A Key Rotation Management System: Design and Implementation for Improved Data Security

Malibongwe Ntalali[1], Noluntu Mpekoa[1] and Sheethal Tom[2]
[1]University of Johannesburg, South Africa
[2]The Independent Institute of Education, Varsity College, Cape Town, South Africa

223081688@student.uj.ac.za
noluntum@uj.ac.za
stom@varsitycollege.co.za

Abstract: Organizations in today's rapidly changing digital world use various channels, such as secure APIs and encrypted communications, to enhance collaborations and information sharing. While these systems assist in enhancing productivity, innovation and collaboration, they can also introduce major security risks for protecting sensitive information. The core of data protection depends on cryptographic key management, with key rotation playing a vital role, yet often neglected. Key rotation, a well-established cryptographic practice, is essential for minimizing data exposure, mitigating key compromise risks and ensuring regulatory compliance. However, inconsistent implementation of key rotation policies across organizations often results in varied security practices. The proposed Key Rotation Management System functions as a solution that automates and standardizes all stages of the key lifecycle process. The system implements a three-tier Model-View-Controller framework which combines different functional components that include user authentication together with role-based control, automated key generation, secure storage and distribution, periodic rotation, dashboard visualization and proactive alert systems. The proposed system resolves fundamental problems which arise from human involvement in administration and irregular scheduling and insufficient user understanding and insecure distribution methods. The solution provides real-time key ageing visibility through its dashboard interface, while its scheduling and reminder features assist in automated rotation interval execution. It includes a centralized service request-response module together with automated email notification to help organizations maintain effective communication and monitor compliance standards. The evaluation results show that the proposed system enhances user responsiveness and decreases expired key occurrences while decreasing manual work and assisting organizations in meeting industry international standards. The system design with role-based access controls provides both security measures and system accountability features. The proposed Key Rotation Management System combines automation, visualization and security to offer a scalable solution that can strengthen the cryptographic strength of digital infrastructures.

Keywords: Key rotation, Cybersecurity automation, Cryptographic key management, Secure data sharing, Key lifecycle management

1. Introduction

Organizations use technologies such as emails, video conferencing and intranet for collaboration (Hurst, 2020). With the increase in collaboration and information sharing, the security of sensitive information has become paramount. Organizations have taken additional precautions to store sensitive information such as passwords, tokens, and private keys in centralized key storage locations, procuring them only during runtime (Ahanger et al, 2024). With the storage and usage of this sensitive information, one critical aspect of maintaining this security is the effective management of cryptographic keys (Ahmad, Mehfuz and Beg, 2023). Key management, structured as a lifecycle includes the secure generation, storage, distribution, use, and destruction of keys (Faisal et al, 2020). The Open Web Application Security Project (OWASP) recommends best practices such as key generation within cryptographic modules, distributing them securely and never storing them in plain text (Sakthivel et al, 2020).

Key rotation, the process of periodically changing encryption keys, is essential to mitigate the risks associated with key compromise and ensure continued data protection. Key rotation is mandated by regulation in some contexts, such as the Payment Card Industry Data Security Standard (PCI DSS) that dictates how credit card data must be secured [PCI16] (Seaman, 2020). Despite its importance, many organizations face challenges with implementing efficient key rotation strategies due to the complexity and resource requirements involved. The most challenging step in the key management life cycle is key rotation, as it requires change management, stakeholder adoption and management, policy enforcement and coordination across different platforms. The challenges faced by the cloud users with key management to securely interact with a range of services and store data created or processed by those services are due to differences in ownership and control of the underlying infrastructures on which the protected resources and the key management system (KMS) are located (Ahmad et al, 2025).

This study aims to address these challenges by proposing a comprehensive Key Rotation Management System (KRMS) that automates and streamlines the key rotation process, thereby enhancing security and reducing operational overhead. The system will serve as a centralized hub for control and communication, facilitating the scheduling of notifications and reminder alerts. Additionally, it will enable the monitoring of cryptographic key aging and provide a comprehensive dashboard for oversight and management.

The remainder of the paper reviews related work (Section 2), introduces the KRMS model and architecture (Section 3), details its implementation (Section 4), presents results and discussion (Section 5) and concludes with key contributions (Section 6).

2. Literature Review

This section focuses on Key Rotation Management System (KRMS) relevant studies, starting with broader aspects of information security before moving to more specific aspects that are related to the proposed system.

