Journal of Informatics and Web Engineering

Vol. 1 No. 2 (September 2022) eISSN: 2821-370X

Secure File Storage On Cloud Using Hybrid Cryptography

Jian-Foo Lai[1], **Swee-Huay Heng**[2*]

[1,2] Faculty of Information Science and Technology, Multimedia University, Malaysia

**corresponding author: (shheng@mmu.edu.my, ORCiD: 0000-0003-3627-2131)*

Abstract - As technology today is moving forward exponentially, data exchange over the Internet has become a daily routine. Furthermore, businesses are growing internationally and offices are being established in a variety of different places throughout the world. This has resulted in the necessity to make data accessible and practical from any place. As a result, information sent via an may lead to critical security problems involving the breach of secrecy, authentication, and data integrity. This paper introduces a cloud storage system by utilising hybrid cryptography approach that leverages both advantages of symmetric key and asymmetric key cryptographic techniques. In our proposed system, the symmetric key algorithm AES is utilised to encrypt data, whereas the asymmetric key algorithm ElGamal is employed to perform key encryption before the data upload into cloud storage. Combining both symmetric key and asymmetric key methods alleviates privacy issues while increasing data confidentiality. In addition, a hash function which is SHA-2 is executed before the encryption process and after the decryption process. Both hash values are derived through a hashing procedure and matched in order to verify the data integrity. However, if the users' accounts were lost or stolen, all encryption would be meaningless. Hence, a Two-Factor Authentication (2FA) is also employed to minimise the abovementioned risk to achieve a greater security over the cloud environment.

Keywords — Cloud Storage, Hybrid Cryptography, Advanced Encryption Standard (AES), ElGamal Encryption, Secure Hash Algorithm-256 (SHA-256).

Received: 17 April 2022; Accepted: 28 June 2022; Published: 16 September 2022

I. INTRODUCTION

A. OVERVIEW

Cloud computing is the technique that provides users access to storage, files, software, and servers through their internet-connected devices such as computers, smartphones, tablets, and wearables. Data stored in the cloud can be accessed or retrieved by other users who are granted permission rather than direct access to the server. Due to its efficiency, many industries and organisations like universities, the military, and corporate have relied on various services and cloud storage in their daily routine. Despite the advantages, there still exist some security and privacy concerns in the services offered by the cloud providers. The data stored in the cloud is visible by the providers and can be easily modified or deleted by them. Besides, stored data could also be shared with third parties if necessary, as permitted in the privacy policy. Last but not least, data leakage is one of the possible risks since large-distributed data servers are being used to store user data. In order to preserve the confidentially of the stored data, cryptographic solutions should be considered.

Journal of Informatics and Web Engineering
https://doi.org/10.33093/jiwe.2022.1.2.1

Published by MMU Press. URL: https://journals.mmupress.com/jiwe

Cryptography, more specifically encryption, is a method of protecting data against harmful behaviour during transmission. More precisely, encryption ensures data confidentially. Generally, it uses an algorithm to perform encryption and decryption to prevent public or unauthorised users from reading the private message. For instance, Transport Layer Security (TLS) or Secure Socket Layer (SSL) is a standard technology or protocol for safeguarding transferred data between server and client over the internet. Since the communication is in encrypted form, an interceptor would not be able to comprehend its meaning unless they possess the decryption key.

Other security objectives besides confidentiality, include data integrity, authentication, and non-repudiation. In terms of data integrity, it means that data or message has not been altered or tampered with during the transmission. Authentication is one of the processes of access control that proves the identity of users are who they claimed to be. Usually, the password method is used in this process. Non-reputation always prevents entities from denying the action they took earlier. Some authorities used digital signatures and audit logs as a technique against the denying of users.

This paper proposes a hybrid cryptography scheme in a cloud environment (see Figure 1) by using a combination of symmetric key cryptography and asymmetric key cryptography to enhance data security and reduce risk. A combined hybrid algorithm tends to achieve better security and efficiency in a cloud environment.

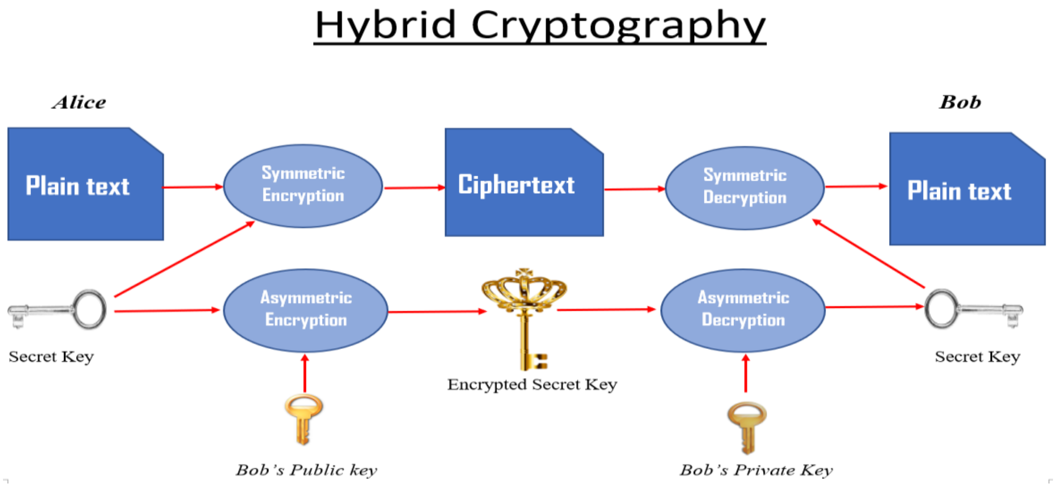


Figure 1. Hybrid Cryptography Model

B. PROBLEM STATEMENT

Numerous security issues are deemed crucial [1] due to the significance of information stored on the cloud and the different services provided to the users. These data can be confidential and extremely sensitive. Once the users store their data on the cloud, there must be an assurance that the data can be accessed only by the authorised user and remain confidential. Hence, security regarding data stored in cloud storage has to be secure against privacy issues, data exposure, and so forth.

