Hybrid Encryption Scheme for Optimizing Data Security and Latency in Mobile Cloud Computing

Msc Research Project
Msc Cloud Computing

Anay Patil
Student ID: 23195975

School of Computing
National College of Ireland

Supervisor: Prof. Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Anay Patil
Student ID:	23195975
Programme:	Msc Cloud Computing
Year:	2024-2025
Module:	Msc Research Project
Supervisor:	Prof. Sean Heeney
Submission Due Date:	12/12/2024
Project Title:	Hybrid Encryption Scheme for Optimizing Data Security and Latency in Mobile Cloud Computing
Word Count:	6170
Page Count:	19

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:	Anay Patil
Date:	12th December 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):	

Hybrid Encryption Scheme for Optimizing Data Security and Latency in Mobile Cloud Computing

Anay Patil
23195975

Abstract

Mobile Cloud Computing (MCC) The Mobile Cloud Computing (MCC) has evolved as revolutionary technology that provides more computing power and storage space to the mobile devices by using cloud. However, this domain introduced growth of various cloud dependent services which then lead to issues related to data security and privacy within mobile devices with context to limited resources and high latency. AES and other traditional cryptographic approaches are consider secure enough to protect information, while the present issue lies in increasing the speed of data processing while considering the protective measures and working on the latency problem. In regards, the study proposes a new approach of a hybrid encryption model for the improvement of security as well as minimising delay in MCC. A AES encryption scheme is proposed for data encryption, ECC for key encryption and the SHA-2 hashing algorithm to validate before and after encryption and decryption process. Moreover, a new approach is implemented for providing a secure way of exchanging session keys between cloud server and mobile devices using Elliptic Curve Diffie–Hellman (ECDH) key exchange. The proposed method aims to solve issue of secure data transmission by integrating ECDH with the chosen hybrid encryption scheme, with efficient key management and providing optimal performance in Mobile Cloud Computing.

1 Introduction

1.1 Background & Motivation

Mobile Cloud Computing (MCC) appears as innovative model that combines the potential of cloud computing with the features of mobile devices. The power of mobile cloud computing lies beneath its ability to allow users to take advantage of vast computational power and storage capacity of distant cloud servers, which makes mobile applications perform better and solve complex tasks. During research, Al_Janabi and Hussein (2020) found that this fast growth in the mobile cloud services comes with significant security and performance issues. The portable nature of many devices along with the limited processing power, storage capacity, and battery life makes it difficult to implement strong security measures. Even though traditional cryptographic methods provide high standard of security, they come with considerable delay which may ahead impact the usage of mobile application. Considering that, Yang et al. (2020) stated that privacy and protection are the key concerns which are regarded to secure transfer and storage of data in cloud. Maray and Shuja (2022) highlights the need of efficient solution that can make balance between security and performance in various scenarios. Although many cryptographic approaches are proposed in recent times, achieving this balance is always being challenging. AES which has high encryption standards and speed is well suited to use in mobile cloud environment.

Mobile Cloud Computing is one of the subfields in cloud computing, in which issues related to security and delay can play a important role in the effective delivery of cloud services. With the constant increase in the amount of data that is being shared between mobile devices and cloud servers, it is necessary to guarantee that data is encrypted while in transmission without any delay. This study aims for proving a Mixed Advanced Encryption (MAE) strategy which will improve the safety of data in MCC and, in the equivalent time and with minimal delay. The proposed method uses AES to encrypt the data before uploading it on cloud to maintain its privacy and hence no data will be accessed in unauthorized way. For internal operations, Elliptic Curve Cryptography (ECC) is used for key encryption, where efficiency is maintained with increase in security, as ECC uses a small key size for strong cryptographic protection, it is well suited for resource-limited mobile applications. To make the data as accurate as possible and increase overall data security, the SHA-2 is used before encryption and decryption as well. These hash values are compared to ensure that data was not altered during transmission using SHA-2 in order to determine the authenticity of the data. As a contribution in this field, this study also employs the Elliptic Curve Diffie-Hellman (ECDH) key exchange protocol used in exchange of cryptographic keys over untrusted channel. ECDH provides a secure way for sharing keys to make use of a secure channel for communication and also overcomes the disadvantages of implementing traditional PKI. AES for encryption, ECC for key management, SHA-2 for data integrity, and ECDH for secure key exchange together provide an efficient and secure framework for mobile cloud applications that eliminates the problem of data security and latency in MCC.

1.2 Research Objectives

The research objectives of this study are as follows:

1. To design a hybrid encryption scheme which would be incorporated by AES, ECC, SHA-2 as well as ECDH for secure data transmission with minimal latency between mobile cloud environments.

2. Explore the performance of AES encryption for data confidentiality, ECC for secure key management and providing strong cryptographic solution in mobile cloud computing.

3. To find why this proposed multi-layered approach more suitable than the traditional encryption methods for addressing challenges of mobile cloud environments.

1.3 Research Questions

What can be the effect of integrating AES, ECC, SHA-2, and ECDH in a hybrid encryption scheme for optimizing data security and latency in Mobile Cloud Computing?

2 Related Work

The study of AlAhmad et al. (2021) states that Mobile Cloud Computing (MCC) consists of using fixed mobile devices, cloud computing resources and wireless networking elements to deliver the computing services, intensive computational capabilities, and vast storage to the user which is beyond what a mobile devices can provide. It uses cloud servers for processing the data and storing it, and the mobile device is used to simply call for resource intensive processing from the cloud. Panigrahi et al. (2021) in their study says that the architecture of MCC typically consists of three layers: portable computers, wireless communication channels, and networks of cloud respectively. Mobile terminal devices are end points of users where even by using WWANs like 4G, 5G or WLAN like Wi-Fi users can get connected to cloud server where computational resources and applications are hosted. The advantages of MCC include the betterment of the services to make it more suitable for the mobile environment, cost savings by resources sharing and unified infrastructure, improvement in scalability options to deliver services and flexibility in the service deployment. However, the study given by Aldmour et al. (2021) involves a number of challenges in MCC like, high latency, concern in data security, and limited use of mobiles. Latency challenges mostly occur due to the delay in accessing either the network or the cloud, which may be due to network traffic or distance respectively. Concerns regarding data security like loss or theft of data during transmission or storage, breaches, eavesdropping, or unauthorized access. Moreover, the battery life of mobile devices may be short, their processing capability may not be robust, and the bandwidth available for using MCC services also may be limited. Solving these challenges needs advancements in safe data transfer, optimization of the encryption method, and optimization of the latency reduction methods including the edge computing where most of the data processing is done near the user.Mobile Cloud Computing (MCC) is expected to grow further, followed by the upcoming mobile technologies and cloud computing features to support real time, bandwidth intensive services such as video streaming, mobile gaming, AI activities and other services, which can be powered by a solution that provides optimized security and performance.

