DOI: https://dx.doi.org/10.4314/gjpas.v31i4.6

GLOBAL JOURNAL OF PURE AND APPLIED SCIENCES VOL. 31, 2025: 683-702
COPYRIGHT© BACHUDO SCIENCE CO. LTD PRINTED IN NIGERIA ISSN 1118 – 0579, e-ISSN: 2992 - 4464
www.globaljournalseries.com.ng, Email: globaljournalseries@gmail.com

CLOUD SECURITY USING CRYPTO-SHREDDING FOR SECURE DATA DELETION: A REDUCTION IN CYBER SECURITY RISKS

LOVETH A. EBONG AND GERTRUDE FISCHER
Email: lovethebong33@gmail.com

(Received 5 February 2025; Revision Accepted 20 March 2025)

ABSTRACT

Cloud computing has revolutionized data management, offering scalability and efficiency. However, it also presents significant security challenges, particularly in data deletion. Traditional deletion methods often leave residual data that can be recovered, increasing the risk of unauthorized access. Ensuring complete and irreversible data disposal is critical for maintaining data security and regulatory compliance. This research investigates crypto-shredding, a technique that enhances data security by destroying encryption keys, rendering the associated data permanently inaccessible. The study focuses on the design, implementation, and evaluation of crypto-shredding techniques within cloud environments, comparing their effectiveness to conventional deletion methods. A structured framework for integrating crypto-shredding into cloud architectures is proposed, ensuring compliance with data protection regulations such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA). The study also explores the role of user authentication mechanisms and content discovery techniques in verifying that deleted files leave no recoverable traces in active storage or archives. Extensive testing and performance analysis will assess the feasibility and reliability of crypto-shredding as a secure data disposal method. The findings of this research aim to strengthen cloud security frameworks, mitigate the risks of data breaches, and contribute to the advancement of secure data management practices in cloud computing.

KEYWORDS: Secure data deletion, Data security, Data disposal, Cloud storage

INTRODUCTION

Cloud computing has significantly transformed how organizations manage and store data (Yang et al., 2017; Sunyaev & Sunyaev, 2020; Obi et al., 2024; Segun-Falade et al., 2024). By leveraging remote servers hosted over the internet, it offers scalability, flexibility, and cost efficiency (Mollah et al., 2012; Attaran, 2017; Alam et al., 2023; Borra, 2024; Kumar et al., 2024). However, this shift also introduces critical challenges, particularly in data security and privacy (Attaran, 2017; Soni & Kumar, 2022; El Kafhali et al., 2022). Organizations entrust sensitive information to third-party service providers, which increases the risk of breaches and unauthorized access. Traditionally, IT infrastructure allowed organizations to maintain direct control over their servers, ensuring greater oversight of data security.

However, cloud computing requires relinquishing that control to external providers, raising concerns about data integrity and confidentiality (Angel et al., 2023; Fernandes et al., 2020; Fraser & Portillo-Dominguez, 2022). One of the most pressing security challenges in cloud environments is data deletion. Conventional deletion methods often leave residual data that can be recovered, posing a significant risk. The inability to physically control storage media further complicates secure data disposal, making it crucial for organizations to adopt effective strategies to ensure that deleted data is permanently and irreversibly removed. These concerns highlight the urgent need for advanced data protection mechanisms to safeguard sensitive information in cloud-based systems.

Loveth A. Ebong, Department of Computer Engineering, Faculty of Engineering and Technology, University of Calabar, Nigeria
Gertrude Fischer, Department of Electrical Electronics Engineering, Faculty of Engineering, University of Cross River State, Nigeria

With the proliferation of cloud computing, the need for robust techniques to ensure the security and integrity of data in cloud environments has become critical (Abdulsalam & Hedabou, 2022, Hazela et al., 2022; Yanamala, 2024). Cloud computing offers scalable resources and flexible access to information, but these benefits come with challenges related to data security and privacy. One of the significant concerns is the secure deletion of data (Shahid et al., 2021; Gupta et al., 2022; Parast et al., 2022; Reddy & Ravindranath, 2024). Conventional deletion methods often leave remnants that can be recovered, posing risks of unauthorized access to sensitive information (Zhang et al., 2018; Mei, 2022). To address this concern, crypto-shredding has emerged as a promising technique. Crypto-shredding involves the destruction of encryption keys used to encrypt data, thereby rendering the data inaccessible and irrecoverable (Godin & Lamontagne, 2021; Zainuddin, 2023). This approach is particularly effective in cloud environments where physical access to storage media is limited or impractical (Zhu & Wen, 2022; Ali et al., 2024; Kostadinovska, 2016). This study explores the integration of crypto-shredding techniques into cloud computing to enhance secure deletion of data. It intends to develop and implement a robust solution for secure data disposal in cloud computing environments.

The aim of this study is to use crypto-shredding to improve secure deletion of data, ensuring that data deleted from cloud systems is irrecoverable, thereby enhancing overall data security in the cloud. Hence, this research seeks to enhance data security and to ensure compliance with regulatory standards.

CONCEPTUAL FRAMEWORK

The conceptual framework for this study is designed to address the critical security concerns associated with cloud computing, particularly in the context of data protection and secure deletion. This framework combines the principles of cloud security, the limitations of traditional data deletion methods, and the introduction of advanced techniques such as crypto-shredding to ensure the effective and irreversible deletion of sensitive data in cloud environments.

Cloud security

Cloud computing has emerged as a popular solution for organizations due to its flexibility and scalability, allowing businesses to manage IT resources more efficiently. However, this shift to cloud-based infrastructure has also introduced significant security concerns, particularly with respect to data protection and privacy. Cloud computing is a model that enables ubiquitous, on-demand access to a shared pool of configurable computing resources, which can be rapidly provisioned with minimal management effort (Mell & Grance, 2011).

Despite these advantages, cloud environments expose data to numerous risks, such as data breaches, data loss, and improper disposal, exacerbated by the multi-tenant architecture of the cloud, remote data storage, and dynamic resource allocation (Yaish, (2014).

Data breaches are a major concern, as cloud data is often distributed across different environments, making it an attractive target for attackers (Darwish et al., 2018). Pearson, (2012) highlights that cloud environments may expose sensitive information to both third-party cloud providers and internal malicious actors. Data loss is another critical issue, which can occur due to accidental deletion or system failures, making secure deletion techniques essential to prevent unauthorized recovery (Grobauer et al., 2011). Denial of Service (DoS) attacks further compound security challenges by exploiting cloud services to overwhelm servers, leading to service outages (Ristenpart et al., 2009). In addition, insider threats are increasingly significant, as cloud infrastructures managed by third-party providers may be vulnerable to abuse by individuals with privileged access (Cloud Security Alliance, 2013). These security risks emphasize the importance of developing robust mechanisms to protect data in cloud environments.

Secure deletion techniques

Secure deletion is crucial for protecting sensitive data. Historically, several methods have been used to securely delete data, including overwriting, degaussing, and physically destroying storage media. Although these methods have proven effective in traditional computing environments, they are often inefficient or impractical for cloud computing, where data may be replicated across multiple servers in different locations (McKee et al., 2017). Overwriting files, for instance, requires direct control over the hardware, which users of cloud services do not typically have, making it difficult to ensure complete data deletion (Reardon et al., 2017). Techniques such as degaussing and physical destruction, though useful for physical media, are not applicable to cloud environments where users do not have access to the underlying hardware (Bowers et al., 2009). These traditional techniques are fraught with limitations in cloud environments where data can be distributed across multiple locations (Reddy et al., 2024; Herman et al., 2020; Mohiddin et al., 2020). Hence, the need for more advanced secure detection technique that can address the specific challenges of cloud storage.

Crypto-Shredding for secure data deletion

Crypto-shredding provides a more practical solution for cloud systems, as it allows users to securely destroy data by deleting the encryption keys, even if the data itself remains stored on distributed servers (Huang et al., 2023).

This technique involves permanently and securely deleting encryption keys, ensuring that the encrypted data becomes irretrievable, even if the physical data remains stored (Gutmann, 1996). Crypto-shredding is a technique where cryptographic methods are employed to render data unrecoverable (Nived et al., 2020). It involves encrypting data and then destroying the encryption keys, making the data indecipherable (Shukla et al., 2020). Ahmad et al. (2023) provided a report on the integration of crypto-shredding into secure deletion practices and its benefits in enhancing data security. Additionally, the effectiveness of crypto-shredding in various scenarios can address the limitations of traditional deletion methods (Mei et al., 2022).

In the context of cloud computing, crypto-shredding is particularly well-suited for ensuring the secure deletion of data. As proposed by (Chow 2009), cloud data can be encrypted, and the encryption keys securely stored in a key management system (KMS). When the data is no longer needed, the encryption keys can be deleted, rendering the data irrecoverable, even if it remains physically stored across multiple virtual machines. This model is highly effective in distributed cloud environments, where traditional data deletion methods would be challenging to implement. Key management is crucial in this context, as encryption keys must be stored and destroyed securely to prevent unauthorized access (Mollah et al., 2012). Many cloud providers use hardware security modules (HSMs) or advanced cryptographic techniques to manage keys securely, ensuring that data remains protected throughout its lifecycle.

Recent studies have explored integrating crypto-shredding into cloud frameworks to improve data security (Fernandes et al., 2020). Dhake et al. (2022) emphasized the significance of incorporating crypto-shredding to enhance the forensic analysis of cloud environments, especially in secure deletion. Crypto-shredding is an effectiveness tool in that, without the encryption keys, the data cannot be decrypted or accessed, making it an ideal method for ensuring secure data destruction (Dinadayalan et al., 2014).