Another problem is that there is a lack of integrity in the data stored in the databases. Ideally, data integrity means protecting data from unauthorised deletion, modification, or fabrication. However, data corruption is a critical issue that usually occurs during data processing or transmission due to hardware and software failure. It would return the unexpected results to the authorised user when they access. Utilising hashing is the common ways to ensure the accuracy and integrity of the data.

Last but not least, the authentication process is also an identified security problem that exists in cloud computing. According to the Data Breach Incident Report by [2], phishing and credential theft are the two most increasing attacks in data breach incidents in 2020. The hackers could expose the data stored in the cloud if they possess the credentials such as username and password. Most cloud authentication procedures nowadays utilise a single user name and password to verify a users' identity. As a result, cloud applications may be vulnerable to some security issues. Hence, an extra emphasis should be placed on adopting a strong password policy, unique passwords and 2FA when feasible.

C. RESEARCH OBJECTIVES

- **To implement a secure and efficient file storage system using hybrid cryptography**

  There is an enormous amount of data that should be transferred and stored in our daily routines. Therefore, the research aims to achieve a secure and efficient file storage system that enables users to transmit and save their data in cloud storage. Although cryptographic encryption is the best method to protect the data or file, using a hybrid encryption technique will offer better security than using a single technique.

- **To employ a hashing algorithm for ensuring data integrity**

  The hashing function is employed in this system to preserve data integrity and ensure no data corruption, unauthorised alternation or fabrication during the transmission.

- **To implement Two-Factor Authentication (2FA) during the verification process**

  Users will receive a Time-based One-Time Password (TOTP) that lasts for a certain period by sending a message to them. They have to use this TOTP to verify their identity. It serves as an additional security layer to resist unauthorised access if the credentials, such as passwords are being compromised or stolen by hackers.

II. LITERATURE REVIEW

A. RELATED WORK

According to [3], a hybrid system utilises the advantages of both asymmetric key and symmetric key encryption schemes. Blowfish symmetric algorithm is used in data encapsulation while RSA for encrypting the secret key and applying Digital Signature Algorithm (DSA). Before performing key exchange, this system will encrypt the data block using the blowfish technique and encrypt the symmetric key using RSA's public key. Subsequently, a hashing function SHA-2 is applied on encrypted files for message digest and utilises the digital signature algorithm once get the message digest. An encrypted data block is concatenated with an encrypted symmetric key, and the whole data block is transmitted to the target system. The receiver decrypts the encrypted symmetric key using their private key and then uses the symmetric key as a decryption key to decrypt the received data block. At the end of decryption, signature verification is performed to compare with the message digest to validate the integrity as well as authenticity. This methodology provides better performance in data confidentiality, integrity, and also authenticity. But blowfish algorithm is not recommended to encrypt files larger than 4GB due to its small block size, according to [4].

[5] focused on implementing a hybrid encryption algorithm (RSA and AES) on the cloud. A combination of RSA and AES algorithms is implemented in their proposed system to enhance the cloud security. Key generation for RSA is based on the system timing; therefore, the redundant key is avoided. In the uploading process, the AES secret key is entered by the user for encryption purposes then the RSA public key, which belongs to the user will then be used to perform encryption. The user must indicate the filename to be downloaded and give a valid RSA private key and AES secret key for decryption in order to get access to and download the data. The advantages of the proposed system are the double layer of encryption and the key generation based on system timing. Uploaded data is stored in an encrypted form so no third party can access it, even the cloud administrator. This proposed system provides high efficiency during data encapsulation because of the use of AES but also slows down due to the complexity of factorisation in RSA.

According to [6], the researchers have performed a hybrid implementation with four well-known algorithms: AES, Twofish, RSA, and ElGamal. A JAVA program was created to research each hybrid scheme's performance. Based on their experimental result, the combination of AES & ElGamal achieved better encryption time while encrypting larger files. In contrast, Twofish & RSA performed well while encrypting smaller files. Furthermore, the result also shows that AES & RSA needs the highest technical resources as compared to the hybrid model of Twofish & RSA, and AES & ElGamal. However, they concluded that AES & RSA scheme is more secure than other schemes, whereas Twofish & RSA achieved better efficiency in terms of speed.

B. COMPARATIVE ANALYSIS

Table 1 shows a comparative analysis according to each algorithm. The comparison table referred from [7] consists of seven different algorithms with the parameters of type, round, efficiency, key size, block size, security level, and power consumption. In addition, the structure, pros, and cons of each algorithm are included in this analysis.

