Transparent Encryption for External Storage Media with Mobile-Compatible Key Management by Crypto Ciphershield

S.RAJU [1], D.SINDHUJA [2]

[1] Dept of Information Technology,
Mahendra Engineering College (Autonomous),Namakkal. Tamilnadu,India.
rasakudil@gmail.com
[2] Mahendra Engineering College (Autonomous)Namakkal Tamilnadu,India.
sindhujad@mahendra.info

ABSTRACT

Evidently, there is a widespread adoption of external storage devices in contemporary situations, including USB sticks, SD cards, and flash memory devices. If, on the other hand, these devices are misplaced or lost, the presence of sensitive data on them can provide a potential security risk. The currently available encryption solutions frequently have usability issues, which necessitate that users continually input keys or login credentials across a variety of portable devices. The study offers a solution that overcomes these difficulties by integrating key caching with time-delayed deletion within the MobileShielded Encryption Framework (MoSEF). This method is our reaction to the problem. There are two distinct iterations of this idea that we have developed. The initial version, which functions flawlessly even in the absence of deliberate user intervention, achieves this by restricting the use of external storage to only temporary data transfers.Within the file system, the second variant makes it possible to handle numerous encryption keys for different files. This is made possible by a trusted host on the file system. By eliminating the need to share keys or passwords, timed key caching significantly improves both the level of security and the level of usability of a system. In addition to ensuring interoperability with mobile devices, our solution, which is incorporated within the MoSEF architecture, provides plaintext encryption for external storage media. To reduce the likelihood of data breaches and illegal access, this strategy offers users a mechanism that is both convenient and safe for managing sensitive data when they are on the move.

Keywords: Transparent encryption; Mobile-compatible; External storage media; Time-delayed deletion; Crypto Ciphershield; Data security; Key caching.

1. Introduction

As a result of the current state of the digital environment, the utilization of external storage devices has grown widespread. USB sticks, SD cards, or flash memory devices have become vital instruments for the purpose of storing and transferring data across a variety of platforms[1]. It is important to note that the convenience of these devices is accompanied by the inherent risk of data disclosure in the event that they are lost or stolen. The existence of personally identifiable information on such external storage media raises a serious security problem, underscoring the urgent need for powerful encryption solutions to protect against data being accessed by unauthorized parties[2].

Despite the fact that encryption technologies provide an essential layer of safety for data that is saved on external devices, the usability of the encryption solutions that are currently available frequently leaves a lot to be desired[3]. Every time they access their encrypted data on a different device, users frequently face difficulties that are related with the laborious process of entering encryption keys or login credentials into the device. This problem with usability not only has the impact of compromising the user experience, but it also affects the efficiency of encryption in reducing the risks associated with security[4]. To make transparent encryption of external storage devices more secure and easier to use, this study presents a new approach.

This technique is proposed as a response to the issues that have been previously mentioned. In order to address the usability difficulties that are inherent in the encryption technologies that are currently in use, the suggested approach makes use of the MobileShielded Encryption Framework (MoSEF) and incorporates key caching with time-delayed deletion[5]. Through the use of this cutting-edge method, customers are able to take advantage of an encryption experience that is both seamless and intuitive, without the requirement of repeatedly entering their keys. The implementation of a two-pronged method is the central premise of the proposed solution, which aims to achieve a balance between usability and security[6]. In the first place, the solution provides a streamlined encryption procedure that involves minimal participation from the user.

This simplifies the user experience and reduces the risk of errors or security breaches occurring. The adoption of key caching technologies that securely store encryption keys on the external storage device enables users to access their encrypted data without the need for manual key entering[7]. This is accomplished by ensuring that the encryption keys are stored in a secure manner. Second, the method implements the idea of time-delayed deletion, which improves the safety of data by automatically erasing cached encryption keys after a set amount of time has passed during which the keys have not been used[8]. This preventative method of key management helps reduce the likelihood of unwanted access occurring in the event that the device is misplaced or stolen, which further strengthens the encryption solution's security posture[9].

In addition, the solution that has been provided provides flexibility and scalability by permitting many iterations that may be customized to meet the requirements of users with varying needs. Users that often transfer data between devices would benefit from a frictionless encryption experience provided by the first version of the solution, which focuses on short-term data transfers[10]. The second iteration, on the other hand, offers comprehensive key management capabilities within the file system. This makes it possible for users to easily handle numerous encryption keys for a variety of files[11]. Additionally, the proposed solution provides compatibility with mobile devices, which enables users to access their encrypted data in a secure manner while they are on the move.