2.1 Encryption Techniques in Data Security

2.1.1 Symmetric Encryption

The first study which is given by Sinha (2022) suggest a study focusing on the application and evaluation of the AES block cipher, which is a type of the symmetric encryption algorithm that has been adopted by e-business and e-commerce environments for converting or encrypting important data. The research focuses on the security risks of using AES in ECB mode, the message content has repeated data blocks as it produces an identical ciphertext block for the corresponding plaintext blocks, compromising the confidentiality of the data. The study also shows that if the message space is less or constrained and only consists the sequences of distinct data blocks, the ECB mode is never compromised. The approach also involve the general introduction for cryptology, mathematical definitions and operational modes of AES, and addressing the questions like, conflict between security and performance in encryption especially in applications. The results demonstrate that one should pay attention while selecting encryption modes based on data characteristics, as some vulnerabilities may occur simultaneously. The study has its own disadvantages because it does not use complex practical applications for validating the obtained theoretical conclusions and it also does not consider other secure modes such as CBC and GCM, which can minimize the described problems.

Another study discussed by Akogun (2020) presents an improved multilevel cryptographic model for cloud storage using RNS with an enhanced AES-256 encryption algorithm. The research is proposed to focus on the upcoming issues of data security and privacy in cloud computing systems, in which data is collected by individuals and business entities. This proposed system first encrypts all the data before storing it in cloud by using Residue Number System (RNS) for forward as well as reverse data conversion with moduli set which are specially designed to integrate with AES-256 so that it can offer high level of cryptographic strength. The system proposed in this research was implemented on the Heroku cloud platform as a SAAS, with encryption/decryption times, throughput, key-size, theoretical cryptographic strength, and attacks tested. According to the results of experimentation, the proposed multilevel architecture makes the security better and provided increased throughput when the size of the file increases and takes less encryption time when compared to other methods. According to the researcher, the most difficult task was to balance the level of security and the time taken to encrypt and decrypt data, especially when the amount of data increasing rapidly. However, there are issues that can arise, such as scalability and feasibility of application with other cloud services and with real-time data processing, due to the added computational burden built into the multilevel scheme used in the approach.

Hamdy (2021) has proposed the idea of employing image steganography as the first layer and AES encryption as the second layer to improve the measures for data security in multimedia environment. The study focuses on the fact that multimedia has widespread, and thus, there is a need to develop the corresponding secure communication system. Into this current approach, steganography is used to send messages inside of a covered medium to keep its existence unknown. On the other side, AES is used for transforming messages into a ciphertext hence, safe against unauthorized attempts. The results provide in numerical form shows that the double-layer scheme is better at achieving optimal level of security regarding in case of private data. However, the difficulty was to provide enough security layers while making the system practical and easy to implement, especially in various multimedia messaging modes (MMMs). Even though the scheme offers a high level of security, it also has some disadvantage. These includes the computational complexity and the impact of the chosen cover medium on the steganographic method.

A design for implementation of Data Encryption Standard (DES) encryption algorithm based on high throughput reconfigurable hardware architecture is presented by Zeebaree et al. (2020) which uses pipeline technique for increasing the speed of operation. The study focuses on minimizing the change in DES so it can be easily be used for encrypting the data on the internet and other application by accelerating through hardware device. The proposed design was developed for the Spartan-3E (XC3S500E) family FPGAs; achieved the encryption and decryption rates of 10,688 Mbps at the clock frequency of 167.448MHz and 167.870 MHz.The architecture is pipelined and increases the overall throughput of the system while also securing DES algorithm. One major technical challenge was making sure that DES could operate efficiently on high-speed platforms in order to protect cryptographic usage. From the results, however, it is clear that compared to other conventional implementations of the algorithm, the proposal performs better but the implementation itself has limitations since DES is considered less secure than other algorithms such as AES. However, apart from limitations of hardware-specific dependence, feasibility has never been an issue in designing the system itself.

2.1.2 Asymmetric Encryption

The asymmetric encryption has the best utilization in IoT applications because there are many devices demanding efficient cryptographic schemes to protect their data. Among the techniques that researchers in the study on enhanced security, authentication besides minimal computational overhead have adopted include Elliptic Curve Cryptography (ECC) and ElGamal encryption system. Preferably ECC is to be used for resource-limited edge node in an IoT environment because it has small key size and high operating speed. These models sufficiently provide lightweight encryption and along with data integrity during transfers from source to destination. Generally, ECC and ElGamal encryption combined gives the most efficient solutions for the increase in security without drastic resource spend.

A secure hybrid encryption technique has been discussed by Kumar and Kumar (2024) which protects the authenticity and integrity of the data. The study utilizes AES for symmetric encryption and ECC for asymmetric encryption, accompanied by Message Authentication Code (MAC) for data integrity and authentication. The primary goal of the scheme is to address the high demand of computational resources for executing cryptographic operations on IoT devices and integration, as well as the exposure of data during transmission process through the cloud servers. The proposed method encrypts the data with AES and ECC and store the encrypted data in cloud servers to avoid the storage overhead and only allow access when the MACs are authenticated. Experimental results also show that the proposed approach offers better security and is faster than the existing methods. Nevertheless, there are some limitations like, the paper does not investigate the network deployment scenarios, which are crucial for the scheme applicability in real life, such as the variability of network conditions, as well as other scalability factors.

A low overhead encryption solution for data protection and authenticity in IoT smart goods transport systems has been proposed by Joglekar et al. (2020), where sensor nodes are also edge nodes which collect and pre-processes an object's data before sending the data to cloud or server for storage and further analysis. Since edge nodes are resource-limited, practical choice of encryption scheme is the Elliptic Curve Cryptography (ECC), specifically for the Elliptic Curve ElGamal encryption scheme, due to its high authentication and data protection features with managable computation load. The key issue this approach is targeting is, how to securely send data to the cloud where most of the computations and storage are typically resident while the edge nodes are constrained with low processing power, memory and energy. With ECC and ElGamal algorithms, encryption and decryption can be done efficiently, considering the qualities which small IoT devices can support. The results also show that the proposed scheme accurately preserves data integrity and security during data transmission. However, its limitation is the conducts a limited number of scenarios with huge networks, and it doesn't consider the different real-world environmental factor including the fluctuations of different IoT devices throughput and the network performance in terms of security and efficiency.