Table 1. Cryptography Algorithm Comparison [7]

Algorithm/ Parameter	**AES**	**DES**	**3DES**	**Blowfish**	**RSA**	**ElGamal**	**ECC**
Type	Symmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher	Symmetric Block Cipher	Asymmetric Block Cipher	Asymmetric Discrete Logarithm	Asymmetric Discrete Logarithm
Structure	Fiestel Network	Fiestel Network	Fiestel Network	Fiestel Network	Exponentiation Congruence	Discrete Logarithm	Elliptic Curves
Key Size (bits)	128, 192, 256	56	168 112	32 to 448	1024	1024	160
Round	10, 12, 14	16	48	16	1	1	16
Block Size (bits)	128	64	64	64	min 512	min 512	64
Efficiency	High	Moderate	Low	High	Low	Moderate	High
Security	Adequately secured	Not secure enough	Not secure enough	Least secure	Least secure	Adequately secured	Adequately secured
Power Consumption	Low	Low	Low	Low	High	Low	Low
Pros	Provide higher security and also the efficiency with large key size	The number of rounds increases the complexity	Easy to implement in both software and hardware	Higher efficiency compared to DES and 3DES algorithm	Computationally infeasible to compute private key given public key.	Provide a different ciphertext (with near certainty) each time it is encrypted	Required less computing power
Cons	Difficult to implement in software	Short key size and considered as insecure	Provide low efficiency due to the number of rounds	64 bits block size makes it vulnerable to birthday attacks	Slower process due to difficulty of factorisation	Ciphertext is twice as long as the plaintext	Increased the size of encrypted messages and more complex to implement compared to RSA

III. RESEARCH METHODOLOGY AND PRELIMINARIES

There are several phases of research activities involved in implementing the cloud storage system. Firstly, literature review with respect to the existing works and the underlying algorithms is conducted, and requirements of the system are gathered and analysed. Based on the gathered requirements, use case diagram and system flowchart are illustrated in order to get a clear picture of the process of the system. In the development phase, the necessary software and tool such as Apache Web Server is installed and configured in order to set up the developing environment. After that, the development of each module is carried out by using PHP and javascript and debugging of each module is performed before the integration. All completed modules are then integrated as a whole and functional testing is performed and documented to make sure the integrated system is free of error. After the integration and testing are completed, the experimental evaluation is conducted to assess the performance of the system and comparison analysis is with the existing solutions is derived.

A. UNDERLYING CRYPTOGRAPHIC ALGORITHMS

The paper presents a new model that implements hybrid cryptography, hash function, and One-Time-Password (OTP) validation. It was developed using PHP, HTML, and Firebase. It utilises two major encryption algorithms, the Advanced Encryption Standard (AES) algorithm, and the ElGamal algorithm. Furthermore, the SHA-256 hash function is utilised to verify data integrity and eliminate the chances of sensitive data being comprised by a third party such as a cloud service provider. Each algorithm was selected through various perspectives of consideration after performing the comparison analysis.

The AES or Rijndael is a symmetric cryptographic scheme that encrypts data in a block called block cipher based on a substitution-permutation network [8]. Rijndael was one of the shortlisted algorithms in a formal call by NIST back in 1999. The NIST had selected it to supersede Data Encryption Standard (DES) and protected classified information after the evaluation of different AES criteria [9]. Hence, AES was widely adopted in software and hardware over many corporates or even governments to encrypt their sensitive data in recent decades.

However, there are three block ciphers contained in AES with different key sizes. Each of the block ciphers can encrypt and decrypt in 128-bit blocks data utilising 128-bit, 192-bit, or 256-bit cryptographic keys. The different key sizes chosen will have a different round of operation on that particular cipher. For example, there are ten rounds in AES-128, twelve rounds in AES-192, and fourteen rounds in AES-192. A round is made up of many steps: byte substitution, permutation, mixing columns and adding the round key. Reverse rounds are used in decryption to convert ciphertext back to plaintext using the same encryption key. Figure 2 depicts the operations involved in AES encryption and decryption.

