National College of Ireland

Secure Cloud Storage with AES and 3DES: A Hybrid Cryptographic Framework

MSc Research Project
Cloud Computing

Aishwarya Gatla
Student ID: X22172297

School of Computing
National College of Ireland

Supervisor: Shreyas Setlur Arun

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Aishwarya Gatla
Student ID:	X22172297
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Shreyas Setlur Arun
Submission Due Date:	14/12/2023
Project Title:	Secure Cloud Storage with AES and 3DES: A Hybrid Cryptographic Framework
Word Count:	4560
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	26th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Secure Cloud Storage with AES and 3DES: A Hybrid Cryptographic Framework

Aishwarya Gatla
X22172297

Abstract

This research focuses on the creation of a cloud-based secure web application that uses a dual-key encryption mechanism and is hosted on AWS EC2 instances. Upon file upload, the system produces two keys for dual encryption: AES coupled with either DES or 3DES. Using PostgreSQL RDS Database within the AWS architecture guarantees safe data storage. The study focuses on the advantages of AES+3DES over AES with a single DES, stressing its innovative performance in data protection inside cloud-based online applications. This implementation emphasizes the importance of hybrid encryption approaches in improving data security and operational efficiency. The use of AES+3DES serves as a distinguishing feature, stressing the important necessity of robust security measures, particularly inside AWS settings, for safe cloud web application deployments. This study not only demonstrates the effectiveness of advanced encryption methods, but it also emphasizes the critical importance of prioritizing data security and encryption methodologies in modern web-based platforms, especially within the AWS ecosystem leveraging RDS Database PostgreSQL for robust and secure data storage on the cloud.

Keywords— AES, Encryption, Decryption, DES, AWS

1 Introduction

1.1 Background

In today's digital ecosystem, data security inside cloud systems is a top priority. The necessity to protect sensitive data stored in the cloud from ever-changing cyber threats has driven the investigation of sophisticated encryption technologies. This research examines the assessment and deployment of a hybrid encryption architecture that combines Advanced Encryption Standard (AES) with Triple Data Encryption Standard (3DES) and is specially designed to improve data security within Amazon Web Services (AWS) Cloud infrastructure.

Cloud computing has transformed data storage and accessibility by providing unprecedented ease and scalability Mrozek (2020). This convenience, however, comes with the essential duty of establishing adequate security measures to protect sensitive data from any intrusions. Encryption is a critical pillar in data security, and the combination of encryption techniques, such as the fusion of AES and 3DES, demonstrates tremendous potential in strengthening cloud-based data.

As a prominent cloud service provider, AWS provides a broad array of services intended to address a wide range of computing requirements. This study investigates the usefulness of the hybrid encryption paradigm in an AWS environment, with the goal of evaluating its compatibility, performance impact, security compliance, and overall efficacy.

1.1.1 Focus on Hybrid Encryption and AWS

The study's key point is the smooth integration of the hybrid encryption approach into AWS infrastructure. The project's major goal is to install and analyse the hybrid encryption architecture within AWS utilising services like as Elastic Beanstalk and EC2 instances, leveraging Python and Django for website development. The study focuses on the model's performance metrics within AWS, interoperability with AWS services, and alignment with AWS security rules. In summary, the purpose of this paper is to give a detailed examination of adopting a hybrid encryption architecture within AWS, with a focus on its potential to strengthen data security and integrity within cloud storage systems.

2 Aim of the study

This research looks at the assessment and deployment of a hybrid encryption model that combines AES (Advanced Encryption Standard) and 3DES (Triple Data Encryption Standard), which is especially developed to improve data security in cloud-based storage systems. The research focuses on analysing the model's influence on security measures, measuring its efficiency, and emphasising its smooth integration into AWS Cloud architecture, with AWS (Amazon Web Services) as the deployment platform. The project focuses on implementing the encryption paradigm utilising AWS services such as Elastic Beanstalk and EC2 instances, leveraging Python and Django for website development. This research focuses on AWS-centric techniques, investigating the model's performance in an AWS environment, adaptation to AWS Cloud architecture, and compliance with AWS security requirements. The research intends to give significant insights into efficiently deploying and enhancing this hybrid encryption architecture within AWS Cloud infrastructure for increased data security and integrity by focusing on AWS-centric deployment methodologies and technologies.

2.1 Research Objectives

The research objectives of this report are:

1. To evaluate the usefulness of the hybrid encryption model (combining AES and 3DES) in cloud storage systems for improving data security, assuring confidentiality, integrity, and resistance to cyber attacks

2. To evaluate the compatibility and effectiveness of implementing the hybrid encryption model—AES paired with 3DES—within AWS, focusing on how well it connects with AWS services such as Elastic Beanstalk and EC2 instances.

3. To assess the hybrid encryption model's conformance to AWS security standards and compliance requirements, and to ensure that it is in accordance with AWS best practises for data protection and privacy rules.

4. To look at the management and protection processes used in the hybrid encryption paradigm for secure key generation, storage, distribution, and key exposure or compromise prevention.

2.2 Research Questions

"What are the security enhancements and performance implications of implementing a hybrid encryption approach (AES followed by 3DES) compared to using AES or 3DES alone, specifically in an AWS environment, including its impact on computational speed, resource utilization, scalability, and deployment efficiency using services like Elastic Beanstalk?"

2.3 Research Gap

While investigating the hybrid encryption combining AES and 3DES within AWS for this study, a few topics required more investigation. To begin, there has been little research on potential vulnerabilities or weaknesses particular to the hybrid encryption approach in real-world cyber threat scenarios, which might give insight into its resilience. Second, while this study mentions AWS integration, there is a vacuum in extensive analysis addressing the practical obstacles or restrictions when applying this approach within multiple AWS services other than Elastic Beanstalk and EC2 instances, potentially overlooking larger compatibility concerns. Furthermore, the study might go deeper into the cost implications and trade-offs related to AWS's encryption policy, addressing how it affects overall operating expenditures or budget issues. Finally, more complete insights into key management within the hybrid encryption architecture are required, with a focus on potential key exposure issues or solutions for strengthening key safety beyond the basic storage and distribution aspects. These gaps in knowledge indicate chances for further refinement and development.

3 Related Work

3.1 Hybrid Encryption Models

Researchers and specialists from diverse disciplines have looked into the complicated terrain of improving data security in the area of cloud computing. Kumar and Badal (2019) stressed the significance of hybrid encryption, which combines AES and FHE, for enhanced cloud storage security. To strengthen cloud data security, Goyal and Kant (2018) called for hybrid encryption that combines homomorphic and Blowfish encryption.
Sajay et al. (2019) investigated the combination of symmetric and asymmetric key methods in cloud security. Ahmad and Garko (2019) highlighted the importance of hybrid cryptography while highlighting limitations in user authentication and algorithm implementation. Viswanath and Krishna (2021) and Denis and Madhubala (2021) concentrated on safeguarding large data and medical imaging in multi-cloud scenarios by utilising encryption techniques such as Discrete Wavelet Transform steganography, AES, and RSA. For improved cloud data security, Abroshan (2021) presented a hybrid encryption scheme that combines Blowfish and elliptic-curve-based algorithms. Seth et al. (2019) enhanced cloud data security by modifying the Paillier Homomorphic method. Rayappan and Pandiyan (2021) created a Lightweight Feistel Structure-based Substitution Permutation

Crypto Model for cloud-based multimedia data protection. Kumar et al. (2021) addressed data security problems by developing a multilayer cryptography-based security solution for cloud storage that employs DES and RSA for better encryption. Each research presented new ideas and viewpoints, focusing on hybrid encryption, customised algorithms, and specialised methodologies to improve cloud computing security. Table 1,2 and 3 shows the Comparison of Studies on Hybrid Encryption for Cloud Security.