2.1 Background on Information Security

Information security protects the confidentiality, integrity, and availability (CIA) of information assets (Mpekoa, 2024). Confidentiality is achieved through encryption, which prevents unauthorized access (Zhang et al, 2018). The protection of data integrity involves protecting it against unauthorized damage, disruption or alteration to maintain its accurate and trusted state. Availability ensures that authorized users, including people and systems maintain access to information during their required times while receiving it in a usable form (Yee and Zolkipli, 2021). The three fundamental principles work together to create a resilient and robust network security framework.

Human beings are the most vulnerable point of information security, as many breaches stem from human errors or carelessness (Hughes-Lartey et al,2021). Hence, Security Education, Training, and Awareness (SETA) is imperative when implementing a security system in any organization. SETA raises the awareness of protecting organizational information, provides employees with the skills and knowledge to work securely and creates a shared organizational culture of security (Bethel, 2020).

2.2 Cryptography and Key Management

Cryptography ensures confidentiality, integrity, authentication, and availability through the use of encryption algorithms. Encryption is a mathematical process used to transform text into a coded format, making it unreadable without the corresponding decryption method. Only those who possess knowledge of the algorithm and all the necessary key material utilized in the process can decode the encrypted text. Encryption algorithms can be categorized as either symmetric or asymmetric algorithms. In a symmetric algorithm, the sender and receiver use a shared secret key for encryption and decryption (Yassein et al, 2017). While this method ensures message privacy, it poses security risks during key distribution. Thus, it is crucial to find a secure way to distribute and protect the key. Asymmetric algorithms use a pair of mathematically related keys instead of a single key. The sender encrypts the message with a public key, while the receiver decrypts it using a private key, known only to them. Public keys can be shared widely, while private keys remain confidential (Hinkes, 2018).

The effectiveness of cryptography relies on strong key management because quantum-resistant primitives are currently emerging. Strong algorithms become weak when poorly managed keys are handled by poor practices (Barker and Barker, 2018). A key management system handles all aspects of cryptographic key operations from secure generation to storage, distribution, usage and eventually key retirement. According to NIST's key management guidelines (SP 800-57 revised 2024), organizations must establish cryptoperiods and perform regular key rotation to reduce the exposure time in case of key compromise (Vlajic and Cianfarani, 2023). Studies (Torres, 2022; Ahmad, Mehfuz and Beg, 2023) emphasize the necessity of automated key lifecycle management in hybrid and cloud-based systems.

2.3 Challenges in Key Rotation

The requirement for key rotation in PCI DSS compliance standards creates major obstacles during implementation. Manual processes in key rotation are error-prone and tend to result in non-compliance and missed rotation deadlines (Kuzminykh, Yevdokymenko and Ageyev, 2020). Research indicates that cloud and hybrid infrastructure key rotation face challenges because of ownership fragmentation and interoperability issues. The "key sprawl" phenomenon becomes more prevalent according to research because it rates difficulties for service orchestration while increasing service disruption risks (Nakamura et al, 2025). The

cryptoperiods should be adaptive, but the practicality of this remains in question due to organizational resource constraints and integration difficulties (Kim et al, 2016).

2.4 Need for Key Rotation Management System

Key management encompasses a comprehensive set of techniques aimed at the initialization, registration, updating, and recovery of cryptographic keys. These techniques are crucial for maintaining the confidentiality, integrity, and authentication of communications among authorized entities (Gautam and Kumar, 2021). However, current commercial solutions and academic literature often overlook the critical aspect of key rotation within the broader framework of key management. The implementation of key rotation policies tends to vary significantly across different organizations, leading to inconsistent security practices that can create vulnerabilities. This disparity highlights a pressing need for standardized approaches to key rotation that can enhance overall security (Joshi, Crowther and Robinson, 2024). Additionally, many existing solutions lack straightforward automation features, particularly those that provide proactive notifications when keys are nearing expiration or require rotation. Furthermore, the literature indicates a significant shortage of expertise and robust systems capable of facilitating an effective lifecycle of key management and compliance monitoring. This gap not only hampers organizations' ability to manage keys efficiently but also increases the risk of security breaches due to outdated or improperly managed cryptographic keys (Radanliev, 2023). Addressing these challenges is essential for organizations seeking to strengthen their security posture in an increasingly complex threat landscape.

3. System Architecture and Model Implementation

The proposed Key Rotation Management System (KRMS) is a comprehensive framework designed to efficiently handle the lifecycle of cryptographic keys. The system encompasses several critical modules, each of which plays a vital role in ensuring robust key management practices. The proposed KRMS implements a Model-View-Controller (MVC) architecture to ensure separation of concerns, maintainability, modularity and scalability. It makes a codebase's concern separation easier by breaking it up into three separate layers:

- the *Model*, which takes care of data management and business logic;
- the *View*, which handles user input and output presentation; and
- the *Controller*, which acts as a mediator between the Model and the View, managing interactions and preserving the application’s state.