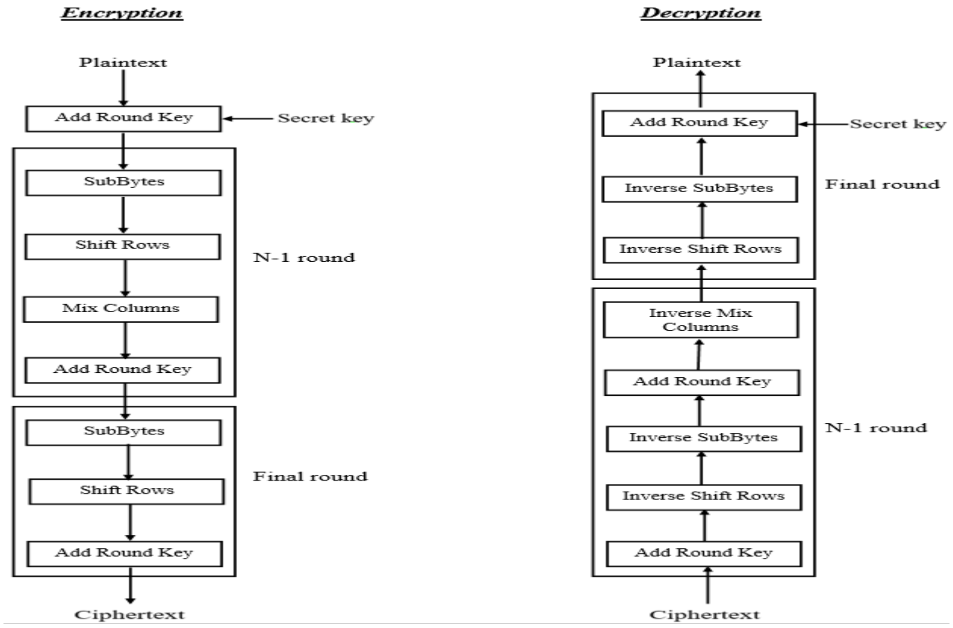


Figure 2. Encryption and Decryption in AES

ElGamal Encryption is a widely used encryption scheme in free GNU Privacy Guard software, Pretty Good Privacy (PGP), and other cryptosystems. ElGamal is an asymmetric key encryption algorithm that encrypts data using the public key and decrypts with the private key based on an Diffie-Hellman key exchange [10]. Although the asymmetric approach requires more intensive calculation than symmetric algorithms, mathematical computations typically take less time. However, the ElGamal scheme is considered robust and harder to break due to the discrete logarithm problem, a known NP-problem. It is difficult to calculate x using a given generator and an equation

$$g^x \, mod \, p \qquad (1)$$

As a result, the scheme becomes more resilient as the discrete logarithm issue becomes more complex.

The US National Security Agency created the Secure Hash Algorithm 2 (SHA-2) family of hashing algorithms to replace the previous hash function, SHA-1 [11]. Hash function provides a collision-resistant hash value of given input as fixed-length output. Even though SHA-2 is much complicated, but it features greater security than SHA-1. There are several hash functions available in SHA-2 series: SHA-224, SHA-256, SHA-384, and SHA-512. The value associated with the hash functions represents the length of the hash value produced. For example, SHA-256 hash values are calculated using 32 bytes, whereas SHA-512 hash values are computed with 64 bytes. SHA-2 is widely adopted with strings in MongoDB, Microsoft SQL Server, and MySQL databases to store user's credentials or sensitive information. SHA-2 is also implemented in protocols and security applications such as TLS as well as SSL.

However, there are a lot of hashing functions available in the market, for instance, MD5, SHA-1, SHA-2, SHA-3 and Whirlpool. [12] conducted a comparative analysis among several hash algorithms with different parameters, as shown in Table 2. Based on the analysis, collisions were found in MD5 and Whirlpool, whereas SHA-1 suffered in the theoretical attack, and a collision was found in [13]. Since collision-free is one of the basic requirements for a hash function, the above mentioned hash functions will no longer be secure. Nevertheless, SHA-2 and SHA-3 are still collision-resistant and perform considerable security in the market. SHA-256 is selected in this system as a balancing of the trade-off between security and efficiency.

Table 2. Hash Algorithm Comparison [12]

Parameters	**MD5**	**SHA-1**	**SHA-2**	**SHA-3**	**Whirlpool**
Block size (bits)	512	512	512, 1024	1600-2*bits	512
Digest size (bits)	128	160	160, 224, 256, 384, 512	160, 224, 256, 384, 512	512
Word size (bits)	32	32	32, 64	64	8
Rounds	4	80	80	24	10
Collision found	Yes	Theoretical attack	None	None	Yes
Operations	and, or, xor, rot	and, or, xor, rot	and, or, xor, rot, shr	and, or, xor, rot, shr	and, or,xor,rot

B. USE CASE DIAGRAM

Figure 3 is a use case diagram that depicts the logic and all functionalities in the developed system. The system is a client-server-based model where the clients can send requests and get a reply from the server accordingly. In our designed system, the clients are able to view, download, and remove their files if they need.

C. SYSTEM FLOWCHART

In the developed system, every authenticated user can store their data or file securely in cloud storage as the data is stored in an encrypted form and can only be accessed by the authorised user. The entire process of the system is categorised into four phases: registration, authentication, encryption, and decryption.

Figure 4 depicts the workflow in the registration process. During registration, the user must provide the necessary information such as email address, password, name, etc. As long as the user clicks the "Register" button, a verification email will be sent to the email address provided in the field to verify the validity. Once verified, the key pair will be generated based on the ElGamal algorithm and stored appropriately in the database. Otherwise, the email address is deemed invalid, and the user needs to provide another valid email address for the registration.

In the authentication process as depicted in Figure 5, users have to enter the validated email and password in order to proceed. If the credentials are verified, OTP will be sent to the registered phone number for verification. The users are granted access after the OTP has been verified; hence a 2FA is completed.

The encryption process as depicted in Figure 6 transforms the original data into ciphertext using the AES algorithm before uploading it to the cloud. Firstly, an authenticated user must select a 128-bit encryption key and a file they would like to store in the cloud server. After that, a hash function SHA-256 is applied to the selected file and the secret key to get its unique fingerprint. Finally, the secret key is encrypted with the user's public key, while the file is encrypted using the secret key.