Table 1: Comparison of Studies on Hybrid Encryption for Cloud Security

Author Name & Year	**Purpose**	**Key Features**	**Strengths**	**Weaknesses**	**Proposed Algorithm**
Kumar and Badal, 2019 Kumar et al. (2021)	Enhance cloud storage security through hybrid encryption	Hybrid encryption (AES, FHE)	Improved data confidentiality, privacy, integrity	Computational complexity	Hybrid approach of AES and FHE
Goyal et al., 2018 Goyal and Kant (2018)	Improve cloud data security using hybrid encryption	Hybrid encryption (homomorphic, Blowfish)	Enhanced data security, redundancy	Computational complexity, integration challenges	Hybrid approach of homomorphic and Blowfish encryption
Sajay et al., 2019 Sajay et al. (2019)	Develop multilevel cryptography for cloud security	Integration of symmetric and asymmetric key algorithms	Enhanced security for cloud-stored data	Neglect of user authentication, implementation challenges	Hybrid encryption with symmetric and asymmetric keys
Ahmad and Garko, 2019 Ahmad and Garko (2019)	Emphasize hybrid cryptography for cloud security	Focus on hybrid cryptography, identification of gaps	Importance of hybrid cryptography, identified gaps	Neglect of user authentication, implementation challenges	Emphasis on hybrid cryptography

3.2 Encryption Algorithms: AES and 3DES in cloud

Several studies have been conducted in this area, with each underlining the need to preserve information kept on the cloud. Elgeldawi et al. (2019) investigated eight symmetric cryptographic algorithms, Al-gohany and Almotairi (2019) and compared the efficiency of DES and AES across varying input sizes, and Commey et al. (2020) examined popular encryption methods for cloud data protection such as Triple DES, AES, Blowfish, and RSA. Both Vennela et al. (2018) and Rahul and Kuppusamy (2021) investigated the efficiency of encryption techniques—AES, DES, 3DES, Blowfish, RC4, and RSA—in cloud

Table 2: Comparison of Studies on Hybrid Encryption for Cloud Security

Viswanath et al., 2021 Viswanath and Krishna (2021)	Secure big data in multi-cloud environments	Use of encryption techniques: Discrete Wavelet Transform steganography, AES, RSA	Enhanced data security in multi-cloud setups	Integration challenges, complexity	Combination of encryption techniques
Seth et al., 2019 Seth et al. (2019)	Modify Paillier Homomorphic algorithm for cloud data protection	Amendment to Paillier Homomorphic algorithm	Strengthened cloud data protection	Implementation challenges	Modified Paillier Homomorphic algorithm
Denis and Madhubala, 2021 Denis and Madhubala (2021)	Secure medical images in the cloud	Employed Adaptive Genetic Algorithms, encryption techniques	Enhanced security for medical data transmission	Implementation complexity	Adaptive Genetic Algorithms, encryption techniques
Abroshan, 2021 Abroshan (2021)	Propose hybrid encryption for cloud security	Combination of Blowfish and elliptic-curve-based algorithms	Improved cloud data security	Integration challenges, complexity	Hybrid approach of Blowfish and elliptic-curve-based algorithms
Karnani et al., 2021 **?**	Introduce multilevel cryptography for cloud security	Use of DES and RSA for multilevel encryption	Enhanced cloud storage security	Computational efficiency, transparency	Hybrid approach of DES and RSA for multilevel encryption

systems, concentrating on characteristics such as performance, security, and algorithmic strengths. Their common purpose is to provide insights on how to use effective encryption algorithms that are matched to the complexities of cloud storage and processing, maintaining data security and integrity while improving speed. Table 4 shows the Comparison of Studies on Encryption Techniques.

3.3 AWS Security and Encryption Services

Mishra et al. (2022) investigate cloud computing security challenges and service providers' methods, whereas Saeed et al. (2019) examine the security and privacy measures of Amazon AWS S3 and Microsoft Azure Blob. Furthermore, Talha et al. (2020)address evolving network and information security concerns, arguing for improved analytical skills to deal with increasing data quantities and novel threats. Boomija and Raja (2023) offer a paradigm for securing patient e-health data in the cloud that combines Secure Par-

Table 3: Comparison of Studies on Hybrid Encryption for Cloud Security

Rayappan and Pandiyan, 2021 Rayappan and Pandiyan (2021)	Develop Lightweight Feistel Structure for multimedia data	Lightweight encryption model, focus on multimedia data	Low computational complexity, enhanced security for multimedia data	Potential limitations in scalability	Lightweight Feistel Structure based encryption for multimedia data

tially Homomorphic Encryption (SPHE) with role-based user regulations, proving its efficacy above previous encryption approaches. Mishra (2023) delves into advanced AWS services for cloud learners, including Elastic Block Storage (EBS), Elastic File System (EFS), AWS networking capabilities, and auxiliary services like as AWS Direct Connect, Snowball, and Storage Gateway. These studies attempt to improve knowledge, provide answers, and address difficulties connected to cloud technology, security, privacy, data encryption, network architecture, and advanced cloud services, while responding to the changing requirements and complexities of the cloud computing ecosystem.

4 Methodology

This study intends to prevent significant security breaches by using a hybrid encryption architecture that employs (AES, DES) and (AES, 3DES) techniques. The methods and standards for using this hybrid approach in cloud data protection constitute the foundation of this research. A comparison analysis was performed to evaluate the efficacy of AES and 3DES separately, revealing their strengths and drawbacks. The hybrid cryptographic paradigm requires the creation of two different keys, one for each AES-DES and AES-3DES combination. The design and algorithm suggested in this study serve as the foundation, highlighting key features and applying a thoroughly planned research technique. The major purpose is to assess the performance of this hybrid paradigm, with an emphasis on data confidentiality, integrity, and resistance to intrusions and assaults. Following that, the study employs dual encryption, utilising both AES and 3DES algorithms, as a critical component in preserving cloud-stored data.

There are both asymmetric and symmetric encryption algorithms in the area of encryption algorithms.

1. Symmetric Encryption Algorithms: These methods encrypt and decode data using a single key. Both persons participating in the conversation utilises the same key. AES (Advanced Encryption Standard), DES (Data Encryption Standard), and 3DES (Triple DES) are some examples that I have used in this report.Refer to figure 1.

