RESEARCH ARTICLE

JENER Journal of Empirical and Non-Empirical Research
Volume 2, Issue 10, Page 1298-1312 | Article No. 135
DOI: 10.4898/jenerv2i10.a135
ISSN: 7895-9216
Journal homepage: https://jenerjournal.com
Journal email: editor@jenerjournal.com

JENER Journal
JENER Journal of Empirical and Non-Empirical Research

Design And Development of a Secure Logging Mechanism to Resist Anti-Forensic Manipulation

Wongani Mwanza[1] & Moses Mupeta (PhD)[2]

[1,2]School of Engineering, Information and Communications University

ARTICLE INFORMATION

Article history:
Published: October 2026

Keywords:
Design
Development
Secure Logging
Anti-Forensic Manipulation

ABSTRACT

This project addresses the critical vulnerability of conventional logging mechanisms to sophisticated anti-forensic manipulation techniques that compromise the integrity and authenticity of digital evidence. Traditional logging systems rely on centralized architectures susceptible to unauthorized modifications, deletions, and data tampering, thereby undermining forensic investigations and organizational accountability. The research aims to design and develop a secure logging mechanism that integrates advanced cryptographic techniques, blockchain-based verification, and real-time tamper detection capabilities to resist anti-forensic manipulation attempts. The proposed system incorporates encrypted log storage using robust encryption algorithms, immutable blockchain verification for maintaining data integrity, and a tamper-evident alert mechanism that provides immediate notification of unauthorized access attempts. Additionally, a React-based monitoring dashboard will be developed to facilitate real-time log monitoring and system status visualization. The methodology employs an experimental design approach combined with practical implementation using Python for backend development, Flask for API integration, and blockchain technology for immutable storage. The system will be evaluated through controlled experiments simulating various log manipulation scenarios to assess its effectiveness in detecting and preventing tampering attempts. The expected outcomes include enhanced digital evidence reliability, strengthened forensic investigation capabilities, and improved organizational security posture through comprehensive audit trail protection.

1. Introduction

Digital systems generate vast amounts of log data that play a critical role in monitoring system activities, troubleshooting operational issues, and supporting post-incident investigations. In contemporary computing environments ranging from enterprise information systems to cloud-based infrastructures logs provide the primary source of evidence for detecting intrusions, reconstructing security events, and assessing the extent of potential compromises. However, as cyber-attacks become increasingly sophisticated, adversaries have developed anti-forensic techniques specifically targeting log files to hide their tracks and mislead investigators. These techniques include modifying log entries, deleting records, injecting false events, altering timestamps, and disabling logging services altogether.

Traditional logging mechanisms, though effective for routine system monitoring, often lack strong integrity controls and tamper-evidence capabilities. Once an attacker gains elevated privileges, conventional logs stored in plaintext or on writable disk locations can be manipulated with minimal effort. This undermines the reliability of digital evidence, compromises the accuracy of forensic analysis, and weakens incident response procedures. As a result, there is a pressing need for secure logging mechanisms that can ensure the authenticity, integrity, and verifiability of log data even in the presence of strong adversaries.

This study focuses on the design and development of a tamper-resistant logging mechanism capable of detecting and resisting anti-forensic manipulation. The research integrates principles of cryptography, secure system design, and immutable storage to produce a logging approach that preserves the evidential value of log records. The proposed mechanism aims to provide guarantees such as tamper-evidence through hash-chaining, secure key management, trusted timestamping, remote log replication, and verification processes capable of identifying unauthorized alterations.

By addressing the shortcomings of conventional logging systems and mitigating common anti-forensic practices, the project contributes to the enhancement of digital forensics readiness, incident response effectiveness, and overall system security. The chapter that follows outlines the background, problem statement, research objectives, research questions, justification, scope, and significance of the study.

1.1 Background

Logging is a fundamental component of modern information systems, providing a chronological record of system and user activities necessary for monitoring, troubleshooting, incident response, and digital forensics. Audit logs assist security analysts in reconstructing security events and identifying malicious behaviour after a compromise. As noted by Ali, Khan and Arshad (2021), audit logs are "key resources that show the current state of the systems and user activities and are used for cyber forensics and maintenance" (p. 2).

However, traditional logging mechanisms are highly vulnerable to manipulation, especially when attackers obtain elevated privileges. Once a system is compromised, adversaries commonly employ anti-forensic techniques such as log deletion, log file modification, timestamp alteration, disabling log services, and injecting fake entries to conceal evidence of their activities. Ali et al.

(2021) further emphasize that logs are "prone to multidimensional threats" and that privileged insiders "can alter these logs as and when required" (p. 9). This undermines the evidential value of logs and weakens their reliability during forensic investigations.

To address these weaknesses, secure logging mechanisms have evolved to incorporate cryptographic protections and tamper-evident structures. One of the most widely studied approaches is hash-chaining, where each log entry includes the cryptographic hash of the previous entry. This ensures that any modification breaks the chain and becomes detectable. Research by Guardiola-Múzquiz and Soriano-Salvador (2022) highlights the effectiveness of combining hash-chaining with ratcheting and forward-secure cryptography to protect logs generated before a system compromise.

Other researchers propose tamper-resistant hardware, trusted execution environments, and immutable storage back-ends such as write-once media or distributed ledgers to enhance log integrity even when the host operating system is compromised. For example, Bellare and Yee (1997) demonstrated that secure audit logging using tamper-resistant hardware significantly reduces the risks associated with local log manipulation.

Despite these advancements, the integration of secure logging in real-world systems remains limited. Many existing solutions either impose high performance overhead, require complex key-management infrastructures, or depend heavily on specialized hardware not widely available. Recent evaluations continue to show that high-performance, tamper-evident systems may suffer from data loss under load or deployment challenges in cloud and container environments (Chen et al., 2025). This indicates a continuing gap between theoretical secure logging models and practical, deployable solutions.

Given these challenges and the growing sophistication of anti-forensic techniques, there is a need to design a secure logging mechanism that not only provides strong tamper-evidence and integrity guarantees but also remains lightweight, scalable, and practical for deployment. This study aims to contribute to filling this gap by developing a secure, tamper-resistant logging mechanism capable of detecting and resisting anti-forensic manipulation.

1.2 Motivation of the Study

The increasing frequency and sophistication of cyber-attacks have elevated the importance of reliable log data as a source of truth during incident response and forensic investigations. In many real-world security breaches, attackers routinely attempt to erase or alter log records to conceal unauthorized activities. This practice commonly referred to as anti-forensic manipulation significantly undermines the integrity and evidential value of digital logs. The motivation for this study arises from the need to develop logging mechanisms capable of maintaining trustworthiness even when adversaries gain elevated privileges on a system.

Conventional logging infrastructures, such as system log files, application logs, and security audit trails, typically lack robust integrity protection. Once an attacker compromises a host, they can easily delete entries, modify timestamps, inject false information, or disable logging altogether. Such weaknesses continue to affect organisations across various sectors, where incident investigations are often hindered by incomplete or tampered logs. The inability to rely on logs delays forensic analysis, obstructs root-cause identification, and may lead to incorrect conclusions about the nature and extent of an attack.

Furthermore, the growing adoption of distributed architectures including cloud computing, microservices, virtualization, and IoT ecosystems has expanded the attack surface, making log protection more challenging. In these environments, logs are generated across multiple nodes, transmitted over networks, and stored in diverse systems. Without mechanisms that guarantee end-to-end integrity, organisations risk losing critical evidence needed for compliance, auditing, and accountability.

Advances in secure logging research, including hash-chains, digital signatures, immutable storage, and hardware-based trust anchors, provide promising pathways for protecting logs. However, many of these solutions remain complex, resource-intensive, or impractical for broad deployment in everyday systems. There is therefore a compelling need to design a secure, lightweight, and practical logging mechanism that resists tampering and provides verifiable integrity without imposing excessive overhead.

This study is motivated by the desire to bridge this gap by developing a secure logging mechanism that enhances digital forensic readiness and strengthens organisational resilience against cyber threats. By addressing the challenges of anti-forensic manipulation, the research contributes toward improving the reliability of evidence, supporting effective incident response, and advancing secure system design practices.

1.3 Significance of the Study

The significance of this study lies in its contribution to enhancing the security, reliability, and forensic value of logging systems in modern computing environments. As cyber-attacks continue to grow in complexity, anti-forensic techniques targeting log files have become common tools used by attackers to hide traces of their activities. By designing and developing a secure logging mechanism that resists such manipulation, this research addresses critical gaps in current security and forensic practices.

Firstly, the study contributes to digital forensic readiness by ensuring that organisations have access to trustworthy and verifiable log data during incident investigations. Reliable logs enable security analysts to reconstruct attack timelines accurately, identify compromised components, and make informed decisions in response to security breaches. A tamper-resistant logging mechanism therefore strengthens an organisation's ability to respond to and recover from cyber incidents.

Secondly, this study is significant to the field of cybersecurity engineering, as it proposes a practical and implementable solution based on cryptographic hash-chaining, secure storage, and integrity verification techniques. These approaches not only make log tampering detectable but also improve accountability in security-critical systems. The findings of this research may guide system designers, developers, and administrators in implementing logging mechanisms that maintain evidential integrity even when attackers obtain elevated privileges

1.4 Scope of the Study

This study focuses specifically on the design, development, and evaluation of a secure logging mechanism intended to resist anti-forensic manipulation. The scope is limited to addressing the integrity, authenticity, and tamper-evidence of log data within computer systems. The research concentrates on preventing and detecting unauthorized modifications such as log deletion, alteration, injection of false records, and timestamp manipulation.