2.2 Optimizing Latency and Efficiency in IoT

The survey presented by Kumaran and Sasikala (2021) provides a systematic overview of various latency reduction strategies in MEC systems and offloading policy using Deep Reinforcement Learning(DRL) with optimized solutions within 5G network. In the paper describes how MEC, associated with cloud infrastructure and User Equipments(UE) can minimize latency, since data processing happens nearer to data origin such as IoT sensors and not in a remote cloud facility. The study presents performance enhancement approaches for the MEC system using efficient offloading heuristics approach that improves terminal processing while reducing user delay. As one of the key issues discussed, including processing of tasks, there is a critical issue on how to determine which tasks should be pushed to the edge and which should remain in the device or be processed at a device level, especially if it is in IoT and 5G systems. The findings prove that the use of deep learning and DRL approaches, significantly decreases latency and increases the efficiency of the general system. However, there are concerns raised in this study regarding scalability and real time decision making especially when the size of the network, that is the number of users and interconnected devices is very large. This makes practical implementation of some techniques to be quite limited in current large-scale MEC systems.

The study by Lakhan et al. (2021) presents a new approach which is Multi-Layer Latency Aware Workload Assignment Strategy(MLAWAS)y to mitigate the problems of latency-sensitive workload in E-Transport IoT applications (including E-Bus, E-Taxi and autonomous vehicle) in distributed cloudlet-based cloud networks. The major problem is the communication delay, round-trip delay and migration delay that takes place during workload running. Actual workloads can also be delayed due to the assignments of computational workloads, MLAWAS aim to eliminate such delays through the application of Q-Learning and iterative approaches. It also involves workload migration and virtual machine (VM) migration for dynamic run time requirements such as overloading and overheating. Since the performance metrics have been defined and quantified, a proof has been provided by means of simulations of the effectiveness of the proposed MLAWAS in terms of average response time for the considered problem with respect to the previous solutions. However, the approach has issues with scalability, especially when use in very large and fast evolving networks where the real-time workload allocation may be very costly and complicated in large networks making it challenging to perform in very large implementations.

Another study proposed by Nagu et al. (2023) describes an ultra-low latency communication technology for Augmented Reality (AR) for Mobile Edge Computing (MEC) systems with focus on reliability, low communication latency. The system is proposed to address main AR workloads on onboard devices and so as to reduce the adaptation demand for customer equipment(CE) and increase the throughput. One of the issues is great overhead and time delay due to the use of wireless media, making it difficult to determine the best payload. This is handled by the use of Bayesian networks for the modeling of code splitting and the dependency models of tasks and a Weighed Particle Swarm Optimization (WPSO) approach to the load of communications. The proposed hybrid WPSO method enhances different performance measures and reportedly minimizes PSF in MEC-enabled AR contexts than other communications technologies in diverse network settings.

Lastly, study given by Caon et al. (2021), proposes a very low latency Earth Observation (EO) satellite onboard data handling architecture to address increasing demand for EO products through improved data compression, encryption as well as efficient management. The architecture which has been developed under the European Union Horizon 2020 EO-ALERT project, address challenges existing in the satellite data chain systems by using consumer available COTS devices for processing data on board, without any compromise in performance. Organized components of the system are the subsystem responsible for CPU scheduling, data compression, data encryption, and data handling of optical as well as SAR data during their processing and transmission. The proposed architecture shows very low latency and therefore the end users will receive the EO products in less than 5minutes. Despite the highly encouraging results, there are factors limiting the method, including the ability to scale up the designed system for other satellite types or operation settings, or the capability of commercial devices to process large scale data or highly complex algorithms under these conditions.

3 Methodology

3.1 Research Flow:

The proposed research method focuses on development and implementation of hybrid encryption technique that optimises data security and latency in domain of Mobile Cloud Computing (MCC). The method consists of Advance Advanced Encryption Standard (AES), Elliptic Curve Cryptography (ECC), Secure Hash Algorithm-2 (SHA-2), and Elliptic Curve Diffie-Hellman (ECDH) inside the mobile-cloud ecosystem. The methodology flow is as follows:

1. **Analysis:**A research on existing encryption techniques based on mobile cloud computing is done to identify the challenges and drawback of those proposed system, well discussed in the Related work Section of this report.
2. **Design:**A hybrid encryption model of AES + ECDH is developed, combining the

cryptographic algorithms and cloud resources to further enhance the security of the data and reduce the computational overhead of the mobile devices.

3. **Implementation:**The mobile Application name SecureData is deployed on Azure Cloud platform which integrates the hybrid encryption model so that the data can be store in a secure manner and less communication delay.

4. **Evaluation:**The prosed method is tested and compared with other traditional encryption method like RSA+Blowfish and RSA+AES,, inside the cloud environment.

3.2 Cloud Integration

To achieve the objectives like scalability, security, and performance optimization, the integration of cloud is very important for this project. By utilizing the services provided by Azure Cloud, the proposed research contributed in solving challenges like secure data transmission and storage, and computational offloading of devices with limited resources. The use of Cloud provided:

1. **Scalability:**The infrastructure of Azure makes sures that the system handles various type of data and requests by user without degrading the performance. Even the demand vary, the flexibility of scaling the resources in a dynamic way is possible due to the serverless architecture of Azure App Service.

2. **Security:**The security features like, encrypting data at rest or in transit, supports the hybrid encryption method. The sensitive data that is related to the users is encrypted and then stored in the SQL database, while the secure API endpoints prevents risk like unauthorized access.

3. **Centralized Management:**Managing the backend infrastructure is easy with Azure platform, which consists of handling the data and updating the system easily. By monitoring the cloud activities, tracking performance and detecting threats can be done.

4. **Reliability:**The Azure architecture has high availability due to which it provides a continuous service and reduces the risk of system downtime. It is more reliable, as the data can be accessed and operation can be performed with the help of backup system of Azure SQL and App Services

3.3 Data Collection

The experiments involves encrypting and decrypting files of various sizes, ranging from 2 KB to 97 KB, so that the latency and performance of the system can be measured. Cloud-based storage and retrieval times are recorded to check the efficiency of the proposed hybrid method. Metrics such as time taken to encrypt or decrypt a file and data transfer speeds were then compared with that of conventional methods using RSA + Blowfish and RSA + AES

3.4 Hybrid Encryption

3.4.1 AES Encryption

AES (Advanced Encryption Standard) is a commonly used algorithm that uses a symmetric key for encrypting the information, this makes information privat and secure and converts it into a text that cannot be understood. It uses a data block size of a fixed size (128 bits) and the different between keys are of 128, 192, and 256 bits to improve the security. According to the architecture proposed in this paper, AES encrypts the data before it can be stored in the cloud storage. The algorithm includes substitution-permutation techniques to encrypt plaintext into ciphertext and is resistant to any cryptographic assault. Due to its symmetric nature, it is amongst the fastest for encrypting and decrypting and is best suitable for mobile applications. AES achieves low latency in the MCC environment by using optimized key expansion along with efficient block processing. By making use of AES, this system focuses on the protection of data confidentiality and, at the same time does not affect the computational capacity of mobile devices.