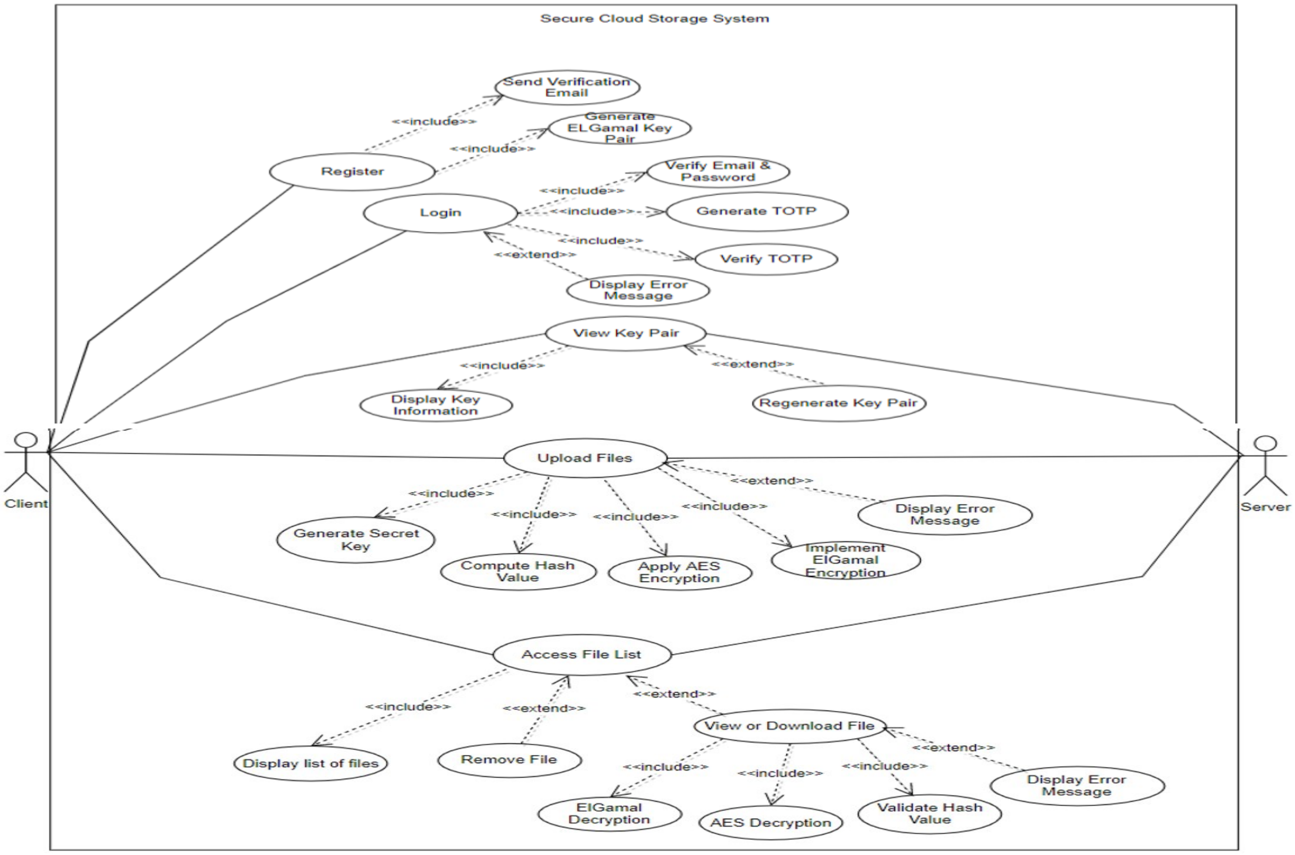


Figure 3. Use Case Diagram

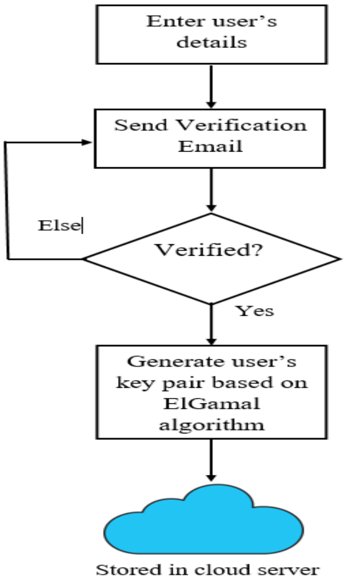


Figure 4. Registration Process

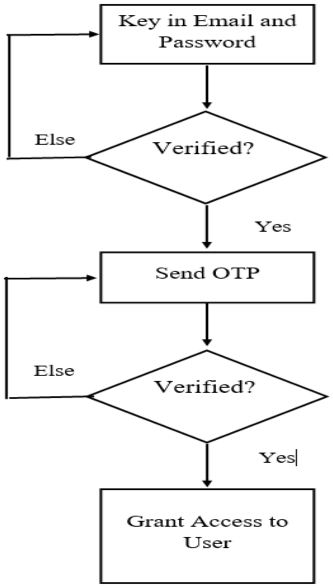


Figure 5. Authentication Process

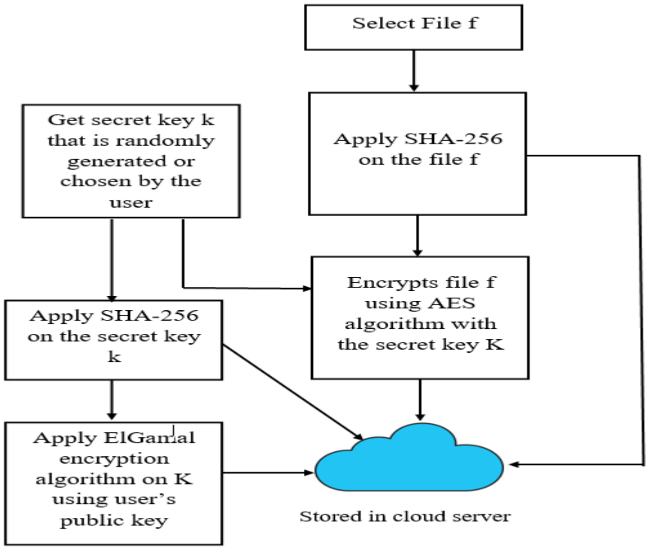

Select File f
Get secret key k that is randomly generated or chosen by the user
Apply SHA-256 on the file f
Encrypts file f using AES algorithm with the secret key K
Apply SHA-256 on the secret key k
Apply ElGamal encryption algorithm on K using user's public key
Stored in cloud server

