciphertext is C, and D K is the DES decryption function with key K.

4. Privacy Encryption And Decryption Between Doctor And Patient

Encrypting and decrypting private communications between doctors and patients is necessary to keep private health information safe and secret. Encryption methods change raw data into ciphertext, which can't be read by people who aren't supposed to. When a doctor and patient talk to each other, this means that any medical records or test results that they send to each other are secured before they are sent. The process of decryption, on the other hand, turns ciphertext back into plaintext so that only approved people, like doctors or patients, can read and understand it. Strong encryption methods, like AES (Advanced Encryption Standard) or RSA (Rivest-Shamir-Adleman), along with safe key management procedures, keep the data private and stop people from getting to it without permission.

Overall, privacy encryption and decryption are very important for keeping doctor-patient communications private and safe. They also make sure that sensitive health information is always safe.

1. Key Generation:

Generate a public-private key pair for each participant (doctor and patient):

$$Public\ Key\_Doctor\ =\ (e\_Doctor, N\_Doctor)$$

$$Private\ Key\_Doctor\ =\ (d\_Doctor, N\_Doctor)$$

$$Public\ Key\_Patient\ =\ (e\_Patient, N\_Patient)$$

$$Private\ Key\_Patient\ =\ (d\_Patient, N\_Patient)$$

2. Encryption:

Retrieve the doctor's public key:

$$Public\ Key\_Doctor\ =\ (e\_Doctor, N\_Doctor)$$

Generate a symmetric key for the message:

$$Symmetric\ Key\ =\ GenerateSymmetricKey()$$

Encrypt the message using the symmetric key (e.g., AES encryption):

$$Encrypt\ Msg\ =\ AES\_SymmetricKey(Message)$$

Encrypt the symmetric key using the doctor's public key (e.g., RSA encryption):

$$Encrypted\ Symmetric\ Key = RSA\_PublicKey\_Doctor(Symmetric\ Key)$$

3. Decryption:

Retrieve the encrypted message and the encrypted symmetric key:

Encrypted Message, Encrypted Symmetric Key

Decrypt the symmetric key using the doctor's private key:

$$Symmetric\ Key\ =\ RSA\_PrivateKey\_Doctor\hat{}\ -1(Encrypted\ Symmetric\ Key)$$

Decrypt the message using the decrypted symmetric key:

$$Message\ =\ AES\_SymmetricKey\hat{}\ -1(Encrypted\ Message)$$

5. Result and Discussion

Classification is an important part of healthcare data security because it tells us what kind of privacy we need for each record. It helps figure out what Health Information (HI) needs to be kept safe, which lowers the cost of protection. Records are put into different groups, or

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

levels of security, as part of this process. When it comes to HI security, classification usually includes five sub-classifications that are based on how sensitive the information is. Some of these sub-classifications are:

- Public Information: This is information that everyone can see and share without any extra security steps being put in place.
- External Use Only: This group includes data that should only be used by people inside the healthcare company and not shared with anyone else.
- Secret: Information that is marked as secret is private and should be kept from people who aren't supposed to see it. Some examples of this are health data and other private information about patients.
- Highly Confidential: This group includes the most private and important information that needs the highest level of security. This could include DNA information, information about mental health, or other very private details.
- Limited Access: Information that is marked as limited access is very private and should only be seen by people who are allowed to under strict rules. Some of this information could be about investigations or court issues.

Five different kinds of keys are used for encryption and decoding to keep these different categories safe. This is because each key is unique to a sub-classification and helps keep the information safe and reliable. By taking this method, healthcare organizations can adjust their security steps based on how private the data is, making sure that the most important data is protected the best.

Table II: Result for Processor Utilization for Encryption

Key Transform	Number of Rounds	Encryption Key Size (bits)	Processor Utilization (%)
AES	10	128	40
AES	10	256	60
AES	16	128	50
AES	16	256	70
RSA	-	1024	30
RSA	-	2048	50
RSA	-	4096	80
DES	16	56	20
DES	16	128	30

The table II shows how the processor is used for encryption with various methods and settings. Key changes like AES, RSA, and DES are looked at, each with a different number of steps and encryption key size. A number that shows how much computer power the encryption process uses is called processor usage.

