National College of Ireland

Data security using a hybrid cryptographic approach in mobile cloud computing

MSc Research Project
Cloud Computing

Sharannya Nair
Student ID: x21154520

School of Computing
National College of Ireland

Supervisor: Vikas Sahni

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Sharannya Nair
Student ID:	x21154520
Programme:	Masters in cloud computing **Year:** 2022
Module:	MSc Research Project
Supervisor:	Vikas Sahni
Submission Due Date:	15/12/2022
Project Title:	Data security using hybrid cryptography approach in mobile cloud computing
Word Count:	6532 words. **Page Count** 20

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: Sharannya Nair

Date: 14 December 2022

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):	

Data security using a hybrid cryptographic approach in mobile cloud computing

Sharannya Nair
x21154520

Abstract

Mobile Cloud Computing is an approach that integrates mobile and cloud computing to offer high-quality services and experiences to mobile users and cloud consumers. However, the drawback of mobile platforms is their constrained processing and storage capabilities. The other inherent issues with mobile cloud computing include security risks, cloud data privacy, intrusion, and other security risks. Consequently, Mobile Cloud Computing (MCC) subscribers' acceptance and growth are hampered (MCC). However, on the platforms utilized for mobile cloud computing, hybrid signature based cryptography guarantees secrecy, authentication, non-repudiation , and integrity. The data encryption approach described in this paper uses cloud storage and a hybrid cryptographic mechanism known as the ASE algorithm. AES algorithm is used to encrypt data, whereas ECC carries out key encryption before uploading data to cloud storage. Additionally, the SHA-2 hashing algorithm is used both before and after the encryption and decryption. To check accuracy of data, both hash values are generated by a hashing technique and compared. In order to guarantee data integrity, authenticity, and non-repudiation, the article compares aspects like the hybrid cryptographic mechanism's execution time with existing systems.

1 Introduction

One crucial use case for cloud computing is mobile cloud computing. As the global adoption of smartphones, tablets, and laptops rises, mobile cloud computing is emerging as a futuristic technology. As the global adoption of tablets, smartphones, and laptops rises, mobile cloud is emerging as a upcoming technology. Depending on amount of storage free space, a mobile device can store a total amount of multimedia content. By routinely eliminating outdated files, the user can create room for new ones to be added. Integration of cloud-based storage will make it simple to solve this issue. The core idea behind mobile cloud computing is to use resource-intensive tasks and reduce resource utilization in mobile devices by using cloud computing techniques to immediately provide computing resources. This paper offers methods for transferring data to the cloud that are more private, secure, and resource-conserving. Data security is significantly aided by cryptography. With the help of cryptography, confidential information may be sent via insecure networks and kept secure without having to worry that someone except the intended recipient would be able to read it.

1.1 Mobile Cloud Computing Issues

Numerous new computing models, including mobile sensing, mobile computing , collaborative computing, pervasive computing, and others, have been sparked by the

widespread use and growing capabilities of mobile devices. One of the concepts that combined cloud computing and mobile computing is called mobile cloud computing. Various issues from the constituting models was passed on to MCC. It inherited the lack of resources from mobile computing, which clouds computing addressed.In MCC, data processing and storage are carried out in the cloud platform. One of the main issues that need to be resolved in MCC is the safeguarding of the privacy and security of user data kept in the cloud. This is because MCC users can keep all private information in the cloud for future use having little to no ability to control it, and a criminal posing as cloud personnel can access the private information, take it, and give it to the user's rivals. Consequently, the safety of cloud-based data storage has become a serious concern. Additionally, it's important to shield user information from cloud service providers as well as from outside intruders.
The difficult problem of the day is data security, which affects computers and communication among other things. A method known as cryptography was created specifically to protect the integrity of data and confidentiality during communication. Key distribution is a crucial factor in determining the type of encryption. The two primary types of cryptography are symmetric and asymmetric, depending on the key distribution type is covered in the below section.

1.2 Cryptography

Cryptography [18] is the science of applying math to transform information in plain text into an unintelligible format for cipher text and then convert the encryption text to plain text , restoring original plain text from the cipher text. Cryptographic algorithms are the procedures or ideas that are applied to encrypt and decrypt communications in a system. Over time, cryptography has been used to offer privacy, confidentiality, and integrity to alleviate security and privacy concerns. Encrypting and decrypting data is a procedure that maintain integrity , secrecy, and privacy is known as cryptography. It is one of the main techniques for resolving data security challenges. Cryptographic algorithms are classified as below:

1.2.1 Symmetric Key Algorithms

Shared key algorithm is another name for the symmetric algorithm. Both the sender and the recipient during data transmission utilize the identical key for encryption and decryption. The key should always be secure if confidentiality is to be maintained. The data could well be taken by the attacker if the communication key is exposed. Symmetric algorithms have the advantage of working quickly and using less computational power to encrypt data. Data Encryption Standard, Advanced Encryption Standard (AES), Blowfish and Triple Data Encryption Standard are a few examples of distinct symmetric methods.
Advance Encryption Standard : AES is an acronym for Advance Encryption Standard. It is a symmetric cipher with keys of 128, 192, and 256 bits with block sizes of 128 bits. The three main categories of encryption algorithms are transposition, substitution, and transposition-substitution technique. The round function used by the AES algorithm differentiate four possible byte-oriented transformations, including sub-byte, shift-row, mix-column, and add round key. The required number of rounds depends on the length of the key; for example, a 10 rounds are needed for a 128-bit key, 12 rounds are needed for a 192-bit key, and 14 rounds are necessary for a 256-bit key.

Comparatively more secure and with a powerful avalanche effect is the AES algorithm. By using a brute force attack, attackers cannot readily decrypt the encrypted text. As a result, AES has been utilized in numerous applications. The AES method is typically coupled with other encryption algorithms to create an onion-layered structure, which boosts security, because of its security.