Figure 6. Encryption Process

In the decryption process as depicted in Figure 7, the user requests the server to retrieve the encrypted file and the secret key. ElGamal decryption is applied to the encrypted key by using the private key of the file owner to recover the secret key. After that, the original file is decrypted through a reverse process of AES using a secret key that is obtained from the previous step. At the end of the decryption process, the system will compute the decoded file's message digest and compare the original message digest to ensure the file is not altered or corrupted during transmission. The file is available for the user to view and download if matched. Otherwise, the file will be discarded, and the user needs to make the request again.

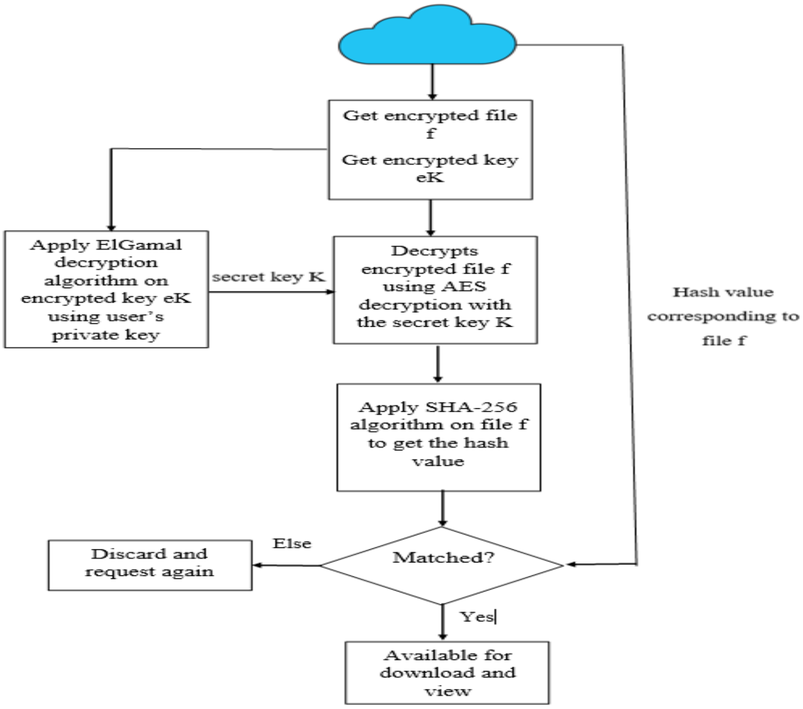

Get encrypted file f
Get encrypted key eK
Apply ElGamal decryption algorithm on encrypted key eK using user's private key
secret key K
Decrypts encrypted file f using AES decryption with the secret key K
Hash value corresponding to file f
Apply SHA-256 algorithm on file f to get the hash value
Discard and request again
Else
Matched?
Yes
Available for download and view

Figure 7. Decryption Process

C. TOOLS AND LANGUAGES

The following language and platforms are used in developing the hybrid encryption web application:

i. PHP

The PHP (Hypertext Preprocessor) is the server-side scripting language used in this web development. PHP is often referred to as a general-purpose scripting language for designing dynamic and interactive web pages. It serves as a server-side language that can integrate with HTML, making a website easier on adding on features without referring to any external files [14].

Moreover, PHP provides an extension that binds OpenSSL library functions for symmetric and asymmetric encryption, decryption, and other cryptographic operations.

ii. FIREBASE REALTIME DATABASE

Firebase is a Backend-as-a-Service (Baas) platform that supports a set of tools and resources to assits developers build high-quality apps. Data is stored in JSON format and synchronised across all linked clients in real-time [15]. The Realtime Database is a NoSQL database, implying it has different optimizations and functionality than relational databases. It will be used to store the user data such as email address, files, and key-pair in this project.