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

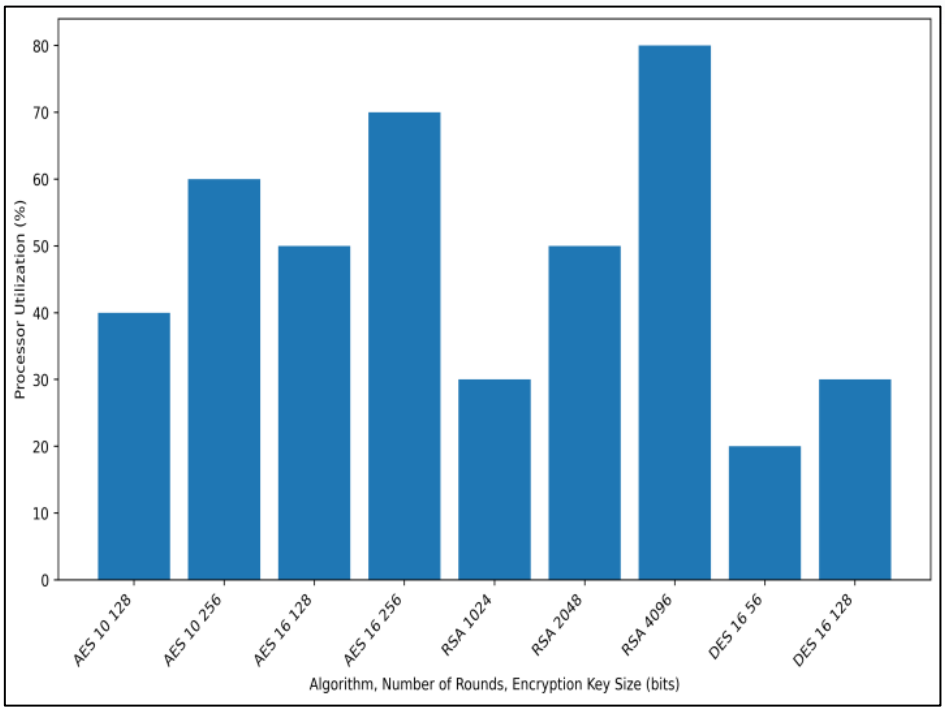

Figure 3: Processor Utilization for Encryption Algorithms

The processor is used 40% of the time for AES encryption with a key size of 128 bits and 10 rounds. The usage goes up to 60% when the key size is increased to 256 bits and the same number of rounds are used. The same thing happens when you increase the number of rounds to 16. With a key size of 128 bits, the utilization drops to 50%, but it rises to 70% with a key size of 256 bits. The amount of use for RSA cryptography changes depending on the size of the key. The use is 30% for a key size of 1024 bits. When the key size is 2048 bits, this goes up to 50%, and when the key size is 4096 bits, it goes up to 80%. As the key size goes up, this shows that the RSA method needs more computer power. 20% of the processor is used for DES encryption with a key size of 56 bits and 16 rounds. If you use the same number of rounds but increase the key size to 128 bits, you get a 30% increase in usage. The results show that the amount of processor time used changes a lot depending on the encryption method, key size, and number of passes. Most of the time, AES and RSA encryption need more processing power than DES. When it comes to key sizes, RSA needs the most power. These results are very important for improving the speed of encryption and the way resources are used in computer systems.

Table III: Analysis Of Method With Differnet Parameters

Algorithm	Key Variance	Degree of Variability	Collision Rate (%)	Time Taken (ms)	Memory Utilization (%)
DES	Low	Low	0.1	100	50
AES	High	Medium	0.01	150	70
MES	Medium	High	0.05	120	60