Benefits and limitations of crypto-shredding

Crypto-shredding offers several key benefits in cloud environments. It is highly efficient, as it only requires the deletion of the encryption key rather than the data itself, making it scalable for large cloud systems (Perlman & Kaufman, 2008). Additionally, once encryption keys are destroyed, the data becomes irrecoverable, providing a high degree of security (Schneier, 2009). This method also aids in compliance with data privacy regulations, such as the General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA), which mandate the secure disposal of sensitive data (Haeberlen, 2010). However, crypto-shredding is not without limitations. The effectiveness of the technique is heavily dependent on the strength of the encryption used, meaning that if the encryption algorithm becomes outdated or vulnerable, the data could potentially be recovered through brute-force attacks (Aly & ElGayyar, 2013). Moreover, if key management is not implemented correctly, encryption keys could be improperly deleted or recovered, undermining the overall security of the system (Grubbs et al., 2017) . Furthermore, despite the deletion of encryption keys, data remnants may persist in backup systems or archives that do not follow the same key management procedures, making it necessary to implement additional measures for complete data destruction.

MATERIALS AND METHODS

Materials

An implementation tool is an assembly of instruments, materials, devices and in most case humans that, when used collectively or separately, can be handy for implementing a new program, practice, project, or initiative. Users can apply the toolkit in its entirety, or they may find certain portions of it particularly informative for their needs. For this research, the system was implemented on windows 7 platform and the programing language used is java jdk 8.2. For the database, MySql was used.

Methods

The following are the steps applied in this study; crypto-shredding technique selection, research design, implementation, and testing and evaluation.

Crypto-shredding technique selection

The selection of an appropriate crypto-shredding technique was a critical step in ensuring the secure deletion of data in cloud environments. This phase involved assessing various cryptographic algorithms to determine the most effective approach for rendering deleted data permanently inaccessible. The evaluation focused on encryption strength, performance efficiency, ease of integration with cloud storage systems, and the reliability of the key destruction mechanism. Encryption strength was a major consideration, as it ensured that data remained protected before the cryptographic key was destroyed. The performance efficiency of each algorithm was also assessed to determine its computational overhead and impact on system performance, particularly in terms of processing time and resource consumption. Additionally, the feasibility of integrating each technique into existing cloud architectures was examined, taking into account compatibility with common cloud storage providers and encryption frameworks.

A major focus of this selection process was the effectiveness of the key destruction mechanism. The crypto-shredding approach relies on the secure elimination of encryption keys, making it impossible to decrypt stored data once the keys have been destroyed. The reliability of this process was analyzed to confirm that no residual traces of the key could be recovered through brute-force mechanism.

Based on these criteria, the most suitable crypto-shredding algorithm was identified and selected for implementation. This choice was made to ensure a balance between security, efficiency, and practical applicability within cloud computing environments.

Research design

The system was designed to include a secure cloud storage environment where sensitive data would be encrypted before being stored. A robust encryption and key management system was implemented to oversee the lifecycle of cryptographic keys, from generation and storage to secure destruction. The design also incorporated a dedicated crypto-shredding module responsible for executing the deletion process through key destruction, ensuring that once a key was removed, the associated data became irretrievable. To prevent unauthorized access and ensure data integrity, access control mechanisms were embedded within the system. These measures restricted key management operations to authorized entities, reducing the risk of accidental or malicious key exposure. The research design also emphasized seamless integration with existing cloud platforms, ensuring that the proposed framework could be adapted to widely used cloud storage services without significant architectural modifications. This structured approach ensured that the crypto-shredding framework could be effectively implemented and evaluated in a real-world cloud computing environment, providing a practical and secure solution for data disposal.

Implementation

The implementation phase involved developing a functional prototype of the proposed crypto-shredding framework and integrating it into a cloud computing environment. This process began with the configuration of cloud storage, encryption services, and key management systems to establish a secure data handling infrastructure. Cloud platforms such as Amazon Web Services (AWS), Microsoft Azure, and Google Cloud were utilized to simulate real-world conditions and evaluate system performance.

The selected crypto-shredding algorithm was integrated into the system, ensuring that encrypted data could only be accessed with a valid cryptographic key. The key destruction process was carefully implemented to ensure that once a key was deleted, the encrypted data became permanently inaccessible. To facilitate secure interactions between different system components, application programming interfaces (APIs) were developed to manage encryption, decryption, and key lifecycle operations. Automation played a crucial role in enhancing the efficiency and reliability of the framework. Key lifecycle management was programmed to enforce secure key generation, periodic key rotation, and irreversible key destruction when data deletion was required. Additional security measures, such as access control policies and authentication protocols, were also incorporated to prevent unauthorized manipulation of encryption keys. By the end of the implementation phase, a fully functional prototype was developed, capable of integrating with existing cloud services while providing a secure and efficient method for crypto-shredding. This prototype formed the foundation for the subsequent testing and evaluation phase, where its performance and security capabilities were assessed.

Testing and evaluation

The testing and evaluation phase was designed to assess the effectiveness of the implemented crypto-shredding framework in ensuring secure data deletion while maintaining optimal cloud system performance. This phase focused on two key aspects: performance testing and security testing. Performance testing was conducted to evaluate the efficiency of the system in handling encryption, decryption, and key destruction operations. The impact of crypto-shredding on system latency, processing time, and cloud resource utilization was measured under varying data loads. The objective was to determine whether the framework introduced any significant performance overhead that could affect real-world usability. Security testing aimed to verify the irreversibility of data once the encryption key was destroyed. Forensic recovery techniques were applied to assess whether deleted data could be retrieved by any means. Simulated attack scenarios were also executed to evaluate the resilience of the system against unauthorized access and potential vulnerabilities. Furthermore, the system was tested for compliance with industry security standards and best practices to ensure that it met the required benchmarks for cloud data protection. The results of these tests provided a comprehensive analysis of the reliability and effectiveness of the system. It demonstrates whether the crypto-shredding technique successfully prevented data recovery while maintaining an acceptable level of performance.

Existing systems

Currently, there are several existing systems. Some are Amazon Elastic Compute Cloud (EC2) and Amazon Simple Storage Service (S3) (Guide, 2008; Amazon, 2009). Most of these file sharing application provides means of transferring data from one end to the other. Most times big and real time organizations do not engage in using most of these publicly used systems in other to secure their data from intruders.

Proposed system

The study propose a robust and adaptable distributed storage system designed to ensure the accuracy and security of client data in the cloud. This system integrates a crypto-shredding technique to enable the complete and irreversible destruction of files or data, preventing any possibility of recovery. To enhance data reliability, erasure coding was incorporated in file distribution, which provides redundancy while significantly reducing storage and communication overhead compared to traditional replication-based methods.

Additionally, my system supports secure and efficient dynamic operations on data blocks, including updates, deletions, and append operations, without compromising performance or security. This flexibility ensures that cloud storage remains both reliable and adaptable to evolving user needs. Comprehensive security and performance analysis demonstrate that my proposed system is highly efficient and resilient against Byzantine failures, malicious data modification attacks, and even collusion-based server attacks. Furthermore, the effective implementation of the crypto-shredding technique guarantees the absolute and irreversible deletion of data, reinforcing the overall security and trustworthiness of cloud storage environments. The data flow diagram (DFD) which provides a structured visualization of how data flows through various processes, storage points, and delays within the system (Fig. 1).

Authentication module

The authentication module is responsible for verifying client identities and controlling access within the application network (Fig. 2). Upon receiving client authentication information, the module interacts with a masquerading router, which serves as a ticketing authority. This process ensures that only authenticated users are granted access to the system. Additionally, the module supports the dynamic updating of client lists, optimizing authentication time. In cases where a client's access needs to be revoked, this module can remove them from the list of valid clients based on specific requests, ensuring secure and timely client management.