1.2.2 Asymmetric Key Algorithms

Public key cryptography is another name for an asymmetric algorithm. It uses both public and private key. A public key known as the cipher text is used by the transmitter to encrypt plain text during data transmission, and a private key is used by the recipient to decrypt this cipher text. Public-key encryption techniques typically employ substantially longer keys, which enhances the security of the data being transmitted. The Elliptic Curve Cryptography is one of the types of asymmetric key algorithm which was used in the proposed solution.
Elliptic Curve Cryptography (ECC): Elliptic curve cryptography uses elliptic curves to operate on a collection of points, allowing for greater security with smaller key sizes. The small key size in ECC decreased the computational power. The hash function SHA-2 reduced the integrity issue.
A hash algorithm processes data by using a mathematical function. When data is subjected to the function, an alphanumerical value is generated. There are several hashing algorithms, each of which yields unique results. A set of data will typically yield the same hash value if the same mathematical function is used, and the contents are left intact. It is possible to imagine the value created as a digital fingerprint. It's helpful to utilize hashing as a checkpoint.

1.2.3 Secure Hash Algorithm

A hash function converts an input of some length into a string of a specific length that serves as a "fingerprint." Such a function is frequently used as an index into a hashtable. Additional characteristics of cryptographic hash functions make them suitable for use in digital signature schemes and as a way to verify the accuracy of messages.

Consider the hypothetical scenario below to see how data integrity can be verified via hashing. An individual has a collection of data in the Cloud that is required to carry out a calculation. The data set is used to generate a hash value before performing the calculation. This value is kept on file for future use. After the calculation is completed, a new hash value is produced from the same data set. The two values should be equivalent if there was no data manipulation during the calculation. It can be determined that something went wrong with the data during the calculation if the two numbers are not equal. The proper procedures can then be taken to fix the issue.

1.3 Research Question

In what ways can data encryption using a hybrid cryptographic technique increase the resource-efficiency, security, and privacy of cloud interaction on mobile?

1.4 Report Objectives

To implement a secure and resource-efficient ways for file storage in cloud environments:
In our daily activities, a tremendous amount of data must be transmitted and stored. The goal of the study is to develop reliable and secure file storage that allows users to send and save their data in the cloud. Using a hybrid encryption strategy provide greater security than doing so with a single technique, even though cryptographic encryption is the best way to safeguard the data or file.

1.5 Report Structure

The following sections make up this report: Related works are in Section 2. In Section 3, the Research Methodology is described. Design specifications are explained in section 4. Section 5 covers implementation, which is Evaluation follows in Section 6.
The research is concluded in Section 7, which also outlines the next work's scope.

2 Related Work

This section reviews all relevant work papers and includes a critical analysis of the weaknesses, problems, and potential future applications. All pertinent author work is segmented and evaluated based on merits, drawbacks, and issues, which aids in the study. The use of symmetric and asymmetric key algorithms in data encryption is compared.

2.1 Background

Elliptic Curve Cryptography *:* Using elliptic bend conditions, the keys are processed. It has more options than Diffie Hellman and RSA computations and can garuantee security using a 164-piece key. Power abuse is minimal, which benefits batteries more. Due to the increased size of the encrypted message and the difficulty of implementation as compared to RSA. To accomplish this, the Elliptic Curve Digital Signature Algorithm is presented. The system is protected from man-in-the-middle attacks thanks to the ECMQV Authenticated key agreement protocol.
SHA-2 Algorithm: In the proposed solution, SHA-2[15] was used to provide message integrity. Data security is provided by the hash function family known as SHA-2, it includes the SHA-224, SHA-256, SHA-384, and SHA-512 hashing algorithms. These applications include PRNG, DSA, and HMAC. Real-world applications also require to compute the short hash value and lengthy messages, as depicted in Fig. 1, in addition to using multiple SHA-2 functions. As seen in Fig. 1, the SHA-2 only calculate one 512/1024-bit data block when computing short messages to produce the output hash. In long message calculation, the message is broken up into numerous 512/1024-bit data blocks, and the data blocks are computed by the SHA-2 one at a time, using the output of the previous block's hash as the starting point for the subsequent block's hash computation. The SHA-2 accelerator should be extremely flexible to carry out many SHA-2 operations and varying message lengths because real applications demand both a variety of SHA-2 operations and long/short messages.

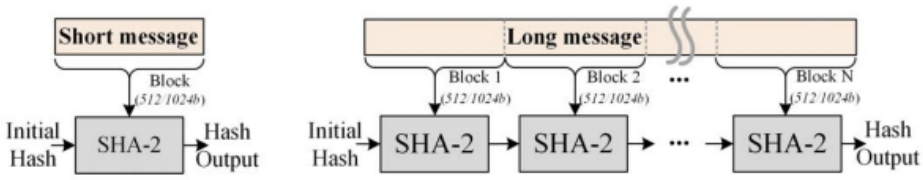

Fig 1 : Creation of short and lengthy message SHA-2 hash values

This section looks into the specifics of the SHA-2 algorithm in order to create a high performance and adaptable SHA-2 accelerator. Six hash functions, specifically SHA224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA512/256, are part of the SHA-2 family. The SHA-224, SHA-256, SHA-384, and SHA-512 operations are the only ones that are covered in this work due to the fact that SHA-512/224 and SHA-512/256 are just shorter variants of SHA-512 and are not frequently utilized. Four hash functions basically operate in the same way, but with slightly different parameters. It should be noted that SHA-256d computes SHA256 twice and is not a SHA-2 family's variant function.

The SHA-2 algorithm, which consists of the three processes of padding, message expansion, and message compression, is shown in Algorithm below.