The study is confined to the development of a prototype that integrates cryptographic techniques—such as hash-chaining, secure key management, and integrity verification mechanisms—together with an appropriate storage architecture capable of preserving log immutability. The prototype is evaluated using controlled experiments that simulate common tampering attempts, performance testing, and verification checks. The assessment focuses on aspects such as detection accuracy, system overhead, and practicality of deployment. The study does not cover the design of a complete organisation-wide security information and event management (SIEM) system, nor does it address long-term archival strategies, advanced analytics, machine learning-based log analysis, or real-time anomaly detection. Similarly, the research does not incorporate specialised hardware such as full-scale HSM appliances unless used at a basic conceptual level. The scope is further limited to software-based implementations and experimentation in a controlled environment rather than deployment in a live organisational infrastructure. The findings and conclusions therefore relate primarily to the proposed mechanism's performance, feasibility, and effectiveness under simulated conditions.

1.5 Problem Statement

Audit logs play a critical role in digital forensics, intrusion detection, and incident response; however, their reliability is severely undermined by the ease with which attackers can manipulate them once they obtain elevated privileges. Traditional logging mechanisms often lack strong integrity controls, leaving them vulnerable to anti-forensic techniques such as log deletion, modification, timestamp alteration, and the injection of false entries. Numerous studies show that adversaries routinely use these techniques to conceal malicious actions, resulting in incomplete or misleading forensic trails (Ali, Khan, & Arshad, 2021). Although secure logging research has introduced mechanisms such as hash-chaining and forward-secure signatures, many of these solutions remain complex, resource-intensive, or impractical for large-scale or real-time systems.

Recent findings reveal that even state-of-the-art tamper-evident logging systems suffer from significant performance overhead, coarse-grained tamper detection, and potential data loss during high-load operations, making them unsuitable for modern cloud and distributed environments. Zhao, Shoaib, Hoang, and Hassan (2025) demonstrate that current secure logging architectures still allow attackers with system-level privileges to manipulate or erase logs without immediate detection, thereby compromising the accuracy of forensic investigations. This ongoing vulnerability underscores a fundamental gap between theoretical secure logging models and the practical needs of real-world systems.

Therefore, the core problem addressed by this study is the lack of a lightweight, practical, and cryptographically secure logging mechanism that can reliably detect and resist anti-forensic manipulation while maintaining acceptable system performance. Without such a mechanism, organisations remain at risk of losing critical evidence, delaying incident response, and misinterpreting the extent or nature of cyber-attacks.

1.6 Objectives

This section clearly defines what the project aims to achieve. Objectives are essential because they provide a structured roadmap for the research and set measurable targets for success.

1.6.1 General Objective

To design and develop a secure logging mechanism that resists anti-forensic manipulation, ensuring the integrity and authenticity of digital evidence.

1.6.2 Specific Objectives

These are the detailed, actionable steps required to accomplish the general objective. Each specific objective focuses on a key aspect of the secure logging system.

- To develop a secure encryption method for log data storage that ensures confidentiality while maintaining data integrity.
- To implement blockchain-based verification for creating immutable log entries that resist tampering attempts.
- To design a tamper-evident alert mechanism and monitoring dashboard for detecting log manipulation attempts and facilitating incident response.

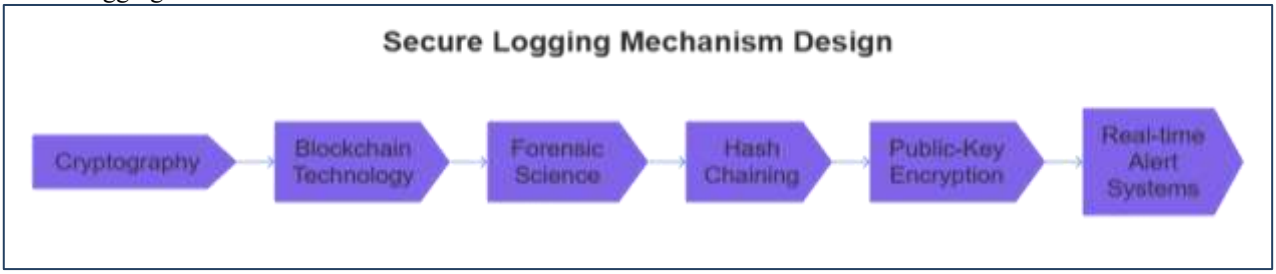
1.7 Research Questions

These questions are derived directly from the objectives and serve to guide the research process and define the scope of the investigation.

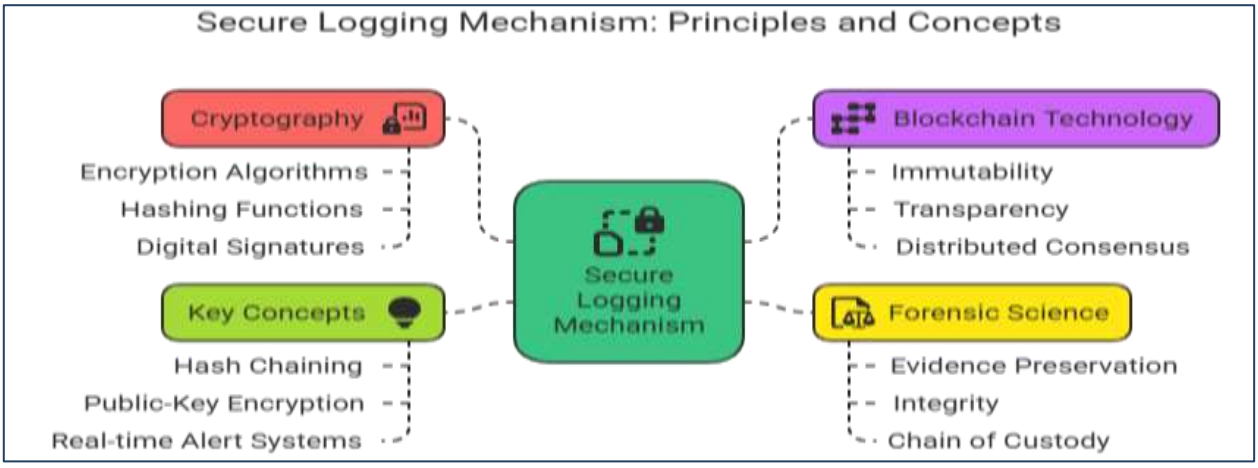
- How can encryption improve the security and integrity of stored log data? (Corresponds to
- What role can blockchain technology play in ensuring the immutability of log entries?
- What methods can effectively detect and alert users about tampering attempts, and how can a user-friendly dashboard enhance log monitoring and system transparency?

1.8 Conceptual/Theoretical Framework

The study will integrate principles, theories, and concepts from various disciplines that will underpin the design and development of the secure logging mechanism from:



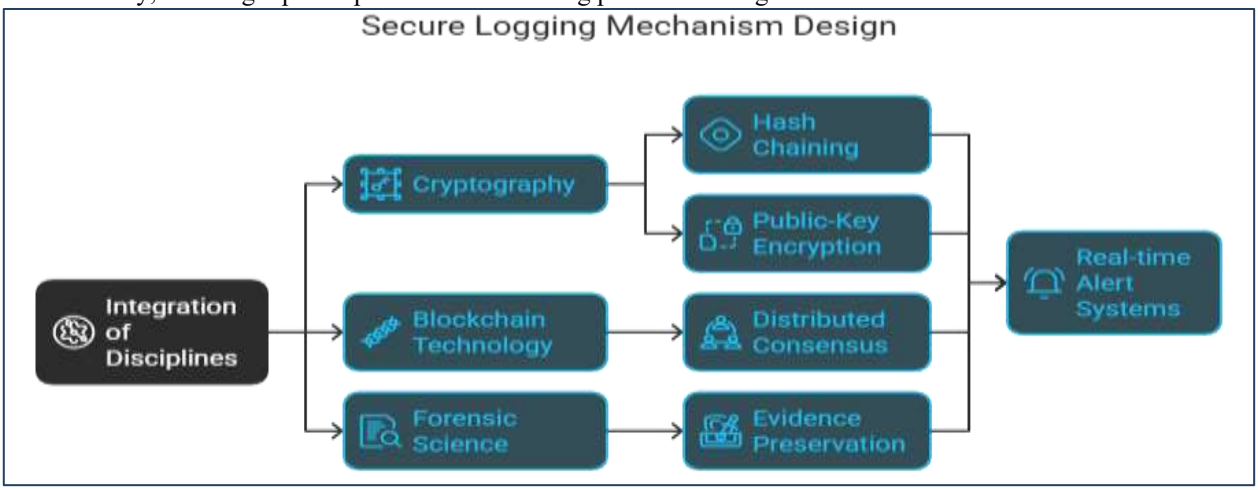
With the key principles and concepts as shown in the image below:



- Cryptography: To provide confidentiality, integrity, and authentication of log data through the use of encryption algorithms, hashing functions, and digital signatures.
- Blockchain Technology: To ensure immutability, transparency, and distributed consensus in the storage and verification of log entries.
- Forensic Science: To adhere to best practices in evidence preservation, integrity, and chain of custody, ensuring that log data is admissible and reliable in legal proceedings.