The solution offers a comprehensive and user-friendly encryption solution that is designed to suit the increasing needs of modern users. It does this by integrating smoothly with the MoSEF architecture, which enables it to provide plaintext encryption for external storage media. In a nutshell, the solution that has been provided is a huge step forward in terms of transparent encryption for external storage devices[12]. It provides a one-of-a-kind mix of security, usability, and compatibility. The solution provides users with an easy and safe mechanism for managing sensitive data when they are on the move. This is accomplished by employing novel technologies like as key caching and time-delayed deletion Therefore, the solution lessens the likelihood of breaches of data and illegal access.The main objectives are:

ISSN: 3006-8894 https://doi.org/10.70023/piqm24302

- ✓ Create an open-source encryption method for portable storage devices to improve the safety and confidentiality of information stored on memory sticks, SD cards, and other similar devices.
- ✓ Address usability challenges associated with existing encryption solutions by implementing key caching with time-delayed deletion within the MobileShielded Encryption Framework (MoSEF), thereby reducing the burden of repeated key or password entry for users.
- ✓ Enhance both security and usability by eliminating the necessity to share keys or passwords, thereby improving the overall user experience while ensuring robust protection against unauthorized access to sensitive data stored on external storage media.

A summary of the research is provided below. In Section 2, the current literature and study techniques are thoroughly examined. The research strategy, methodology and processing procedures are detailed in Section 3. The results analysis is covered in Section 4. Part 5 explores the main conclusion and Future work.

2. Research Methodology

Yi et al. [13] presented the goal of developing a teacher workload management system that is based on jQuery Mobile was to increase the standardization and efficiency of the process of managing workloads for college classroom instructors. In addition to being able to manage ordinary instruction, projects, data filling, summary, statistics, and inquiries, the system is also capable of running on mobile devices and personal computers. The system's ability to work with departments, training centers, instructors, and teaching affairs offices means that users have 24/7 access. There is less labor and less time spent processing transactions because users can access the system whenever they like. As an added bonus, this system makes things easier for the user.

Luceri et al. [14] showed that the use of mobile devices has had a substantial impact on the shopping experiences of customers. According to the findings of a meta-analysis that included 207 publications and 228 research, technological advancements and the functionality of devices had an impact on mobile shopping. According to the results of the study, it is recommended that businesses concentrate their efforts on utilitarian and hedonistic factors in order to maximize the intention to continue using mobile devices. Enhancing the mobile shopping experience can also be accomplished by increasing client satisfaction through the enhancement of the mobile channel's quality and delight.

Curum et al. [15] displayed the Mobile-assisted learning is gaining popularity as a result of its capacity to facilitate education in a variety of settings and to establish a realistic learning environment. Previous study, on the other hand, has demonstrated that poor dynamic content adaptation is the result of bad design of learning aspects in mobile systems. In this study, a model chart of learning efficiency is proposed with the intention of enhancing mobile learning experiences. This is accomplished by examining many common learning theories and comparing them with the proposed chart. The goal of this inquiry is to increase the quality of the learning experience while simultaneously reducing the complexity of mobile learning systems.

Iwaya et al. [16] presented the Data security and privacy must be improved as mHealth and uHealth technologies become more popular. A systematic mapping study (SMS) on 365 qualitatively selected research identified, classified, compared, and evaluated m/uHealth system security and privacy. Results showed that research focuses on control families, system and information protection, access control, authentication, person participation, and privacy

authorization. Data governance, security and privacy rules, and program management are underrepresented despite their importance. Lack of real-world evaluation drives most research to propose novel ideas with inadequate validation. This SMS helps researchers and practitioners develop privacy and security for next-generation m/uHealth systems.

Nadeem et al. [17] provided the healthcare business prioritizes patient data security and privacy. This study proposes a system that secures and privacys patients' medical records and allows healthcare professionals to share data. The study suggests switching from healthcare industry-managed electronic health records to patient-centric apps that put people in charge. This research aims to construct an Ethereum-based EHR system and smart contract to eliminate third-party systems. Using this technology, medical professionals can access patient records and ask for permission to use them. As a result of patient empowerment, HER system data sharing is expedited. Patient information is stored in the decentralized ledger system that communicates amongst peers. By centralizing the collection and storage of patient medical records, the proposed patient-centric HER platform allows for interoperability across healthcare providers and is accessible via personal computers and mobile devices.

Hati et al. [18] presented the Cryptography has created encryption systems with varied performance and security benefits. Cryptography is really important in revealing the benefits of encryption techniques by utilizing them appropriately. Time of use and administration are equally crucial as encryption method strength. This section covers hybrid encryption. Hybrid encryption(HE) techniques enhance data transmission safety and speed by combining several encryption methods at the right time. This article examines the causes behind hybrid encryption methods and introduces a new one. The novel hybrid architecture combines the new encryption system with RSA, an asymmetric encryption scheme.

As a reliable certification body, Pandey et al. [19] proposed digital certificate issuance. The four PAIN characteristics—privacy, authenticity, integration, and non-repudiation—are given via certificates. Unfortunately, CAs can be dishonest, negligent, or both, which can lead to the issuance of fraudulent certificates. This can jeopardize the PKI environment and cause millions of people to fall victim to bogus websites and forged signatures, among other things. Transparency in certificate issuance, annulment, and audits is provided by measures such as Certificate Transparency (CT). But there are still limitations. In order to create a full-fledged PKI ecosystem for certificates, this paper suggests PRAMANIK, an online system for pre-issuing certificate validation, a storage for certificates on the blockchain, a validator, and a querier (with the addition of database operation and API support).