This structure makes code easier to organize and more maintainable (Nahhas, 2021). The three-layer architecture is illustrated in Figure 1 below.

The Presentation Layer provides a user interface through a Flask-based web application. The responsibilities of this layer include rendering web pages based on user requests, collecting and validating user inputs before passing them to the business logic layer, and displaying results or errors based on responses from the business logic layer. The Business Logic Layer contains the fundamental KRMS operations, which generate keys automatically and schedule rotations and enforce security protocols. The system uses this layer to link user requests with system operations and maintain compliance with cryptographic standards. The data access layer enables safe database interactions with the KRMS system. The data access layer communicates with the business layer to obtain database instructions, which it then uses to communicate with a MySQL database via a Flask-based Python application using a MySQL connector.

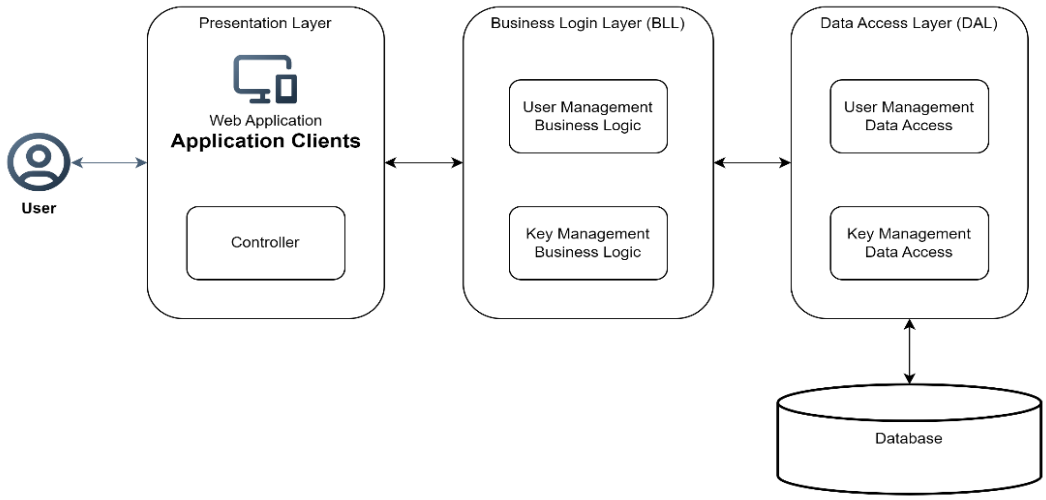

Figure 1: Key Rotation Management System Architecture Diagram

The layered design of KRMS allows organisations to monitor cryptographic keys from a central point while providing features like automation and proactive notification capabilities.

4. Proposed Solution - Key Components

The KRMS provides automation alongside visualization tools and notification features through an adaptable centralised platform. The KRMS implements practical and scalable lifecycle management through user-friendly interfaces which connect security features to usability. The key modules within the KRMS are outlined and described in detail as follows:

4.1 User Management Module

The User Management Module is a critical component of the KRMS that plays a central role in overseeing the creation, management, and authentication of user accounts within the system. Its primary purpose is to ensure that only authorized individuals can access sensitive operations related to key management. This module handles authentication (verifying user identity), authorization (enforcing role-based access controls) and user management (creation, updating, and deletion of user accounts). Key features of the module include user registration with secure handling of passwords (hashing and salting), login-based user authentication and Role-Based Access Control (RBAC) with defined user roles (e.g., Admin, User). User profile management is another crucial feature, enabling users to view and update their profiles, including personal information and security settings, and providing functionalities for password changes and account recovery. Finally, the module supports account deactivation and deletion, ensuring that user accounts can be deactivated or removed when no longer needed, in compliance with data retention policies.

4.2 Key Management Module

The Key Management Module ensures key generation, distribution, storage, rotation, and retirement. Key generation is handled by utilizing the Python secrets library, which is designed to generate cryptographically secure pseudo-random numbers. This library ensures that the keys used for managing secrets, such as authentication tokens or API tokens, meet the necessary length and complexity standards. By configuring the token key length appropriately, the generated keys adhere to the required security measures. In terms of key distribution, it is essential to ensure that keys are decrypted in a secure environment to prevent unauthorized access. The decryption process reverses the encryption steps: it starts by decoding the Base64-encoded encrypted key into bytes, verifies the secret key, and if correct, extracts the initialization vector (IV) and encrypted data. An AES cipher is then used to decrypt the data, removing any added padding to retrieve the original key value, which is decoded from bytes to a UTF-8 string. The KRMS enables secure key availability through its rotation system which protects keys from interception. The storage process uses AES-CBC encryption to safeguard the keys, which are then combined with IVs prior to Base64 encoding for storage and transmission with error handling mechanisms. The key rotation process follows established policies through automated systems, which minimizes human errors. Through decommissioning, deletion or archiving techniques, the retirement process safely eliminates obsolete keys; however, in the event of a breach, manual intervention is necessary.