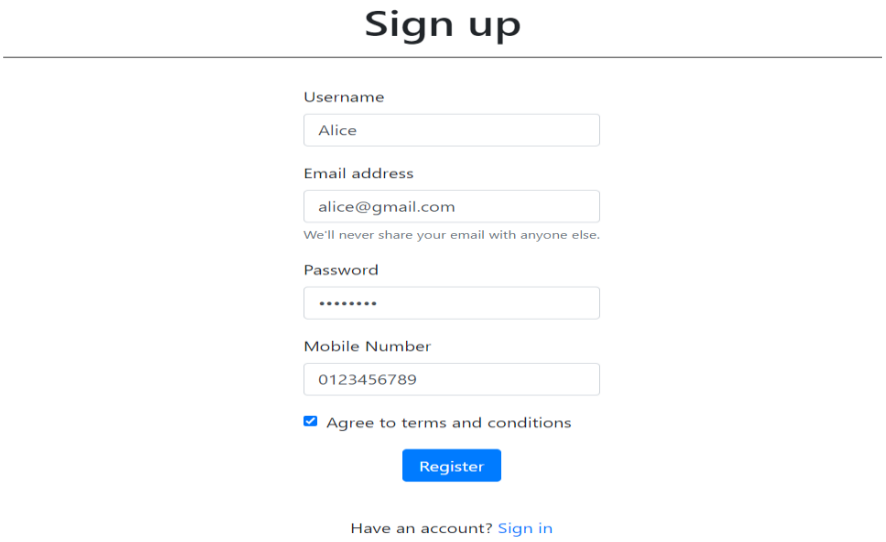
iii. AMAZON EC2

Amazon Elastic Compute Cloud (Amazon EC2) is a service that offers scalable computing resources in the Amazon Web Services (AWS) Cloud. It provides and launches the virtual server and allows customers to configure storage, processing speed, and networking according to their needs. The cost of acquiring hardware in an organisation or business has dramatically reduced by setting up virtual servers on the cloud. Hence, Amazon EC2 will be utilised in this application for website hosting purposes.

IV. PROPOSED SOLUTION

A. USER INTERFACE AND FUNCTIONALITY

Figure 8 is the registration page for users to register their accounts. Users must provide information such as username, email address, password, and phone number to proceed with the registration. Once registered successfully, a notification will be prompted. They will receive a verification email associated with a link for verification. After the link is clicked, the user's account is verified, and the user is able to sign in.



Sign up

Username: Alice

Email address: alice@gmail.com

We'll never share your email with anyone else.

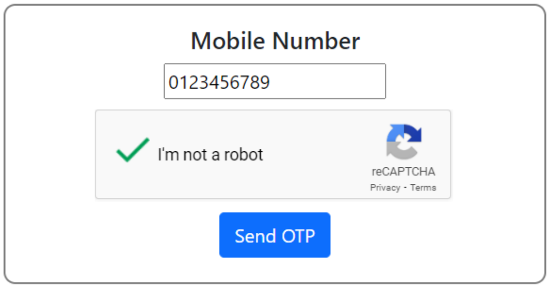
Password: ••••••••

Mobile Number: 0123456789

☑ Agree to terms and conditions

Figure 8. Sign up Page

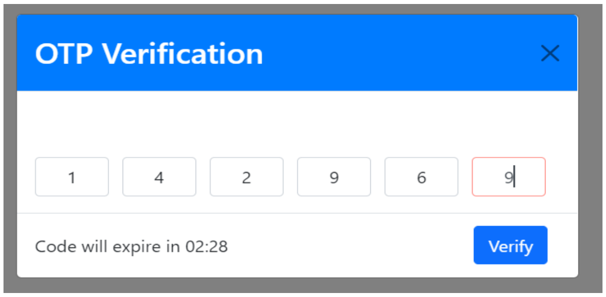
In this system, a Two-Factor Authentication (2FA) with the combination of password and Time-based One Time Password (TOTP) is applied (see Figure 9). The system will send a 6-digits password that lasts for 3 minutes to the user's phone number, which is registered in the registration process. The user is only granted access where the TOTP is verified; otherwise, the TOTP will be resent again (see Figure 10).



Mobile Number: 0123456789

✓ I'm not a robot

Figure 9. TOTP Verification Page



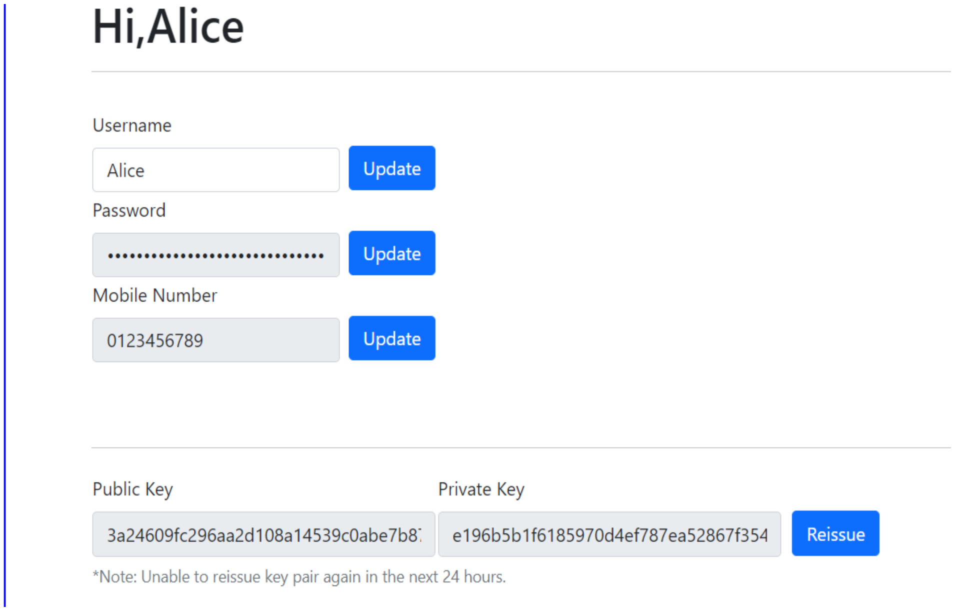
OTP Verification

1 4 2 9 6 9

Code will expire in 02:28

Figure 10. Verifying TOTP

After users have successfully signed in, they will be navigated to the page that shows their information. Besides, users can update their username, password, phone number and reissue their key pair (refer to Figure 11). In order to ensure security, the users should validate their current password before replacing the new password. As for phone number, users need to verify the new phone number through TOTP before updating. Furthermore, the key pair can be reissued every 24 hours if they think their key might not be secured anymore.