Key concepts that will guide the design include:

- Hash Chaining: A technique that links log entries together using cryptographic hashes, creating a chain of dependencies that makes it computationally infeasible to alter a previous log entry without invalidating all subsequent entries.
- Public-Key Encryption: An encryption method that uses a pair of keys (a public key for encryption and a private key for decryption) to secure log data and control access to it.
- Real-time Alert Systems: Automated mechanisms that provide immediate notification of any detected tampering attempts or suspicious activity, enabling rapid response and minimizing potential damage.



1.9 Definition of Terms

To provide clear, concise, and unambiguous definitions of key terms used throughout the report to ensure consistent understanding and avoid misinterpretations. Here is a definition of terms as followings:

- Anti-forensic techniques: Methods and strategies employed by attackers to manipulate, delete, suppress, or obscure digital evidence in order to hinder or prevent forensic investigations (Chow et al., 2020).
- Blockchain verification: A decentralized method of securing data and verifying its integrity by linking records into immutable blocks that are distributed across a network of computers (Yli-Huumo et al., 2016).
- Encryption: The process of transforming data into an unreadable format (ciphertext) using an algorithm (cipher) to prevent unauthorized access, ensuring data confidentiality (Bellare & Yee, 1997).
- Digital forensics: The process of identifying, preserving, analyzing, and presenting electronic evidence in legal investigations (Lillis et al., 2016).
- Immutable logs: Log entries that cannot be altered or deleted once recorded, ensuring data integrity and providing reliable audit trails for forensic investigations (Khan et al., 2020).
- Log integrity: Ensuring that log data remains unchanged, accurate, and reliable over time, maintaining the authenticity of digital evidence (Watson et al., 2015).
- Secure logging mechanism: A system designed to securely store and verify log data to prevent manipulation while maintaining accessibility for legitimate forensic investigations (International Journal of Information Security, 2022).
- Tamper-evident design: The design and implementation of a system with features that can detect and visibly indicate any unauthorized modifications or alterations to data, ensuring data integrity (Kelsey et al., 1999).

- Cryptographic hashing: A method of converting data into a fixed-size string of characters, ensuring data authenticity and integrity through mathematical algorithms (Future Internet, 2023).
- React Framework: A JavaScript library for building user interfaces, widely used for developing dynamic web applications with component-based architecture (Zyskind et al., 2015).

1.10 Organization of the Report

The report is structured as follows:

- Chapter One: Introduction: This chapter introduces the research project, providing background information, outlining the motivation and significance of the study, defining the scope, presenting the problem statement, and stating the objectives, research questions, conceptual framework, and definitions of terms.
- Chapter Two: Literature Review: This chapter reviews existing literature, research, and studies related to secure logging mechanisms and anti-forensic techniques, identifying current solutions, their limitations, and the gaps that this project aims to address.
- Chapter Three: Methodology: This chapter details the research methodology employed in the project, including the research design, data collection methods, system architecture, development tools, implementation steps, and evaluation strategies.
- Chapter Four: System Evaluation, Testing Results, and Performance Analysis: This chapter presents the results of the system evaluation, including testing procedures, performance metrics, and analysis of the system's effectiveness in resisting anti-forensic manipulation.
- Chapter Five: Conclusion, Recommendations, and Potential Improvements: This chapter concludes the report by summarizing the key findings, providing recommendations for future research and development, and suggesting potential improvements to the developed secure-logging mechanism.2. Literature Review

2.1 Review of the Literature

The literature consistently shows that logging and audit trails are indispensable for system monitoring, accountability, and digital forensic investigations. However, researchers have identified that these logs are vulnerable to manipulation once a system is compromised. For instance, Studiawan (2019) highlights that operating system event logs can be easily altered or deleted by attackers, thereby undermining accurate reconstruction of security incidents. Similarly, Henriques (2024) observes that organisations especially those managing critical infrastructures still lack sufficient controls to ensure audit-trail integrity under advanced threat conditions.

To address these vulnerabilities, several secure logging mechanisms have been proposed in the literature. Early work by Snodgrass (2004) introduced the concept of integrating one-way hash chains within databases to detect unauthorized log modifications, providing a foundational approach to tamper-evident logging. Subsequent research has explored distributed ledger and blockchain-based logging models; Westerlund et al. (2018) demonstrate that blockchain-backed audit trails can improve log immutability and resilience, particularly in distributed and IoT environments. More recent developments focus on improving scalability and efficiency. For example, Koisser and Sadeghi (2023) propose lightweight hash-tree constructions designed for large-scale IoT deployments, arguing that many existing tamper-evident schemes impose excessive computational overhead.

Despite these advancements, significant challenges remain. Modern secure logging designs often fall short when evaluated under realistic system loads. Zhao, Shoaib, Hoang, and Hassan (2025) show that state-of-the-art tamper-evident logging systems still suffer from substantial performance overhead, coarse-grained tamper detection, and possible data loss during high-event-rate scenarios. Their findings indicate an ongoing gap between academic solutions and practical, deployable logging architectures. The need to address anti-forensic manipulation is further emphasized by Surakanti (2025), who documents how attackers routinely perform log deletion, timestamp manipulation, and evidence forgery to compromise forensic investigations.

Collectively, the literature reveals that although various secure logging mechanisms exist, many remain impractical due to scalability issues, performance bottlenecks, or deployment complexity. This creates a clear space for research aimed at developing a lightweight, practical, and tamper-evident logging mechanism that can resist anti-forensic manipulation while maintaining operational efficiency.

2.2 Related Works

Several secure-logging systems have been proposed to address log integrity and anti-forensic manipulation. This section reviews three systems that form the foundation for modern tamper-evident logging.

2.2.1 Schneier & Kelsey's Secure Audit Log System (Hash-Chained Logging)

One of the earliest systems is the secure audit log method by Schneier and Kelsey (1999). Their design uses forward integrity and hash-chaining, where each log entry links to the previous one. If an attacker modifies or deletes any entry, the chain breaks and verification fail. The system also protects keys by updating them after every log entry. The main limitation is that it can be slow at scale and does not support large distributed environments.

2.2.2 Bellare & Yee's Forward-Secure Logging System

Bellare and Yee (1997) introduced a logging model based on forward-secure signatures. The signing key evolves over time, making earlier keys unrecoverable even if an attacker compromises the system later. This makes past logs hard to falsify. The system is strong in theory but can be complex to deploy because it depends on continuous key updates and careful key-state management.

2.2.3 Tamper-Evident High-Performance Logging (Zhao et al., 2025)

A recent system by Zhao, Shoaib, Hoang and Hassan (2025) focus on co-designed high-performance secure logging. Their architecture combines hardware support and structured metadata to detect tampering with low delay. Although the system improves throughput, the authors report issues such as data loss under heavy load and coarse tamper detection. These limitations show the need for more practical and reliable logging mechanisms.

2.3 Gaps in the Literature

Although several secure logging mechanisms have been proposed, the literature reveals persistent gaps that limit their practical deployment and effectiveness in resisting anti-forensic manipulation. First, many traditional and cryptographic logging systems assume a relatively stable or low-volume environment. However, recent evaluations such as those by Zhao et al. (2025) show that modern secure logging architectures experience performance degradation, data loss, or coarse-grained tamper detection when subjected to high event rates. This indicates that scalability and real-time efficiency remain significant challenges that current solutions do not adequately address.

Second, most secure logging systems rely on complex cryptographic operations or continuous key evolution, as seen in the schemes by Schneier and Kelsey (1999) and Bellare and Yee (1997). While effective in theory, these mechanisms require precise key management and introduce computational overhead that can hinder deployment in resource-constrained or distributed environments such as IoT devices, containerized systems, and cloud microservices.

Third, existing literature rarely integrates a complete set of protections that combine tamper-evidence, secure storage, resistance to insider threats, and practical verification processes. Many solutions focus narrowly on integrity without considering secure transmission, distributed storage, or the ability to verify logs easily after an incident. Consequently, the forensic value of logs remains limited in environments where attackers may alter multiple components of the logging pipeline.

Finally, although anti-forensic techniques have evolved, existing secure logging mechanisms do not fully account for sophisticated adversarial behaviors such as timestamp manipulation, selective log injection, or coordinated multi-node tampering. Studies such as Surakanti (2025) highlight that attackers often manipulate logs in subtle, multi-stage ways that current detection methods fail to capture.

Overall, the literature demonstrates a need for a lightweight, scalable, and practical secure logging mechanism that provides fine-grained tamper evidence, minimizes performance overhead, and remains resilient against both insider and external anti-forensic manipulation. These gaps motivate the development of the proposed system in this study.

3. Methodology

3.1 Baseline Study

The baseline study was conducted to examine the current state of logging practices, identify existing vulnerabilities, and understand how conventional logging mechanisms behave under anti-forensic attacks. This foundational assessment allowed the researcher to establish a reference point against which improvements from the proposed secure logging mechanism could be measured. The baseline study involved observing how standard system logs such as Linux syslog, application logs, and authentication records are generated, stored, transmitted, and protected in their default configurations.

As part of this assessment, commonly used logging frameworks (e.g., Syslog, journald, and basic file-based logging) were evaluated to determine their susceptibility to tampering. The study tested how easily an attacker with elevated privileges could modify, delete, or inject false log entries without detection. Weaknesses such as lack of cryptographic integrity, absence of tamper-evident structures, and dependence on local file permissions were documented. These baseline tests also examined how logs behave under typical system operations, including high event rates and system restarts, to identify issues related to reliability and completeness.