2. Asymmetric Encryption Algorithms: These algorithms rely on a pair of keys, one public and one secret. The public key is open to the public and is used for encryption, but the private key is kept private and is used for decryption. However, we did not employ asymmetric encryption techniques in my report.Refer to figure 2.

Table 4: Comparison of Studies on Encryption Techniques

Author & Year	**Purpose**	**Key Features**	**Strengths**	**Weaknesses**	**Proposed Approach**
Elgeldawi et al., 2019 Elgeldawi et al. (2019)	Comparative analysis of symmetric cryptographic algorithms	Evaluated 8 symmetric algorithms (DES, 3DES, Blowfish, Twofish, RC2, RC5, RC6, AES)	Comprehensive assessment, algorithmic structure analysis	Limited focus on symmetric algorithms only	DES, 3DES, Blowfish, Twofish, RC2, RC5, RC6, AES
Gohany & Almotairi, 2019 Algohany and Almotairi (2019)	Comparison of DES and AES efficiencies across different input sizes	Contrast of DES and AES encryption/decryption times for varied input sizes	Detailed analysis of encryption/decryption times	Limited focus on two specific algorithms	DES, AES
Commey et al., 2020 Commey et al. (2020)	Survey of encryption methods (Triple DES, AES, Blowfish, RSA) for cloud data protection	Evaluation of popular encryption algorithms, emphasis on performance comparison	Wide coverage of prevalent algorithms	Limited coverage of newer algorithms	Triple DES, AES, Blowfish, RSA
Vennela et al., 2018 Vennela et al. (2018)	Assessing encryption efficiency (AES, DES, 3DES, Blowfish, RC4, RSA) in cloud environments	Analysis of encryption algorithms' performance in cloud settings	Focus on varied file types (text, image, video)	May lack comprehensive coverage of newer techniques	AES, DES, 3DES, Blowfish, RC4, RSA

4.1 Encryption Phase

The sender's plaintext document is compressed to reduce its size during the data encryption phase of the hybrid method applied in the web app, streamlining the encryption process. The hybrid approach then generates a random session key that serves as the symmetric key. This session key is critical for data security and is handled using asymmetric encryption algorithms—AES plus either single DES or 3DES—to generate a one-time key. This one-time key is then used by the symmetric encryption technique to encrypt the plaintext document, converting it to ciphertext and maintaining its secrecy. Once the document has been encrypted, the emphasis changes to protecting the session key. It is encrypted using the receiver's public key, guaranteeing that the session key can only be decrypted by the intended recipient who also has the associated private key. This encrypted session key is sent to the recipient together with the ciphered document, forming a secure communication channel. This methodology offers strong data security by combining symmetric and asymmetric encryption algorithms with the web app's encryption procedure.

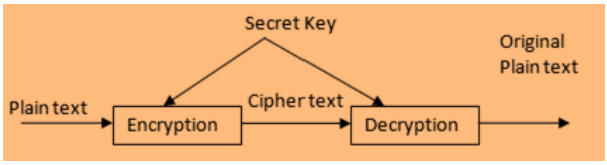


Figure 1: Symmetric Encryption Flow Diagram

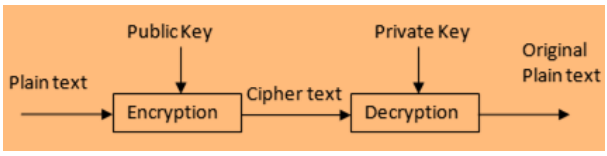


Figure 2: Asymmetric Encryption Flow Diagram

4.2 Decryption Phase

The recipient uses their private key—a critical component of asymmetric encryption—to obtain the session key encrypted with their public key during transmission during the decryption phase within the web app's hybrid algorithm. This retrieval enables the suggested approach to get access to the session key, which is used to decode the systematically encrypted cypher document. Using this session key, the technique uses the symmetric encryption counterpart to reverse the encryption applied to the document during transmission. The recipient successfully restores the original plaintext document via this inverse application of encryption methods, providing safe and accurate data retrieval within the online application.

4.3 Methodologies Used

There are two algorithms used; AES and DES.

4.3.1 AES

The Advanced Encryption Standard (AES) is critical in this process for safeguarding data within the hybrid encryption strategy. The symmetric encryption technique AES is used to encrypt the compressed plaintext document and ensure its secrecy during transmission. AES was chosen because to its powerful security features, processing efficiency, and broad usage as a trusted encryption technology. This approach assures a unique key for each data transmission by producing a Key SK for AES encryption, enhancing security safeguards against possible assaults. Furthermore, AES's computational efficiency and resilience to known weaknesses greatly contribute to the algorithm's efficacy in protecting sensitive information within the web application.

4.3.2 DES

The incorporation of the Data Encryption Standard (DES) with the Advanced Encryption Standard (AES) in this hybrid encryption process adds an extra degree of protection and compatibility. DES is a hybrid method that works with AES to provide a one-time key that is used in combination with AES encryption. This technique improves the overall

resilience of the encryption process by introducing a new cryptographic layer that ensures interoperability with systems or settings where DES is still used or enforced. While AES is the primary symmetric encryption algorithm because of its superior security, efficiency, and widespread acceptance, the inclusion of DES in this hybrid method demonstrates a comprehensive approach that caters to diverse security requirements or legacy systems within the web application environment.

The process of using AES, then Single DES, and finally Triple DES (3DES) encryption is a layered encryption system that seeks to improve security by utilising several encryption processes. This procedure is broken down as follows:

1. AES Encryption: To create ciphertext, the plaintext document is encrypted using the Advanced Encryption Standard (AES) method. AES is a secure and efficient symmetric encryption technique that is current and resilient.

2. Single DES Encryption with AES: The ciphertext generated by AES is encrypted further using the Data Encryption Standard (DES) algorithm, a less secure but still usable symmetric encryption approach. This phase seeks to add additional degree of encryption to the data that has previously been encrypted.

3. Triple DES Encryption (3DES) After Single DES: Following the Single DES encryption, the data is encrypted again using Triple DES (3DES), a symmetric encryption process that employs DES three times in a row. Despite its antiquity, 3DES is noted for its increased security as a result of many rounds of encryption.