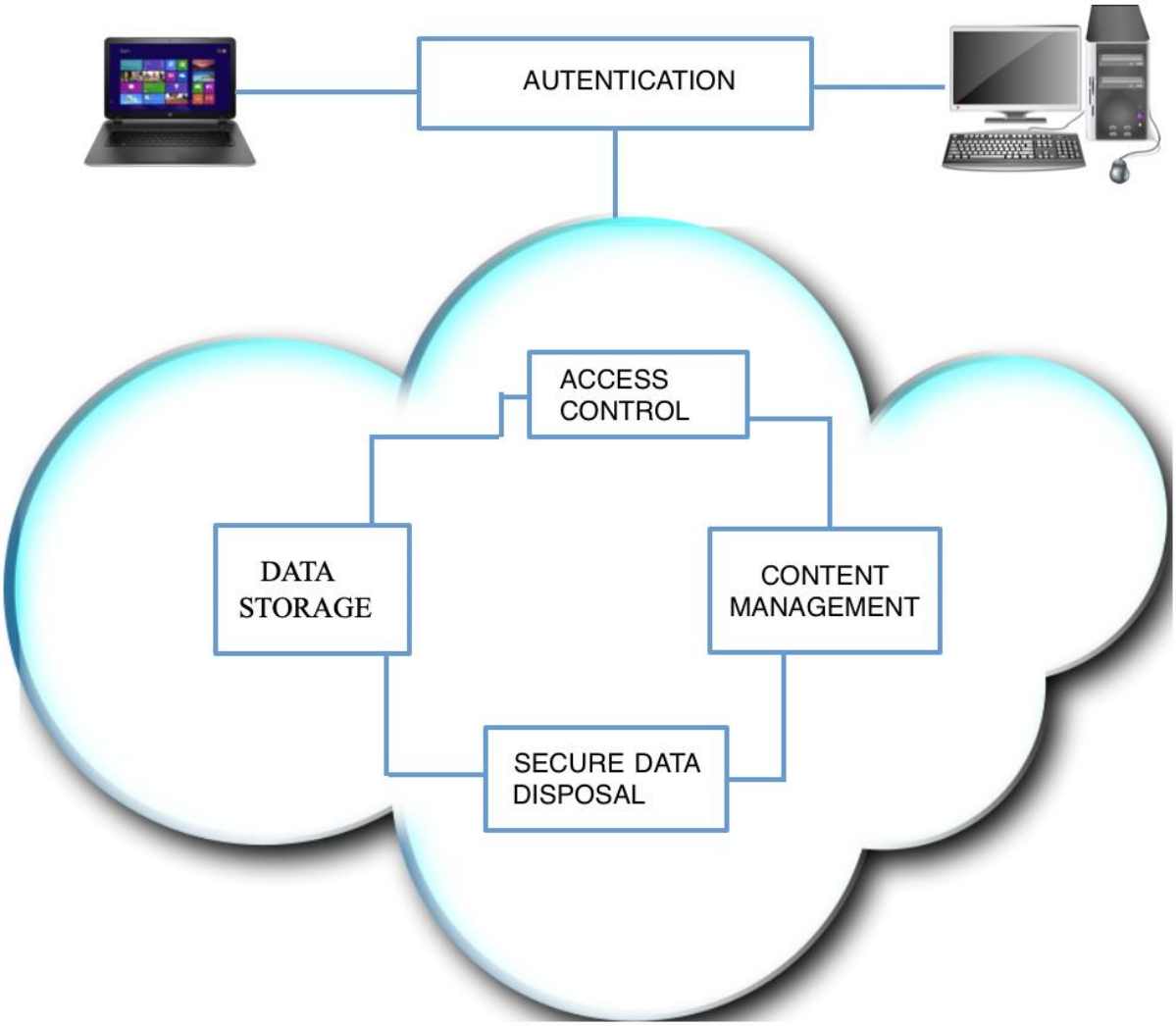


Fig. 1: Workflow diagram of crypto-shredding

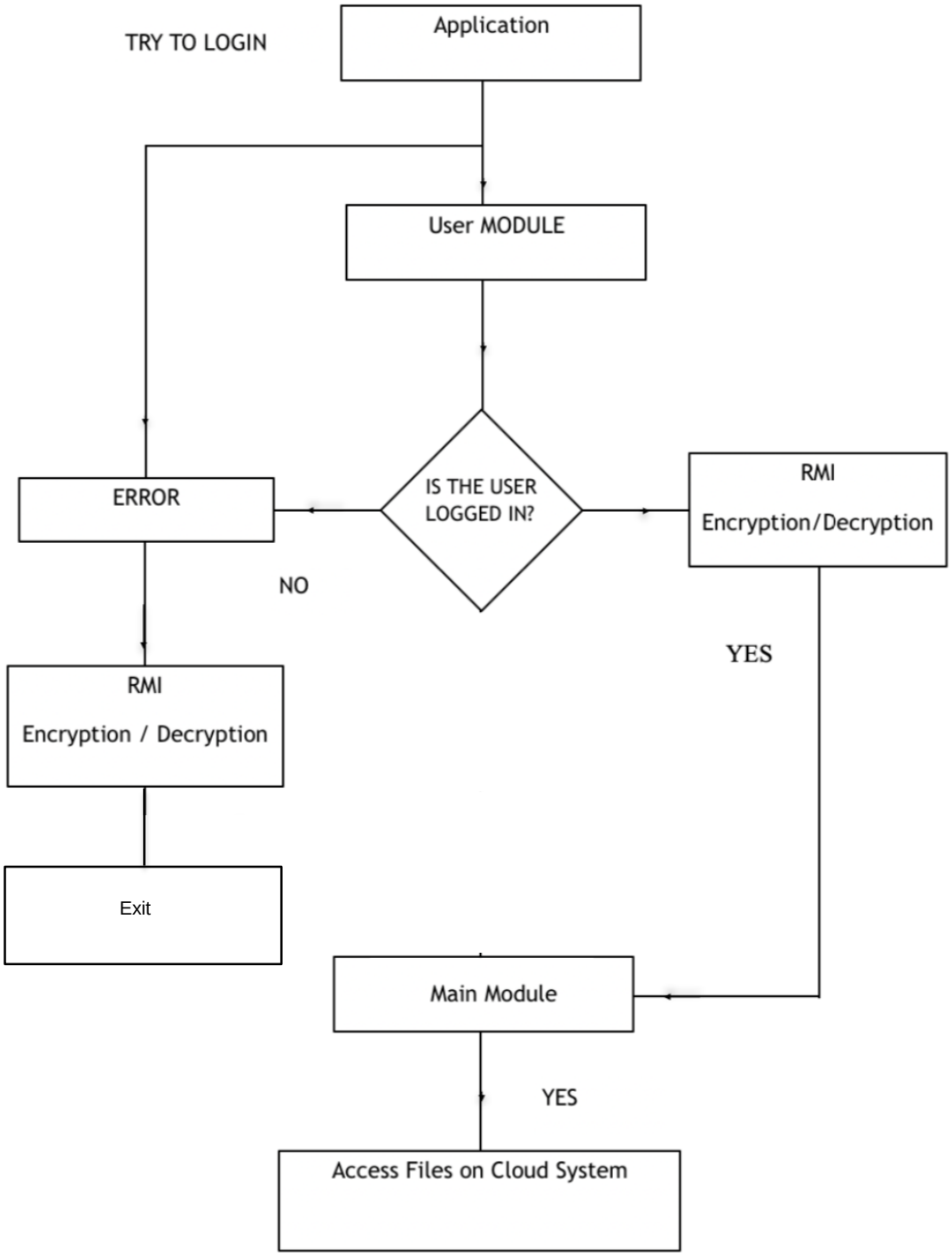


Fig. 2: Block diagram of proposed system

Data storage module

The data storage module allow users to store their data through a cloud service provider across multiple cloud servers. Users interact with these servers via the cloud service provider to access or retrieve their data. For additional security, users can perform block-level operations on their data.

To ensure data integrity, users are equipped with mechanisms to continuously monitor the correctness of their stored data, even in the absence of local copies. In cases where users are unable to monitor their data, the system allows for delegation to a trusted Third Party Auditor (TPA), ensuring that data reliability is maintained without requiring constant user involvement.

Access control module

The Access Control Module governs the authorization process before allowing clients to access their stored data. When a client submits a query, the module checks the client's credentials (username and password) to verify their identity. Upon successful authentication, the server searches for the requested file and sends it to the client. If an unauthorized intruder attempts access, the server is designed to detect this and redirect the intruder's access attempt through an alternative path, ensuring the security and integrity of the system's data.

Secure disposal module

The secure disposal module employs crypto-shredding and content discovery techniques to validate the fact that none of the disposed data is remaining in active storage or archives. Once all the data has been destroyed, the encryption keys become unavailable and the encryption protocol used is broken and cannot be guessed through brute force mechanisms, making the data irrecoverable. These modules collectively form the backbone of a secure, efficient, and reliable data management system, ensuring that all processes—authentication, data storage, access control, and data disposal—are carried out in a secure and effective manner.

The crypto shredding algorithm is as shown below

```
public class RNG Crypto Service Provider {int times ToCrypt =0;
/** Crypto Shredding Service Constructor
*/public RNGCrypto Service Provider (int times ToCrypt)
{this. Times ToCrypt = times ToCrypt;}
* Crypto Encryption technique, used to damage a
* file so it's content cannot be restored
* Note: when a file is parsed to be shredded,
* it changes all the pointer location of that file
* and also dumps in dummy data into the file,
* which now makes it un recoverable...
public String crypto(String str) {byte b[] = new byte[str.length()];
byte result[] = new byte[str.length()];
b=str.getBytes();
for(int i=0;i<str.length();i++) {result[i] = (byte) ((byte) b[i] - (byte) timesToCrypt);
System.out.println(b[i]+"-"+result[i]);}
return ( new String(result) );}
/*
* Note: decrypto is added to see if we can create
* a reverse enginering of our crypto-shredded file
 public String decrypto (String str) {
 byte b[] = new byte[str.length()];
 byte result[] = new byte[str.length()];
 b=str.getBytes();
 for(int i=0;i<str.length();i++){
 result[i] = (byte) ((byte) b[i] + (byte) timesToCrypt);
 System.out.println(b[i]+"-"+result[i]);}
return ( new String(result) );}}
```

Table 1 shows a comparison between the existing systems, namely Amazon Simple Storage Service (S3) and Amazon Elastic Compute Cloud (EC2), and the proposed system. In the second scenario, both the existing systems and the proposed system demonstrated the capability to delete files. However, in the existing systems, deleted files could be easily recovered, whereas in the proposed system, deleted files were rendered irrecoverable. Furthermore, in the third scenario, the existing systems lacked the ability to shred files, while the proposed system incorporated file shredding, ensuring that data recovery was irrevocable

.