3.4.2 ECC Key Exchange

Elliptic Curve Cryptography (ECC) is an enhanced version of digital asymmetric encryption which is designed for exchanging keys efficiently. ECC is based on the theory of elliptical curves in finite fields, such that an highly secure system can be achieved through the usage of relatively shorter keys than those used in RSA. As in the current corrected hybrid scheme. ECC is used for securely transferring the encrypted key between the mobile device and the cloud server. It has short keys which has enhanced ability to reduce computational overhead and latency, making it suitable for MCC. Despite the fact that keys are intercepted during transmission, they can also be made more secure via ECC. Thus the whole process is the best for encryption as it gives faster and better data protection and acquisition of resources, most suited for mobile environments.

3.4.3 SHA-2 Hashing

SHA-2 refers to a family of secure hashing algorithms for the generation and verification of the same-sized fixed value hash value for data. Before encryption, data is passed through SHA-2, and after it stays in-reception for decryption; it creates hash extensions that compare with each other and indicate that the data merely existed in its unaltered condition while being transmitted. This the makes it rather difficult for attackers to create similar hashes by spoofing the same hash for two different inputs, thus making the pegging much stronger. This has proven to be faster than all the rest on the Small-Mobile and cloud-integrated platforms, thus proving its scalability (CodeSigning; 2024). Therefore, the inclusion of SHA-2 in the hybrid scheme guarantees the complete integrity and verification of transactions toward end users, thereby instilling confidence with which the user receives data from the data source.

3.4.4 ECDH for Key Exchange

The Elliptic Curve Diffie-Hellman (ECDH) algorithm is a method used for cryptographic key exchange protocol. It uses elliptic curve mathematics to allow two parties to agree on a shared secret between. ECDH in the hybrid encryption allows mobile devices and cloud servers to obtain the same encryption key without the need to transmit it, which prevent the key from getting expose to hackers. The process used both the client and the server to create private and public key pairs, and then using mathematical computation to be done at the server side to generate secret key. The application of elliptic curves by ECDH offers a more secure defence with reduced key sizes, hence lowering computations and latency. This efficient and secure key exchange method is important to the performance of the hybrid system, allowing extensive secure communication while considering the computational limitations of smart mobile devices.

4 Design Specification

The architectural layout of hybrid encryption model and entire proposed system is illustrated in Fig. 1 and Fig. 2 respectively, and includes multiple elements that make the actual storage and encryption of data secure.

4.1 Hybrid Model Design

Looking at the architecture of the encryption mechanism, at the center of system implementation is the Key Generator which creates the necessary cryptographic keys. It is to be noted that the establishment of a secret key between the two parties is carried out through the ECDH Mechanism based on elliptic curve cryptography. This shared key is then taken into the SHA hashing function to produce the digital signature and the encrypted key.

AES Cryptographic module has been used in this solution to encrypt the data, so as to ensure the data is only accessible by authorized individual. The resulting secured data and key are then stored, which can be widely used as a backend for storing other important data securely. Based on the proposed architecture, the general approach used to protect the data is well designed and systematically makes use of commonly used cryptographic tools including ECDH, SHA, and AES. When integrated together, these components allow the system to provide data integrity, confidentiality and availability, which makes it ideal for applications where data security is the most priority.

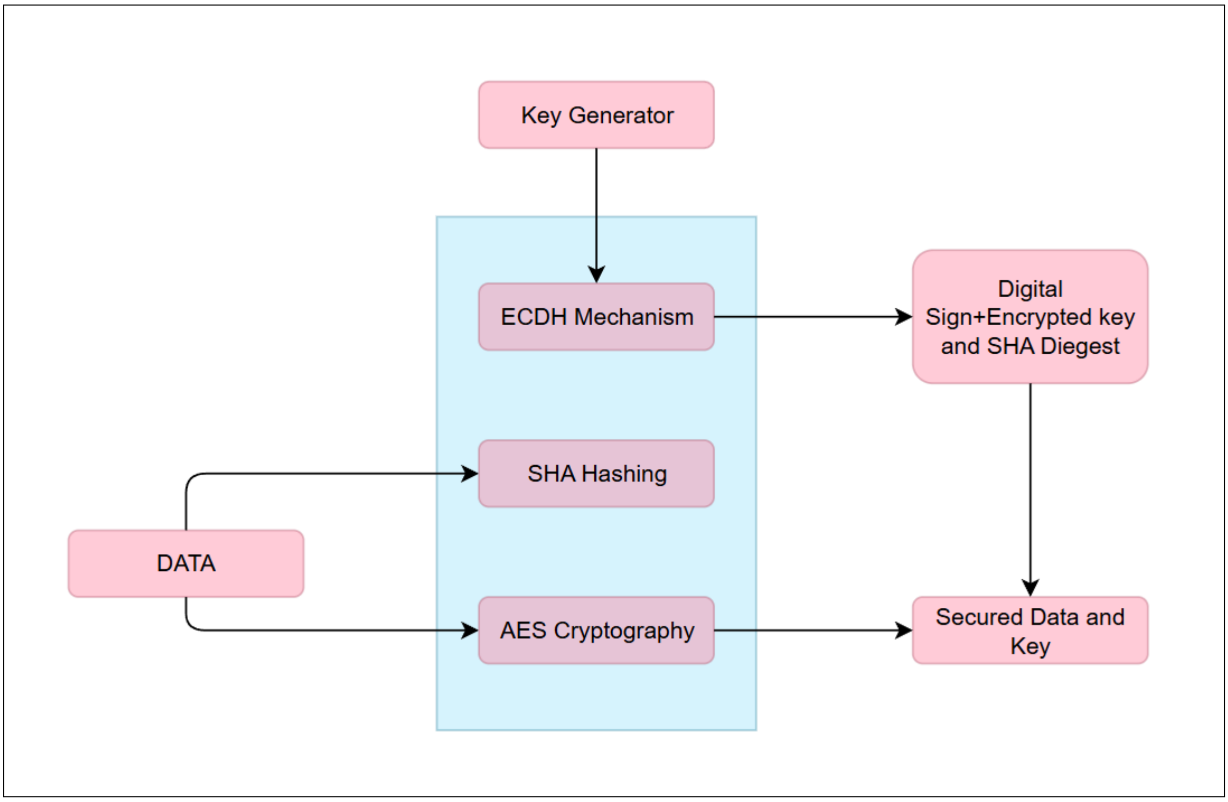

Figure 1: Encryption Architecture

4.2 Proposed system Architecture

The proposed hybrid encryption system or mobile cloud computing (MCC) consists of 3 main layer:

1. **Client side:** At the client of the design, user interaction and data preparation is handled. User interacts with the interface to register, login, upload, download, and view the files. The hybrid module secures the files using combination of encryption algorithms like ECC for public key encryption, SHA-2 for data integrity, AES for symmetric encryption,and ECDHA for exchanging key securely. The Background task handler manages uploading and downloading in the background. The Data Logger collects the performance metrics like encryption time, file size, transfer time.