4.3 Service Request - Response Module

The Service Request/Response Module within the KRMS serves as an essential communication channel, facilitating collaboration between clients and administrators. This module is designed to streamline and manage the exchange of information and requests related to key management and system operations. This module enables clients to submit detailed requests to administrators. The module produces both system updates and automatic email notifications to keep administrators immediately aware of each request. The KRMS allows administrators to review requests while clients receive email updates that also get logged in the module for future reference. This module gives administrators a controlled system to handle client requests for support. The module maintains a clear record of all responses that administrators make. The system automatically generates an email notification to clients after submission, which enables prompt and transparent documentation for efficient issue resolution.

4.4 Reminder Capability Module

The KRMS reminder function enables administrators to schedule key updates through rotation schedule-based reminders, which help maintain timely key updates. The system adjusts schedules based on defined parameters, which include timing, key groups and notification content that can be modified as needed. Additionally, reminders can be removed from the system once they have served their purpose. This includes automatic removal after a reminder is sent, as well as manual deletion of outdated or irrelevant notifications. This automated cleanup helps maintain an organized and efficient reminder system.

4.5 Dashboard and Monitoring Module

The Dashboard and Monitoring Module in the KRMS is a vital feature that provides administrators and clients with a comprehensive overview of key statuses and ageing. This module is designed to facilitate efficient key management by visually categorizing keys based on their lifecycle stages, allowing for timely key rotations and proactive maintenance.

Dashboard Overview

The dashboard offers a summary view of the key statuses, using a color-coded system to differentiate between various stages of key validity:

- **Valid Keys (Green):** Keys that are currently active and within their designated validity period are displayed in green. The green indicator helps administrators quickly identify which keys are currently operational and in good standing.
- **Close to Expiry (Amber):** Keys that are approaching their expiration date are highlighted in orange or amber. The amber alert allows administrators to plan ahead and initiate key rotation processes before these keys expire, thereby avoiding potential disruptions or security issues.
- **Expired Keys (Red):** Keys that have surpassed their validity period and are no longer usable are marked in red. The red status ensures that expired keys are not used inadvertently and prompts administrators to take necessary actions, such as removing or replacing the expired keys.

Monitoring and Alerts

The monitoring aspect of the module extends beyond visual status indicators to include dynamic tracking and alerting functionalities:

- **Real-Time Monitoring**: The module continuously updates key statuses in real-time, providing administrators with the most current information. This real-time tracking ensures that administrators are always aware of the current condition of each key.
- **Alerts and Notifications**: Automated alerts are configured to notify administrators when keys are approaching expiration or have expired. These alerts can be customized based on specific thresholds and criteria, ensuring that administrators receive timely reminders to take appropriate actions.

4.6 Scheduler Module

The Scheduler Module is responsible for automated key rotation through scheduled alerts which are uniformly applied across systems and services, thereby reducing the risk of key compromise. The Scheduler Module encompasses key functions such as defining the frequency of key rotations—whether daily, weekly, or monthly—and allowing for customization to set different rotation schedules based on the type of key or service and its criticality. Additionally, it includes features for setting reminders and alerts, which automatically notify

stakeholders of upcoming key rotations and provide configurable reminders to facilitate timely preparation and execution of key rotation tasks.

5. Results and Discussion

Usability testing was not conducted on the proposed solution; instead, the focus was exclusively on functional testing to evaluate its performance and reliability. Functional testing involves assessing specific features and functionalities to ensure they operate as intended under various conditions. The results of this testing are detailed in the sections below.

5.1 System Functionality

The functional testing proved that the KRMS executed cryptographic key rotation based on established schedules. Figure 2 below displays the Key Generation and setup screen. The user enters the service name, selects the key administrator, key type, key value, rotation frequency and lastly, schedule type.