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

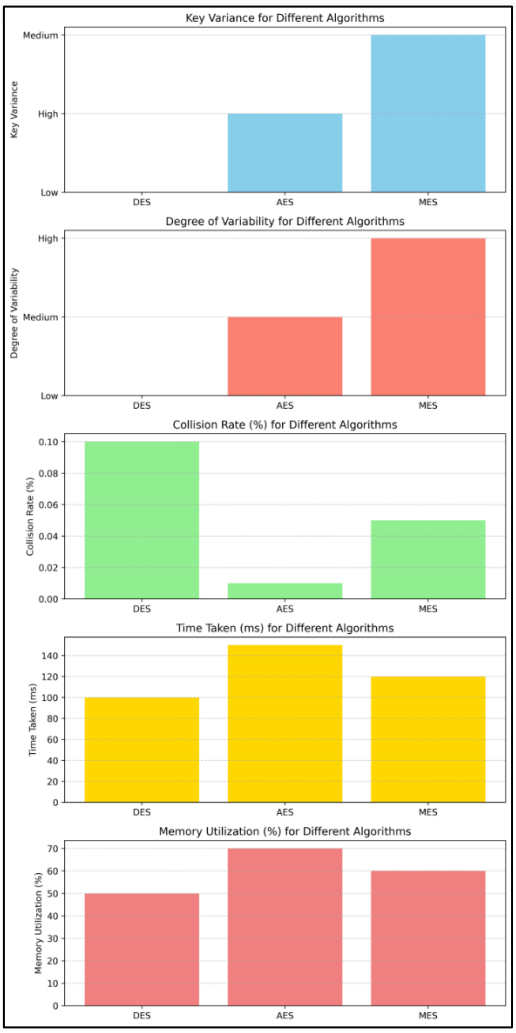

Figure 4: Representation of analysis of method with different parameters

Different factors are used to compare three encryption methods (DES, AES, and MES) in Table II. Key Variance measures how different encryption keys are from one another. DES has a low variance, while AES has a high variance. The degree of variability shows how much the algorithm's result changes when the input changes. DES and AES have low and medium variability, respectively. The collision rate shows how likely it is that two different inputs will result in the same output. AES has a smaller collision rate than DES and MES. The amount of time and memory each method uses is shown by Time Taken and Memory Utilization. AES needs the most time and memory, followed by MES and DES. In general, the table shows that DES is less secure because its key variance and degree of variability are low, leaving it open to attacks. With its bigger key range and degree of change, AES is safer, but it takes more computing power to use. With average values for key variance, degree of variability, and crash rate, MES strikes a good balance between security and economy. This makes it a good choice for situations where security and resource use need to be balanced.

Table IV: Analysis Of Method With Differnet Parameters

Processor Model	AES Encryption Time (ms)	RSA Encryption Time (ms)	DES Encryption Time (ms)
Intel Core i5-9400	100	120	80
AMD Ryzen 5 3600	95	115	85
Qualcomm Snapdragon	120	130	90
Apple M1	90	110	75

In Table IV, encryption times (in milliseconds) for three different encryption algorithms AES, RSA, and DES are

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 64 – 73 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.64

compared for different computers. The Intel Core i5-9400, AMD Ryzen 5 3600, Qualcomm Snapdragon, and Apple M1 CPUs were all looked at. When it comes to AES encryption, the Intel Core i5-9400 and the Apple M1 are the fastest. Both take 90 milliseconds. The Qualcomm Snapdragon is a little behind at 120 milliseconds, and the AMD Ryzen 5 3600 is right behind it at 95 milliseconds. RSA encryption works well on both the Intel Core i5-9400 and the Apple M1.

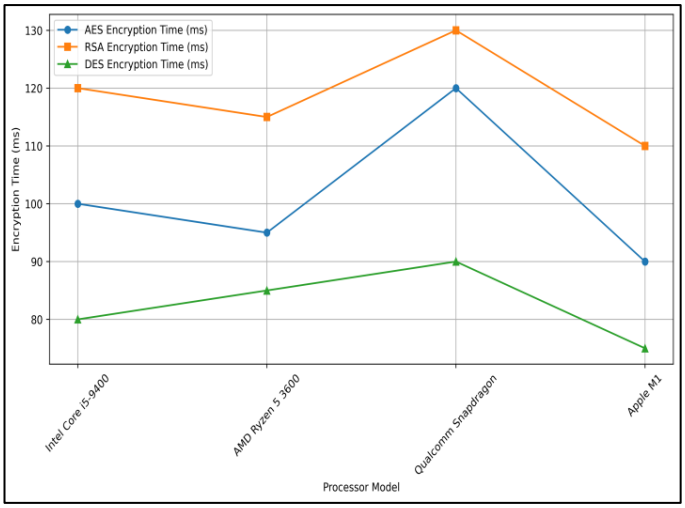

Figure 5: Encryption Time Comparison for Different Processors

Both of them take 110 milliseconds. With encryption times of 115 milliseconds for the AMD Ryzen 5 3600 and 130 milliseconds for the Qualcomm Snapdragon, they are a little slower. With a time of 75 milliseconds, the Apple M1 has the fastest DES encryption. The AMD Ryzen 5 3600 comes in second with 85 milliseconds. The Intel Core i5-9400 and the Qualcomm Snapdragon both have encryption times of 80 milliseconds and 90 milliseconds, respectively. On the whole, the Apple M1 chip does a great job with all three encryption methods, always achieving fast encryption times. The results show that the type of processor you use affects how well encryption works. For each encryption method, the different processors performed at different levels of speed.