2. **Network Layer:** The network layer ensures that the transmission between client and cloud is secure by using a Web API (RESTful APII) service. This service uses HTTP protocol for transferring files securely.

3. **Cloud Layer:** At the server part of the project, the encrypted file is received and processed by the Receiver Module and stores the file securely in the Azure SQl database. When a encrypted file is requested back, it is Decrypted by using the key generated to ensure the file is accessed by the original user.

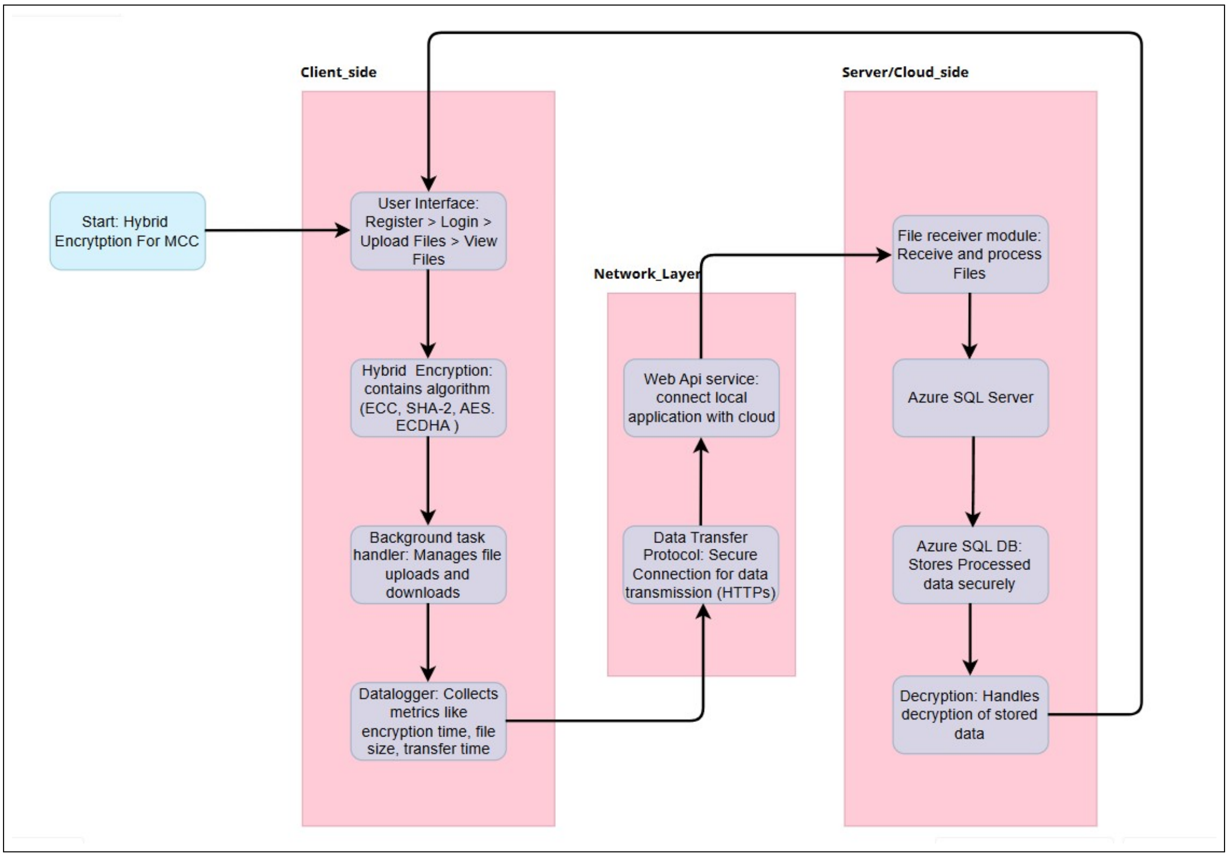
Client_side
Network_Layer
Server/Cloud_side
Start: Hybrid Encrytption For MCC
User Interface: Register > Login > Upload Files > View Files
Hybrid Encryption: contains algorithm (ECC, SHA-2, AES. ECDHA)
Background task handler: Manages file uploads and downloads
Datalogger: Collects metrics like encryption time, file size, transfer time
Web Api service: connect local application with cloud
Data Transfer Protocol: Secure Connection for data transmission (HTTPs)
File receiver module: Receive and process Files
Azure SQL Server
Azure SQL DB: Stores Processed data securely
Decryption: Handles decryption of stored data

Figure 2: System Architecture

5 Implementation

5.1 Tools and Technologies

Tool/Technology	Purpose
Android Studio	Development environment for building the CrypographyApp mobile application.
Android OS	Platform for running the mobile application.
Azure Cloud	Cloud service for storing and managing encrypted data.
Azure App Service	Hosting platform for the API middleware used in the application.
Azure SQL Database	Cloud-based relational database for storing application data.
ASP.NET Framework	Framework for developing the middleware (Web API layer).
Visual Studio 2022	Integrated development environment (IDE) for backend and middleware development.
AesCryptoServiceProvider	Library used for implementing AES encryption in the application.
ECDiffieHellmanCng	Library used for secure key exchange using ECDH algorithm.
SHA256Managed	Library for implementing SHA-256 hashing to ensure data integrity.
Java	Programming language for developing the Android application.
C#	Programming language for implementing the middleware using ASP.NET.
RESTful API	Communication protocol between the mobile app and cloud services.

Table 1: Tools and Technologies

Table 1 shows tools and technologies in this study which ensure security, smooth application performance and all.

5.2 Implementation of SecureDataApp: A Secure Mobile Application

The presented hybrid encryption approach was deployed on virtual Android device application called SecureDataApp, which aims to provide the confidentiality and integrity of data processing within the Mobile Cloud Computing system. The application was build using:

1. The application was built using the Android operating system, with a help of supporting application known as the Android Studio. The interface includes the registration, login, and dashboard of the platform, user can upload files, and view or download the files.
2. The data uploaded through the application is stored on Azure Cloud which is the backend that has been developed using multiple Azure services.
3. An API has been deployed on Azure App Service that acts as the intermediate platform for the mobile application and, an Azure SQL database for effectively managing the data.
4. The middleware application was built using the ASP.NET framework in Visual Studio 2022 that uses AesCryptoServiceProvider for AES and ECDH for key exchange.SHA256manage, which is a element of the SHA-2 hashing family has also been used to check the integrity of data and determines weather or not the data has been tampered.
5. The implementation of Web API layer consists of the UserController class which is under the controller package, where the main operations include user management and secure data transfer.