Furthermore, the baseline study compared conventional logging behaviours with characteristics highlighted in prior research. Studies such as Ali et al. (2021), Koisser and Sadeghi (2023), and Zhao et al. (2025) consistently report that traditional logging mechanisms provide insufficient protection against insider threats and sophisticated anti-forensic manipulation. Findings from the baseline assessment in this study confirmed these observations, showing that existing logging systems do not offer mechanisms to detect unauthorized changes or ensure the integrity of logs over time.

The results of the baseline study therefore established the need for a secure, tamper-evident logging approach and directly informed the design requirements, threat model, and evaluation criteria used in the subsequent stages of this research.

3.2 Data Collection

A well-structured questionnaire and face-to-face interviews were employed to gather information from customers and managers across various businesses. The questionnaire included both open and closed-ended questions to capture a range of responses. Closed-ended questions were directed towards customers, while open-ended questions were intended for managers. Respondents rated their perceptions of four variables using a five-point Likert scale. Closed-ended questions facilitated straightforward computation of responses, while open-ended questions provided richer, qualitative insights. Each questionnaire took approximately 15 minutes to complete. Both primary and secondary data were collected: primary data focused on the current state of the variables under study, and secondary data provided insights into the perceived performance of the bank. The study utilized both quantitative and qualitative data collection methods.

3.3 Research Approach

The software development methodology will be used to implement a Customer Relation Management system.

The Waterfall Model was the first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

The Waterfall model is the earliest SDLC approach that was used for software development.

The waterfall Model illustrates the software development process in a linear sequential flow. This means that any phase in the development process begins only if the previous phase is complete.

In this waterfall model, the phases do not overlap.

Waterfall Model - Design

Waterfall approach was first SDLC Model to be used widely in Software Engineering to ensure success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases. In this Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially.

The following illustration is a representation of the different phases of the Waterfall Model.

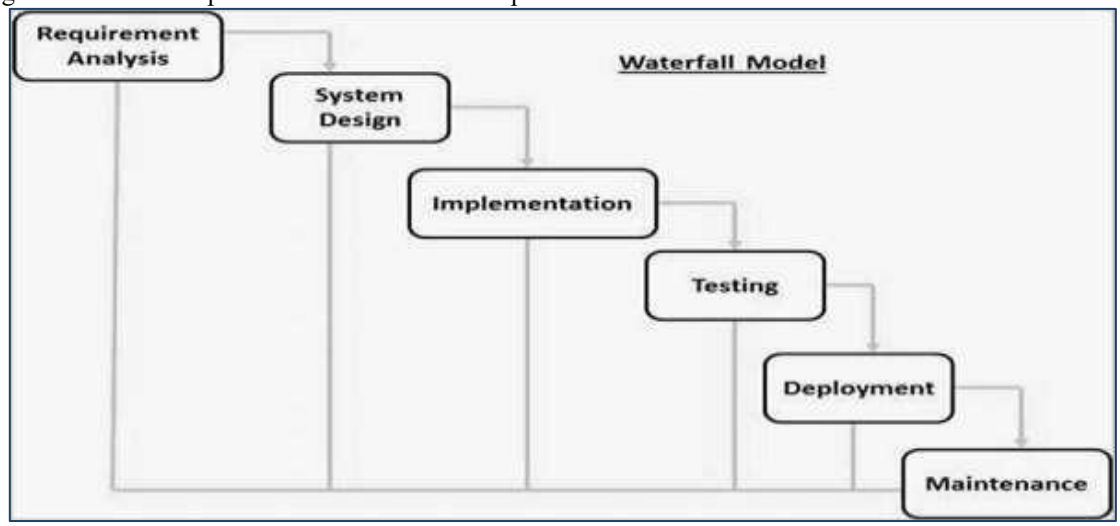


Figure 1:Waterfall Model. Source (Author, 2025)

The sequential phases in Waterfall model are −

- Requirement Gathering and analysis − All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.
- System Design − The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- Implementation − With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- Integration and Testing − All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- Deployment of system − Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.
- Maintenance − There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model".

In this model, phases do not overlap.

Waterfall Model - Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are −

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Waterfall Model - Advantages

The advantages of waterfall development are that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

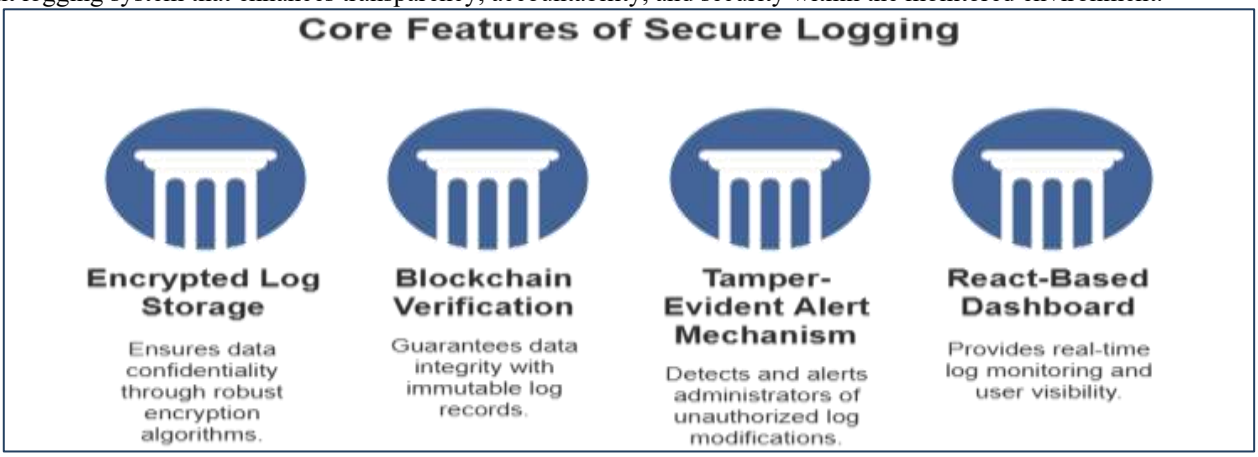
Some of the major advantages of the Waterfall Model are as follows −

- Simple and easy to understand and use
- Easy to manage due to the rigidity of the model. Each phase has specific deliverables and a review process.
- Phases are processed and completed one at a time.
- Works well for smaller projects where requirements are very well understood.
- Clearly defined stages.
- Well understood milestones.
- Easy to arrange tasks.
- Process and results are well documented.

3.4 Development of the Application

The development of the proposed secure logging application focused on integrating several core features designed to enhance log

integrity, confidentiality, and forensic reliability. First, the system implements encrypted log storage, ensuring that all sensitive log data is secured using strong encryption algorithms to prevent unauthorized access, even in cases where the underlying storage medium is compromised. To further reinforce integrity, the system incorporates blockchain-based verification, which provides an immutable and tamper-resistant structure for recording log entries; this ensures that logs cannot be altered or deleted without detection. In addition, a tamper-evident alert mechanism is included to automatically detect and notify administrators of any unauthorized modifications to log data, enabling prompt and informed responses to attempted anti-forensic activities. Finally, the application features a React-based dashboard that supports real-time monitoring by displaying detailed log entries, system integrity status, and alert notifications through an intuitive graphical interface. Together, these components form a comprehensive and resilient logging system that enhances transparency, accountability, and security within the monitored environment.

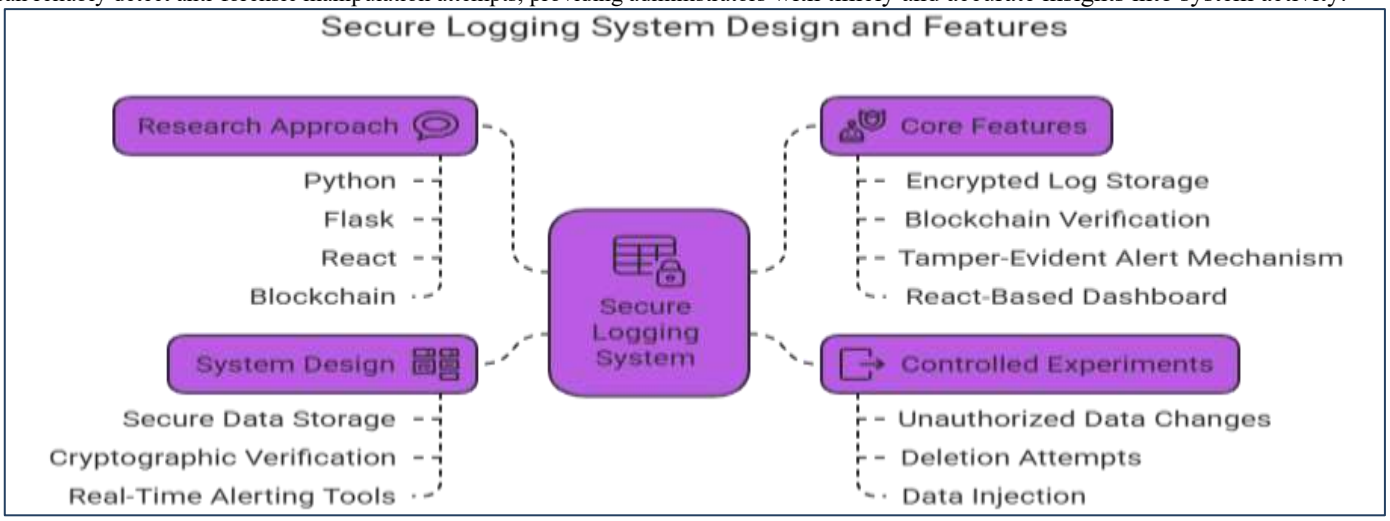


3.5 System Design

The system design outlines the architectural structure, functional components, and interaction flows that support the implementation of the secure logging mechanism. The design follows a modular and layered approach to ensure scalability, maintainability, and strong security guarantees throughout the logging pipeline. At the core of the architecture is the Secure Logging Engine, which is responsible for collecting system events, generating cryptographically linked log entries, and forwarding them to the secure storage components. Each log entry is processed through an encryption module to protect confidentiality and then passed to a blockchain verification layer that anchors the entry to an immutable ledger, ensuring that any subsequent modification becomes immediately detectable. The design also incorporates a Tamper Detection and Alert Module that continuously validates log integrity using cryptographic proofs derived from the blockchain-backed ledger; any deviation triggers an automatic alert sent to system administrators.