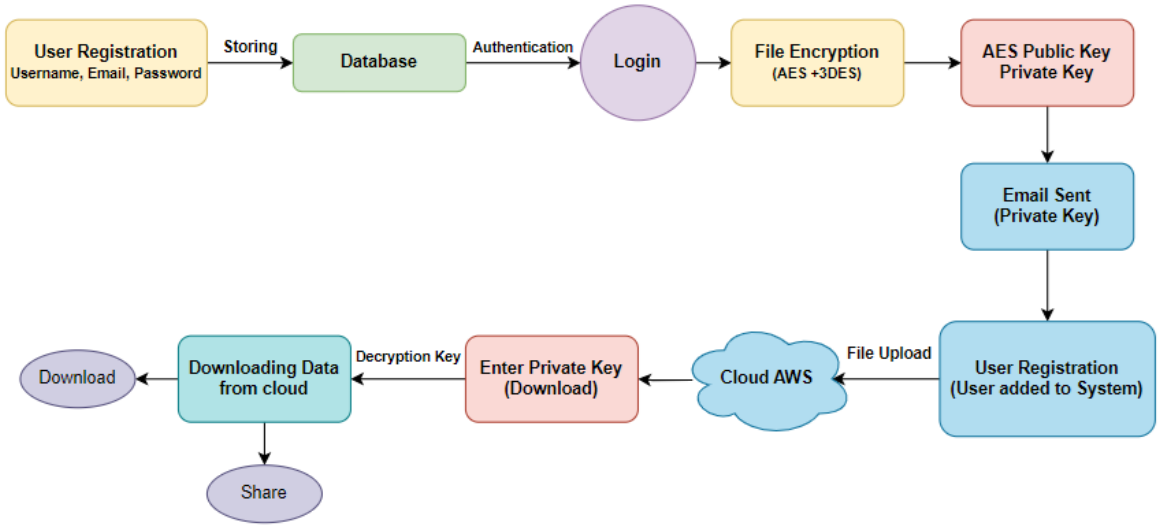
This multistep encryption procedure produces a nested or layered security strategy in which the plaintext is encrypted using AES first, then with Single DES and finally Triple DES, increasing complexity and making it more difficult for unauthorised organisations to read the original plaintext. However, while numerous layers of encryption might boost security, they can also reduce performance and efficiency. In practise, striking the correct balance between security and performance is critical.

5 Design Specification

To enable efficient operation and safe data management, the suggested system design includes numerous critical functionalities. Initially, the administrator registers on the online application by entering required credentials such as a username, name, email address, and password, as well as conducting a password confirmation check. These credentials are safely kept in the database after successful registration for subsequent authentication. Subsequent admin logins are validated against these saved credentials to get access to the application's main page. The administrator determines the type of data or file to be encrypted, launching the first degree of protection with a hybrid approach—AES followed by Triple DES. During this procedure, the application's database holds both the public and private keys of AES, guaranteeing safe data management.

Furthermore, the admin sends the private key to users via email once they successfully register for the online application, along with a unique password for further protection. Users enter their credentials and gain access to their allocated accounts. Users enter their private key in the download area. Following accurate input, the system uses DES to query the AES public key straight from the database.

This architecture implements strict security mechanisms. Users can only download the plain text file if both keys (private and queried public) are entered correctly. The system architecture prioritises secure communication between users and the application, safe key storage, and key validation for secure data retrieval. The architectural diagram, shown in Figure 4.1, visualises the movement and interaction of these components, demonstrating the system's methodical approach to encryption, decryption, and safe data access.



Block Diagram for the proposed system

The design specifications chapter provides a thorough blueprint for installing a hybrid encryption model on Amazon Web Services (AWS), focusing on its interface with current infrastructure and the functionality of the accompanying web application. This section describes the technical standards, architectural structure, and operational methods needed to install the hybrid encryption model, which combines Advanced Encryption Standard (AES) and Triple Data Encryption Standard (TDES) (3DES). It digs into the technicalities of using the Python and Django frameworks for web application development, as well as designing the system's architecture with AWS services such as Elastic Beanstalk and EC2 instances. It also describes the web application's user interface design and functions, with an emphasis on user experience, file management, and encryption procedures. The chapter concludes with a detailed breakdown of technical specifications, system components, integration procedures, and user interactions, laying the groundwork for deploying the hybrid encryption architecture on AWS.

6 Implementation

The implementation chapter deals with the methodical execution of the given system design. It starts with setting Amazon RDS and PostgreSQL to securely store encrypted data and user credentials. EC2 instances are provided with essential security parameters, Python/Django frameworks, and application requirements using AWS services. Elastic Beanstalk is used for web deployment, which streamlines the process while maintaining scalability and load balancing. The implementation integrates strong encryption techniques, such as (AES+DES) and (AES+3DES), inside the application's source, assuring

data security. Integration testing checks the smooth interaction of components, with a focus on safe key exchange and data retrieval. The user interface design is iteratively refined within Django to provide a user-friendly experience.

6.1 Limitations of Single DES and Implementation of Triple DES

Single DES (Data Encryption Standard) has various drawbacks, the most significant of which is its set key length of 56 bits. This key length is vulnerable to brute-force attacks in today's computing world, undermining its security. Because of the comparatively short key length, it is vulnerable to exploitation via quicker processing powers, making it less effective against newer threats. The introduction of powerful processing technology has expedited the ability to decrypt DES-encrypted data within a reasonable timescale, while decreasing its dependability, as we have shown in previous research work. To mitigate these issues, Triple DES (3DES) encryption was required. 3DES applies the DES algorithm three times in a row, considerably increasing its security by encrypting data in several passes and utilising multiple keys. This technique mitigates the weaknesses associated with single DES's short key length, increasing resistance against sophisticated cyber threat. So that is why this research is implementing with novelty AES with triple DES

7 Evaluation

7.1 User Registration

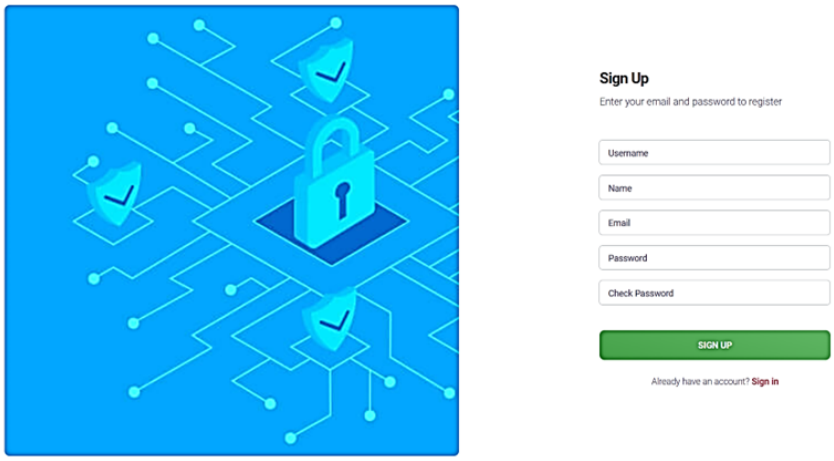


Figure 3: User registration screen

The user registration output page, shown in Figure 3 starts the system's user registration procedure. The user interface allows the user to enter important information such as their login, name, email address, password, and a confirmation entry for password validation. This extensive set of fields guarantees that all relevant user information is collected for account creation. After providing these data, the user advances by selecting the "Sign Up" button, which initiates the registration procedure for new users. This action verifies

the user's desire to register and authorizes the system to process the submitted information, run validation tests such as checking password consistency, and establish a new user account within the program. The screen provides a simple and user-friendly interface, leading users through the necessary steps for successful registration and, once completed, granting them access to the system's functionality and features.