Algorithim :

Padding:

L = length in bits (message)

k = B - (1 + D + (L mod B))

Pad = {1, zeros(1,k), L}

M[0:N−2] = message[0 : ((N − 2) ∗ B) − 1]

MN−1 = {message[(N − 2) ∗ B : L − 1], Pad}

for t ← 0 to (N-1) do

Message Expansion (ME):

for i ← 0 to (R-1) **do**

if i < 16 **then** $W_i = M^t_{[i\times D:(i+1)\times D]}$

else $W_i = W_{i-16} + \sigma 0(Wi-15) + Wi-7 + \sigma 1(Wi-2)$

Message Compression (MC):

a=H^t_0,b=,H^t_1c=H^t_2,d=H^t_3

e=H^t_4,f=H^t_5,g=H^t_6,h=H^t_7

for i ← 0 to (R-1) **do**

=h+ Σ_1(e) + Ch(e,f,g) + K_i + W_i

=h+ Σ_0(a) + Maj(a,b,c)

h = g, g = f, f = e, e = d + T_1,

d = c, c = b, b = a, a = T_1 + T_2

H^0_{t+1} = H^0_t+ a, ... H^{t+1}_7= H^t_7 + h

return Hash = {H^{N-1}_0 , ...,H^{N-1}_{nD-1}}

Where , Block size (B),Message size (M),Word size (D),Word number (nD),Digest size (D×nD) and Round number (R)

The Proposed solution in this research has combined the Elliptic Curve digital signature algorithm, SHA-2, and Advanced Encryption Standard (AES) algorithms , reffered as ASE algorithm. AES algorithm is used to encrypt data, whereas ECC carries out key encryption before uploading data to cloud storage. Additionally, the SHA-2 hash function is used both before and after the encryption and decryption . In the paper, to improve the effectiveness of

security in IoT networks, the suggested mechanism makes use of a hybrid encryption function that was created utilizing AES, ECC, and SHA-2. Advanced Encryption (AES) AES is being developed to replace the established Data Encryption Standard . AES is a block cipher that accepts 128-bit plaintext as input and encrypts it using 128 bits, 192 bits, and 256 bits depending on whether there are 10, 12, or 14 rounds.

2.2 Literature Review

S.Subhasree [7] recommends a hybrid strategy using Dual RSA, Elliptic curve encryption, and MD5 for increased security and integrity. It also provides the three cryptographic primitives of secrecy, authentication, and integrity. Hybrid cryptography, according to Manorama Chauhan [8], has been introduced. By integrating the MD5 hash generation algorithm and the ECC encryption-decryption technology, the method provides a more effective and secure cryptographic methodology. The suggested strategy asks for the development of security solutions using the MD5 hash-generating method and common encryption techniques (ECC). A hybrid approach to cryptography is provided by Rathod [9] et al. for RSA and AES. AES only needs one key, whereas the hybrid technique needs three.

When transferring and receiving files from the server, both the RSA public key and the AES secret key are utilized. These algorithms have the benefit of ensuring the availability, security, and confidentiality of the data. The drawback is that it takes a long time to process data encoding and decoding.

Hybrid encryption, which combines four different encryption algorithms, is used by the Mehul et al. [10] system. The fundamental idea behind hybrid encryption is to mix several encryption methods to strengthen cloud security and protect personal data. These four algorithms are utilized by this model: Steganography, RC4, DES, and AES. The sentences are categorized into three segments, the first of which uses RC4 encryption, the second DES encryption, and the third AES encryption.The secret message will contain the key, which uses steganography, a method where the sender and recipient have cause to feel that it is there. The study concludes that the DES and 3DES algorithms are not recommended for usage in hybrid cryptographic approaches because of their poor performance and large file sizes. The study found that the 2-tier method outperforms the 3-tier method. The study is unable to adequately overcome the three-tier hybrid models' performance constraint despite combining numerous different approaches. This investigation will address this drawback. According to the study's conclusions, AES is the best symmetric key encryption algorithm in terms of price, security, and ease of use. An approach based on elliptical curve cryptography is developed and put to the test to ensure cloud security. SHA-2 is utilized to lessen integrity issues. There are numerous instances where cloud computing and social networking platforms have been combined.

An AES-based, Message Digest-based hybrid cryptosystem (MD5) , and ECC to guarantee information security in the IoT environment was proposed by Chanal and Kakkasageri [11]. In its three-phase encryption process, this system between the source and destination nodes, utilizes a geo-tag. For this system, a key generation method has also been created that results in key sizes for encryption and decryption that are more efficient. To increase overall security and processing performance, Aghajanzadeh et al. [25] developed a hybrid system using the AES, Serpent, and RC4 algorithms. This approach makes use of the extremely secure and resistant to cryptographic attacks Serpent algorithms and original AES. Data security, with this

cryptosystem, and speedier encryption and decryption are achieved using three symmetric approaches in combination.

Hybrid cryptography has been utilized by Wani, S.M.K., and Kumar, A [13] to guarantee data security in cloud storage. Since the data is encrypted and can only be read and downloaded by the authorized user, this system allows any authorized user to safely store their data or file in cloud storage, eliminating privacy issues. Even before uploading to the cloud, the original data is converted into an incomprehensible format with the appropriate secret key information using AES encryption. The owner's ElGamal public key, which is created upon registration, is utilized to encrypt the associated secret key. Afterward, an automatic decryption operation will be carried out to restore the original data and compare its hash value with the generated hash value and saved in the database during the upload process. Only in cases where the SHA-2 algorithm successfully decrypts and validates the file will it be made accessible for download. It's to make sure that data wasn't altered or tampered with during transmission.

V. S. Mahalle and A. K.Shahade [14] focused on using RSA and AES as a hybrid encryption scheme in the cloud. In their suggested approach, the RSA and AES algorithms are combined to increase cloud security. When RSA generates keys depending on system timing, duplicated keys are avoided. The user enters the AES secret key during the uploading procedure for encryption, and the user's RSA public key is then used to carry out the encryption. The user must specify the filename to be downloaded together with a functioning RSA private key and an AES secret key for decryption in order to access and download the data. The twofold encryption layer and the system timing-based key generation are two benefits of the suggested system. Data that is uploaded is kept in an encrypted format so that no one, not even the cloud operator, can access it. The inclusion of AES makes the proposed system highly efficient during data encapsulation, but it also causes a slowdown owing to the intricate factorization in RSA.