Table 1: Comparison between existing system and proposed system

SCENARIO/ PARAMETRES	EXISTING SYSTEM	PROPOSED SYSTEM
DELETE FILE	YES	YES
RECOVER FILE	YES	NO
SHRED FILE	NO	YES

Testing and evaluation

Testing is an essential phase in software development that ensures the system functions as expected and meets the defined requirements. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. Testing presents an interesting anomaly for the software engineer. It serves as a validation process where the implemented system is evaluated against its performance and security goals. Testing provides a structured approach to identifying and addressing potential issues, ensuring that the system operates efficiently and securely. Performance testing assesses how well the system handles various operations, ensuring that all modules function correctly under different conditions. Security testing focuses on the integrity of data disposal, ensuring that files, once shredded, cannot be recovered. The results obtained from these tests provide insights into the system's overall reliability and effectiveness.

The following sections present the testing methods applied, the test cases examined, and the outcomes observed during the evaluation process. This system was examined based on two categories of testing which are performance testing and security testing.

Performance testing

To carry out certain functions and achieve the set goals of the system, performance testing was carried out to ensure that all the various modules in the system were properly analyzed, all criteria were met and the expected results were produced. The various criteria in the test cases table shows all the various performance tests that were carried out.

Security testing

To evaluate the functionality and security of the system, a series of test cases were performed. These test cases are outlined in the Table 2 and serve to assess the effectiveness of the system in achieving its goal of ensuring irreversible file destruction.

Table 2: Test case scenarios, expected results and actual outcomes

Scenario	Test Step	Expected Result	Actual Outcome
Verify that login form can accept a username and password	Login to application with username and password	Application should be able to verify username and password information	Application then decides if the provided username and password is correct.
Verify that login form can accept a username and password	Login to application with username and password	Application should NOT proceed if no username and password was provided	Application then decides if the provided username and password is correct.
Verify that login information was accepted	Processed Information	Application should display the landing page for user to navigate	A complete Landing page with options to navigate to other parts of the application
Starting the server	Clicks on Start Server Button	Application should be able to start the server successfully	Server Started and is running
Starting the server	Clicks on Start Server Button	Application could not start the server due to other constraints	Server could not be started.
Stopping the server	Clicks on Stop Server Button	Application should be able to stop the server successfully	Server Stopped and is not running
Stopping the server	Clicks on Stop Server Button	Application could not stop the server due to other constraints	Server could not be stopped and is still running.
Starting the Alert Service.	Clicks on the Start alert service	Application should be able to start an alert service successfully.	Alert service started, to stop service, just stop the entire server
Stopping the Alert Service.	Clicks on the Stop alert service	Application could not stop the alert service due other constraints	Alert service could not be stopped
Starting the Crypto Shredding Technique	Clicks on the Crypto-Shred On	Application should be able to activate the Crypto shred button for shredding	Crypto Shred button is active and ready to shred data
Stopping the Crypto Shredding Tool	Clicks on the Stop Crypto-Shred	Application could not stop due to other constraints	Crypto Shred could not be stopped
File and its content	After Shredding	Contents in the file should be lost, i.e. not readable and recoverable	Contents were lost
File and its content	After Shredding	Contents in the file should was not lost and could still be seen	Contents were not lost

RESULTS AND DISCUSSION

Fig. 3 shows the authentication GUI based on information entered, the system determines if the user is a valid one or not once the user clicks on the "LOGIN" button, a "CLEAR" button is also provided in cases where the user might have inputted a wrong text and wants to reset the form. If

Username : loveth

Password: loveth

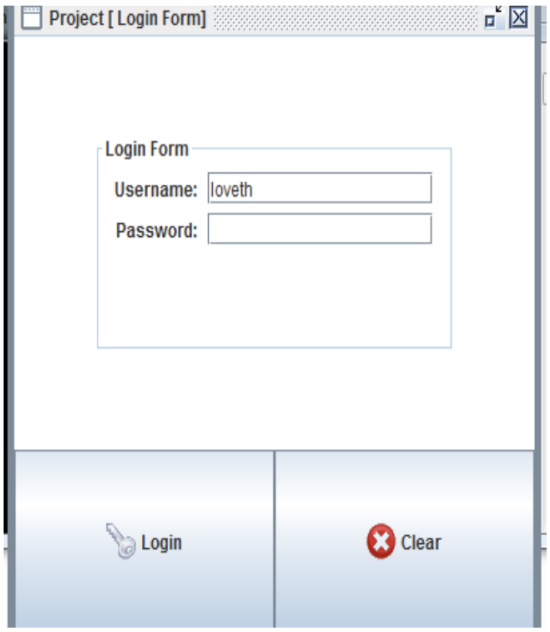


Fig. 3: Login Form

valid, then the system processes the account and moves to the next phase. This GUI interface is displayed if the previous GUI (login form) was successful. The landing page/GUI for the application is shown in Fig. 4.

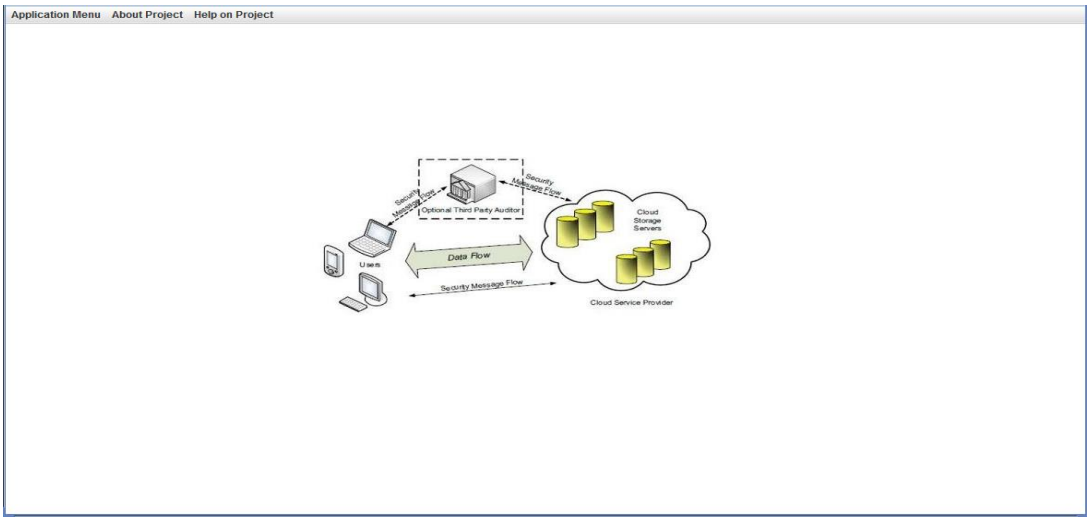


Fig. 4: Landing page for a successful logged in user

On this page, which has 3 menus bars (application menu, about project and help on project). The main application interface (i.e., starting the server) shows a layout of the entire application structure and also allows the user to actually select a particular module of interest. At this stage, the user has to click on the "START SERVER" to get the application running (Fig 5).

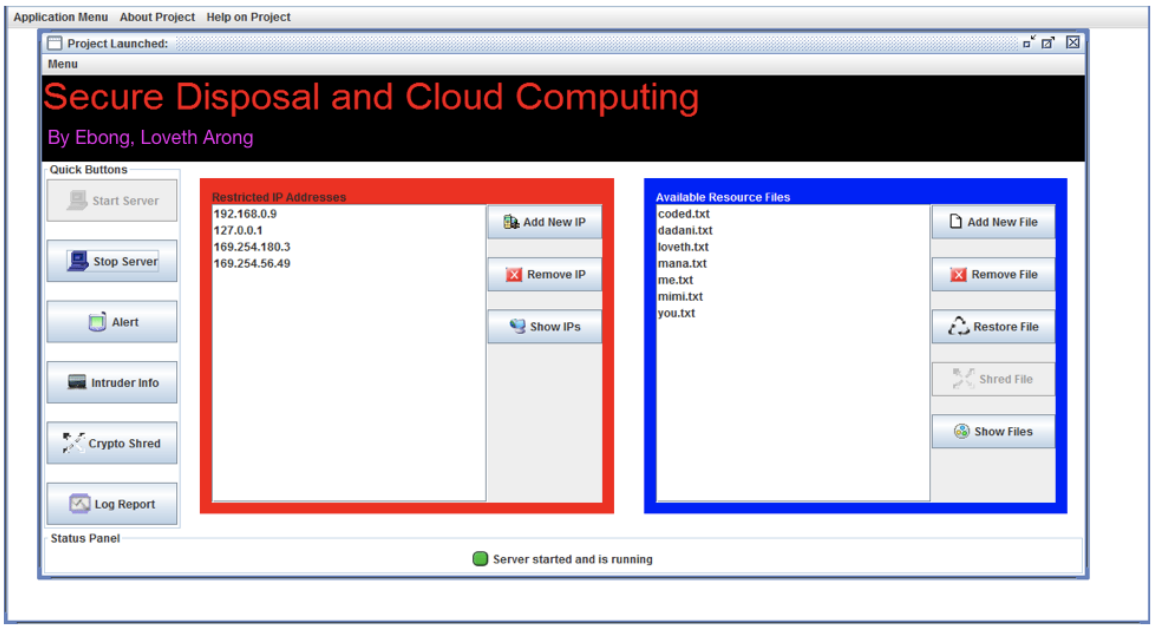


Fig. 5: Main application interface

When the server is started in the previous page, a user can click on the "STOP SERVER" to terminate all process and disconnect every connection made to the system (Fig. 6).