7.2 User Login

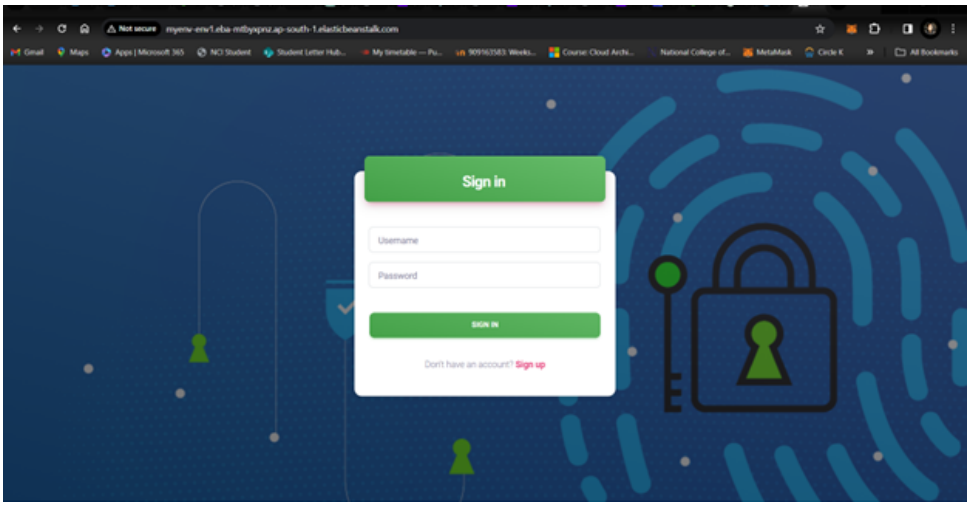


Figure 4: User Login Screen

Figure 4 depicts the system's user login output screen. The admin assigns each user a unique user ID and password to access their allotted account. When users visit the login screen, they enter the allocated username and password into the appropriate areas. Users proceed by clicking the "Sign In" button after inputting their login credentials. This initiates a validation procedure in which the system compares the entered login and password to the database data. The system runs a verification check, comparing the submitted credentials to the entries stored in the database. The user receives successful access to the system's functionality if the supplied username and password match the stored data, validating their correctness.

7.3 User File List

Figure 5 depicts the interface of the file-sharing system, notably the "My Files" area, where users may view and manage the files they've posted. This area acts as a consolidated repository for uploaded files related with the user's account. Each file entry provides important information such as the file name and a description, providing a clear image of the uploaded files' contents and purpose. Users may see and organize their submitted files inside this interface, allowing for convenient access and administration.

7.4 Data Encryption

The interface for uploading files to the cloud storage system is shown in Figure 6. It demonstrates the capability where users may pick a file, in this example labelled "TEST

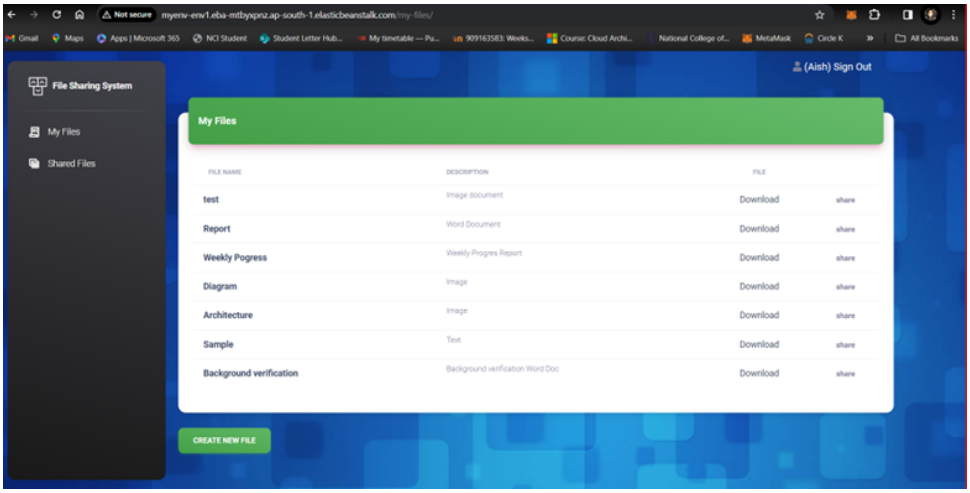


Figure 5: My Files Uploaded File Management Interface

DATA," for encryption and subsequent cloud storage. The interface provides a simple way for users to commence the encryption and saving procedure for the selected file. Users may interact with this interface by selecting the file they want to upload and encrypt and then starting the encryption process.

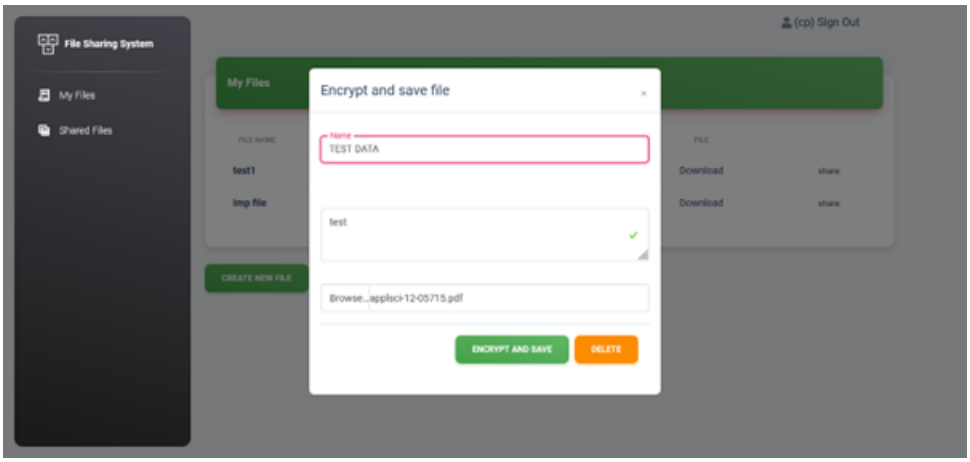


Figure 6: Upload and Encrypt to Cloud Interface

. After uploading the file figure 7 shows that hybrid encryption has been performed having both algorithm names (AES+DES) and (AES+3DES) followed by their time consumption and file sizes in bytes to safeguard the file's content which is storing it in the cloud storage.

7.5 Shared Files

Figure 8 shows all shared files which the user has shared.