Table 1: Overview of literature review

Sr no	Title	Algorithm used	Strength	Weakness	Author
1	Survey Paper on Different Type of Hashing Algorithm	Hashing algorithms – SHA1, SHA2, SHA3	It recognizes the varying length of the information and produces a typically small, settled size yield, or message process.	Hash capabilities depend on MD5, SHA1, SHA2, SHA3, and other algorithms, making them vulnerable to attacks.	Kale, A.M et.al (2018) [16]
2	Secure File storage in Cloud Computing using Hybrid Cryptography Algorithm	Blowfish, SRNN public key	There are issues with information security and protection on every level, including SPI benefit conveyance methods.	Time performance is improved by Srnn.	B.Swathi and Bhaludra [16]

3	A multilevel encryption technique in cloud security	AES, ECC	Mutiple level encryption technique – double authentication	The effectiveness of the encryption process is not raised	B. Jana, J. Poray, T. Mandal and M. Kule [17]

3 Research Methodology

The provided methodologies have served as the foundation for a comprehensive study. Cloud cryptography has been employed in several works, but not many have used the combination of algorithms presented in this paper. The research has produced a method to prevent crucial security breaches. The SHA-2 algorithm, ECSDA, and AES in the ASE algorithm are all components of the hybrid cryptographic model. In the suggested mobile cloud computing storage architecture, mobile devices link to the data storage using a mobile network or Wi-Fi and employ the ASE algorithm.

The original data is first hashed using SHA, which is used to check the accuracy of the data. The elliptic curve technique is used in the second stage to create digital signatures and secure the private key. The EC will run at a specific time shortened because we must maintain a high standard of security, short keys and hash codes. The original data is then encrypted using the symmetric AES technique, which has a quick encryption and decryption time. The cryptographic approach was implemented using below three phases:

3.1 Digital Signature

To ensure data integrity, this paper combined a hashing function for a digital signature. In the study, the message is hashed first, and then it is signed. As a result, the attacker is powerless to alter or harm the message content by using a phony signature because it does not match the result of the hashed message. As a result, adopting a digital signature is a powerful technique for protecting cloud computing. A flowchart for the suggested solution's use of a digital signature and hash code is shown in Fig. 2.

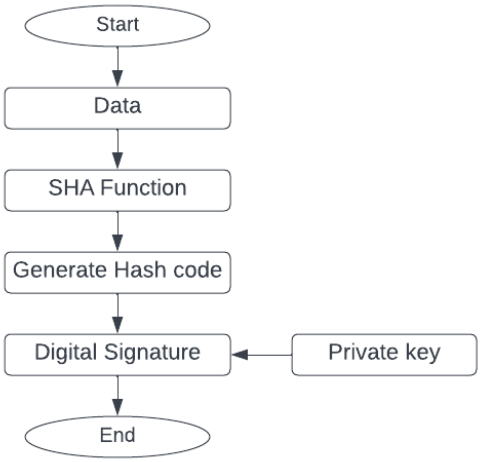

Fig 2. Digital signature and hash code flowchart

3.2 Data Encryption

This study suggests a quick-to-implement secure cryptography method. The AES algorithm was utilized in the article. It is a symmetric cipher with block sizes of 128 bits and keys of 128, 192, and 256 bits. The three main categories of encryption algorithms are transposition, substitution, and transposition-substitution technique. The round function used by the AES algorithm compares four possible byte-oriented modifications, including sub-byte, shift-row, mix-column, and add round key. The number of rounds to be employed depends on the key's length; for example, a 128-bit key requires 10 rounds, a 192-bit key requires 12 rounds, and a 256-bit key requires 14 rounds.

The proposed mobile cloud computing storage architecture use the AES , ECC to encrypt data, later perfor m hashing for integrity and connect using a wireless or mobile network, to the cloud storage. To guarantee a higher degree of trust, security, and privacy for the user, Reference[19] recommended encrypting personal data that the cloud service provider is unaware of. Therefore, while encrypting data, the ASE algorithm is used on the mobile cloud before being uploaded to the Azure Cloud storage environment, and data is likewise decrypted on the mobile after being read from the cloud storage.

It has two processes: encryption and decryption

A. Encryption Process: Fig 3. shows the process of the encryption. It involves:
    1. Key Generation : A public key is generated
    2. Encrypt file using ECC algorithm and above generated key.
    3. Private Key Generation: Encrypted key is generated using ECC
    4. Using the private key, sign the data hash.
    5. The file is encrypted using AES operations and encrypted key
    6. The encrypted file is uploaded to the cloud data storage

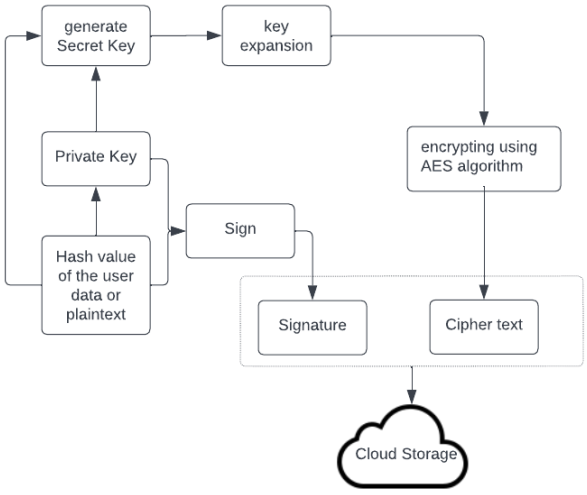

Fig 3: Encryption process

B. Decryption Process: Fig 4. shows the process of the decryption.
    1. Input secret key generated using encryption
    2. The encrypted data is decrypted using ECC algorithm
    3. Verifying the hash value generated while encrypting.
    4. Terminate if unsuccessful or lese go to next step

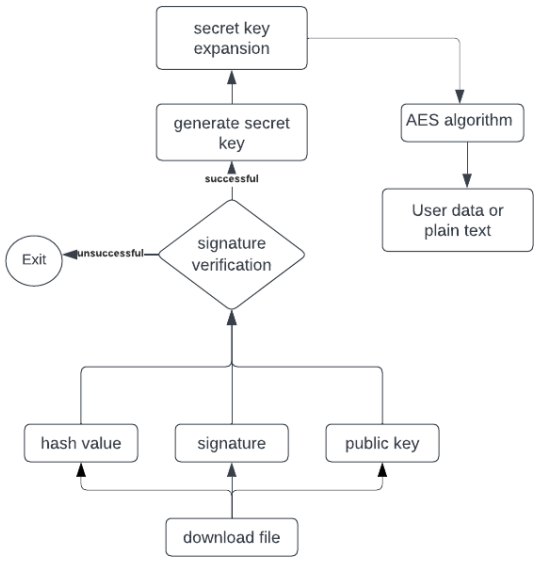

Fig 4: Decryption process

3.3 Securing the Symmetric key

It is dangerous to transfer the private key for symmetric-key based algorithms like AES because hackers likely to steal the key (e.g., man in the middle attacks). So, in this method, to secure the private key and hash code, we employ an asymmetric cryptography approach based on elliptic curves. This approach uses a 164-bit key to deliver better performance than existing approaches. High levels of security are provided by elliptic-curve-based cryptography methods, which also consume little memory and bandwidth. The use of cryptography in place of finite fields gives this encryption a benefit. The procedure is explained in full in the steps that follow.