The alert service allows the system to send notifications to the admin once an intruder tries to gain access into the system (Fig. 7a). The threat triggers a log report that enables the administrator to visualize traces of the intruder and take further actions. The threat report (i.e., the user that tried to gain unwanted access to the system) is presented in Fig. 7b. The report also provides the name of the file the intruder was trying to access, ip address, the username, password and the date the attack was lunched.

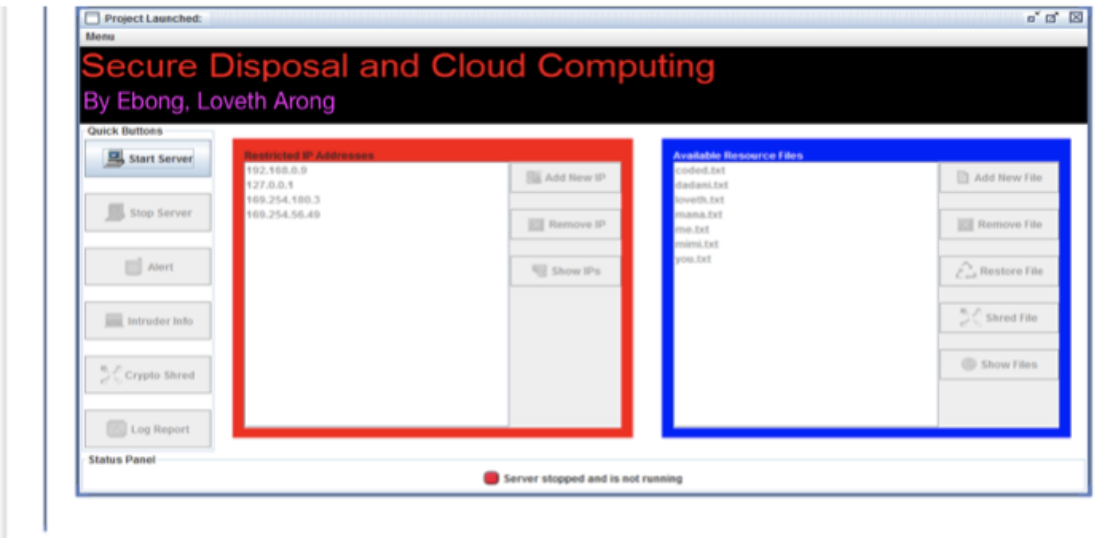


Fig. 6: Stopping the server

Fig. 7a: Alert service

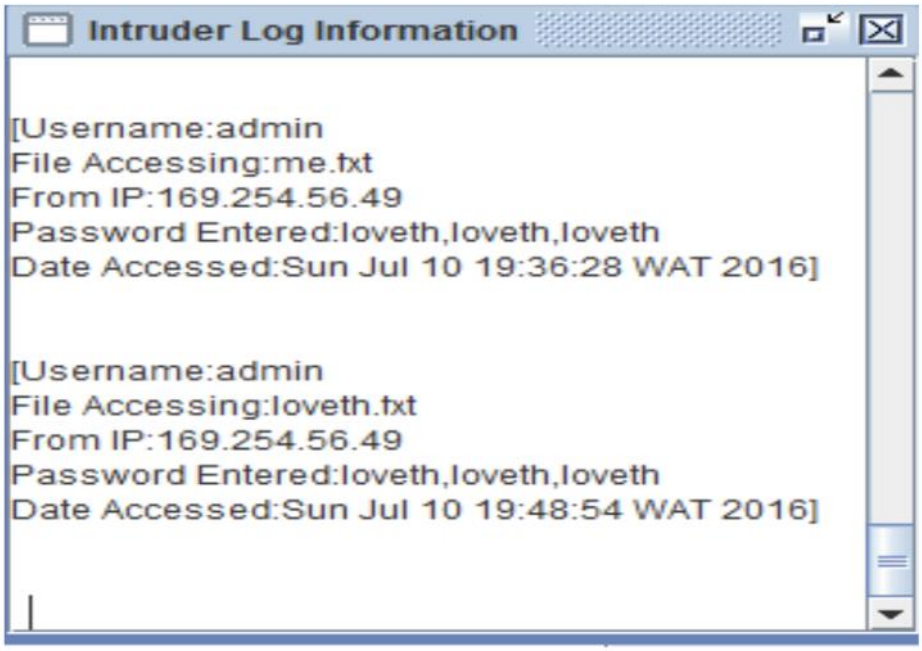


Fig. 7b: Log report

To begin the crypto-shredding process, the admin must first click on the “CRYPTO SHRED ON” button. This process activates the shredding button so that the file to be shred can be selected (Fig. 8).

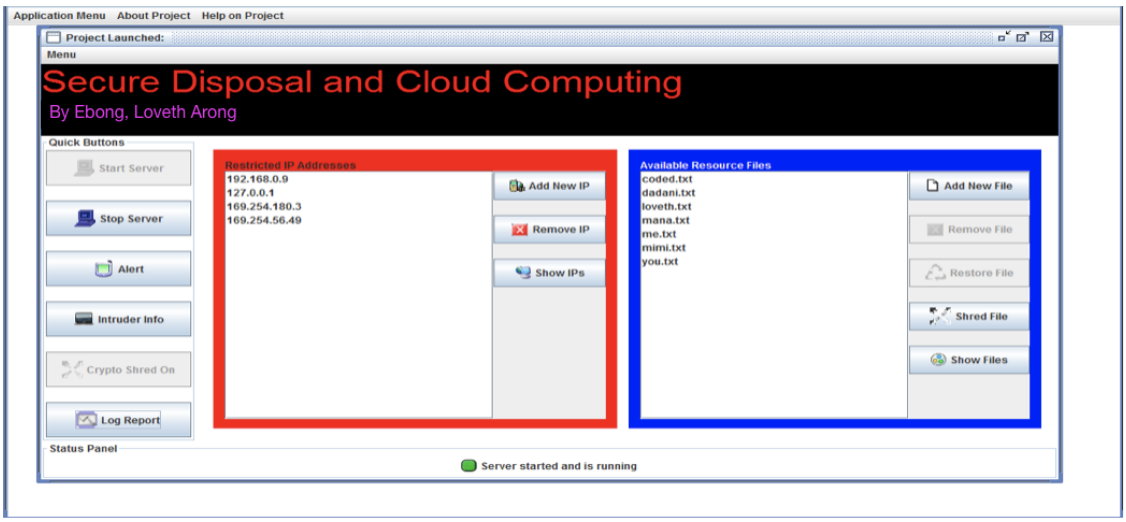


Fig. 8: Shredding technique activated (Crypto Shredding)

For the process to be initiated, some verifications are required. Fig. 9. shows an interface that allow users to perform re-authenticate before gaining access into the control center to carry out the shredding process. The system allows the re-authenticated user to provide a server key, which may differ from the password, depending on the admin's choice (Fig. 10).

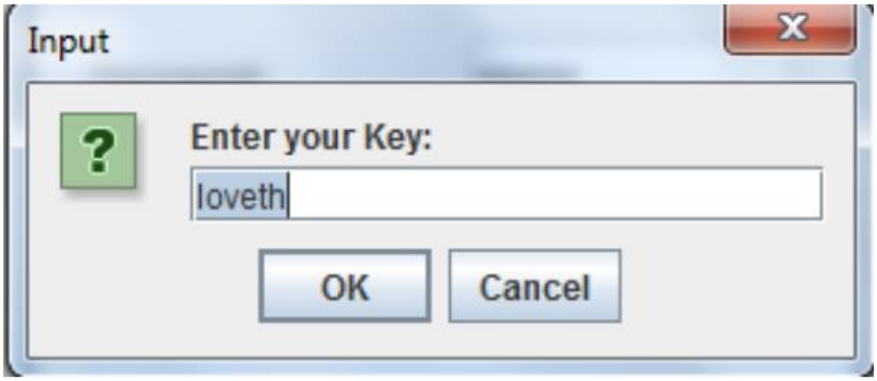


Fig. 9: Restricted log area

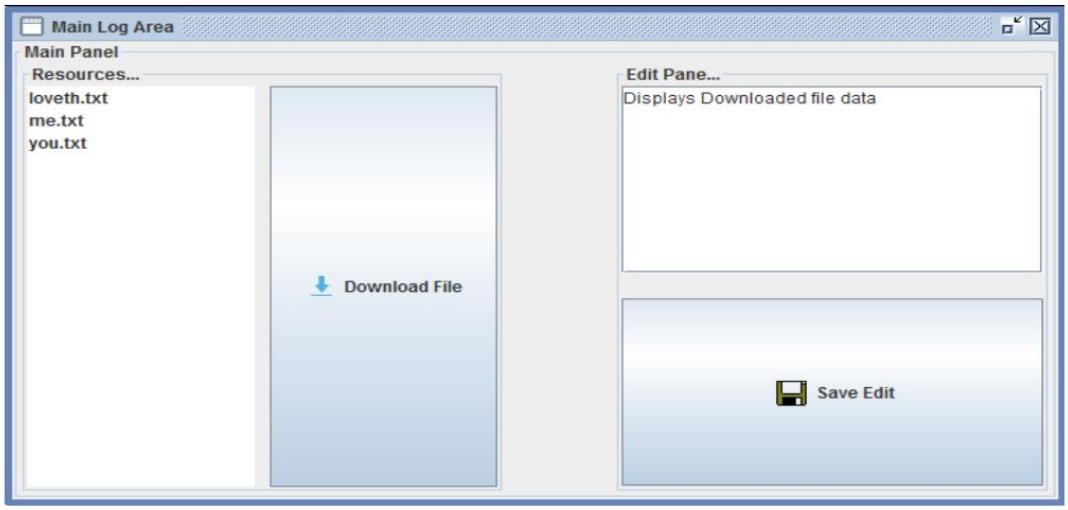


Fig. 10: Admin server key

However, the server key is secret key that provides another level of authentication and should be different from the admin password which was previously created. This is to ensure that no intruder gains access into the server. The main log panel enables the visualization of the names of all the files existing on the server (Fig. 11). The main log, display files based on IP addresses. So, once the IP address of a server changes, all files must also change their addresses, if they are to be accessed.