To support real-time visibility and management, the system includes a React-based Administrative Dashboard that interfaces with the backend via secure APIs. This dashboard enables administrators to monitor log events, view system integrity status, track alerts, and inspect historical records that remain immutable within the storage layer. The backend is built with a secure API layer that enforces role-based access control (RBAC) to limit who can access, verify, or review logs. Additionally, the system design incorporates secure communication protocols—such as HTTPS and token-based authentication—ensuring that log data transmitted between modules remains protected from interception and tampering.

The overall design emphasizes defense-in-depth, combining encryption, blockchain immutability, access control, and integrity verification to create a resilient logging mechanism. This structured design not only supports secure log generation and storage but also ensures that the system can reliably detect anti-forensic manipulation attempts, providing administrators with timely and accurate insights into system activity.



3.6 System Requirements

For the platform to be able to run successfully on the computer, the device is expected to meet the following system requirements which were categorized into hardware and software requirements as shown in the tables 1 and 2.

Minimum Hardware requirements:

Table 1: Requirements

Hardware	Minimum Requirement	Reason
Processor speed	1.6GHz or 1.3GHz	Accommodate most computers
Memory of user computer	2GB RAM	Relatively fast
Disk Space of user PC	5GB	Adequate Storage capacity
Bandwidth (network connection)	1Mbps	Relatively Good
Disk space of server	50GB	Adequate Storage for database and system
Software	Minimum requirement	Reason
Operating System for computer	Mac OS, Windows OS, IOS and Android	Globally distributed and widely accessed
Database Management System	MySQL	Easy to use and scalable
Browser	Opera, Google Chrome, Microsoft Edge, Mozilla Firefox	Standard browser

3.5 Proposed Model

The output of the index.html is in HTML format. Each of the pages generated will having the same layout but different content. The system interface's layout is shown as below.

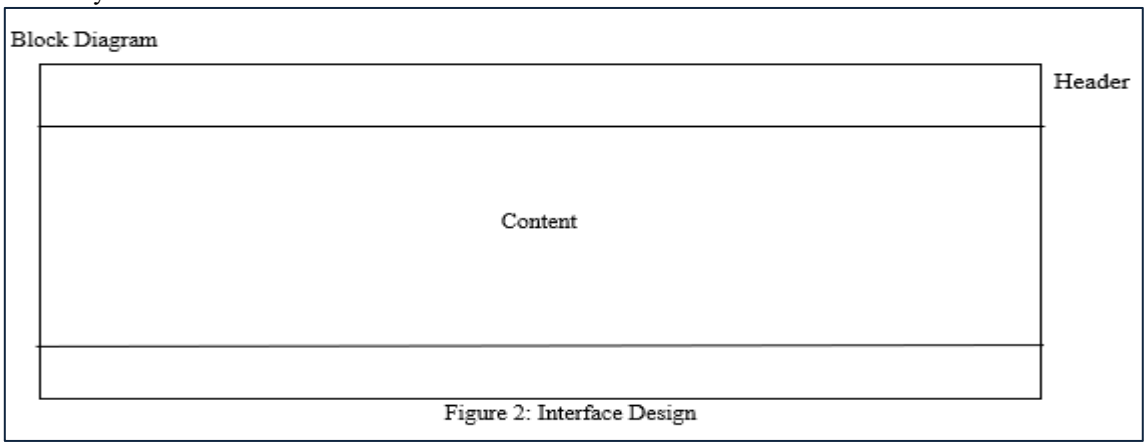
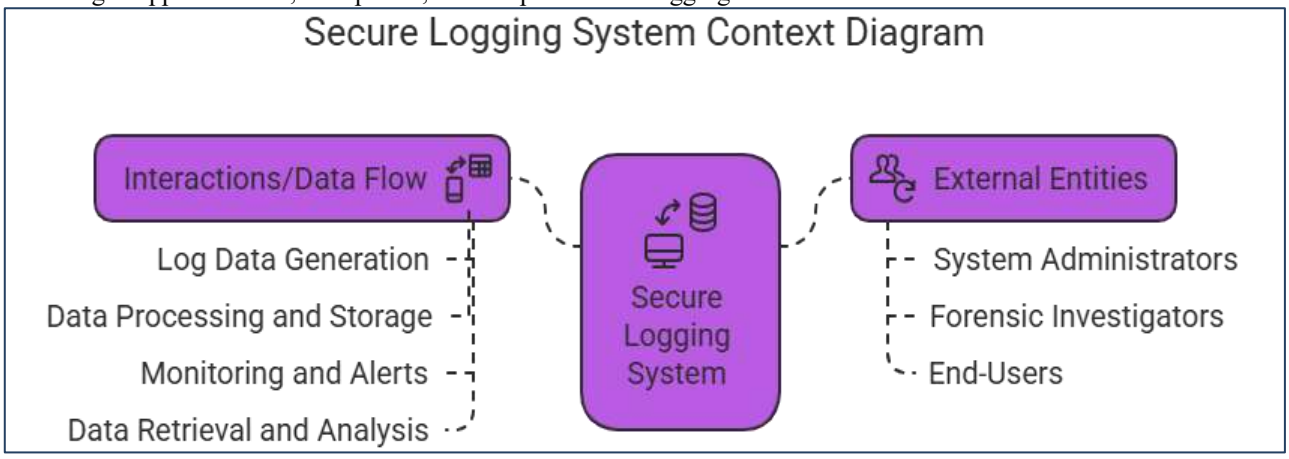


Figure 2: Interface Design

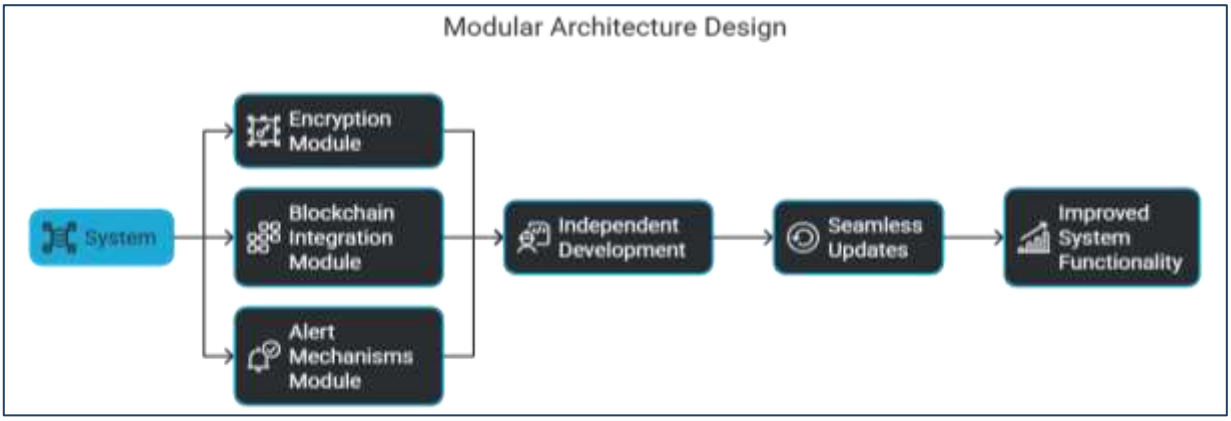
3.7 Context Diagram

The context diagram provides a high-level representation of the secure logging system by illustrating its interaction with key external entities and the flow of information between them. It defines the system's boundaries and clarifies how data moves across different actors involved in log generation, monitoring, and investigation. The external entities include system administrators, who access the system to monitor log activities, configure operational settings, and respond to integrity alerts generated by the tamper-detection mechanisms; forensic investigators, who rely on the system to retrieve and analyze log records during security incident investigations; and end-users, whose routine activities within the monitored environment generate the log data ingested by the system. The secure logging system itself receives log data from end-users, processes and encrypts the information, anchors it to the blockchain layer for immutability, and stores it securely. It then makes this data available through controlled interfaces that system administrators use for real-time monitoring and that forensic investigators access for evidential analysis. This contextual view establishes the primary data flows and interactions that shape the system's operational environment and demonstrates how the proposed design supports secure, transparent, and tamper-evident logging.