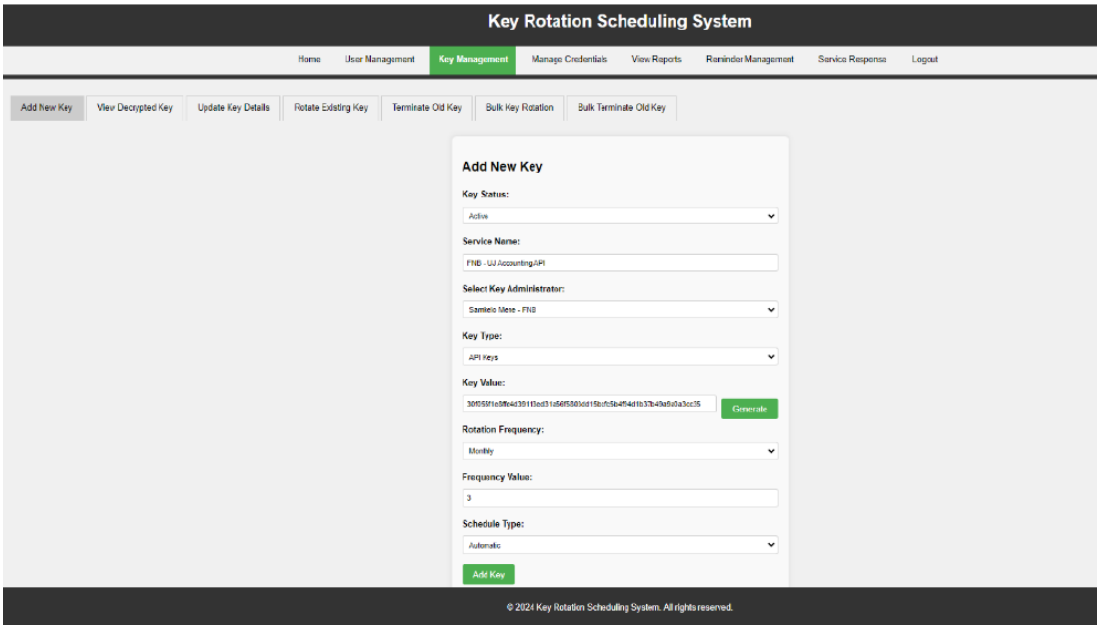

Figure 2: Key Generation and Setup

Figure 3 below provides a real-time overview of key statuses ('Valid', 'Close to Expiry', 'Expired'). But over and above this available facility, the system automates email notifications and directly informs users about key statuses, prompting timely actions and adherence to security protocols.

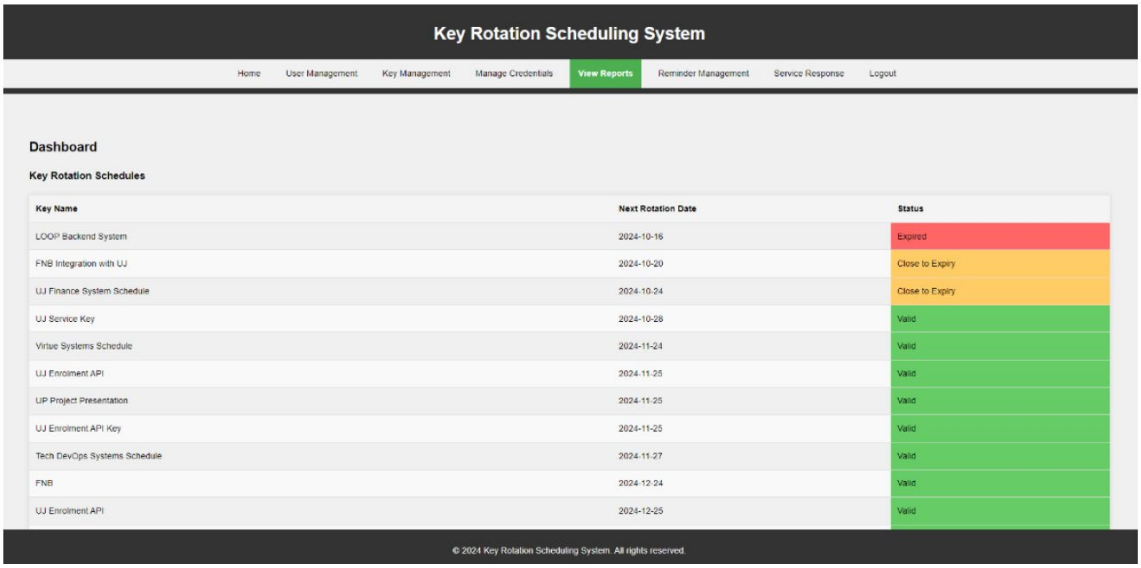

Figure 3: Key Rotation Scheduling

These outputs show that the KRMs provide real-time visibility as well as proactive alerts and verifiable evidence of compliance. The system maintained consistent logs of all lifecycle events, including key generation, storage, rotation and termination, which created a system of accountability while minimizing audit ambiguities.