Step 1: Prior to encryption, a hash code for the original data will be created (using SHA). This code will be used to increase the effectiveness of data integrity checks and digital signatures.

Step 2: After the hash code has been encrypted with the private key, the data will be issued with a digital signature.

Step 3: Elliptic-curve cryptography will be used to encrypt the AES private key.

Step 4: The symmetric cryptography algorithm will be used to encrypt the data (AES). Using the symmetric key ,we encrypt the initial data.

Step 5: Using a reverse algorithm and the private key, the receiver decrypts the data they have received.

Step 6: Using the AES private key to decrypt the original data and the hash function to execute the validation and verification process (digital signature).

4 Design Specification

This section elaborates on the layout and structure of the suggested strategy that was covered in the methodology section. The two operations that make up the suggested approach are encryption and decryption. ECC and AES are used in both procedures. The following explains why we selected these particular algorithms: ECC is a public-key algorithm that may be applied to both digital signature and encryption procedures. The ECC algorithm's key management

system is a crucial component. The security of the ECC method is based on how challenging it is to factor huge integers.
AES delivers a very high performance and makes better use of resources in addition to being a safe cipher. The US government has approved its usage for top-secret information encryption because it is robust enough. The only known successful assault against AES to yet has been a brute-force attack, which has not yet been discovered.
Fig 5. illustrates an architectue of the proposed solution. The solution comprises of key generation, encryption and decryption using AES and ECC ,hashing data.

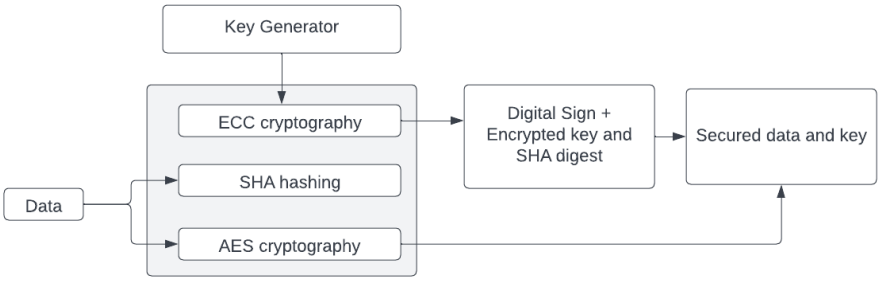

Fig 5: Architecture of the proposed solution

Fig 6. illustrates the flow diagram for the proposed solution. The registration process must be finished by the User via the mobile interface. The user can log in to the program after registering. The document may be uploaded by the user. The ASE algorithm is being used to encrypt this text. The user's registered email address receives the key for later usage. The user will have the ability to view and download documents from the cloud. The application uses the ASE algorithm to decrypt data in the same way. Thus, the user's data saved in the cloud is protected from unauthorised access and is authenticated, authenticated, and non-repudiated.

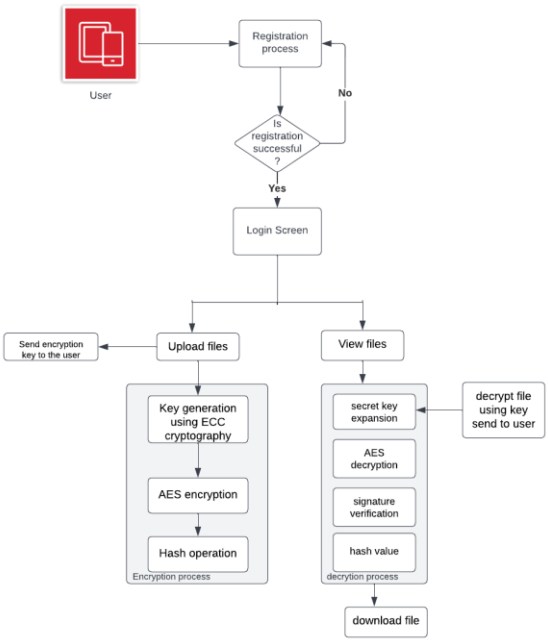

Fig 6: Flow diagram for proposed solution

5 Implementation

The cryptographic approach was implemented as an android mobile application called CrypographyApp. Android O.S was used for the application development. Android Studio was

used as a development tool. The interface of mobile application includes registration screen, login screen, dashboard screen, upload screen ,view files screen and download screen.

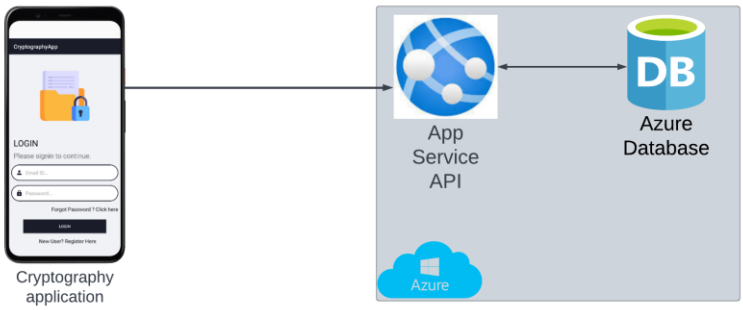

Fig 7: Cryptograhy implementation in cloud

The data is uploaded to Azure cloud, as seen in Fig. 7. The mobile interface makes use of an API that is housed on an Azure App Service tool, which internally accesses an Azure SQL database.