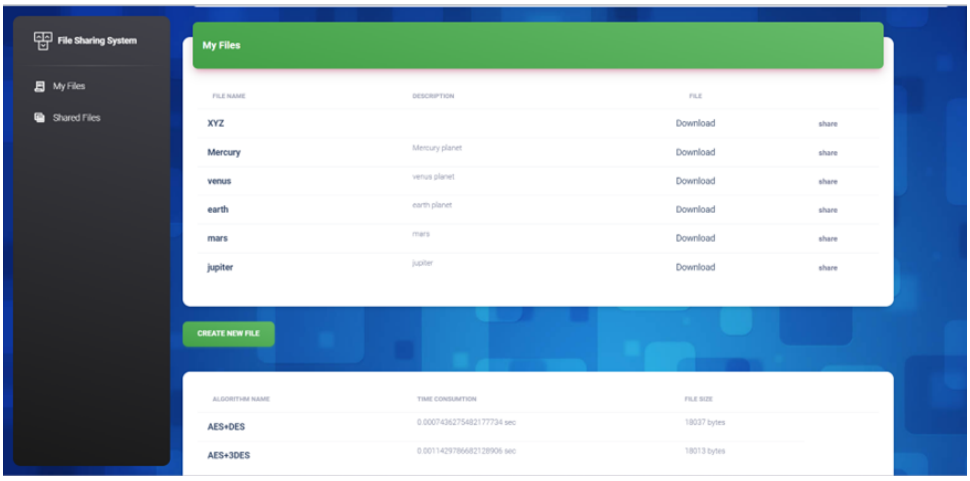


Figure 7: Encryption Performed (HYBRID Encryption)

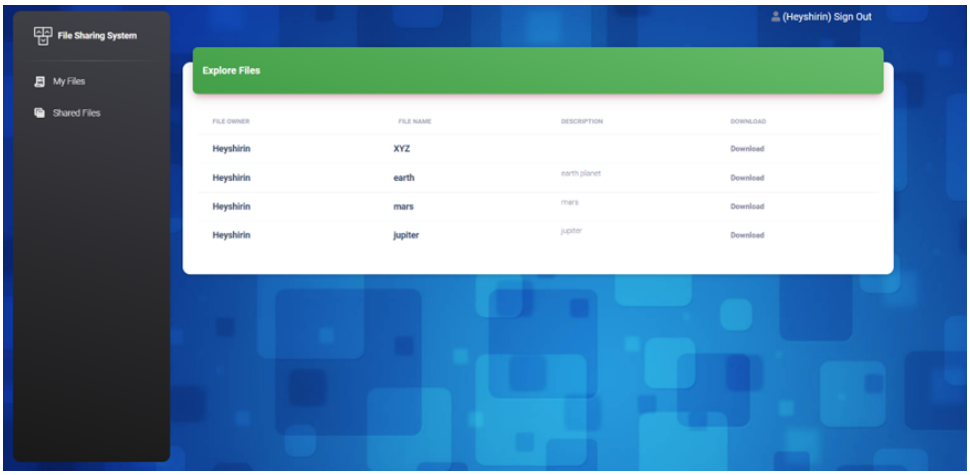


Figure 8: Shared Files

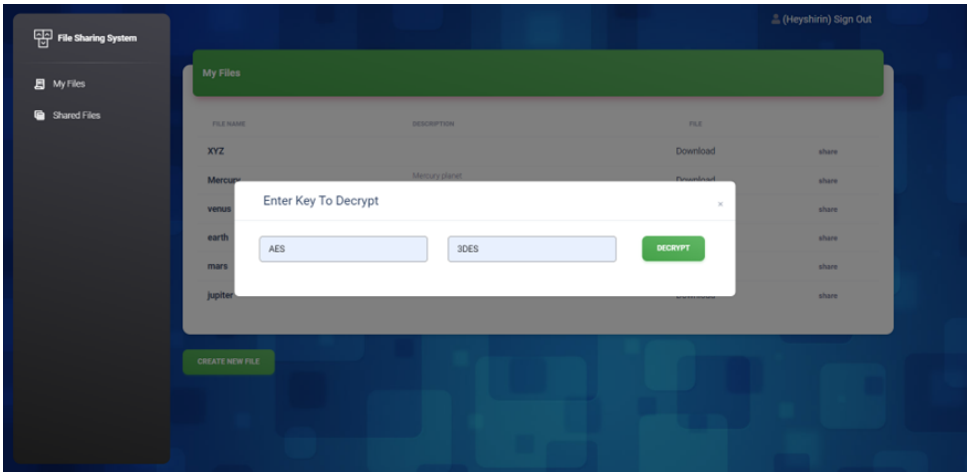


Figure 9: Data Decryption

7.6 Data Decryption

Figure 9 depicts the system's data decryption interface. Users are required to provide the decryption key, which is most likely the private key previously supplied or related with the encrypted file to be decrypted via email. This interface acts as a safe conduit for users to enter the decryption key needed to unlock and access encrypted data stored within the system. Users interact with this interface by entering the proper decryption key into the appropriate area, so starting the decryption process. Once the right decryption key is input, the system decrypts and retrieves the original plaintext data using the relevant decryption algorithm—potentially AES and 3DES in reverse order.

7.7 AES+DES Results

7.7.1 Analysis of AES+DES with Image format

Table 5 and Figure 10 shows the encryption timings, measured in seconds when combining AES and DES encryption methods on image files ranging in size from 6KB to 132 KB. The encryption time displays how long it takes to encrypt data using the combined AES+DES method. There is a notable incremental trend in encryption time as the file size grows, indicating a relationship between file size and encryption duration. Small files, such as 6KB and 7KB, have exceptionally low encryption times, measured in milliseconds, emphasising the encryption process's efficiency for smaller data sets. However, when larger files, such as 132KB, are used, the encryption time increases somewhat but stays relatively quick, proving the usefulness of the combined AES+DES technique even with greater data quantities. While encryption time increases with larger file sizes, the AES+DES combination maintains a decent degree of efficiency across changing file sizes, demonstrating its ability to handle encryption quickly and effectively over a range of data sizes.

File Size (KB)	Encryption Time (s)	File Size (bytes)
6	0.00042	15444
7	0.00074	18013
52	0.00528	151004
79	0.00588	231753
132	0.01243	386327

Table 5: Encryption Time of AES+DES (Image)

7.7.2 Analysis of AES+DES with Doc format

Table 6 and Figure 11 only applies to DOC file types with file sizes ranging from 37KB to 620 KB. The encryption time indicates how long it takes to encrypt DOC files using the AES+DES combination. The data shows a consistent pattern in which, as the file size grows, there is a modest but apparent increase in encryption time. Smaller DOC files, such as the 37KB and 100KB ones, have comparatively quick encryption speeds, measured in milliseconds, emphasising the rapid encryption procedure for smaller-sized documents. However, as the file size climbs to 620KB, the encryption time increases considerably, indicating a proportionate link between file size and encryption time.

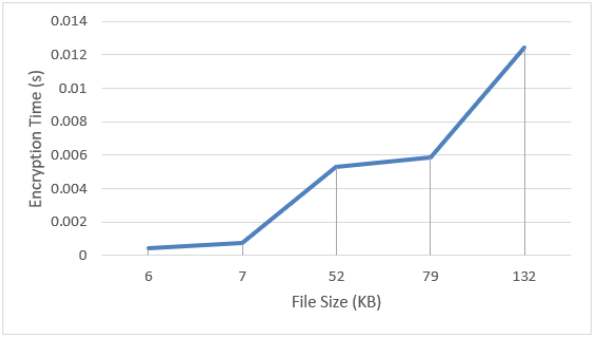


Figure 10: Graph of the encryption time of AES+DES of Image

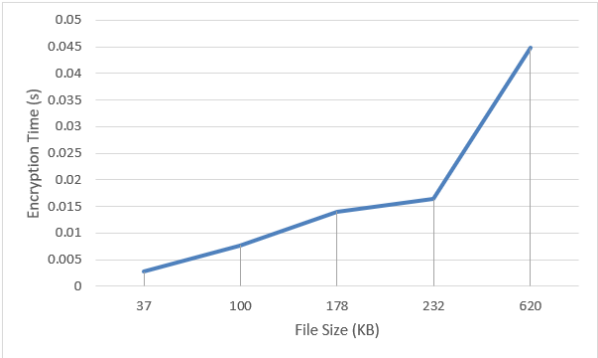


Figure 11: Graph of the encryption time of AES+DES of Document

File Size (KB)	Encryption Time (s)	File Size (bytes)
37	0.00275	107135
100	0.00771	291807
178	0.01409	521031
232	0.01635	679863
620	0.04479	1823131