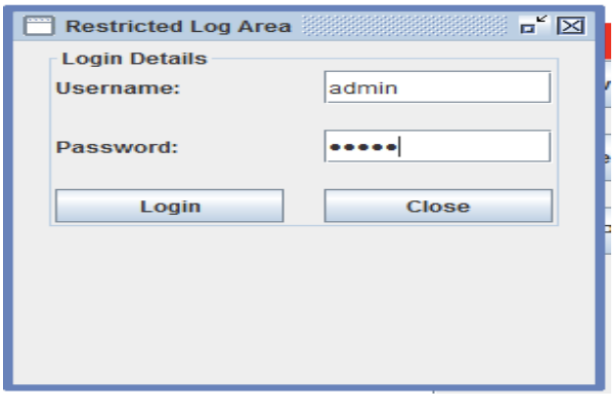


Fig. 11: Main log area

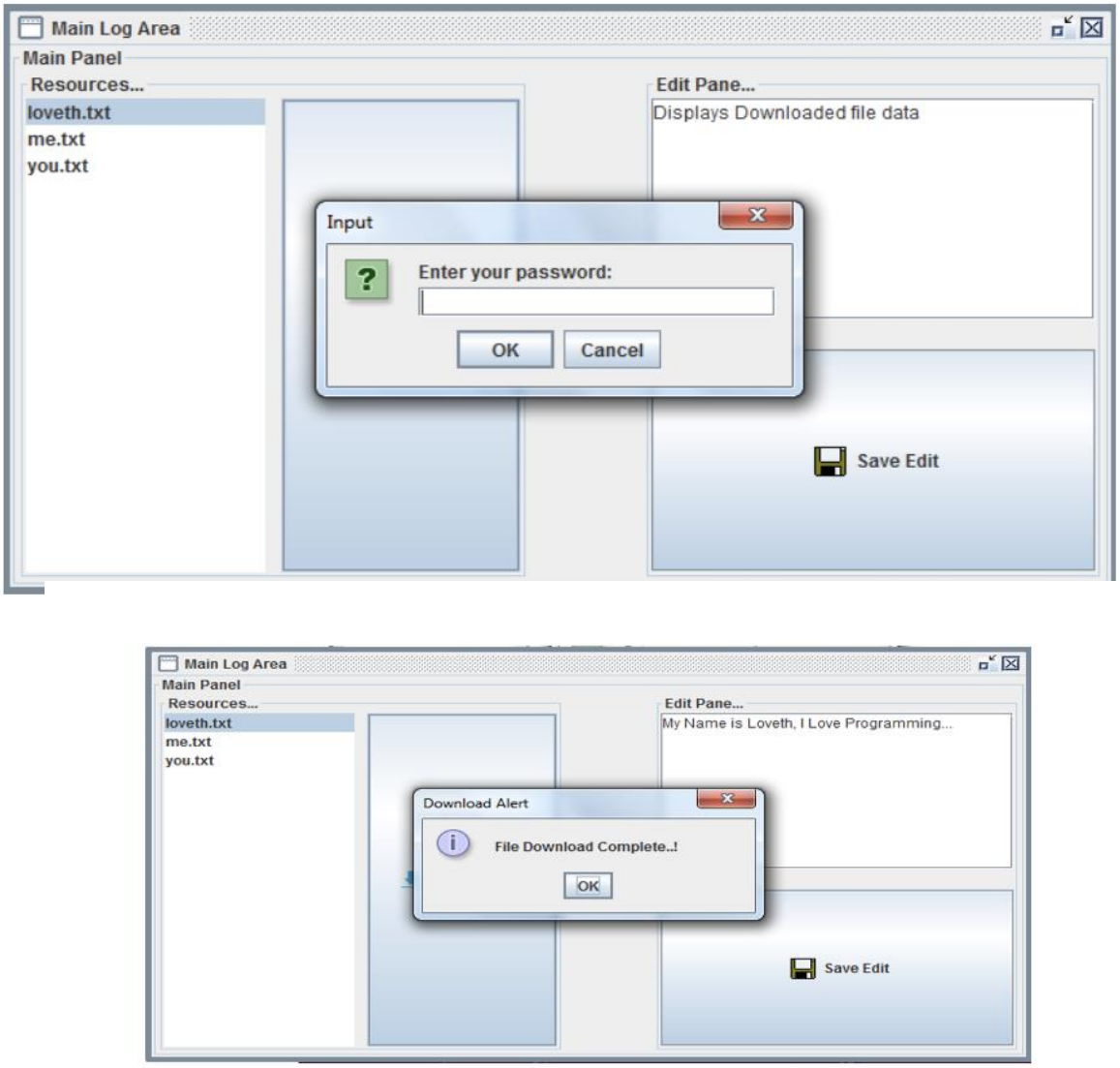


Fig. 13: Download successful

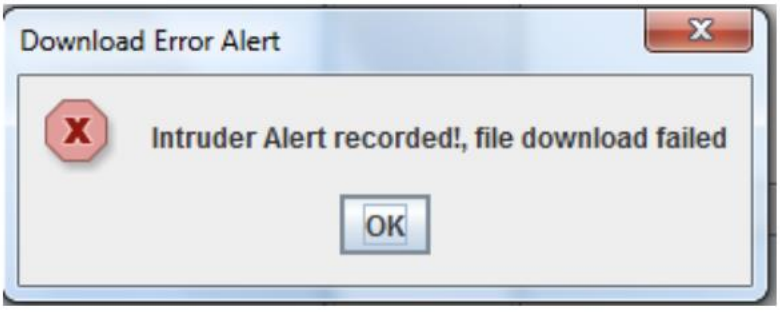
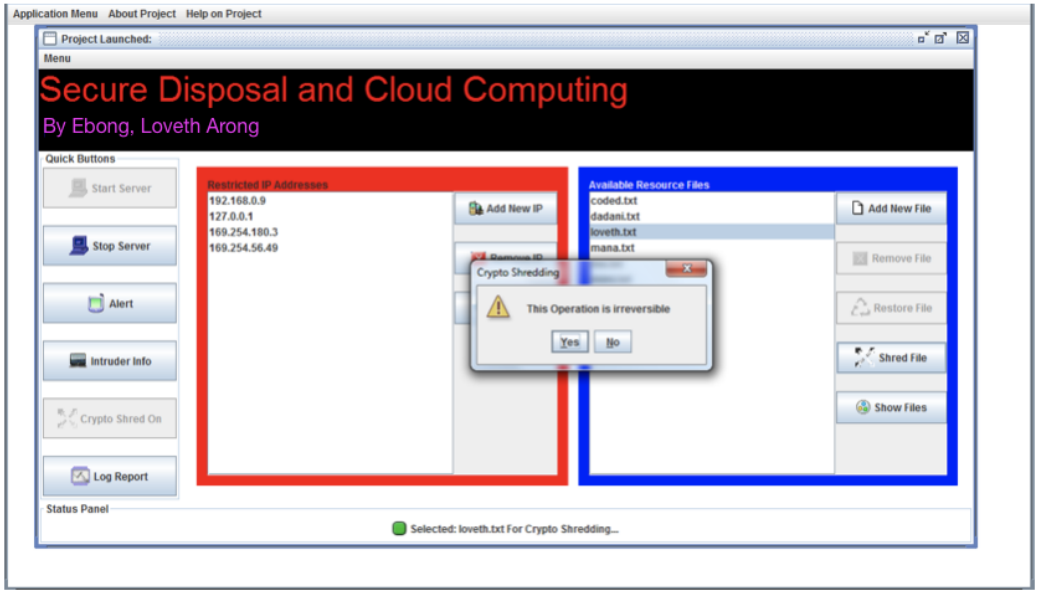