5.2 System Interaction

The keys are stored in an encrypted form as illustrated in Figure 4 below. It is essential to ensure that keys are encrypted and decrypted in a secure environment to prevent unauthorized access.

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	expiration_date	System_Service_Name	key_type	new_key	old_key	key_status	rotated_datetime	schedule_status
39	2024-10-16	LOOP Backend System	RSA	0TUVzFulAer26s1+XCvqCULrln0ripv6wtaKp…	NULL	1	2024-10-16 09:41:28	manual
40	2024-11-24	Virtue Systems Schedule	Encryption Keys	FDGXhXiC8EqhpaAbx70eeZSyvXZ6GEqjMPo…	NULL	1	2024-09-24 14:08:53	automatic
42	2024-11-27	Tech DevOps Systems Schedule	API Keys	REoLcwc5MgjsEoxU/HQHOPd1KiJPs7x4JshQ…	NULL	1	2024-08-28 15:48:33	automatic
44	2025-09-24	Virtue Systems OPT	CORE	xu6VpZ/NhMwypJwoAmhADCy+b8bf+k+Me…	NULL	1	2024-09-24 17:34:08	manual
45	2025-07-28	InfoSec Team Service	CORE	Ly4lvZLvWh1VZS3yX3xp8/CUiwR5UCuZqlDV…	NULL	1	2024-08-28 15:48:33	manual
47	2024-10-24	UJ Finance System Schedule	Asymmetric Keys	u2D/9y4GTiq7HavfUMhwGQDmaKJ8nNUoz…	NULL	1	2024-09-24 14:09:01	automatic
48	2024-10-28	UJ Service Key	API Access Key	wRPNaz5kHYCFTWTj84I5OEj1oh9v4WS+/y…	NULL	1	2024-08-28 19:45:46	manual
49	2025-10-15	UP Loyalty Service Schedule	Symmetric Keys	TYboKm5w/tARlZDFuUIVqOu2tvldv+qEZBn…	NULL	1	2024-10-15 20:17:34	automatic
50	2025-08-22	Virtue Systems OPT	CORE	fd+UnrWkbjuPbe+Lt1nMDcU0KR6QN7N/NI…	NULL	1	2024-08-28 15:48:33	manual
51	2024-11-25	UJ Enrolment API	ACCESS	483BM6060TXKJSDILt8Nxarje603GSxVHNP8…	NULL	1	2024-09-25 08:46:09	manual
52	2024-12-24	FNB	API Access Key	qpNk5bukTPtumNSchQGO0CotAX3Wd+OjL9…	NULL	1	2024-09-24 17:32:59	manual
53	2024-12-25	UJ Enrolment API	API Access Key	5jxvHhnOs9BGzm6sRIYjN7UsVQDG8Au+fYD…	NULL	1	2024-09-25 08:40:32	manual
54	2024-11-25	UP Project Presentation	API Access Key	LNTDRsnmY+Z+7yb9il0cRvRUIK58uVjC6JDJ…	NULL	1	2024-09-25 09:42:44	manual
56	2024-11-25	UJ Enrolment API Key	API Access Key	yYqvUMsnZwH5m/CAYyeqy8KF62jgN9CFbi2…	NULL	1	2024-09-25 11:18:29	manual
57	2025-10-20	UJ Enrolment API with Home Affairs Sch…	API Keys	xfbCAx6Kokx34a5qEOyjrfvF74smJd4FhSxR…	qBjHlYA9rcnZ0POb+LoVN7uSU8TDohbi…	1	2024-10-20 17:50:16	automatic
60	2025-01-16	UJ Enrolment API with Home Affairs	API Keys	FDPxED3YRiYLXqjgZQV7+Uc2s3GuU9KVrSp…	NULL	1	2024-10-16 09:40:22	automatic
62	2024-10-20	FNB Integration with UJ	Session Keys	NBKj0BMbLESVCbUulYIwzRb/H3xdsAMJVN0s…	8f9x4de6gHWKThEO0AhmgiVmeCBWx…	1	2024-10-20 17:00:25	automatic
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 4: Cryptographic Key Storage

The system achieved real-time capabilities alongside lifecycle visibility (key generation, storage, distribution, rotation, and termination) by using multiple channels that supported immediate reaction and long-term oversight. The proposed KRMS provides a centralised platform for all system communications, which eliminates the requirement for administrators to track multiple information streams. The stream implements role-based views that grant administrators and super administrators access to appropriate tools and information based on their job responsibilities.