Secure encryption methods for IoT devices, with a focus on the white-box idea, were provided by Asanuma et al. [20] under the acronym WBE-IoT. The initial plan uses a combination of a space-hard cypher and a physiologically unclonable function (PUF) to ward off attacks that attempt to extract keys or lift codes. This technique keeps the security level unchanged from the Even-Mansour cipher by routinely altering the whitening key without updating the encryption key to the space-hard cipher. Theft of the key does not compromise this security. Substantial protection against secret extraction but code lifting may be achieved with this strategy, and the low cost of altering only the whitening keys is a major plus.

3. Proposed Methodology

The proposed work aims to create a secure external storage solution that combines robust security measures with user-friendly functionality. It focuses on data confidentiality, key management, and transparency for end users. The framework uses advanced key management, encryption, and decryption processes, enhancing data security. The study also offers multiple

ISSN: 3006-8894 https://doi.org/10.70023/piqm24302

key options and time-delayed key expiration to protect stored data from potential breaches. The system is user-friendly, integrating seamlessly with mainstream operating systems and ensuring secure data access with minimal user interaction. Its authentication mechanism ensures secure data access with minimal user interaction. The system also offers scalability, enabling efficient handling of diverse data sizes and storage demands. The framework optimizes energy usage for sustainable operation.

Crucial elements of the suggested approach to mobile-compatible key management by Crypto Ciphershield for transparent encryption of external storage media with time-delayed deletion include two separate iterations, the elimination of key sharing, compatibility with mobile devices, and plaintext encryption of external storage media. Secure storage of encryption keys is assured by key caching within the MobileShielded Encryption Framework (MoSEF), and they are automatically deleted after a set length of time thanks to time-delayed deletion. In its second iteration, the approach can manage several encryption keys for distinct files within the file system, and in its first, it works flawlessly for fast data transfers without user input. Users who need safe data management on the go will appreciate the method's interoperability with mobile devices and the timed key caching that improves security and usability. An all-inclusive method of safe and easy-to-use encryption of external storage devices, especially in mobile settings, is the goal of the research.

a. Integration of Key Caching with Time-Delayed Deletion

The method integrates key caching with time-delayed deletion within the MobileShielded Encryption Framework (MoSEF). This approach allows for the secure storage of encryption keys while ensuring that they are automatically deleted after a certain period, enhancing security.

The time-delayed deletion process is mathematically represented using an exponential decay function, as expressed by the equation:

$$K(t) = K_0 \times e^{-\lambda t} \quad (1)$$

In equation 1, The key strength that is still available at time t is denoted by the symbol $K(t)$.
Initially, the key strength is denoted by the value K_0. λ represents the decay constant.
t represents the amount of time that has passed since the production of the key.

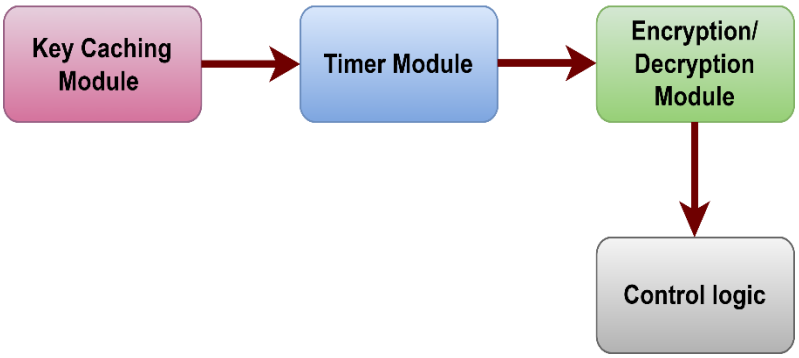

Figure 1. Integration of Key Caching with Time Delayed Deletion

In figure 1, Key caching and time-delayed deletion are both components that are used into the MobileShielded Encryption Framework (MoSEF) in order to improve security. The block diagram is made up of three different components: The Key Caching Module, which is responsible for the secure storage of encryption keys; the Timer Module, which is in charge of the time-delayed deletion process; the Encryption/Decryption Module, as it is responsible for encrypting and decrypting data using the stored keys;and the Control Logic, which is in charge

of managing the key caching, deletion process, and encryption/decryption operations. In order to formally express the time-delayed deletion process, an exponential decay function can be utilized.

An integral part of the integration process is key caching, which involves the secure storage of encryption keys within the Key Caching Module. Other components of the process include time-delayed deletion, secure deletion, and re-encryption and decryption. When the timer reaches its expiration point, the Control Logic will initiate the deletion procedure, which will ensure that the encryption key is removed from the Key Caching Module in a safe and secure manner. In the final analysis, the MobileShielded Encryption Framework (MoSEF) improves security by automatically discarding encryption keys after a predetermined amount of time. This reduces the likelihood that unauthorized individuals would gain access to critical data.

b. Two Distinct Iteration

The first version restricts the use of external storage to data transfers that are only conducted for a brief period of time and does not require any intentional input from the user. When it comes to transferring data quickly, it functions in a fluid and effective manner. The second version is a variant that allows for the management of several encryption keys for various files that are contained within the file system. When it comes to managing encrypted data, having a trusted host on the file system makes this functionality easier to implement, which in turn allows for increased security and flexibility.