This architectural approach allows smooth integration of the mobile application with the cloud services and offers security integration as per the correct security standards. The cryptographic framework is designed within the application and the provided time and computational overhead are both low, making it ideal for mobile deployment. The design of SecureDataApp showcases how the hybrid encryption scheme can be used effectively for secure data exchange, efficient storage and integrity checks with newly developing cloud technologies, all integrated into an Android application.

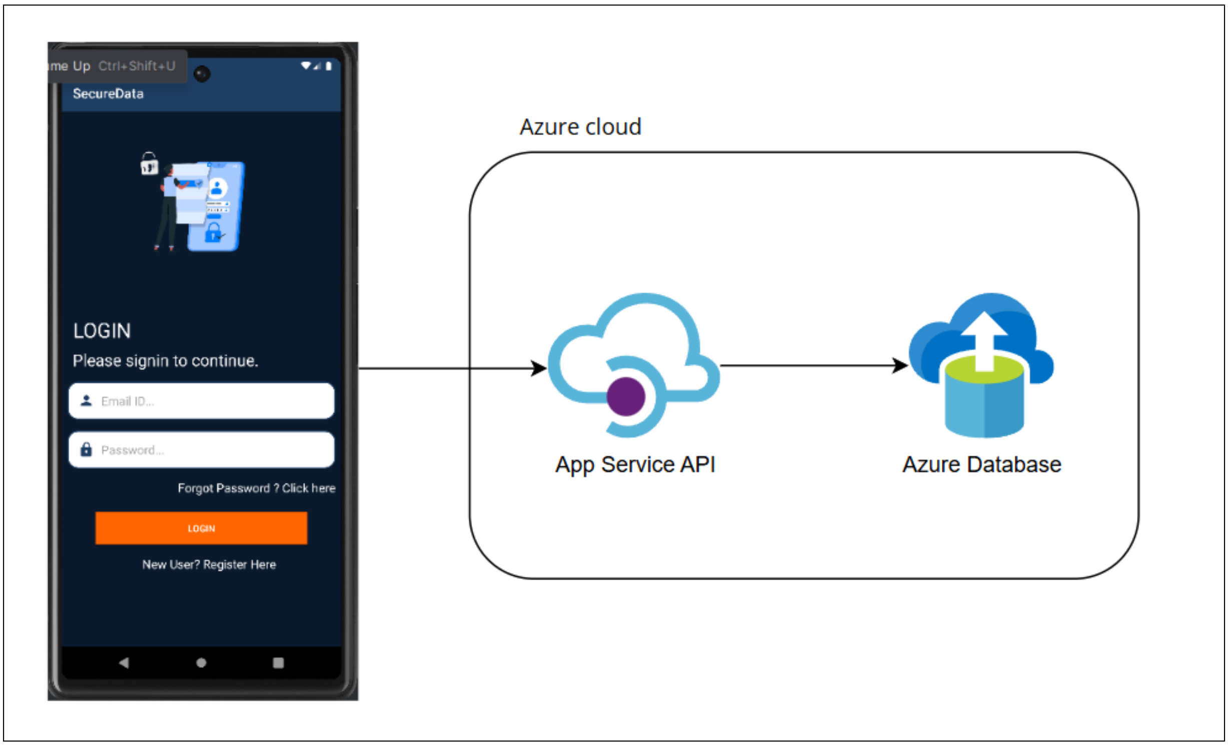

Figure 3: Azure Integrated Cryptography Application

The above Fig. 3 shows a mobile cryptography application with interconnection with an Azure-based cloud system. On the left side, a mobile device is shown which is running the developed cryptography application on it, and is responsible for collecting the input given by users, as well as handles the encryption and decryption process of the data. Next to it, the App Service API denotes a serverless computing environment, within the Azure cloud environment. This API servers as a interface between the mobile application and the actual Azure Database where data will be stored. The interaction between the mobile application and the App Service API is carried through secure means, ideally through the use of the Hypertext Transfer Protocol Secure or HTTPS. Azure Database shown on the right side stores the data that has been processed by the application. This database is responsible for secure storage of the keys, metadata, and other essential information, which can become necessary for a mobile application. In general, the design of the entire solution shows the usage of Cloud services and highlights the ability of Azure Cloud to implement data storage and protection systems, ensuring the mobile application acts as a reliable component for the user environment.By storing and processing the data of user in cloud, the mobile application can simplify it work and concentrate on performing the cryptographic operations with the support of cloud infrastructure, reducing the problems of the application like availability, scalability, and security of user data. Such architectural pattern matches the recent trends in mobile application development, since the local encryption and cloud-based services can create a very secure yet highly available solution for applications which involve processing of very secure and protected data.

6 Evaluation

6.1 Experiment Setup

The below table 2 shows the elements used for this project:

Aspect	Description
Hardware	Android 6 Pixel device for testing the CrypographyApp functionality. ASUS Tuf with AMD processor used for developing and testing.
Software	SQL Server Management Studio for managing the database Android Studio for mobile app development. Visual Studio 2022 for middleware and Web API development.
Cryptographic Libraries	AesCryptoServiceProvider for AES encryption and decryption. ECDiffieHellmanCng for secure key exchange. SHA256Managed for implementing SHA-256 hashing.
Cloud Services	Azure Cloud for encrypted data storage. Azure App Service for hosting APIs and communication layers. Azure SQL Database for structured data storage.

Table 2: Experiment Setup Table

6.2 Experiment/ Case Study 1

From the table 3, it can be seen that as the size of the file increases, the encryption time of AES + ECDH gradually increases, which is the proposed algorithm. But when compared with the RSA + Blowfish and RSA + AES proposed schemes, and the difference is relatively small.As the size of the file grows, the encryption time taken by RSA + Blowfish and RSA + AES increases more significantly. This indicates that the proposed approach, where AES is used for data encryption and ECDH is used for key exchange, has better scalability in cases with larger file sizes. When compared with the encryption time of the other algorithm, the proposed algorithm is more favorable for the larger data in the mobile cloud computing environments, although is more time consuming in case of small files. Graphical representation of the results can be seen in Fig. 4.