Fig. 14: Download error report



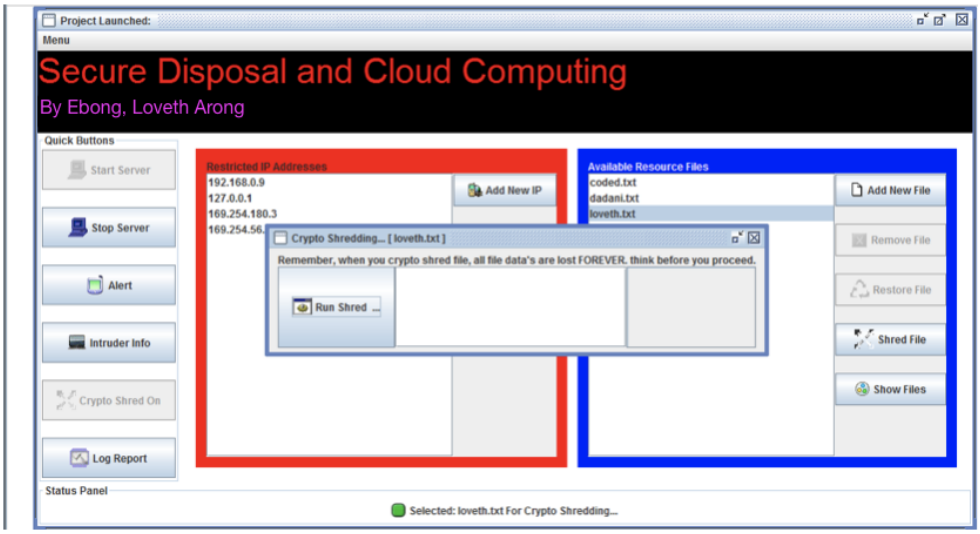


Fig. 16: Crypto-shredding initialization

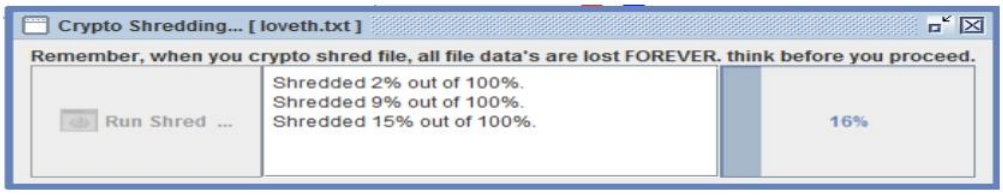


Fig. 17a: Crypto shredding in progress

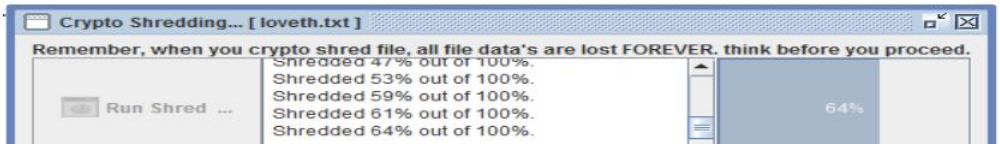


Fig. 18: Shredding successful

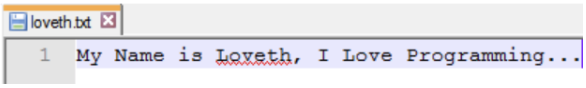


Fig. 19a: Preview of file and its content before shredding

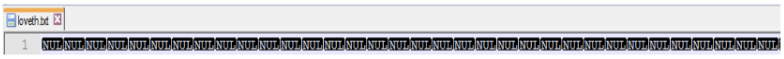


Fig. 19b: Preview of the file and its content after shredding

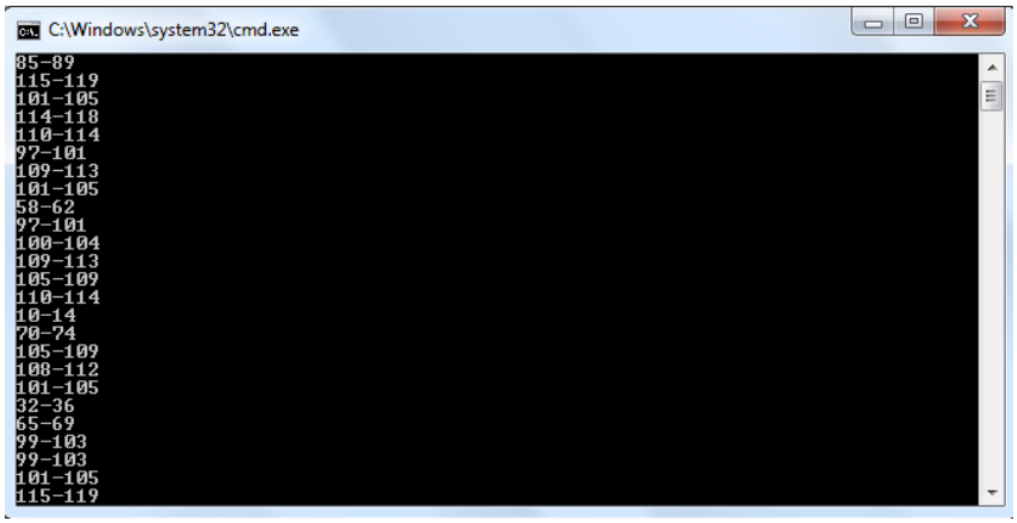


Fig. 20: MSDoS-preview of the intruders attack

When a file is to be downloaded, the system requires the user to input a password. This password must match the required password for download to be possible (Fig. 12). In this project, the password key is “loveth”, ”loveth”, ”loveth”. Once download is successful, the system displays a message “File Download Complete…” (Fig. 13). In a situation where the file download fails, an error report is displayed (Fig. 14). This occurs when an invalid password or key is entered during the authentication phase. However, when it goes through without an error report, a dialog confirmation to proceed with the operation, or cancel the operation is displayed, since the operation is irreversible (Fig. 15). Figs 16, 17a & b, shows the progress during the shredding process from initialization to completion. Once crypto-shredding has been completed, a confirmation message is displayed (Fig. 18), indicating that the process has been carried out successfully and that the file content has been destroyed. Fig. 19a & b shows the preview of the file and its content before and after the shredding process was carried. The summary of the information stored from the intruder’s attack, which was generated by the system is displayed in Fig. 20.

LIMITATIONS

This research on crypto-shredding presents several limitations that impact its practical implementation. A major limitation is its dependence on encryption key management. The security of crypto-shredding relies on the proper handling of encryption keys; if compromised before deletion, unauthorized access remains a risk. Additionally, accidental loss of keys before intentional deletion can result in permanent data loss, which may be problematic when data recovery is necessary. Another challenge is the lack of standardization across cloud platforms. Different cloud service providers use varying architectures and data management policies, making it difficult to develop a universally applicable crypto-shredding framework. This variability may limit the scalability and adaptability of the proposed system. Furthermore, the implementation of crypto-shredding introduces processing overhead, as the encryption and key management processes require additional computational resources, which could impact system performance, especially in high-volume data environments. Additionally, the irreversible nature of crypto-shredding poses a risk, as deleted data cannot be recovered if mistakenly removed, making it unsuitable for cases where recovery mechanisms are required. Legal and regulatory compliance also presents a challenge. Many industries must adhere to data retention laws, such as GDPR, HIPAA, and PCI-DSS, which require the preservation of certain records. The permanent deletion enforced by crypto-shredding may conflict with these regulations, limiting its use in regulated environments. The research has been conducted in a controlled environment, and its effectiveness in large-scale, real-world cloud infrastructures has not been fully tested. Future studies should focus on real-world deployment to assess its performance, scalability, and compliance with industry standards. Addressing these limitations can be essential for refining crypto-shredding as a reliable and adaptable method for secure data deletion in cloud computing.

CONCLUSION

This research examined the security challenges associated with cloud computing, particularly focusing on the risks of improper data deletion and unauthorized access.

Traditional data deletion methods often leave recoverable traces, creating vulnerabilities for sensitive information. To address this issue, the study explored crypto-shredding as a secure deletion technique. By permanently destroying encryption keys, crypto-shredding ensures that deleted data becomes irretrievable, even if physical copies remain stored on cloud servers. The implementation of crypto-shredding within the proposed system demonstrated its effectiveness in mitigating risks associated with data breaches. Unlike conventional cloud storage services, where deleted files can often be recovered, the proposed system successfully rendered deleted files irretrievable, thereby enhancing data security. Additionally, the system incorporated authentication measures to prevent unauthorized access, further strengthening its security framework. Cloud data security remains an area of ongoing research, with evolving challenges that require continuous advancements. As cloud computing becomes more widely adopted, ensuring privacy and security is crucial. Future research should focus on testing additional security techniques to identify vulnerabilities and develop more advanced frameworks for protecting sensitive data. Addressing these challenges will be essential in establishing more robust security mechanisms and safeguarding data integrity in cloud environments.

REFERENCES

Abdulsalam, Y.S. and Hedabou, M., 2021. Security and privacy in cloud computing: technical review. Future Internet, 14(1), p.11.

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, 79(7), pp.7377-7413.

Alam, M., Mustajab, S., Shahid, M. and Ahmad, F., 2023, November. Cloud computing: architecture, vision, challenges, opportunities, and emerging trends. In 2023 international conference on computing, communication, and intelligent systems (ICCCIS) (pp. 829-834). IEEE.

Ali, S.I., Abdulqader, D.M., Ahmed, O.M., Ismael, H.R., Hasan, S. and Ahmed, L.H., 2024. Consideration of web technology and cloud computing inspiration for AI and IoT role in sustainable decision-making for enterprise systems. Journal of Information Technology and Informatics, 3(2), p.4.

Aly, H. and ElGayyar, M., 2013, June. Attacking aes using bernstein's attack on modern processors. In International Conference on Cryptology in Africa (pp. 127-139). Berlin, Heidelberg: Springer Berlin Heidelberg.

Angel, S., Basu, A., Cui, W., Jaeger, T., Lau, S., Setty, S. and Singanamalla, S., 2023. Nimble: Rollback protection for confidential cloud services. In 17th USENIX Symposium on Operating Systems Design and Implementation (OSDI 23) (pp. 193-208).

Attaran, M., 2017. Cloud computing technology: leveraging the power of the internet to improve business performance. Journal of International Technology and Information Management, 26(1), pp.112-137.