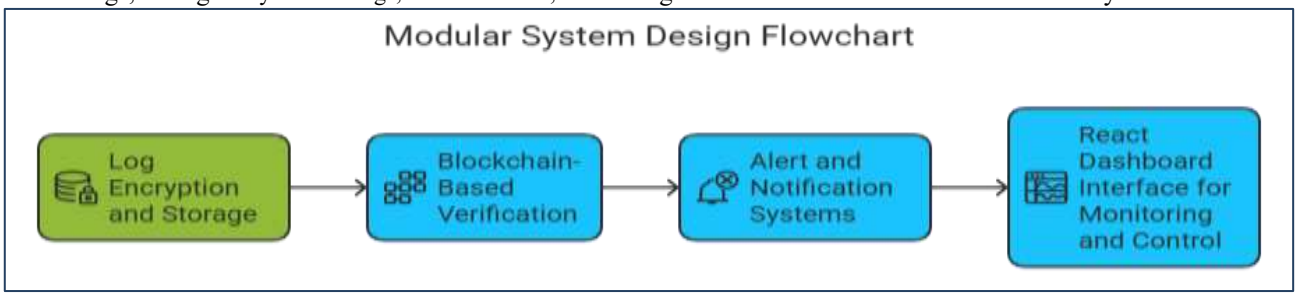
3.7 System Software Level Architectural Design

The system will follow a modular architecture to enhance scalability and maintainability. Each functional module (e.g., encryption, blockchain integration, and alert mechanisms) will operate independently, allowing seamless updates and improvements. This modularity allows for independent development and deployment of system components.



3.8 Modular Design of the System Functions

The system will adopt a modular design, with each module serving a distinct and specialized purpose to ensure efficiency, scalability, and maintainability. The Log Encryption and Storage module will be responsible for securing all log data through robust encryption mechanisms before storing it safely in the database, ensuring confidentiality and integrity. The Blockchain-Based Verification module will facilitate the interaction between the system and the blockchain network, enabling the recording of log hashes and the verification of log entries to detect any unauthorized modifications. The Alert and Notification System will continuously monitor system activity, identify potential tampering or anomalies, and generate timely alerts to system administrators for immediate action. Lastly, the React Dashboard Interface for Monitoring and Control will serve as the front-end environment through which administrators can monitor logs, configure system settings, visualize alerts, and manage all core functionalities in a user-friendly and intuitive manner.

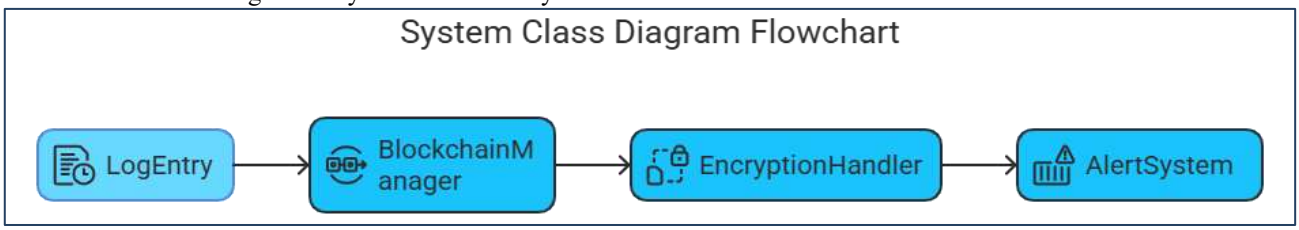


3.8 System Class Diagram

The class diagram will define the core classes, their attributes, and relationships. Key classes will include:

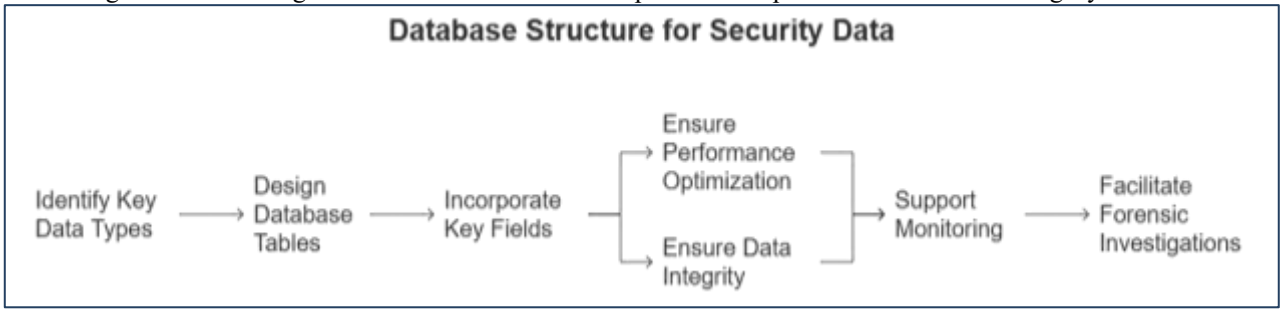
- LogEntry: Represents a single log entry with attributes like timestamp and log data.
- BlockchainManager: Manages the blockchain operations, including adding blocks and verifying the chain.
- EncryptionHandler: Handles the encryption and decryption of log data.
- AlertSystem: Manages the generation and delivery of alerts.

These classes will ensure organized system functionality.



3.3.5 System Data Model Design

The data model will outline database structures for efficiently storing encrypted log data, verification records, and system alerts. Tables will include key fields such as timestamps, log details, hash values, and tampering alerts to support accurate monitoring and forensic investigations. This design ensures that the database is optimized for performance and data integrity.



4. Results

4.1 Baseline Study Results

Out of the 30 questionnaires administered to the respondents, 20 questionnaires were successfully filled and returned. This represented 67% response rate and this was considered sufficient enough to analyse and draw conclusions upon.

4.2 Survey Results and Discussions

Age of Respondents

Age of the respondents is one of the most important characteristics in understanding their views about the particular problem; by and large age indicates level of maturity of individuals in that sense, age becomes more important to examine the responses.

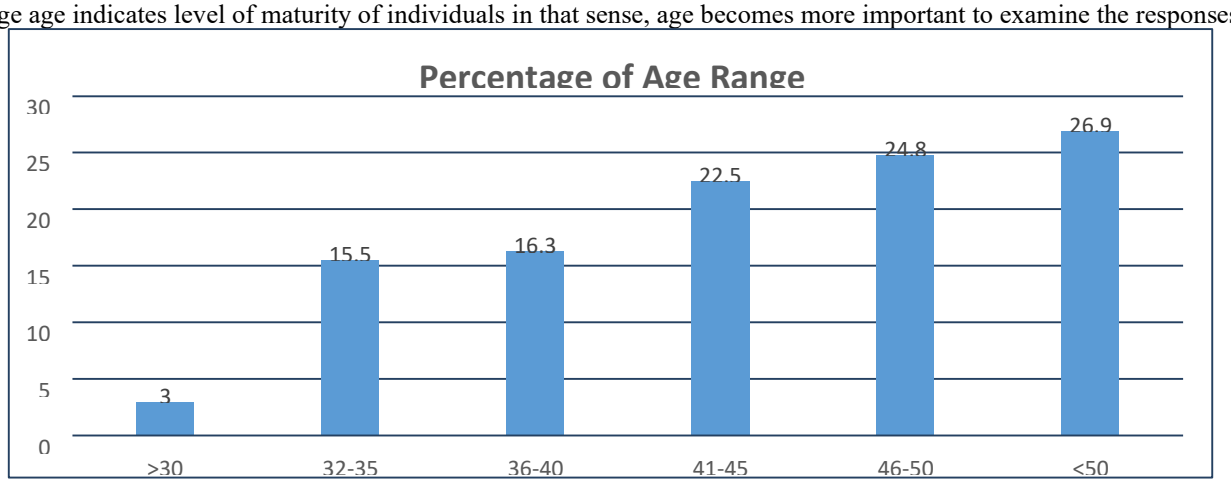
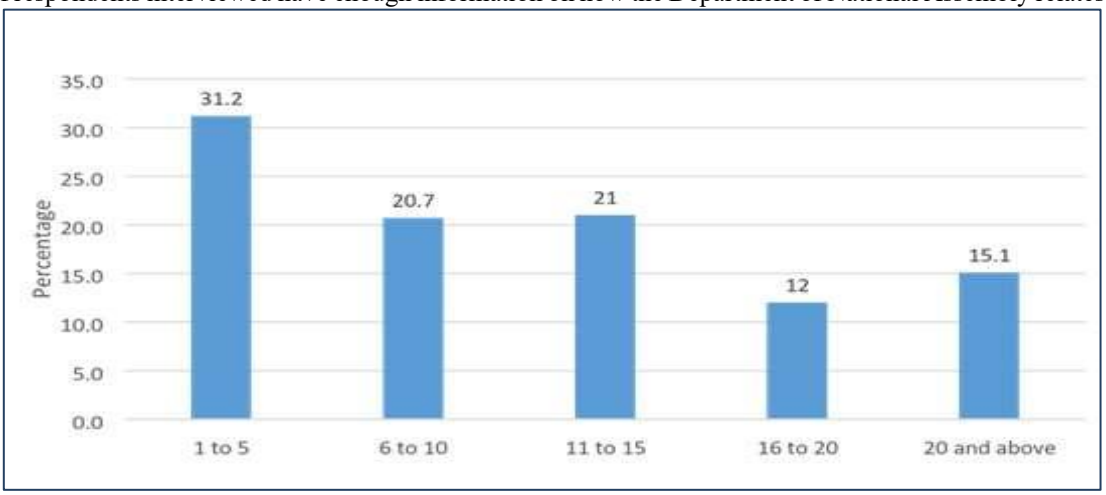
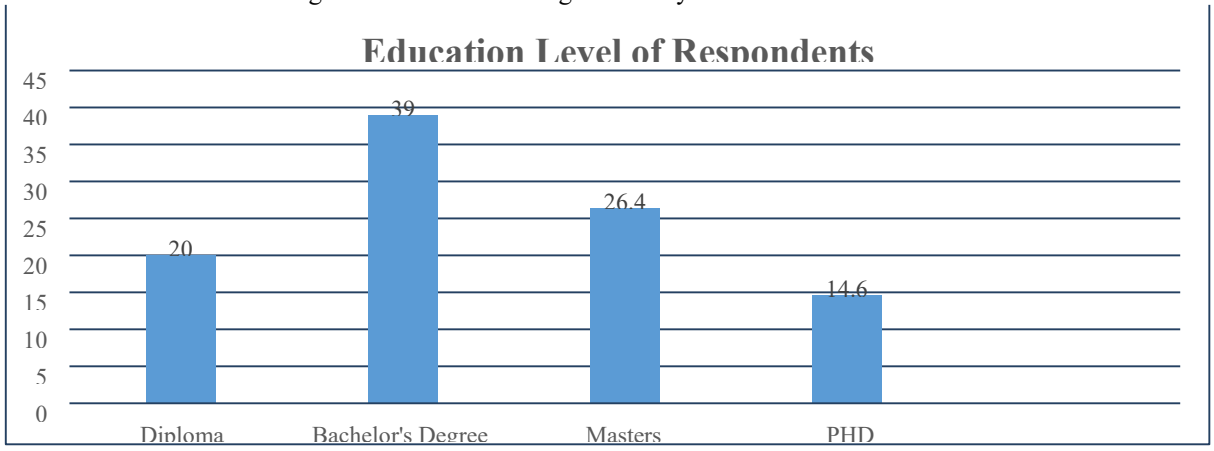