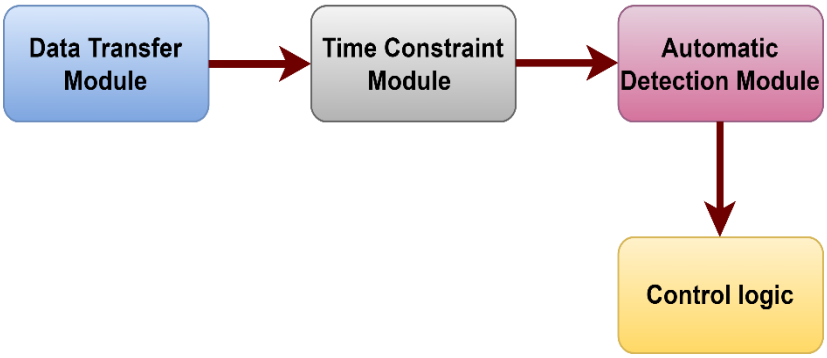

Figure 2. Efficient Short-Term Data Transfer

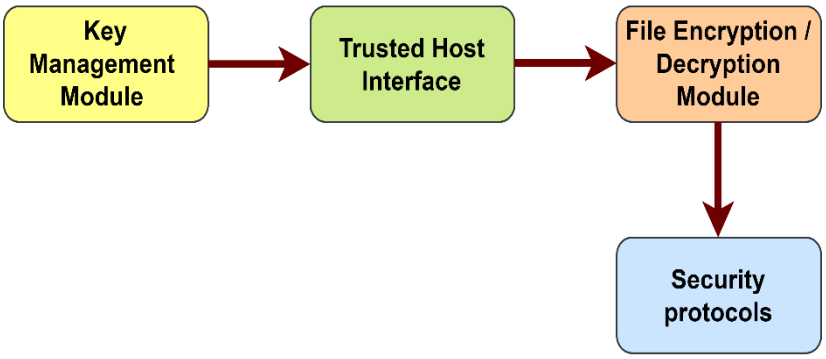

Figure 3. Secure Handling of Multiple Encryption Keys

The solution that has been developed focuses on two main iterations: the efficient transfer of data over short periods of time and the secure management of multiple encryption keys. In figure 2, The first version limits the use of external storage to data transfers that are only temporary and do not require input from the user. This ensures that the operations are both

ISSN: 3006-8894 https://doi.org/10.70023/piqm24302

frictionless and efficient. Having a time-based constraint on the length of storage guarantees that data will be deleted automatically after a predetermined amount of time has passed. A Data Transfer Module, a Time Constraint Module, an Automatic Deletion Module, Control Logic, and Flow are the components that make up the block diagram. In figure 3, One of the goals of the second edition is to make it possible to manage several encryption keys for various files that are stored within the file system.

Enhanced security and flexibility are achieved through the utilization of a trusted host on the file system for the management of encryption keys. For the purpose of producing and safely storing numerous encryption keys, the equation might contain key management methods or cryptographic functions. Key Management Module is responsible for the generation and storage of multiple encryption keys for various files. Meanwhile, Trusted Host Interface is responsible for interacting with the file system in order to handle these keys in a secure manner. For the purpose of encrypting and decrypting files, the File Encryption/Decryption Module makes use of the proper key, thereby ensuring the safety of their contents. Through these elaborations, a visual and conceptual comprehension of the two unique iterations of the proposed technique is provided. The emphasis is placed on the functionality and execution of these iterations for the purpose of ensuring secure data management on external storage media.

c. Elimination of Key Sharing

Through the elimination of the requirement to disclose keys or passwords, the strategy improves both the security and the usability of the system. The effectiveness of the system in terms of both its usability and its overall security is significantly enhanced by the utilization of time-based key caching.The equation may involve key generation algorithms or cryptographic functions to create individual keys for each user or entity.

$$K_{user1} = f(user1_{credentials}) \quad (2)$$

In equation 2, K_{user1} is the unique key for $User\ 1$ derived from their credentials using a cryptographic function f.

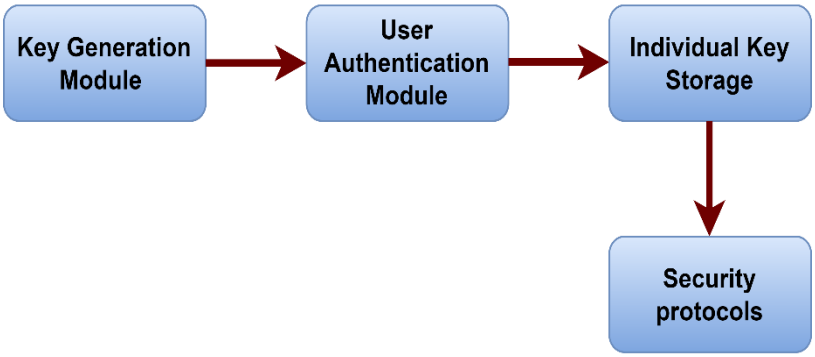

Figure 4. Elimination of Key Sharing

In figure 4, The elimination of the requirement to share keys or passwords is the primary objective of this system, which aims to improve both security and usability. One way to accomplish this is by putting in place a system that allows each person or entity to have their own unique key without allowing them to share important information. It is possible that the system will make use of cryptographic functions or algorithms for key generation in order to generate unique keys for each user or entity. The Key Generation Module, the User Authentication Module, the Individual Key Storage, and the Security Protocols are some of the

components that make up the block diagram. The User Authentication Modules are responsible for verifying the identity of the user and granting access to the individual key.