The middleware(Web API layer) of the application was developed in ASP.NET framework using Visual Studio 2019 (Web Assembly API framework). The cryptographic functions are provided by AesCryptoServiceProvider and ECDiffieHellmanCng libraries . The hashing for data integrity is done by SHA256Managed. The Web API layer implementation contains the controller package with the class "UserController". This class contains the end point implementation of the API. The list of Web API endpoints is as given below:

- Register API – endpoint used to register user
- Login API – endpoint used to login user
- Upload API – endpoint used to upload files to the cloud
- Download API –endpoint used to download file from the cloud
- GetFiles API – endpoint used to fetch file details of the user from the cloud

The middleware layer was deployed to App Service in Azure cloud. The database layer for the application was created using SQL Database in Azure.

6 Evaluation

6.1 Experiment Setup

The implementation of the hydrid cryptography approach is done using Android 13 Pixel device , Mac 11.6.4. The input files are text files and image files ranging from ranging till 3000 KB. In this experiment, encryption and decryption speed , throughput are all evaluated factors.

6.2 Experiment / Case Study 1

The experiment's evaluation criteria takes into account the speed of encryption and decryption. The evaluation hybrid algorithms are Blowfish and RSA[2], AES and RSA [26], 3DES and ElGamal [27] and ASE (proposed algorithm)

Blowfish and RSA, AES and RSA, 3DES and ElGamal, and proposed algorithm (AES, ECC, and SHA2) are the hybrid methods that have been assessed. The solution was tested using

small (50 KB) data to evaluate its performance, and then the test was repeated using larger data (ie. 1024 KB). The encryption time for each of the different techniques is shown in Table 2.

Table 2 : Encryption time result (ms)

File Size/Scheme	Blowfish & RSA	AES & RSA	3DES & ElGamal	Proposed algorithm (ASE)
1 KB	15.8	23.8	21	508
19 KB	80.4	93.6	86	518
36 KB	153.0	165.2	147.6	549
99 KB	387.8	411.8	415.0	555
465 KB	1580.0	1629.8	1764.2	750
2305 KB	8104.2	7517.2	8008.4	953

The time taken to execute ASE encryption is shown in a graphic form in Figure 8. The time is displayed on the Y axis in milliseconds, and the X axis displays the file size in KB. The encryption speeds for files larger than 100KB, the developed ASE performed noticeably faster to other algorithms.

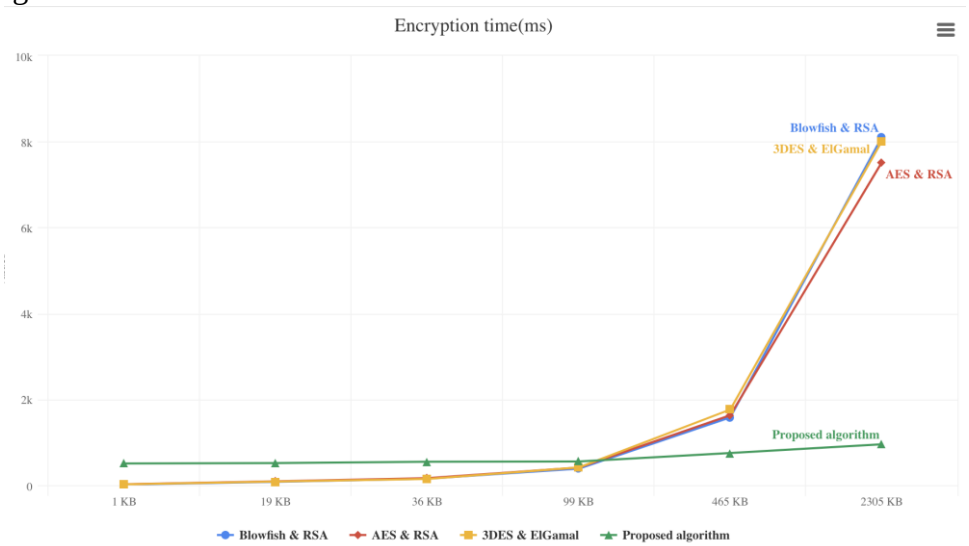

Fig 8: Encryption execution time

Table 3 displays the decryption time for all of the various techniques.

Table 3 : Decryption time result (ms)

File Size/Scheme	Blowfish & RSA	AES & RSA	3DES & ElGamal	Proposed algorithm(ASE)
1 KB	77.4	78.8	6.6	9
19 KB	170.4	152.4	78.6	9
36 KB	197.4	186	160.0	13
99 KB	410.0	390.6	363.8	22
465 KB	1652.0	1532.6	1569.4	59
2305 KB	7459.4	7078.0	7519.6	333

The decryption execution times of ASE are shown graphically in Fig. 9 (execution timings of ASE). The X-axis shows the file size in KB and Y axis shows the time in milliseconds. The

decryption speeds for files larger than 100KB, the developed ASE performed noticeably faster than other algorithms.

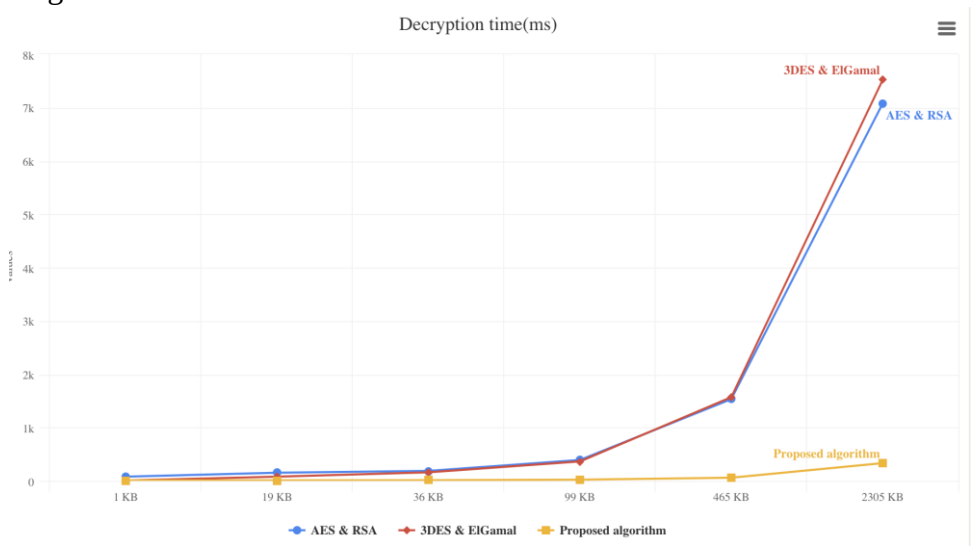

Fig 9: Decryption execution time

According to Tables 1 and 2, the average time for proposed hybrid approach for encrypting and decrypting files is a little bit long for smaller file sizes, but it is less time than the current system for larger file sizes.