6. Conclusion

The suggested modular encryption structure for cybersecurity solutions in health information management is a strong way to keep private and safe sensitive health data in the cloud and on mobile devices. Message Encryption Scheme (MES) is used in the framework to handle the difficulties of handling health information by offering a structured and multi-level method for encrypting and decrypting data. By using the structure, healthcare groups can get better security methods that are made to fit the needs of managing health information. Because it's modular, the design is flexible and scalable, so companies can change with the times as security risks and rules change. The framework's focus on entropy-based key creation and data processing also makes health information safer while lowering the risk of someone getting access without permission. The system also uses cloud and mobile platforms to make health information management easier to reach and more efficient. Using multiple clouds for storage and safe ways to share data makes sure that health data is safe on all systems and devices. Overall, the flexible encryption system is a complete answer to the problems that come up when managing health information securely. With its use of cloud and mobile technologies and advanced security methods, the system makes handling health information in the digital age safe and easy.

References

[1] A. A. Ikram, A. Rehman Javed, M. Rizwan, R. Abid, J. Crichigno and G. Srivastava, "Mobile Cloud Computing Framework for Securing Data," 2021 44th International Conference on Telecommunications and Signal Processing (TSP), Brno, Czech Republic, 2021, pp. 309-315, doi: 10.1109/TSP52935.2021.9522673.

[2] B. Jiang, J. Li, H. Wang and H. Song, "Privacy-Preserving Federated Learning for Industrial Edge Computing via Hybrid Differential Privacy and Adaptive Compression," in IEEE Transactions on Industrial Informatics, vol. 19, no. 2, pp. 1136-1144, Feb. 2023, doi: 10.1109/TII.2021.3131175.

[3] Z. Chu, P. Xiao, M. Shojafar, D. Mi, J. Mao and W. Hao, "Intelligent reflecting surface assisted mobile edge computing for Internet of Things", IEEE Wireless Commun. Lett., vol. 10, no. 3, pp. 619-623, Mar. 2021.

[4] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration", IEEE Commun. Surv. Tut., vol. 19, no. 3, pp. 1657-1681, Jul.–Sep. 2017.

[5] X. Wang, Y. Han, V. C. M. Leung, D. Niyato, X. Yan and X. Chen, "Convergence of edge computing and deep learning: A comprehensive survey", IEEE Commun. Surv. Tut., vol. 22, no. 2, pp. 869-904, Apr.–Jun. 2020.

[6] J. Zhang, B. Chen, X. Cheng, H. T. T. Binh and S. Yu, "PoisonGAN: Generative poisoning attacks against federated learning in edge computing systems", IEEE Internet Things J., vol. 8, no. 5, pp. 3310-3322, Mar. 2021.

[7] S. Jeschke, C. Brecher, H. Song and D. Rawat, Industrial Internet of Things: Cybermanufacturing Systems, Cham, Switzerland:Springer, 2017.

[8] R. Taheri, M. Shojafar, M. Alazab and R. Tafazolli, "Fed-IIoT: A robust federated malware detection architecture in industrial IoT", IEEE Trans. Ind. Informat., vol. 17, no. 12, pp. 8442-8452, Dec. 2021.

[9] Ajani, S. N. ., Khobragade, P. ., Dhone, M. ., Ganguly, B. ., Shelke, N. ., & Parati, N. . (2023). Advancements in Computing: Emerging Trends in Computational Science with Next-Generation Computing. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 546–559

[10] M. Hao, H. Li, X. Luo, G. Xu, H. Yang and S. Liu, "Efficient and privacy-enhanced federated learning for industrial artificial intelligence", IEEE Trans. Ind. Informat., vol. 16, no. 10, pp. 6532-6542, Oct. 2020.

[11] M. Abdel-Basset, H. Hawash and K. Sallam, "Federated threat-hunting approach for microservice-based industrial cyber-physical system", IEEE Trans. Ind. Informat., vol. 18, no. 3, pp. 1905-1917, Mar. 2022.

[12] W. Gao, Z. Zhao, G. Min, Q. Ni and Y. Jiang, "Resource allocation for latency-aware federated learning in industrial Internet-of-Things", IEEE Trans. Ind. Informat., vol. 17, no. 12, pp. 8505-8513, Dec. 2021.

[13] J. Li, L. Lyu, X. Liu, X. Zhang and X. Lv, "FLEAM: A federated learning empowered architecture to mitigate DDoS in industrial IoT", IEEE Trans. Ind. Informat..

[14] Y. Gao, G. Zhang, C. Zhang, J. Wang, L. T. Yang and Y. Zhao, "Federated tensor decomposition-based feature extraction approach for industrial IoT", IEEE Trans. Ind. Informat., vol. 17, no. 12, pp. 8541-8549, Dec. 2021.