File Size / Scheme	RSA & Blowfish	RSA & AES	My Proposed Algorithm (AES+ECDH)
2 KB	21.7	22.7	512
21 KB	78.2	97.2	521
38 KB	156.2	161.2	526
97 KB	361.5	410.8	543

Table 3: Encryption Time Results in (ms).

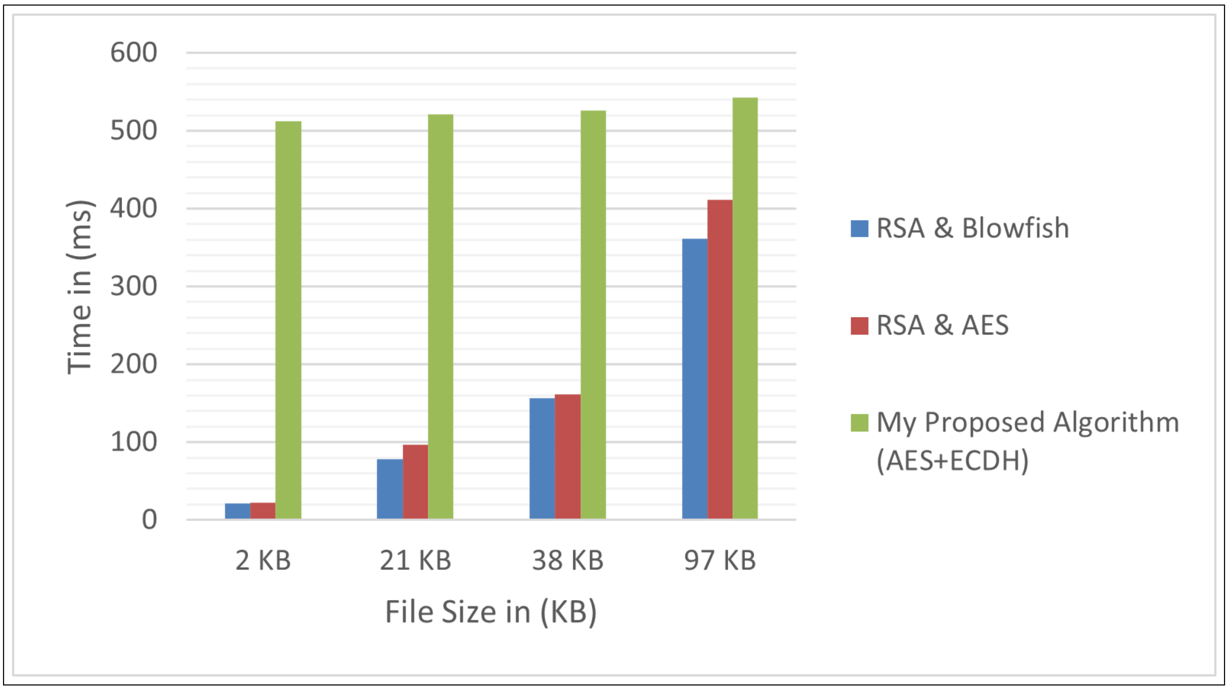

Figure 4: Encryption time result chart

The results of decryption time are shown in table 4, shows that the proposed algorithm AES + ECDH has much lower decryption time than RSA + Blowfish as well as RSA + AES for all the file sizes. For example, for a file size of 97KB the proposed approach requires only 24ms, when compared with RSA + Blowfish and RSA + AES , time taken is 413.2 ms and 389.1 ms respectively. This shows that even if the file size increase, the decryption time of the proposed system still remains less, which shows the efficiency of the method. This highlights that the hybrid approach of AES + ECDH is even more scalable, with better performance in context with decrypting large size files. Graphical representation of the results can be seen in Fig. 5

File Size / Scheme	**RSA & Blowfish**	**RSA & AES**	**My Proposed Algorithm (AES+ECDH)**
2 KB	76.1	77.9	8
21 KB	171.2	155.2	10
38 KB	196.2	187	12
97 KB	413.2	389.1	24

Table 4: Decryption Time Results in (ms).