5.3 System Validation and Reliability

The functionality testing process confirmed that the KRMS executed all its fundamental lifecycle operations as specified in the design. The distribution process proved to be streamlined, allowing for safe and efficient key sharing with authorized parties. Key rotation was automated, reducing manual intervention and minimizing the risk of human error. Finally, the key termination process guarantees that expired or compromised keys are promptly and securely deactivated, enhancing overall data protection. Collectively, these capabilities contribute to the robust security framework provided by the KRMS, ensuring that all aspects of key management are effectively handled.

5.4 Critique and Analysis

The automation of key rotation greatly enhances efficiency, reducing the risk of human error and ensuring keys are rotated at appropriate intervals without manual intervention. This consistency is crucial for adhering to key rotation security policies. The proactive notification system, such as email reminders, helps inform users of upcoming expirations and necessary actions, fostering a culture of security awareness and compliance. Audit trails also help meet regulatory requirements, providing evidence that key management policies are being followed. Error handling further improves the system's reliability by managing failures gracefully, reducing the likelihood of complete system shutdowns, and ensuring smoother operations. The results of this study proved technical feasibility, yet several areas require improvement.

The lack of integration with calendars and the selection of business days creates inefficiencies in scheduling. Relying on hard-coded start dates for tasks makes adapting the system to different environments challenging, necessitating frequent updates or maintenance. Limited scalability is another concern; performance might degrade under high volumes of reminders or key rotations, indicating potential bottlenecks. Ambiguities in handling time zones complicate scheduling tasks across different geographical locations, especially in global organizations. Lastly, the reliance on a single instance of the scheduler presents a vulnerability—if that instance fails, all scheduled tasks may not execute.

To address these issues, future research could explore more advanced scheduling algorithms that adapt to load and resource availability, improving efficiency and responsiveness. Understanding user behaviour through analytics could provide insights into enhancing the user experience and identifying potential security risks. Integrating the KRMS with other security frameworks, such as Security Information and Event Management

(SIEM) systems, could strengthen the overall security posture and incident response capabilities. SIEM will improve the KRMS because it can collect, store, aggregate, and correlate events and information generated by a managed system infrastructure. Developing smarter notification systems that adapt based on user preferences and behaviours may increase engagement and compliance rates. Additionally, incorporating more detailed reporting and analytics capabilities would allow for better insights into key management activities and trends.

6. Conclusion

Organizations must maintain effective key management systems because the changing information security environment demands protection of sensitive data during collaborative technology adoption. The proposed KRMS solves key rotation problems by implementing a systematic, automated system for cryptographic key management throughout its entire existence. The KRMS uses automated integration with existing IT systems to simplify key rotation processes while reducing human involvement and minimizing errors. The Model-View-Controller pattern-based layered architecture provides scalability alongside maintainability and clear separation of concerns. Key components of the KRMS, including the User Management Module, Key Management Module, Service Request/Response Module, Reminder Capability Module, Dashboard and Monitoring Module, and Scheduler Module, each play a crucial role in enhancing the security and efficiency of key management practices. These components collectively support robust key lifecycle management by facilitating secure key generation, distribution, storage, rotation, and termination. Additionally, the system's reminder and scheduling features help maintain compliance with regulatory standards and organizational policies, ensuring timely key updates and minimizing security risks.

The study found that the KRMS effectively addresses the initial challenges identified, including the necessity for manual administrator intervention for key updates, a lack of awareness about key rotation processes, inadequate monitoring of key age, insecure key distribution methods, and insufficient communication channels. The implementation of a single point of communication, automated user notifications, and automated key rotation has significantly improved user engagement, efficiency, security, user satisfaction, and awareness of key management processes. The KRMS demonstrated high success rates in minimizing human error and enhancing operational efficiency through automated key rotations. While initial qualitative feedback is positive, further research with a broader user base is necessary to fully validate these findings. KRMS demonstrates through its automated key lifecycle management that organisations can achieve better security. The KRMS offers an extensible automated key rotation system which enhances information security practices and enables organisations to handle rising collaborations and technology complexity.

Acknowledgements

Not Applicable

Ethics Declaration: Ethical clearance was not required for this study as it did not involve human participants.

AI declaration: Grammarly was utilized for editing and grammar refinement.

References

Ahanger, A.S., Masoodi, F.S., Khanam, A. and Ashraf, W. (2024) Managing and securing information storage in the Internet of Things, in Internet of Things Vulnerabilities and Recovery Strategies, Auerbach Publications, pp. 102–151.

Ahmad, S., Mehfuz, S. and Beg, J. (2023) "Hybrid cryptographic approach to enhance the mode of key management system in cloud environment", The Journal of Supercomputing, Vol. 79, No. 7, pp. 7377–7413.

Ahmad, S., Nazim, M., Arif, M., Ahmad, J., Mehfuz, S. and Ansari, M.A. (2025) "Protecting data in the cloud: A systematic literature review of key management", Concurrency and Computation: Practice and Experience, Vol. 37, Nos. 21–22, p. e70223.