Table 6: Encryption Time of AES+DES (Document)

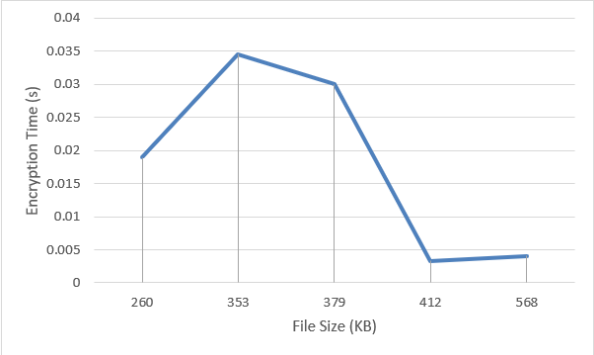


Figure 12: Graph of the encryption time of AES+DES of Audio

7.7.3 Analysis of AES+DES with Audio format

Table 7 and Figure 12 shows various file sizes ranging from 260KB to 568KB. The encryption time represents the time it takes to encrypt audio files using the combined AES+DES encryption method. Notably, the table shows a shifting trend in audio file encryption times relative to file size. Encryption time increases noticeably with larger audio file sizes such as 260KB, 353KB, and 379KB, indicating a proportionate link between file size and encryption duration. Smaller audio files, such as 412KB and 568KB, have much lower encryption periods, measured in milliseconds, indicating a speedier encryption procedure for these smaller audio file types. It's important to notice that the AES+DES encryption method has varying encryption periods for different audio file sizes, demonstrating its versatility while also indicating a proportionate increase in encryption duration with larger audio file sizes.

File Size (KB)	Encryption Time (s)	File Size (bytes)
260	0.01892	7764295
353	0.03455	1039138
379	0.03001	1213466
412	0.00333	1478188
568	0.00410	1689321

Table 7: Encryption Time of AES+DES (Audio)

7.7.4 Analysis of AES+3DES with Image format

The following table 8 and Fiure 13 displays encryption durations in seconds for using both AES and 3DES encryption methods in tandem, applied especially to image file

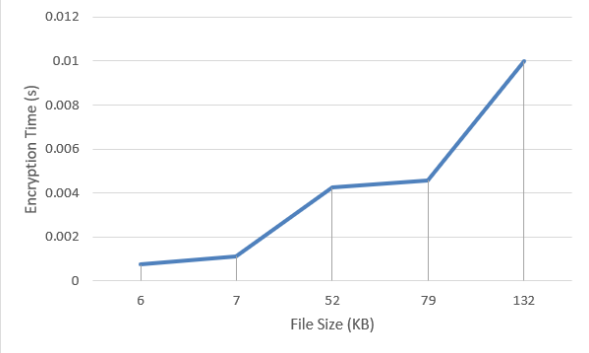


Figure 13: Graph of the encryption time of AES+DES of Image

types ranging in size from 6KB to 132KB. This encryption time indicates how long it takes to encrypt image files using the combined AES+3DES encryption algorithm. The data shows a constant pattern in which smaller image files, such as those at 6KB and 7KB, have significantly shorter encryption times, measured in milliseconds, emphasizing the efficacy of encryption for lower image file sizes. With images at 132KB, the encryption time increases somewhat, indicating a proportionate link between file size and encryption duration. Even with larger image file sizes, however, the AES+3DES encryption technique retains decent efficiency, demonstrating its ability to handle encryption well across diverse image file sizes while keeping a continuous trend of encryption duration proportionate to file size.

File Size (KB)	Encryption Time (s)	File Size (bytes)
6	0.00076	15555
7	0.00114	18037
52	0.00424	151361
79	0.00458	231605
132	0.00999	387036

Table 8: Encryption Time of AES+3DES (Image)

7.7.5 Analysis of AES+3DES with Document format

Table 9 and Figure 14 shows document file types in various file sizes ranging from 37KB to 620 KB. These encryption times indicate how long it takes to encrypt documents using the AES+3DES encryption algorithm. The data shows a consistent trend in which lower document file sizes, such as 37KB and 100KB, display comparatively quick encryption times, measured in milliseconds, demonstrating the effectiveness of encryption for smaller document file sizes. As the file size climbs to 620KB, the encryption time increases gradually but noticeably, illustrating a proportional link between file size and encryption duration.

7.7.6 Analysis of AES+3DES with Audio format

The information in table 10 and Figure 15 applies exclusively to audio file formats with file sizes ranging from 260KB to 568KB. These encryption timings indicate how long it

File Size (KB)	Encryption Time (s)	File Size (bytes)
37	0.00206	107325
100	0.00584	292339
178	0.01025	520454
232	0.01283	681268
620	0.03495	1823686

Table 9: Encryption Time of AES+3DES (Document)

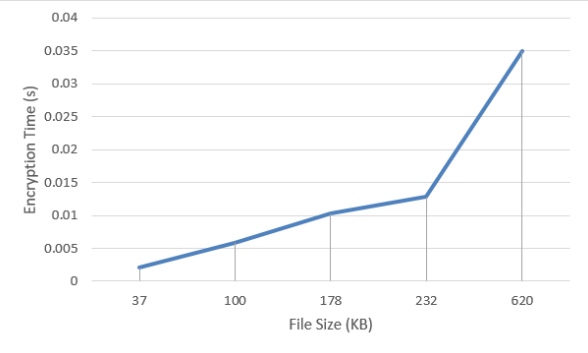


Figure 14: Graph of the encryption time of AES+DES of Document

takes to encrypt audio files with the AES+3DES encryption algorithm. Larger audio file sizes, such as those at 260KB, 353KB, and 379KB, indicate a considerable increase in encryption time, demonstrating a proportionate link between file size and encryption duration. Smaller audio files, such as those weighing 412KB and 568KB, have much lower encryption periods, measured in milliseconds, indicating a comparatively quicker encryption procedure for these smaller audio file types.