The Key Generation Module is responsible for the creation of one-of-a-kind keys based on the user's credentials. Because each user has their own key that is stored in a secure location, there is no need to facilitate key exchange. By utilizing timed key caching, the system also intends to enhance both the overall security and the usability of the system. This entails caching keys for a predetermined amount of time in order to strike a balance between usability and security. This ensures that keys are accessible whenever they are required and are automatically removed after a predetermined amount of time has passed. The Key Cache Module is responsible for storing keys for a predetermined amount of time, which is determined by the Timer Module. When the caching duration is over, the Key Deletion Module deletes the cached key in a safe and secure manner. Incorporating these equations and block diagrams into the system allows for the impact of removing key sharing and adopting timed key caching to be viewed and comprehended within the system.

d. *Interoperability with Mobile Devices*

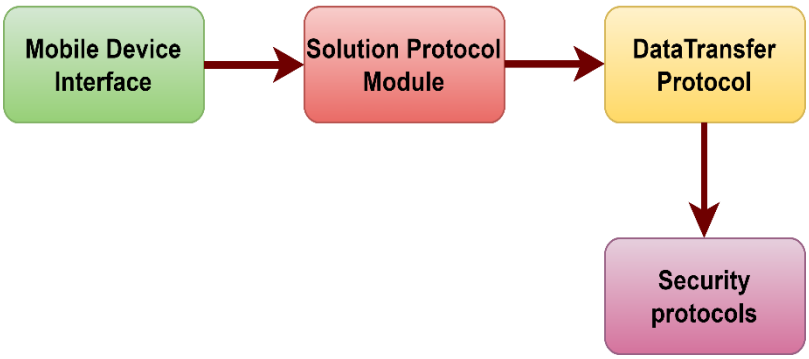

Figure 5. Interoperability with Mobile Devices

In figure 5,As a way to ensure secure data management when on the move, this study provides an overview of the goals and procedures involved in guaranteeing interoperability with mobile devices. The solution and mobile devices are able to establish communication and data transmission thanks to the equations and block diagrams that are provided in this document. Additionally, the Solution Protocol Module assures interoperability and data transfer protocols between the mobile platform and the solution. The Mobile Device Interface is responsible for establishing connection with the mobile device. By including the essential security protocols, the Data Transfer Module guarantees the safety of the data transfer that takes place between the solution and the mobile device. For the purpose of assuring compatibility with mobile devices, the text tries to visualize and comprehend the way that allows for smooth integration and functionality within the system, which enables secure data management when on the move.

e. *Plaintext Encryption for Extend Storage Media*

Through the use of this technology, plaintext encryption is provided for external storage media. This ensures that sensitive data is safeguarded from unwanted access as well as data breaches.

ISSN: 3006-8894 https://doi.org/10.70023/piqm24302

The solution ensures that it is compatible with mobile devices, which satisfies the requirements of customers who need to handle their data in a secure manner when they are on the move.

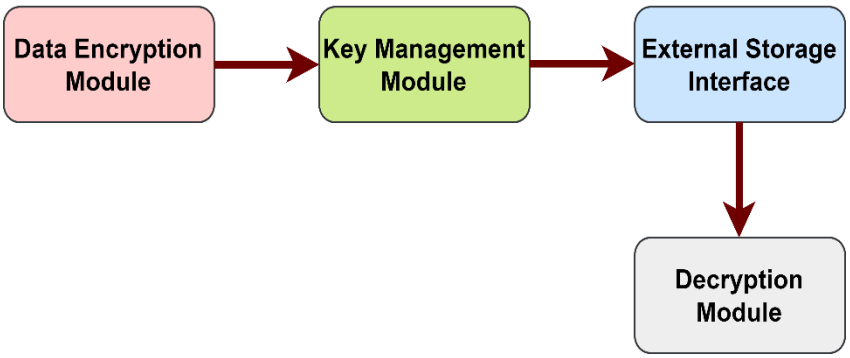

Figure 6. Plaintext Encryption for External Storage Media

In figure 6, The purpose of providing plaintext encryption for external storage devices is to safeguard sensitive data from being accessed by unauthorized parties and from potential breaches. This document provides an overview of the motivation for this practice. In order to ensure that only authorized users are able to access and decode the information, the method comprises encrypting the data in plaintext format before making it available for storage on an external device. In order to facilitate the transformation of plaintext data into ciphertext for the purpose of secure storage, the equation might incorporate encryption methods or cryptographic functions.