6.3 Experiment / Case Study 2

The experiment's evaluation criteria takes into account is the throughput of each algorithm. The evaluation hybrid algorithms are Blowfish and RSA[2], AES and RSA [26], 3DES and ElGamal [27] and ASE (proposed algorithm). According to table 4 below, the throughput (execution time for a file per unit size) of each technique for encryption and decryption was calculated and compared.

Table 4 : Throughput (kb/ms) for encryption and decryption process

File Size/Scheme	Blowfish & RSA		AES & RSA		3DES & ElGamal		Proposed algorithm (ASE)	
	En	De	En	De	En	De	En	De
1 KB	0.063	0.012	0.042	0.012	0.047	0.15	0.0001	0.11
19 KB	0.236	0.111	0.202	0.111	4.526	0.24	0.0366	2.11
36 KB	0.235	0.18	0.22	0.19	0.243	0.225	0.065	2.77
99 KB	0.255	0.24	0.24	0.25	0.238	0.27	0.17	4.5
465 KB	0.29	0.28	0.285	0.3	0.26	0.29	0.62	7.89
2305 KB	0.284	0.3	0.306	0.32	0.287	0.3	2.42	6.92

The algorithms throughput was calculated using:

$$Throughput = t_p / e_t$$

where et is the execution time in milliseconds and tp is the file size (in KB).

ASE has higher throughput compared to other hybrid algorithms at file sizes higher than 465KB both during encryption and when decrypting files larger than 100 KB, as shown in Figs. 10 and 11. While it has a close throughput with other algorithms during encryption at file sizes less than 100KB, as shown in Figs. 10 and 11.

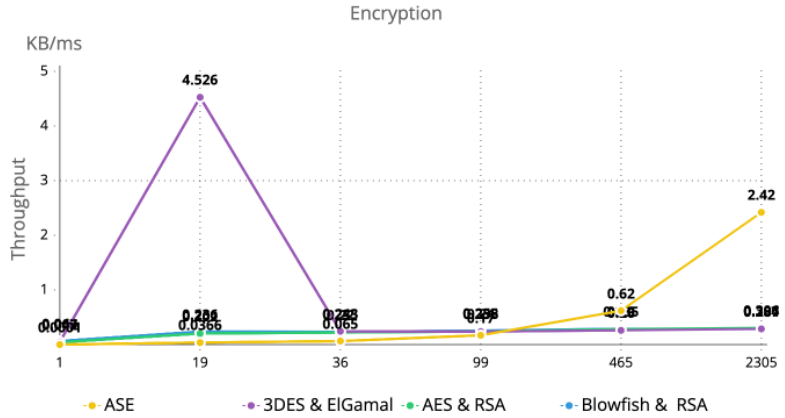

Fig 10: Encryption throughput comparision

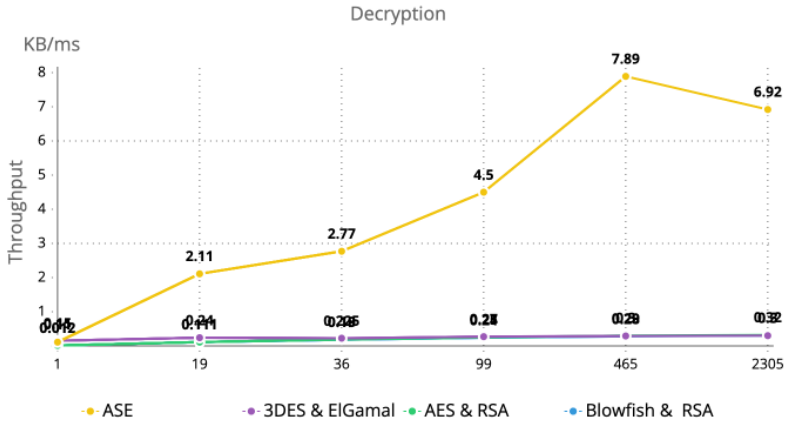

Fig 11: Decryption throughput comparision

6.4 Discussion

We have run files ranging in size from 1KB to 3000KB using a hybrid cryptography approach. The comparison is based on the throughput and execution times for each algorithm. The ASE algorithm takes 953 ms to encrypt and 333 ms to decode a 3000 KB file, compared to more than 7000 ms for other algorithms. For files larger than 2000 KB, the throughput of the ASE algorithm for encryption and decryption is 2.42 kb/ms and 6.92 kb/ms, respectively, compared to less than 0.35 kb/s for other algorithms. By having a successful hash match, the suggested algorithm (ASE) offers confidentiality and security. It also offers improved time and validation for data integrity.

7 Conclusion and Future Work

Hybrid cryptographic algorithm have been used to safeguard user data in the cloud from theft and unauthorized users, but the only surefire way to maintain complete security and support MCC adoption is for users to encrypt and digitally sign their data before uploading it to the

cloud. In this study, we have seen the source file is encrypted using the secret key, the recipient encrypted the file before running a hash comparison. A successful hash match show that the data has not been altered. To guarantee the secrecy of sensitive data, this process performs encryption and decryption. It protects private information against tampering and illegal access. It offers authentication, faster time, and data integrity verification. In this study, one of the effective methods for providing security, authentication, integrity, and non-repudiation on resource-constrained mobile cloud computing devices was developed. It is based on signatures and is robust and resource-efficient.
Future research will compare the ASE to other hybrid cryptographic methods with larger file size.