File Size (KB)	Encryption Time (s)	File Size (bytes)
260	0.01477	7649591
353	0.02968	1037929
379	0.02031	1213466
412	0.00213	1378188
568	0.00392	1589321

Table 10: Encryption Time of AES+3DES (Audio)

7.8 Comparative Analysis: AES+DES vs. AES+3DES Performance

The comparison of AES+DES and AES+3DES encryption algorithms demonstrates unique performance characteristics over a wide range of file types and sizes. Encryption timings for AES+DES are usually efficient, especially for lower file sizes, demonstrating quick encryption operations across image, document, and audio file types. However, as file sizes get larger, the encryption duration increases proportionally, while being quite efficient. Notably, AES+3DES exhibits flexibility but exhibits a persistent tendency of

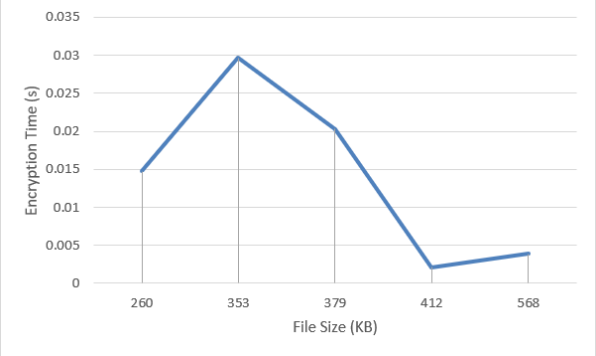


Figure 15: Graph of the encryption time of AES+DES of Audio

longer encryption times compared to AES+DES, especially with smaller file sizes, indicating significantly worse encryption performance. As file size increases AES with 3DES gives better result, less time-taking also performing good.

8 Conclusion and Future Work

8.1 Conclusion

The primary focus on security and speed in delivering this web application led to the construction of a dual-key system leveraging Python and Django technologies within the AWS Cloud via EC2 Instances. The production of two different keys—one for AES and one for DES or 3DES—during file upload enables a strong dual encryption operation. AWS's infrastructure, particularly the EC2 Instances, played a critical role in guaranteeing the secure execution of various encryption processes inside this framework. The decisive findings highlighted the AES+3DES hybrid encryption technique's considerable superiority over AES with single DES, demonstrating its new performance and dependability in safeguarding data within online applications. The investigation of this encryption technology within the AWS environment not only confirmed its effectiveness, but also underscored the need of prioritising security measures in data transit and storage within web-based platforms. This study highlights the limits of single DES owing to its susceptible key length, requiring the use of Triple DES for greater security. While Triple DES provides more resistance, observations indicate that lightweight algorithms like as Blowfish outperform DES in terms of efficiency.

8.2 Limitations

The research is limited by a set file size constraint of 5 MB, which limits the scope of the investigation. Furthermore, despite emphasising the benefits of Triple DES over single DES, the paper emphasises the necessity for lightweight algorithms without delving further into their actual implementation and real-world usefulness.

8.3 Future Works

Future research will go deeper into lightweight algorithms such as Blowfish, examining their implementation for greater efficiency across a range of data sizes or for reduced file sizes. The investigation of techniques that are especially suited for smaller data file sizes will be a focus, addressing the reported difference in encryption durations between smaller and larger files. Notably, for practical purposes, this study limits file sizes to 5 MB, highlighting the importance of creating encryption systems that respond to varying data quantities for increased efficiency.

References

Abroshan, H. (2021). A hybrid encryption solution to improve cloud computing security using symmetric and asymmetric cryptography algorithms, *International Journal of Advanced Computer Science and Applications* **12**(6): 31–37.

Ahmad, S. A. and Garko, A. B. (2019). Hybrid cryptography algorithms in cloud computing: A review, *2019 15th International Conference on Electronics, Computer and Computation (ICECCO)*, IEEE, pp. 1–6.

Al-gohany, N. A. and Almotairi, S. (2019). Comparative study of database security in cloud computing using aes and des encryption algorithms, *Journal of Information Security and Cybercrimes Research* **2**(1): 102–109.

Boomija, M. D. and Raja, S. K. (2023). Securing medical data by role-based user policy with partially homomorphic encryption in aws cloud, *Soft Computing* **27**(1): 559–568.

Commey, D., Griffith, S. and Dzisi, J. (2020). Performance comparison of 3des, aes, blowfish and rsa for dataset classification and encryption in cloud data storage, *Int. J. Comput. Appl* **177**(40): 17–22.

Denis, R. and Madhubala, P. (2021). Hybrid data encryption model integrating multi-objective adaptive genetic algorithm for secure medical data communication over cloud-based healthcare systems, *Multimedia Tools and Applications* **80**: 21165–21202.

Elgeldawi, E., Mahrous, M. and Sayed, A. (2019). A comparative analysis of symmetric algorithms in cloud computing: a survey, *International Journal of Computer Applications* **975**: 8887.

Goyal, V. and Kant, C. (2018). An effective hybrid encryption algorithm for ensuring cloud data security, *Big Data Analytics: Proceedings of CSI 2015*, Springer Singapore, pp. 195–210.

Kumar, L. and Badal, N. (2019). A review on hybrid encryption in cloud computing, *2019 4th International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU)*, IEEE, pp. 1–6.

Kumar, S., Karnani, G., Gaur, M. S. and Mishra, A. (2021). Cloud security using hybrid cryptography algorithms, *2021 2nd International Conference on Intelligent Engineering and Management (ICIEM)*, IEEE, pp. 599–604.

Mishra, P. (2023). Advanced aws services, *Cloud Computing with AWS: Everything You Need to Know to be an AWS Cloud Practitioner*, Apress, Berkeley, CA, pp. 247–277.

Mishra, S., Kumar, M., Singh, N. and Dwivedi, S. (2022). A survey on aws cloud computing security challenges & solutions, *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, pp. 614–617.

Mrozek, D. (2020). A review of cloud computing technologies for comprehensive microrna analyses, *Computational biology and chemistry* **88**: 107365.

Rahul, B. and Kuppusamy, K. (2021). Efficiency analysis of cryptographic algorithms for image data security at cloud environment, *IETE Journal of Research* pp. 1–12.

Rayappan, D. and Pandiyan, M. (2021). Lightweight feistel structure based hybrid-crypto model for multimedia data security over uncertain cloud environment, *Wireless Networks* **27**: 981–999.

Saeed, I., Baras, S. and Hajjdiab, H. (2019). Security and privacy of aws s3 and azure blob storage services, *2019 IEEE 4th International Conference on Computer and Communication Systems (ICCCS)*, IEEE, pp. 388–394.

Sajay, K. R., Babu, S. S. and Vijayalakshmi, Y. (2019). Enhancing the security of cloud data using hybrid encryption algorithm, *Journal of Ambient Intelligence and Humanized Computing* pp. 1–10.

Seth, B., Dalal, S. and Kumar, R. (2019). Hybrid homomorphic encryption scheme for secure cloud data storage, *Recent Advances in Computational Intelligence* pp. 71–92.

Talha, M., Sohail, M. and Hajji, H. (2020). Analysis of research on amazon aws cloud computing seller data security, *International Journal of Research in Engineering Innovation* **4**(3): 131–136.

Vennela, G. S., Varun, N. V., Neelima, N., Priya, L. S. and Yeswanth, J. (2018). Performance analysis of cryptographic algorithms for cloud security, *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, IEEE, pp. 273–279.

Viswanath, G. and Krishna, P. V. (2021). Hybrid encryption framework for securing big data storage in multi-cloud environment, *Evolutionary Intelligence* **14**: 691–698.