The Module of Data Encryption, the Key Management Module, the External Storage Interface, and the Decryption Module are the three components that make up the block diagram. When permitted, the Data Encryption Module will encrypt plaintext data by utilizing a key that has been specified by the Key Management Module. On the other hand, the Decryption Module will decode the data by utilizing the key that corresponds to the encrypted data. This approach makes it possible to obtain a deeper comprehension of the encryption procedure as well as the data protection procedures that have been put in place to prevent unwanted access and data breaches.

4. Results and Discussion

a. Scalability

Scalability is a statistic that measures the way a system can handle increased workloads or traffic volume while still retaining its performance and efficiency. When it comes to systems that have shifting demand, it is essential. Among the most used scalability metrics is speedup, which measures the degree to which the performance of the system improves in proportion to the growth in the number of processing elements or the workload. The metric is used to determine how well a system can adjust to shifting requirements without sacrificing its performance or efficiency.

In figure 7, A comparative examination of four different encryption frameworks is presented in the picture. These frameworks are "HE," "CT," "WBE-IoT," and "MoSEF." The security performance of these frameworks is evaluated on three separate levels such as low level, medium level, and high. The rankings are based on the rankings. This section provides the numerical scores for each framework, which indicate the strengths and shortcomings of each framework in terms of managing security problems. Comparatively, it has been discovered that the "MoSEF" framework is more effective than the other frameworks on a number of different levels. The range of possible scores is from 80 to 90, with "HE" receiving a score of

80 at the low level, "CT" receiving a score of 83 at the medium level, and "WBE-IoT" receiving a score of 86 at the superior level. The figure emphasizes the significance of having a comprehensive grasp of the disparities in security performance that exist between various levels of encryption and decryption overhead.

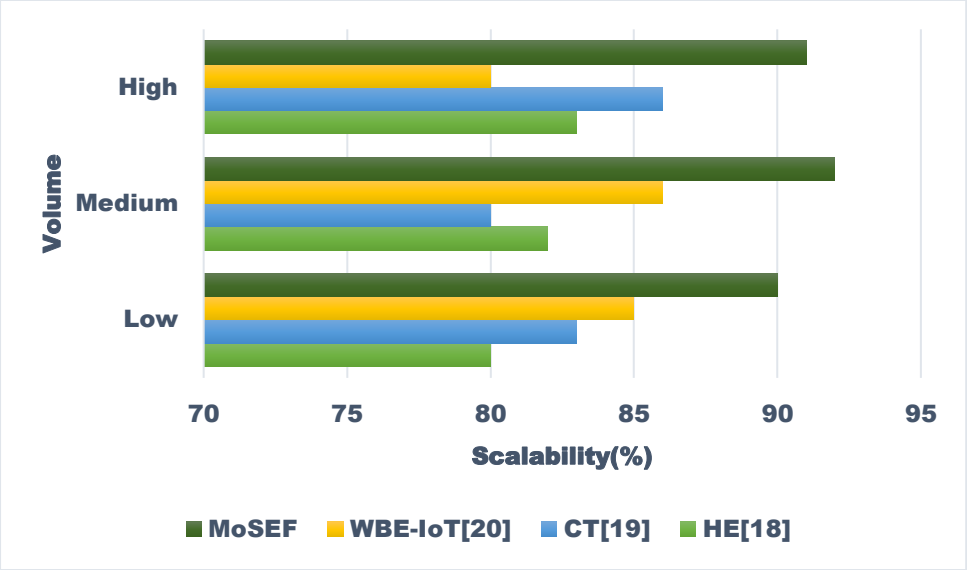

Figure 7. Scalability Improvement

In figure 7, A comparative examination of four different encryption frameworks is presented in the picture. These frameworks are "HE," "CT," "WBE-IoT," and "MoSEF." The security performance of these frameworks is evaluated on three separate levels such as low level, medium level, and high. The rankings are based on the rankings. This section provides the numerical scores for each framework, which indicate the strengths and shortcomings of each framework in terms of managing security problems. Comparatively, it has been discovered that the "MoSEF" framework is more effective than the other frameworks on a number of different levels. The range of possible scores is from 80 to 90, with "HE" receiving a score of 80 at the low level, "CT" receiving a score of 83 at the medium level, and "WBE-IoT" receiving a score of 86 at the superior level. The figure emphasizes the significance of having a comprehensive grasp of the disparities in security performance that exist between various levels of encryption and decryption overhead.

b. Encryption/Decryption Overhead

In the context of encrypting and decrypting data with the help of an encryption framework or algorithm, the term encryption/decryption overhead refers to the additional computational resources and time demands that are required. It has an immediate and direct effect on the performance of system, efficiency, and usage of resources. The performance of the system may be negatively impacted by an increase in the overhead, particularly in situations involving real-time data processing or low-latency processes. In order to evaluate this cost, it is necessary to evaluate the influence that encryption methods, key lengths, cryptographic primitives, and computational complexity have on the performance of the system. In order to achieve optimal security and operational efficiency, it is essential to take into consideration factors such as the efficiency of algorithms, the processor power, the memory utilization, and the overall performance of the system.