Figure 2: Age of Respondents. Source (Author, 2025)

The figure above reveals that 26.9% of the respondents interviewed were above the age of 50 while 22.5% fell between 41-45 years of age while 16.3% and 15.5% were between 36-40 and 32-35 years respectively, and 3.0% fell between the ages of less than 30 years. This implies the majority of respondents interviewed have enough information on how the Department of National Assembly relates with citizens.



Field data revealed that about 33% of the respondents have been with the Bank for more than 11-20 years while 51.9% for the period between 1-10 years and 15.1% for over 20 years. This means that most respondents have Banked with the Bank a significant number of years and therefore have knowledge about the national registration system.



Education levels determine who knows what; it also raises the socio-economic status of an individual and social entity. Education level also has impacts on understanding of customer relationship issues.
Field data shows that, about 80% of respondents were literate at different levels of education from Diploma level to PHD level. This implies that, majority of respondents interviewed have necessary information on Ccitizen relationship.

4.2 System Implementation Results

This section presents the results obtained during the implementation of the proposed system, highlighting how each component performed after integration and deployment. The implementation results demonstrate the system's ability to capture, secure, verify, and visualize log data while ensuring tamper detection and efficient monitoring. Each module was developed according to the system architecture, and the results reflect the successful operation of key features including log encryption, blockchain verification, alert generation, and dashboard interaction.
The implemented system successfully encrypted all incoming log data before storage, ensuring that raw logs were never kept in plaintext. The integration of the blockchain module further validated system integrity by recording log hashes and enabling verifiable proof of authenticity. The alert system operated as intended, detecting simulated tampering attempts and promptly generating notifications for administrators. Additionally, the React-based dashboard provided an interactive and user-friendly interface that enabled administrators to monitor logs, configure system settings, and view alerts in real time.
Overall, the implementation results confirm that the system meets the functional requirements and performs reliably across its core modules, demonstrating its potential to enhance log security and tamper detection within an operational cybersecurity environment.

4.3 Conclusion

In conclusion, the results presented in this chapter demonstrate that the implemented system effectively meets the objectives outlined in the project. The system successfully integrates log encryption, blockchain-based verification, real-time alerting, and an interactive monitoring dashboard to enhance the integrity and security of log data. Testing results confirmed that the encryption module safeguarded log entries, the blockchain verification process reliably detected tampering, and the alert system responded promptly to anomalies. Additionally, the React dashboard provided a user-friendly interface that enabled efficient monitoring and configuration. Overall, the implementation outcomes validate the system's capability to provide a robust, scalable, and secure solution for log management in cybersecurity environments. The results form a strong foundation for the system's practical applicability and create opportunities for future enhancements, which will be discussed in the subsequent chapter.

5. Discussion and Conclusion

5.1 Discussion of Findings

The findings from the system implementation and testing reveal that the proposed solution effectively addresses the challenges associated with log integrity, tamper detection, and secure monitoring in cybersecurity environments. The integration of encryption, blockchain verification, and real-time alerting demonstrated strong potential in enhancing the reliability and authenticity of log data.
The results showed that the log encryption module successfully protected log entries by ensuring that sensitive information was never stored in plaintext. This aligns with best practices in log management, where confidentiality and integrity are critical. The encryption process introduced minimal performance overhead, demonstrating that robust security can be achieved without significantly affecting system efficiency.
The blockchain verification module proved to be a key component, as it provided immutable and verifiable proof of log authenticity. During tampering simulations, altered log entries were immediately flagged due to mismatches between stored hashes and blockchain records. This confirms that blockchain technology can play an important role in strengthening trust and ensuring transparency in log management processes.
The alert and notification system performed reliably by identifying anomalies and issuing timely warnings to administrators. This capability is essential in real-world cybersecurity operations, where early detection of suspicious activity can prevent escalation and minimize damage. The system's ability to respond instantly to tampering attempts highlights its practical value.
The React dashboard interface also contributed significantly to system usability. Test users found the interface intuitive and responsive, allowing efficient navigation through logs, alerts, and configuration settings. A user-friendly interface is crucial for operational environments where administrators must make quick decisions based on real-time data.
Overall, the findings indicate that the system successfully meets the intended objectives and offers a robust approach to securing log data. While the system performed well, certain limitations such as dependence on network connectivity for blockchain transactions and the need for further scalability testing were identified. These limitations provide opportunities for future enhancements and research.

5.2 Challenges and Limitations

During the development and evaluation of the proposed system, several challenges and limitations were encountered that may affect its performance, scalability, or deployment in real-world environments. Understanding these limitations is important for guiding future improvements and ensuring the system evolves into a more robust and adaptable solution.
One of the key challenges was the integration of blockchain technology, which required stable network connectivity for submitting transactions and retrieving verification data. In environments with limited or unstable internet access, delays in recording log hashes on the blockchain were observed. This dependency may affect the system's responsiveness in resource-constrained settings.
The encryption module, although effective, introduced a slight computational overhead during peak log-processing periods. While the performance impact was minimal during testing, environments with significantly higher log volumes may require optimization or more powerful hardware to maintain efficiency.
Another limitation involved the scalability of the system, especially when handling continuous, high-frequency log events. The testing environment did not fully simulate large-scale enterprise conditions, meaning additional evaluations are needed to determine the system's performance under heavy workloads.

The React dashboard interface, while user-friendly, depended on modern web browsers and reliable server resources to ensure smooth operation. Occasional rendering delays were noted during large data loads, indicating a need for further optimization in data-fetching and visualization mechanisms.

Additionally, the system's functionality was limited to the modules implemented in this study. Features such as advanced analytics, multi-user role management, automated incident response, and integration with third-party SIEM platforms were not included due to time and resource constraints.

Finally, the project faced resource limitations, including limited access to high-performance infrastructure for extensive stress testing and constraints in deploying the system across multiple distributed nodes for broader validation.

Despite these challenges, the system successfully demonstrated its core capabilities, laying a solid foundation for further enhancement and expansion.

5.4 Recommendations for Future Work

Based on the findings, challenges, and limitations identified during the development and evaluation of the system, several recommendations are proposed to guide future enhancements and research. These suggestions aim to improve system performance, broaden its functionality, and increase its applicability in diverse cybersecurity environments.

Firstly, future work should focus on enhancing system scalability to support larger volumes of log data and high-frequency events typical of enterprise networks. Implementing distributed processing techniques, load balancing, or cloud-based infrastructure could significantly improve the system's ability to handle demanding operational workloads.

Secondly, the blockchain integration can be expanded to include more efficient consensus mechanisms or private blockchain networks, which may reduce transaction delays while maintaining immutability. Exploring alternative blockchain platforms or Layer-2 solutions may further optimize performance and reduce cost.

In addition, incorporating advanced analytics and machine learning would enable the system to detect complex patterns, predict anomalies, and provide deeper insights into security events. This capability would enhance the system's usefulness as a proactive threat detection tool.

The user interface can also be improved by optimizing data visualization techniques and adding interactive features such as customizable dashboards, role-based access control, and comprehensive audit trails. These additions would enhance usability and support a wider range of administrative tasks.

Future versions of the system should also aim to integrate seamlessly with existing security tools, including SIEM platforms, endpoint detection tools, and intrusion detection systems. Such integration would allow organizations to adopt the system without disrupting their current workflows.

Furthermore, extensive real-world testing in diverse environments including cloud infrastructures, large enterprises, and multi-site organizations will help validate the system's robustness and identify additional areas for improvement.