Borra, P., 2024. An Overview of Cloud Computing and Leading Cloud Service Providers. International Journal of Computer Engineering and Technology (IJCET) Volume, 15, pp.122-133.

Bowers, K.D., Juels, A. and Oprea, A., 2009, November. HAIL: A high-availability and integrity layer for cloud storage. In Proceedings of the 16th ACM conference on Computer and communications security (pp. 187-198).

Chow, R., Golle, P., Jakobsson, M., Shi, E., Staddon, J., Masuoka, R. and Molina, J., 2009. Controlling data in the cloud: outsourcing computation without outsourcing control. In Proceedings of the 2009 ACM workshop on Cloud computing security (pp. 85-90).

Cloud Security Alliance, 2013. The notorious nine: Cloud computing top threats in 2013. Cloud Security Alliance.

Darwish, M.A., Yafi, E., Almasri, A.H. and Zuhairi, M.F., 2018. Privacy and security of cloud computing: a comprehensive review of techniques and challenges. International Journal of Engineering Trends and Technology, 7(4.29), pp.239-246.

Dhake, B., Limaye, H. and Motwani, D., 2022, January. Cloud Forensics: Threat Assessment and Proposed Mitigations. In 2022 International Conference for Advancement in Technology (ICONAT) (pp. 1-6). IEEE.

Dinadayalan, P., Jegadeeswari, S. and Gnanambigai, D., 2014, February. Data security issues in cloud environment and solutions. In 2014 World Congress on Computing and Communication Technologies (pp. 88-91). IEEE.

El Kafhali, S., El Mir, I. and Hanini, M., 2022. Security threats, defense mechanisms, challenges, and future directions in cloud computing. Archives of Computational Methods in Engineering, 29(1), pp.223-246.

Fernandes, R., Colaco, R.M., Shetty, S. and Moorthy, R., 2020. A new era of digital forensics in the form of cloud forensics: A review. In 2020 second international conference on inventive research in computing applications (ICIRCA) (pp. 422-427). IEEE.

Fraser, S. and Portillo-Dominguez, A.O., 2022. A Systematic Review of How Cloud Infrastructure and GDPR have affected Digital Investigations in a Multinational Business Context.

Godin, J. and Lamontagne, P., 2021, December. Deletion-compliance in the absence of privacy. In 2021 18th International Conference on Privacy, Security and Trust (PST) (pp. 1-10). IEEE.

Grobauer, B., Walloschek, T. and Stocker, E., 2010. Understanding cloud computing vulnerabilities. IEEE Security & privacy, 9(2), pp.50-57.

Grubbs, P., Sekniqi, K., Bindschaedler, V., Naveed, M. and Ristenpart, T., 2017, May. Leakage-abuse attacks against order-revealing encryption. In 2017 IEEE symposium on security and privacy (SP) (pp. 655-672). IEEE.

Gupta, I., Singh, A.K., Lee, C.N. and Buyya, R., 2022. Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions. IEEE Access, 10, pp.71247-71277.

Gutmann, P., 1996, July. Secure deletion of data from magnetic and solid-state memory. In Proceedings of the sixth USENIX security symposium, san jose, CA (Vol. 14, pp. 77-89).

Haeberlen, A., 2010. A case for the accountable cloud. ACM SIGOPS Operating Systems Review, 44(2), pp.52-57.

Hazela, B., Gupta, S.K., Soni, N. and Saranya, C.N., 2022. Securing the confidentiality and integrity of cloud computing data. ECS Transactions, 107(1), p.2651.

Herman, M., Herman, M., Iorga, M., Salim, A.M., Jackson, R.H., Hurst, M.R., Leo, R., Lee, R., Landreville, N.M., Mishra, A.K. and Wang, Y., 2020. Nist cloud computing forensic science challenges. Gaithersburg, MD, USA: US Department of Commerce, National Institute of Standards and Technology.

Huang, B., Gao, J. and Li, X., 2023. Efficient lattice-based revocable attribute-based encryption against decryption key exposure for cloud file sharing. Journal of Cloud Computing, 12(1), p.37.

Kostadinovska, I., 2016. Cloud security-An approach with modern cryptographic solutions. Unpublished M.Sc. Thesis submitted to the Faculty of Computer and Information Science, University of Ljubljana, Slovenia

Kumar, S., Dwivedi, M., Kumar, M. and Gill, S.S., 2024. A comprehensive review of vulnerabilities and AI-enabled defense against DDoS attacks for securing cloud services. Computer Science Review, 53, p.100661.

McKee, S., Kissel, E., Meekhof, B., Swany, M., Miller, C. and Gregorowicz, M., 2017, October. OSiRIS: a distributed Ceph deployment using software defined networking for multi-institutional research. In Journal of Physics: Conference Series (Vol. 898, No. 6, p. 062045). IOP Publishing.

Mei, R., Yan, H.B., He, Y., Wang, Q., Zhu, S. and Wen, W., 2022, August. Considerations on evaluation of practical cloud data protection. In China Cyber Security Annual Conference (pp. 51-69). Singapore: Springer Nature Singapore.

Mell, P., and Grance, T., 2011. The NIST definition of cloud computing.

Mohiddin, S.K. and Babu, Y.S., 2020. Role of cloud forensics in cloud computing. In Soft Computing for Problem Solving: SocProS 2018, Volume 2 (pp. 91-107). Springer Singapore.

Mollah, M.B., Islam, K.R. and Islam, S.S., 2012. Next generation of computing through cloud computing technology. In 2012 25th IEEE Canadian conference on electrical and computer engineering (CCECE) (pp. 1-6). IEEE.

Nived, T.M., Tiru, J.J., Jayapandian, N. and Balachandran, K., 2020. Secure Data Processing System Using Decision Tree Architecture. New Trends in Computational Vision and Bio-inspired Computing: Selected works presented at the ICCVBIC 2018, Coimbatore, India, pp.173-180.

Obi, O.C., Dawodu, S.O., Daraojimba, A.I., Onwusinkwue, S., Akagha, O.V. and Ahmad, I.A.I., 2024. Review of evolving cloud computing paradigms: security, efficiency, and innovations. Computer Science and IT Research Journal, 5(2), pp.270-292.

Parast, F.K., Sindhav, C., Nikam, S., Yekta, H.I., Kent, K.B. and Hakak, S., 2022. Cloud computing security: A survey of service-based models. Computers and Security, 114, p.102580.

Pearson, S., 2012. Privacy, security and trust in cloud computing. In Privacy and security for cloud computing (pp. 3-42). London: Springer London.

Perlman, R. and Kaufman, C., 2008, March. User-centric PKI. In Proceedings of the 7th Symposium on Identity and Trust on the Internet (pp. 59-71).

Reardon, J., Basin, D. and Capkun, S., 2013, May. Sok: Secure data deletion. In 2013 IEEE symposium on security and privacy (pp. 301-315). IEEE.

Reddy, P.R.S. and Ravindranath, K., 2024. Enhancing Secure and Reliable Data Transfer through Robust Integrity. Journal of Electrical Systems, 20, pp.900-910.

Ristenpart, T., Tromer, E., Shacham, H. and Savage, S., 2009, November. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. In Proceedings of the 16th ACM conference on Computer and communications security (pp. 199-212).

Schneier, B., 2009. Schneier on security. John Wiley and Sons.

Segun-Falade, O.D., Osundare, O.S., Kedi, W.E., Okeleke, P.A., Ijomah, T.I. and Abdul-Azeez, O.Y., 2024. Assessing the transformative impact of cloud computing on software deployment and management. Computer Science and IT Research Journal, 5(8).

Shahid, M.A., Islam, N., Alam, M.M., Mazliham, M.S. and Musa, S., 2021. Towards Resilient Method: An exhaustive survey of fault tolerance methods in the cloud computing environment. Computer Science Review, 40, p.100398.

Shukla, M.K., Dubey, A.K., Upadhyay, D. and Novikov, B., 2020, November. Group key management in cloud for shared media sanitization. In 2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC) (pp. 117-120). IEEE.

Soni, D. and Kumar, N., 2022. Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy. Journal of Network and Computer Applications, 205, p.103419.

Sunyaev, A. and Sunyaev, A., 2020. Internet computing (pp. 237-264). New York, NY, USA: Springer International Publishing.

Yaish, H., 2014. A multi-tenant database framework for software and cloud computing applications. University of Technology Sydney (Australia).

Yanamala, A.K.Y., 2024. Emerging challenges in cloud computing security: A comprehensive review. International Journal of Advanced Engineering Technologies and Innovations, 1(4), pp.448-479.

Yang, C., Huang, Q., Li, Z., Liu, K. and Hu, F., 2017. Big Data and cloud computing: innovation opportunities and challenges. International Journal of Digital Earth, 10(1), pp.13-53.

Zainuddin, A.A., Muhammad, N.F.A., Hasli, A.H.M., Zaini, N.A.J.A., Karimudin, N.B.B., Bahri, H.S.S., Nizam, M.K.M.K., Kamarudin, S.I. and Ghazalli, N., 2023. Empowering smart city governance through decentralized blockchain solutions for security and privacy in IoT communications. Bulletin of Social Informatics Theory and Application, 7(2), pp.104-117.

Zhang, J., Chen, B., Zhao, Y., Cheng, X. and Hu, F., 2018. Data security and privacy-preserving in edge computing paradigm: Survey and open issues. IEEE access, 6, pp.18209-18237.