ISSN: 3006-8894 https://doi.org/10.70023/piqm24302

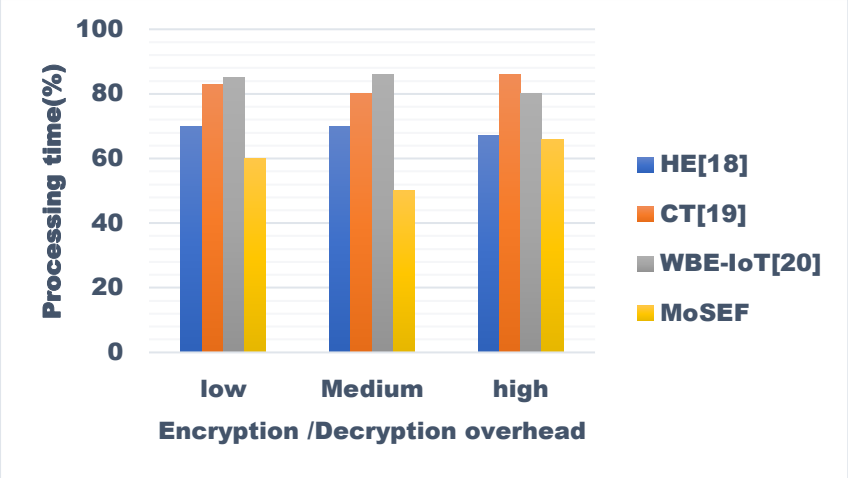

Figure 8. Encryption/Decryption Overhead

Figure 8 presents a comparison of four different encryption frameworks: HE, CT, WBE-IoT, and MoSEF. These frameworks are compared across three different levels of encryption/decryption overhead: low, medium, and severe. The numerical scores that are assigned to each framework are determined by the framework's performance at each level. The lower the scores, the better the performance, which probably indicates that there are fewer computational resources. When selecting the encryption framework that is most suited to meet particular security requirements, it is important to take into account a number of aspects, including interoperability, scalability, and security robustness, in addition to concerns regarding overhead.

c. Resource Utilization

Resource utilization is the efficient allocation of computational, memory, storage, and network resources within a computing system or application. It ensures optimal performance, responsiveness, and scalability while minimizing waste. In encryption frameworks, it involves CPU processing power, memory usage, disk I/O operations, and network bandwidth consumption. Efficient resource utilization is crucial for maintaining system performance, responsiveness to user requests, and accommodating increasing workloads without performance degradation or resource exhaustion. Monitoring and optimizing resource utilization are critical aspects of system administration and software development.

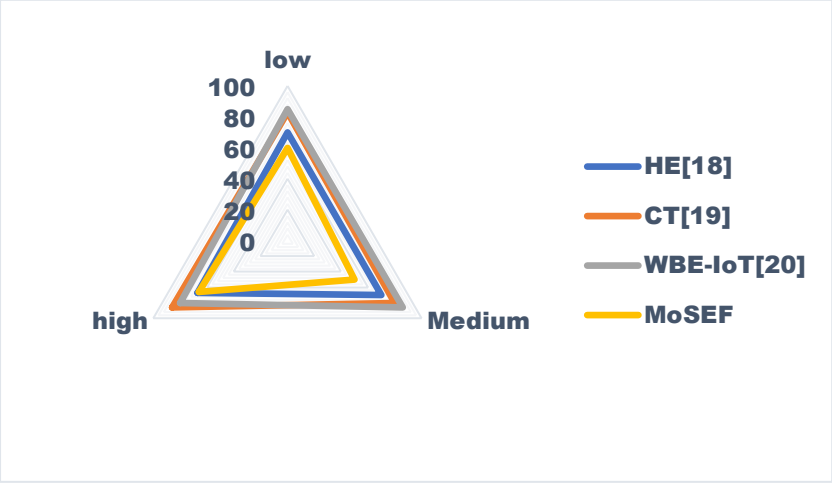

Figure 9. Resource Utilization

In figure 9, Resource usage is an encryption framework's computational efficiency for encryption and decryption. Four encryption frameworks—HE, CT, WBE-IoT, and MoSEF—are tested at Low, Medium, and High resource use. Each framework receives numerical scores for its level performance. Higher scores imply greater resource utilization, reducing encryption and decryption computational resources. In addition to resource utilization, security performance, scalability, and compatibility should be addressed while choosing an encryption framework for specific security and operational demands. Encryption/Decryption Overhead" is the computational resources and time required to encrypt and decrypt data using an encryption framework or technique. It affects system efficiency, performance, and resource use. In real-time data processing or low-latency processes, more overhead might reduce system performance. This overhead is determined by how encryption methods, key lengths, cryptographic primitives, and computational complexity affect system performance. Optimal security and operational efficiency depend on algorithm efficiency, processing power, memory utilization, and system performance.