Barker, E. and Barker, W. (2018) Recommendation for Key Management, Part 2: Best Practices for Key Management Organization (Draft), NIST Special Publication (SP) 800-57 Part 2 Rev. 1, National Institute of Standards and Technology, Gaithersburg.

Bethel, K.L. (2020) An Evaluation of Organizational Culture: Its Influence on Security Culture: A Case Study, Doctoral dissertation, Northcentral University.

Faisal, M., Ali, I., Khan, M.S., Kim, J. and Kim, S.M. (2020) "Cyber security and key management issues for internet of things: Techniques, requirements, and challenges", Complexity, Vol. 2020, No. 1, p. 6619498.

Gautam, A.K. and Kumar, R. (2021) "A comprehensive study on key management, authentication and trust management techniques in wireless sensor networks", SN Applied Sciences, Vol. 3, No. 1, p. 50.

Hinkes, A.M. (2018) "Throw away the key, or the key holder? Coercive contempt for lost or forgotten cryptocurrency private keys, or obstinate holders", Northwestern Journal of Technology & Intellectual Property, Vol. 16, pp. 225.

Hughes-Lartey, K., Li, M., Botchey, F.E. and Qin, Z. (2021) "Human factor, a critical weak point in the information security of an organization's Internet of Things", Heliyon, Vol. 7, No. 3.

Hurst, E.J. (2020) "Web conferencing and collaboration tools and trends", Journal of Hospital Librarianship, Vol. 20, No. 3, pp. 266–279.

Joshi, S., Crowther, K. and Robinson, J. (2024) "Tradeoffs in key rotation strategies for industrial internet of things devices and firmware", Applied Sciences, Vol. 14, No. 21, p. 9942.

Kim, H., Wasicek, A., Mehne, B. and Lee, E.A. (2016) "A secure network architecture for the internet of things based on local authorization entities", in 2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud), IEEE, pp. 114–122.

Kuzminykh, I., Yevdokymenko, M. and Ageyev, D. (2020) "Analysis of encryption key management systems: strengths, weaknesses, opportunities, threats", in 2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T), IEEE, pp. 515–520.

Mpekoa, N. (2024) "An analysis of cybersecurity architectures", in International Conference on Cyber Warfare and Security, Academic Conferences International Limited, March, pp. 200–207.

Nahhas, S. (2021) "MVC architecture from maintenance quality attributes perspective", International Journal of Computer Science and Security (IJCSS), Vol. 15, No. 5, pp. 169–181.

Nakamura, Y., Fukuda, T., Yang, X. and Takefuji, Y. (2025) "SSHAA: A Python package index for visualizing features of SSH attacks with text mining in classification", Information Security Journal: A Global Perspective, Vol. 34, No. 1, pp. 50–62.

Radanliev, P. (2023) Cyber-attacks on Public Key Cryptography.

Sakthivel, M., Sivanantham, S., Bharathiraja, N., Krishna, N.B., Kamalraj, R. and Kumar, V.S. (2024) "Ensuring web application security: An OWASP driven development methodology", in 2024 International Conference on Knowledge Engineering and Communication Systems (ICKECS), IEEE, April, Vol. 1, pp. 1–7.

Seaman, J. (2020) PCI DSS: An Integrated Data Security Standard Guide, Apress.

Torres, A.G. (2022) "Encryption key lifecycle management and best practices for maintaining trusted e-commerce services in the cloud", Journal of Artificial Intelligence and Machine Learning in Cloud Computing Systems, Vol. 6, No. 11, pp. 1–8.

Vlajic, N. and Cianfarani, G. (2023) "Risk-based methodology for optimal cryptoperiod calculation in ICSs under data siphoning attack", in Proceedings of the 2024 Workshop on Re-design Industrial Control Systems with Security, November, pp. 87–95.

Yassein, M.B., Aljawarneh, S., Qawasmeh, E., Mardini, W. and Khamayseh, Y. (2017) "Comprehensive study of symmetric key and asymmetric key encryption algorithms", in 2017 International Conference on Engineering and Technology (ICET), IEEE, August, pp. 1–7.

Yee, C.K. and Zolkipli, M.F. (2021) "Review on confidentiality, integrity and availability in information security", Journal of ICT in Education, Vol. 8, No. 2, pp. 34–42.

Zhang, Q., Jia, S., Chang, B. and Chen, B. (2018) "Ensuring data confidentiality via plausibly deniable encryption and secure deletion – a survey", Cybersecurity, Vol. 1, No. 1, p. 1.