References

[1] Beloglazov, A. and Buyya, R. (2015). Openstack neat: a framework for dynamic and energy-efficient consolidation of virtual machines in openstack clouds, *Concurrency and Computation: Practice and Experience* 27(5): 1310–1333.

[2] Timothy, D.P. and Santra, A.K., 2017, August. A hybrid cryptography algorithm for cloud computing security. In 2017 International conference on microelectronic devices, circuits and systems (ICMDCS) (pp. 1-5). IEEE.

[3] Gomes, D. G., Calheiros, R. N. and Tolosana-Calasanz, R. (2015). Introduction to the special issue on cloud computing: Recent developments and challenging issues, *Computers & Electrical Engineering* 42: 31–32.
Jana et al
[4] Kune, R., Konugurthi, P., Agarwal, A., Rao, C. R. and Buyya, R. (2016). The anatomy of big data computing, *Softw., Pract. Exper.* 46(1): 79–105.

[5] William, P., Choubey, A., Chhabra, G.S., Bhattacharya, R., Vengatesan, K. and Choubey, S., 2022, March. Assessment of Hybrid Cryptographic Algorithm for Secure Sharing of Textual and Pictorial Content. In *2022 International Conference on Electronics and Renewable Systems (ICEARS)* (pp. 918-922). IEEE.

[6] KOTEL, S. and SBIAA, F., 2022. A Data Security Algorithm for the Cloud Computing based on Elliptic Curve Functions and Sha3 Signature. *International Journal of Advanced Computer Science and Applications*, *13*(3).

[7] Subasree, S. and Sakthivel, N.K., 2010. Design of a new security protocol using hybrid cryptography algorithms. IJRRAS, 2(2), pp.95-103.

[8] Chauhan, M.M., 2016, August. An implemented of hybrid cryptography using elliptic curve cryptosystem (ECC) and MD5. In 2016 International Conference on Inventive Computation Technologies (ICICT) (Vol. 3, pp. 1-6). IEEE.

[9] Rathod, S., Raut, S. and Yadav, D., Hybrid Cryptographic Technique for File Storage in Cloud Computing. JOURNAL OF ENGINEERING AND SCIENCES, p.26.

[10] B. Mehul, D. Prayas, R. Lalit & K. Rohini, “Secure File Storage In Cloud Computing Using Hybrid Encryption Algorithm”, International Journal of Computer Engineering and Applications,9(6),2018

[11] Nagar, N. and Suman, U., 2016. A secure mobile cloud storage environment using encryption algorithm. International Journal of Computer Applications, 140(8), pp.0975-8887

[12] AbdElnapi, N.M., Omara, F.A. and Omran, N.F., 2016. A hybrid hashing security algorithm for data storage on cloud computing. International Journal of Computer Science and Information Security (IJCSIS), 14(4).

[13] Wani, S.M.K. and Kumar, A., 2022. Secure File Storage on Cloud Using a Hybrid Cryptography Algorithm. International Journal of Research in Engineering, Science and Management, 5(5), pp.35-39.

[14] V. S. Mahalle and A. K.Shahade, “Enhancing the data security in the cloud by implementing hybrid (RSA& AES) encryption algorithm,”2014 International Conference on Power, Automation and Communication(INPAC),2014, pp. 146-149

[15] M. A. Kale and S. Dhamdhere, “Survey Paper on Different Type of Hashing Algorithm”, Int. J. Adv. Sci. Res. Eng. Trends, Vol. 3, no. 2, pp.14-16, Feb. 2018.

[16] B. Swathi, S.D Bhaludra, S. Raveendranadh, “Secure File Storage In Cloud Computing Using Hybrid Cryptography Algorithm”, International Journal of Advance Research in Science and Engineering 6(11), 2017.

[17] Jana, B., Poray, J., Mandal, T. and Kule, M., 2017, November. A multilevel encryption technique in cloud security. In 2017 7th International Conference on Communication Systems and Network Technologies (CSNT) (pp. 220-224). IEEE.

[18] A. Bhardwaja, G. V. B. Subrahmanyam, V. Avasthi and H. Sastry, ―Security algorithms for cloud computing‖, Procedia Computer Science, vol. 85, pp. 535-542, 2016.

[19] Seyyed Yasser hashemi, Parisa Sheykhi Hesarlo,"Security, Privacy and Trust Challenges in Cloud Computing and Solutions", IJCNIS, vol.6, no.8, pp.34-40, 2014. DOI: 10.5815/ijcnis.2014.08.05

[20] Pham, H.L., Tranl, T.H., Le, V.T.D. and Nakashima, Y., 2022, May. A Coarse Grained Reconfigurable Architecture for SHA-2 Acceleration. In 2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW) (pp. 671-678). IEEE.

[22] Varma, V., Patil, M., Patil, S., Patil, M. and Kadam, A., 2022. Data Storage Security in Cloud Computing Using AES Algorithm and MD5 Algorithm. International Journal for Research in Applied Science and Engineering Technology, 10(5), pp.5052-5055.

[23] Akomolafe, O.P. and Abodunrin, M.O., 2017. A hybrid cryptographic model for data storage in mobile cloud computing. International Journal of Computer Network and Information Security, 9(6), p.53.

[24] Wani, S.M.K. and Kumar, A., 2022. Secure File Storage on Cloud Using a Hybrid Cryptography Algorithm. International Journal of Research in Engineering, Science and Management, 5(5), pp.35-39.

[25] Lai, J.F. and Heng, S.H., 2022. Secure File Storage On Cloud Using Hybrid Cryptography. Journal of Informatics and Web Engineering, 1(2), pp.1-18.

[26] Mahalle, V.S. and Shahade, A.K., 2014, October. Enhancing the data security in Cloud by implementing hybrid (Rsa & Aes) encryption algorithm. In 2014 International Conference on Power, Automation and Communication (INPAC) (pp. 146-149). IEEE.

[27] Jintcharadze, E. and Iavich, M., 2020, September. Hybrid implementation of Twofish, AES, ElGamal and RSA cryptosystems. In 2020 IEEE East-West Design & Test Symposium (EWDTS) (pp. 1-5). IEEE.