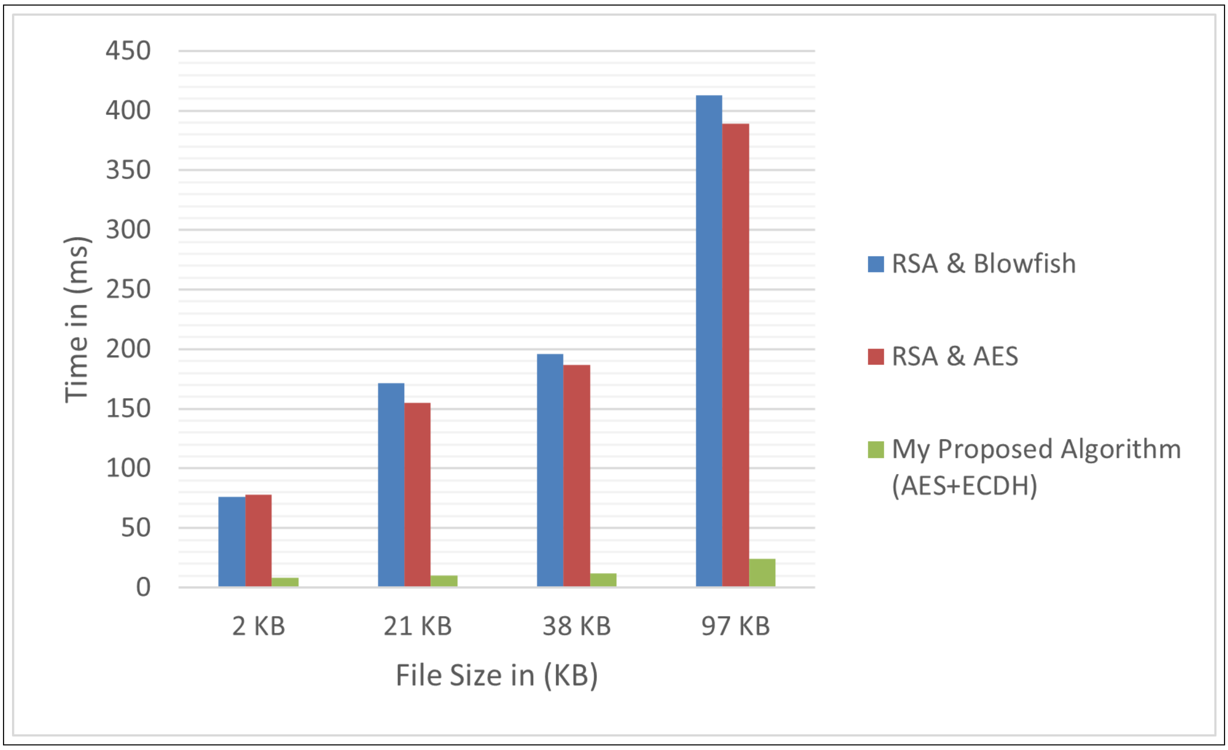

Figure 5: Decryption time result chart

6.3 Discussion

The findings of the research prove that the proposed hybrid encryption scheme combining AES for data encryption and ECDH for secure key exchange shows better performance concerning the encryption and decryption as compared to RSA + Blowfish and RSA + AES encryption schemes. Whereas its encryption time increases slightly with small files as suggested in the proposed approach, its performances surpass the other alternatives as the file sizes increases and hence making it ideal for large data sets. In addition, the number of decryption time is surprisingly low for all the sizes of the file which shows the effectiveness of the proposed approach. The findings indicate that incorporating AES and ECDH achieve a good security to computation trade off, thereby addressing the latency issues synonymous with mobile cloud computing. It is denoted that ECDH effectively improves the security for key exchange with low cost overhead, while AES efficiently performs encryption with high speed. Also, the integration of the SHA-2 hashing algorithm maintains data integrity; thus, the scheme fits in well with any dynamic application that needs a secure data relay system. In sum, the analysis supports the effectiveness of the proposed hybrid approach, especially in processing large-scale data operations and confirms the adoption of the proposed solution as an efficient method for data protection and latency in mobile cloud computing.

7 Conclusion and Future Work

At last, the proposed hybrid encryption scheme, which is the CryptographyApp for MCC, uses AES, ECC, Diffie-Hellman (ECDH), and SHA-2 for adaptive and efficient protection, optimally balancing performance and latency. The integration of Azure cloud services allows the user to maintain the security of the data during upload and download. It provides a user-friendly interface to carry out important functionalities like registration, uploading, and downloading the files. The users are secured by the cryptographic libraries, such as AesCryptoServiceProvider and ECDiffieHellmanCng, which encrypt the key and take care of the high-level encryption process. Data integrity is achieved efficiently and effectively through SHA256. It addresses safety issues in very minimalist MCC environments, delivering a large number of multimedia data to multiple devices of low computational capability.

However, there are several trends for the further upgrade that can be considered. The first possible enhancement is to enhance the encryption a step further by decreasing the computational resources load on the mobile device. This can be achieved by using a lightweight cryptographic algorithms in the system or offloading the computation to the device's cryptographic accelerator hardware. Furthermore, the investigation of how blockchain technology could be used in the system for further improvement of security layer, that guarantees the immutability of the records and offers validation of file transactions in a decentralised manner. Also, the future work could be focusing on adapting the real time security threat detection by using AI based anomaly detection system. Moreover, the application can be extended by using cross-platform support, making it available for not only the android, but all the mobile operating systems. Lastly, enhancing the mobile cloud security standard can benefit in terms of compatibility and adaptability for the proposed framework and redevelop it into a flexible model which can be integrate with various cloud platforms. By expanding to these areas, SecureDataApp can become a more solid and scalable solution which could improve protection on mobile cloud data even in complex environments.

References

Akogun, D. N. (2020). *Enhancing data security in cloud storage using residue number system and advanced encryption standard*, Master's thesis, Kwara State University (Nigeria).

AlAhmad, A. S., Kahtan, H., Alzoubi, Y. I., Ali, O. and Jaradat, A. (2021). Mobile cloud computing models security issues: A systematic review, *Journal of Network and Computer Applications* **190**: 103152.

Aldmour, R., Yousef, S., Baker, T. and Benkhelifa, E. (2021). An approach for offloading in mobile cloud computing to optimize power consumption and processing time, *Sustainable Computing: Informatics and Systems* **31**: 100562.

Al_Janabi, S. and Hussein, N. Y. (2020). The reality and future of the secure mobile cloud computing (smcc): survey, *Big Data and Networks Technologies 3*, Springer, pp. 231–261.

Caon, M., Ros, P. M., Martina, M., Bianchi, T., Magli, E., Membibre, F., Ramos, A., Latorre, A., Kerr, M., Wiehle, S. et al. (2021). Very low latency architecture for earth observation satellite onboard data handling, compression, and encryption, *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, IEEE, pp. 7791–7794.

CodeSigning (2024). Hash algorithm comparison. Accessed: 2024-12-10.
URL: *https://codesigningstore.com/hash-algorithm-comparison*

Hamdy, A. (2021). Image processing and aes for secure communications.

Joglekar, J., Bhutani, S., Patel, N. and Soman, P. (2020). Lightweight elliptical curve cryptography (ecc) for data integrity and user authentication in smart transportation iot system, *Sustainable Communication Networks and Application: ICSCN 2019*, Springer, pp. 270–278.

Kumar, D. and Kumar, M. (2024). Hybrid cryptographic approach for data security using elliptic curve cryptography for iot', *International Journal of Computer Network and Information Security (IJCNIS)* .

Kumaran, K. and Sasikala, E. (2021). Learning based latency minimization techniques in mobile edge computing (mec) systems: A comprehensive survey, *2021 International conference on system, computation, automation and networking (ICSCAN)*, IEEE, pp. 1–6.

Lakhan, A., Dootio, M. A., Groenli, T. M., Sodhro, A. H. and Khokhar, M. S. (2021). Multi-layer latency aware workload assignment of e-transport iot applications in mobile sensors cloudlet cloud networks, *Electronics* **10**(14): 1719.

Maray, M. and Shuja, J. (2022). [retracted] computation offloading in mobile cloud computing and mobile edge computing: Survey, taxonomy, and open issues, *Mobile Information Systems* **2022**(1): 1121822.

Nagu, B., Arjunan, T., Bangare, M. L., Karuppaiah, P., Kaur, G. and Bhatt, M. W. (2023). Ultra-low latency communication technology for augmented reality application in mobile periphery computing, *Paladyn, Journal of Behavioral Robotics* **14**(1): 20220112.

Panigrahi, C. R., Sarkar, J. L., Pati, B., Buyya, R., Mohapatra, P. and Majumder, A. (2021). Mobile cloud computing and wireless sensor networks: A review, integration architecture, and future directions, *Iet Networks* **10**(4): 141–161.

Sinha, R. (2022). *Symmetric key cryptology: Implementation of aes block cipher*, Master's thesis, Lamar University-Beaumont.

Yang, P., Xiong, N. and Ren, J. (2020). Data security and privacy protection for cloud storage: A survey, *Ieee Access* **8**: 131723–131740.

Zeebaree, S. et al. (2020). Des encryption and decryption algorithm implementation based on fpga, *Indones. J. Electr. Eng. Comput. Sci* **18**(2): 774–781.