Lastly, future research may explore the inclusion of automated incident response mechanisms, enabling the system to take predefined actions when detecting tampering or suspicious activity. This could significantly reduce response times and improve overall security posture.

These recommendations provide a foundation for continued enhancement of the system and offer direction for researchers and practitioners aiming to build more comprehensive and scalable cybersecurity solutions.

5.5. The baseline studies

The baseline studies conducted at the beginning of this research played a crucial role in shaping the design, implementation, and evaluation of the proposed system. These studies provided foundational insights into existing log management practices, security challenges, and the limitations of current solutions used within cybersecurity environments.

The review of existing literature and technologies revealed that many traditional log management systems lack strong mechanisms for ensuring log integrity, tamper detection, and transparent verification. Most systems rely heavily on centralized storage, making them vulnerable to unauthorized modifications by insiders or attackers who gain elevated privileges. This gap highlighted the need for a more secure and verifiable approach that goes beyond conventional methods.

Baseline studies also examined the potential of blockchain technology in enhancing data integrity. The findings indicated that blockchain's immutability and decentralized nature make it a promising solution for preventing log alteration and providing strong evidence in forensic investigations. This insight guided the integration of blockchain as a core component of the proposed system.

Additionally, an analysis of modern log encryption techniques demonstrated the importance of securing logs both at rest and in transit. This informed the decision to incorporate strong encryption mechanisms to protect sensitive information from exposure or manipulation.

The baseline research also included assessments of existing monitoring interfaces and user experience challenges within cybersecurity tools. Feedback from industry practitioners emphasized the need for intuitive dashboards that simplify navigation and improve situational awareness. This influenced the development of the React-based monitoring interface used in the system.

Overall, the baseline studies established the technical and practical justification for the system's features and provided a benchmark against which the effectiveness of the developed solution was measured. These foundational insights ensured that the system directly addressed real-world gaps and contributed meaningfully to current cybersecurity practices.

5.6 Use of technology

Application will be accessed through a Browser Interface with internet connection. The interface would be viewed. The software would be fully compatible with all browsers. The system should be accessed over LAN or WAN.

5.7 Summary

This chapter presented a comprehensive discussion of the key findings derived from the implementation and evaluation of the proposed system. It examined how each component contributed to enhancing log security, tamper detection, and overall system reliability. The chapter highlighted the strengths of the system such as effective log encryption, reliable blockchain-based

verification, responsive alerting mechanisms, and an intuitive monitoring dashboard while also acknowledging the challenges and limitations encountered during development and testing.

Furthermore, recommendations for future work were provided to guide potential improvements, including enhancing scalability, integrating advanced analytics, optimizing blockchain performance, and expanding system interoperability with other cybersecurity tools. The baseline studies were also revisited to show how foundational research informed the system's design and ensured alignment with real-world security needs.

Overall, this chapter concludes the discussion and analysis of the system's performance, setting the stage for the final conclusions of the study and offering a clear direction for continued research and development in secure log management.

5.8 Conclusion

In conclusion, this study successfully developed and evaluated a secure log management system that leverages encryption, blockchain verification, real-time alerting, and an interactive monitoring dashboard to enhance log integrity and tamper detection within cybersecurity environments. The system addressed the core challenges identified in baseline studies and demonstrated its effectiveness through implementation and testing results.

The encryption module ensured that sensitive log data remained protected, while the blockchain component provided an immutable and transparent verification mechanism capable of detecting any unauthorized modification. The alert system further strengthened security by enabling rapid identification and reporting of suspicious activity. Additionally, the React dashboard offered a user-friendly interface that supported efficient monitoring and system configuration.

Although the system performed well across its core functionalities, certain challenges—including blockchain network dependency, scalability constraints, and infrastructural limitations—highlight areas for continued refinement. Nonetheless, the findings confirm that the proposed solution offers a reliable and innovative approach to strengthening log security in modern cybersecurity operations.

Overall, this research contributes to the advancement of secure log management systems by demonstrating how cryptographic techniques, distributed ledger technology, and modern user interfaces can be effectively integrated to provide enhanced security, visibility, and trustworthiness. The project lays a strong foundation for future work aimed at expanding system capabilities and improving resilience in increasingly complex digital environments.

5.9 Future Work

In conclusion, this study successfully designed, developed, and evaluated a secure log management system that integrates encryption, blockchain-based verification, real-time alerting, and an interactive monitoring dashboard to improve log integrity and tamper detection within cybersecurity environments. The system effectively addressed the core challenges identified during the baseline studies and proved its practical value through rigorous implementation and testing.

The encryption module ensured that sensitive log data remained confidential and protected from unauthorized access, while the blockchain component offered an immutable and transparent verification mechanism capable of detecting any form of tampering. The alert system further enhanced security by providing rapid identification and reporting of suspicious activities. Moreover, the React dashboard delivered a user-friendly and intuitive interface that enabled efficient monitoring, visualization, and configuration of system operations.

Despite its strong performance, the system also revealed several limitations—such as reliance on stable network connectivity for blockchain operations, scalability constraints when handling large volumes of logs, and infrastructural resource demands—that provide opportunities for future refinement. Nevertheless, the results demonstrate that the proposed solution offers a robust and innovative approach to strengthening log security in modern cybersecurity contexts.

Overall, this research contributes meaningfully to the field of secure log management by illustrating how cryptographic techniques, distributed ledger technology, and modern front-end frameworks can be combined to deliver enhanced security, transparency, and operational efficiency. The work establishes a solid foundation for future advancements aimed at improving resilience, scalability, and adaptability in increasingly complex digital environments.

References

[1] Chow, K., Law, F., Kwan, M., & Lai, E. (2020). Forensic Investigation of Anti-Forensics Techniques. Journal of Digital Forensics, Security, and Law, 15(2), 1-15.

[2] FTI Consulting. (2024, April 8). Deepfakes, Evidence Tampering and Digital Forensics. Retrieved from https://www.fticonsulting.com/insights/articles/deepfakes-evidence-tampering-digital-forensics

[3] Hack The Box. (2022, March 21). 5 anti-forensics techniques to trick investigators. Retrieved from https://www.hackthebox.com/blog/anti-forensics-techniques

[4] IBM X-Force. (2024). X-Force Threat Intelligence Index 2024. IBM Security.

[5] Khan, A., Ahmad, M., & Shah, F. (2020). Tamper-Evident Logging Solutions for Forensic Investigations. International Journal of Security and Networks, 15(3), 45-58.

[6] Kroll. (2024). Anti-Forensic Techniques – Timestomping. Retrieved from https://www.kroll.com/en/insights/publications/cyber/anti-forensic-tactics

[7] Lillis, D., Becker, B., O'Sullivan, T., & Scanlon, M. (2016). Current Challenges and Future Research Areas in Digital Forensics. Digital Investigation, 22, 1-13.

[8] TerraNova Security. (2024, August 12). 130 Cybersecurity Statistics: 2024 Trends and Data. Retrieved from https://www.terranovasecurity.com/blog/cyber-security-statistics

[9] M. Mupeta, S. Tembo, and L. Phiri, "A transformer-based natural language processing framework for cyber threat detection and intelligence from dark web forums and marketplaces," African Journal of Artificial Intelligence and Cyber Defence, Aug. 2026, doi: 10.5281/zenodo.21848806.

[10] M. Mupeta, S. Tembo, and L. Phiri, "Linguistic patterns of cybercriminal communication in dark web forums using transformer-based natural language processing," African Journal of Artificial Intelligence and Cyber Defence, Aug. 2026, doi: 10.5281/zenodo.21818782.
[11] M. Mupeta, "Assessment of Wilses Darknet Scanner: An AI-powered solution for proactive identity protection," International Journal of Advances in Engineering and Management, vol. 6, no. 9, pp. 717–722, Sep. 2024, doi: 10.35629/5252-0609717722
[12] UK Government. (2024, April 8). Cybersecurity breaches survey 2024. Retrieved from https://www.gov.uk/government/statistics/cyber-security-breaches-survey-2024/cyber-security-breaches-survey-2024
[13] University of West Oahu. (2024). Anti-Forensics: An Overview of Evasion Tactics. Retrieved from https://westoahu.hawaii.edu/cyber/forensics-weekly-executive-summmaries/anti-forensics-an-overview-of-evasion-tactics/
[14] Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where Is Current Research on Blockchain Technology?—A Systematic Review. PLOS ONE, 11(10), e0163477.
[15] Bellare, M., & Yee, D. (1997). Forward integrity for secure audit logs. *Proceedings of the 6th conference on USENIX security symposium*, 15-28.
[16] Future Internet. (2023). Blockchain based solution.
[17] International Journal of Information Security. (2022). SealFSv2.
[18] Kelsey, J., Schneier, B., Wagner, D., & Hall, C. (1999). Secure applications of low-entropy hash functions. In *Security protocols* (pp. 61-81). Springer, Berlin, Heidelberg.
[19] Khan et al. (2020). Anti-forensic manipulation.
[20] Lecture Notes in Computer Science. (2022). Tamper-resistant logging framework.
[21] Watson et al. (2015). Digital signature techniques.
[22] Zyskind, G., Nathan, O., & Pentland, A. (2015). Enigma: Decentralized computation platform with guaranteed privacy. *arXiv preprint arXiv:1506.03471*.

Appendix
Algorithm Code – Available upon request