5. Conclusion and Future Study

The Crypto Ciphershield architecture has introduced mobile-compatible key management and transparent encryption for external media for storage to remedy the security issues connected with lost or misplaced devices housing important data. The MobileShielded Encryption Framework (MoSEF) provides a viable option for safeguarding data kept on external storage devices by incorporating key caching and time-delayed erasure. We no longer need to share keys or credentials because our system enables the maintenance of various encryption keys as well as the brief transfer of data. In addition to supporting mobile device interoperability, it provides plaintext encryption with external storage media, allowing users to easily handle vital data while on the go. Future work will include investigating more sophisticated key management methods, improving encryption and decryption, testing for usability, integrating along with cloud services, making sure it can scale to different devices and OSes, and maintaining regular security audits to find and fix vulnerabilities. The development of a transparent encryption system for external storage media compatible with mobile key management might proceed by focusing on these factors. People who handle sensitive information on mobile devices will benefit from this in terms of enhanced security, functionality, and usefulness.

REFERENCES

[1]. Alazab, Ammar, Ansam Khraisat, and Sarabjot Singh. "A Review on the Internet of Things (IoT) Forensics: Challenges, Techniques, and Evaluation of Digital Forensic Tools." (2023).

[2]. George, A. Shaji, AS Hovan George, and T. Baskar. "Digitally immune systems: building robust defences in the age of cyber threats." *Partners Universal International Innovation Journal* 1.4 (2023): 155-172.

[3]. Mousavi, Seyyed Keyvan, et al. "Security of internet of things based on cryptographic algorithms: a survey." *Wireless Networks* 27.2 (2021): 1515-1555.

[4]. RS, Anthony Raj Fathima Khanum Dixitha, and Kousalya N. Sarala Chaithra. "CLOUDMOAP: Multilayer Security by Online Encryption and Auditing Process in Cloud." (2023).

[5]. Tramèr, Florian, Dan Boneh, and Kenny Paterson. "Remote {Side-Channel} attacks on anonymous transactions." *29th USENIX security symposium (USENIX security 20)*. 2020.

[6]. Ning, Li, et al. "A hybrid MCDM approach of selecting lightweight cryptographic cipher based on ISO and NIST lightweight cryptography security requirements for internet of health things." *IEEE Access* 8 (2020): 220165-220187.

[7]. He, Ying, et al. "An efficient ciphertext-policy attribute-based encryption scheme supporting collaborative decryption with blockchain." *IEEE Internet of Things Journal* 9.4 (2021): 2722-2733.

ISSN: 3006-8894 https://doi.org/10.70023/piqm24302

[8]. Qowi, Z., and N. Hudallah. "Combining caesar cipher and hill cipher in the generating encryption key on the vigenere cipher algorithm." *Journal of Physics: Conference Series*. Vol. 1918. No. 4. IOP Publishing, 2021.

[9]. Wulandari, Septi Yana. "Cryptography: A Combination of Caesar and Affine Cipher to Conceal the Message." *Proceeding International Conference on Science and Engineering*. Vol. 3. 2020.

[10]. Dwivedi, Ashutosh Dhar. "Brisk: dynamic encryption based cipher for long term security." *Sensors* 21.17 (2021): 5744.

[11]. Subaselvi, S., et al. "VLSI Implementation of Triple-DES Block Cipher." 2023 7th International Conference on Computing Methodologies and Communication (ICCMC). IEEE, 2023.

[12]. Al-Moselly, Muntaser, and Ali Al-Haj. "High-Performance Hardware Implementation of the KATAN Lightweight Cryptographic Cipher." Journal of Circuits, Systems and Computers 32.01 (2023): 2350017.

[13]. Yi, Yangyang, Yiding Liu, and Ying Wang. "Design and Implementation of Teacher Workload Management System Based on JQuery Mobile." International Conference on Artificial Intelligence and Security. Cham: Springer International Publishing, 2022.

[14]. Luceri, Beatrice, et al. "What drives consumers to shop on mobile devices? Insights from a Meta-Analysis." Journal of Retailing 98.1 (2022): 178-196.

[15]. Curum, Brita, and Kavi Kumar Khedo. "Cognitive load management in mobile learning systems: principles and theories." Journal of Computers in Education 8.1 (2021): 109-136.

[16]. Iwaya, Leonardo Horn, Aakash Ahmad, and M. Ali Babar. "Security and privacy for mHealth and uHealth systems: a systematic mapping study." IEEE Access 8 (2020): 150081-150112.

[17]. Nadeem, Mohd, and Deven Shah. "Pre settlement in Insurance for Hospital Treatment Using Cryptocurrency and Blockchain Technology." International Journal of Research in Engineering, Science and Management 4.10 (2021): 136-140.

[18]. Hati, Nargiz Khankishiyeva. "HYBRID APPROACHES IN CRYPTOGRAPHY." Journal of Modern Technology & Engineering (2023).

[19]. Pandey, Anoop Kumar, et al. "Pramanik-Cloud Based Certificate Repository." 2022 IEEE 4th International Conference on Cybernetics, Cognition and Machine Learning Applications (ICCCMLA). IEEE, 2022.

[20]. Asanuma, Takaki, and Takanori Isobe. "Even-Mansour Space-hard Cipher: White-box Cryptography Cipher Meets Physically Unclonable Function." Journal of Information Processing 31 (2023): 88-96.