Hi,Alice

Username
Alice

Password
••••••••••••••••••••••••••••••

Mobile Number
0123456789

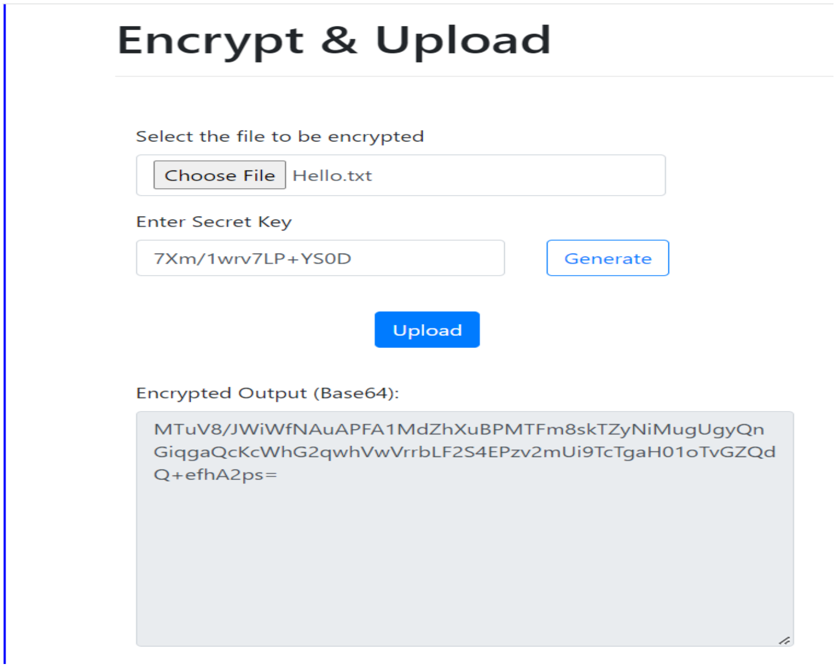
Public Key
3a24609fc296aa2d108a14539c0abe7b8

Private Key
e196b5b1f6185970d4ef787ea52867f354

*Note: Unable to reissue key pair again in the next 24 hours.

Figure 11. Profile Page

To upload a file to the server, users must select the file they want to upload and provide a 128-bit secret key for the encryption process. The encrypted data will be displayed in base 64 format after the file has been successfully uploaded (refer to Figure 12).


Encrypt & Upload

Select the file to be encrypted
Hello.txt

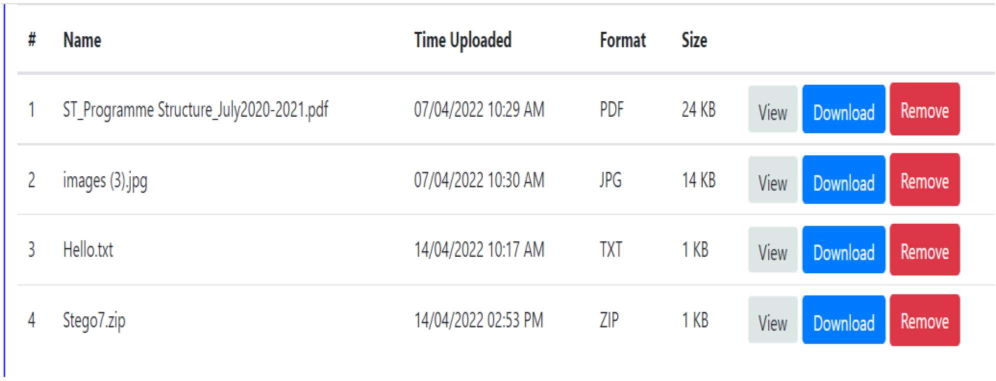
Enter Secret Key
7Xm/1wrv7LP+YS0D

Encrypted Output (Base64):

MTuV8/JWiWfNAuAPFA1MdZhXuBPMTFm8skTZyNiMugUgyQn
GiqgaQcKcWhG2qwhVwVrrbLF2S4EPzv2mUi9TcTgaH01oTvGZQd
Q+efhA2ps=

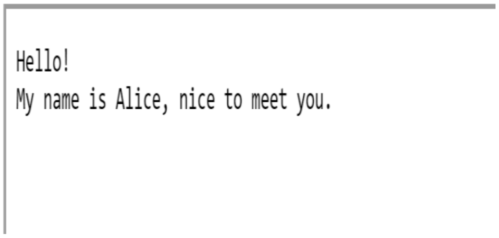
Figure 12. Uploading File

Once users have uploaded their files, they could view their files by clicking the "View" button as shown in Figure 13. For example, the content of file is displayed in Figure 14 after the decryption process. The decryption process is performed whenever the owner requests only; the content stored in the cloud server will always remain encrypted. However, not all types of files are supported on the browser. Hence, an alert message might be prompted, if the file type is not supported, such as .docx, .xlsx, .zip, and so forth.



#	Name	Time Uploaded	Format	Size	
1	ST_Programme Structure_July2020-2021.pdf	07/04/2022 10:29 AM	PDF	24 KB	View Download Remove
2	images (3).jpg	07/04/2022 10:30 AM	JPG	14 KB	View Download Remove
3	Hello.txt	14/04/2022 10:17 AM	TXT	1 KB	View Download Remove
4	Stego7.zip	14/04/2022 02:53 PM	ZIP	1 KB	View Download Remove

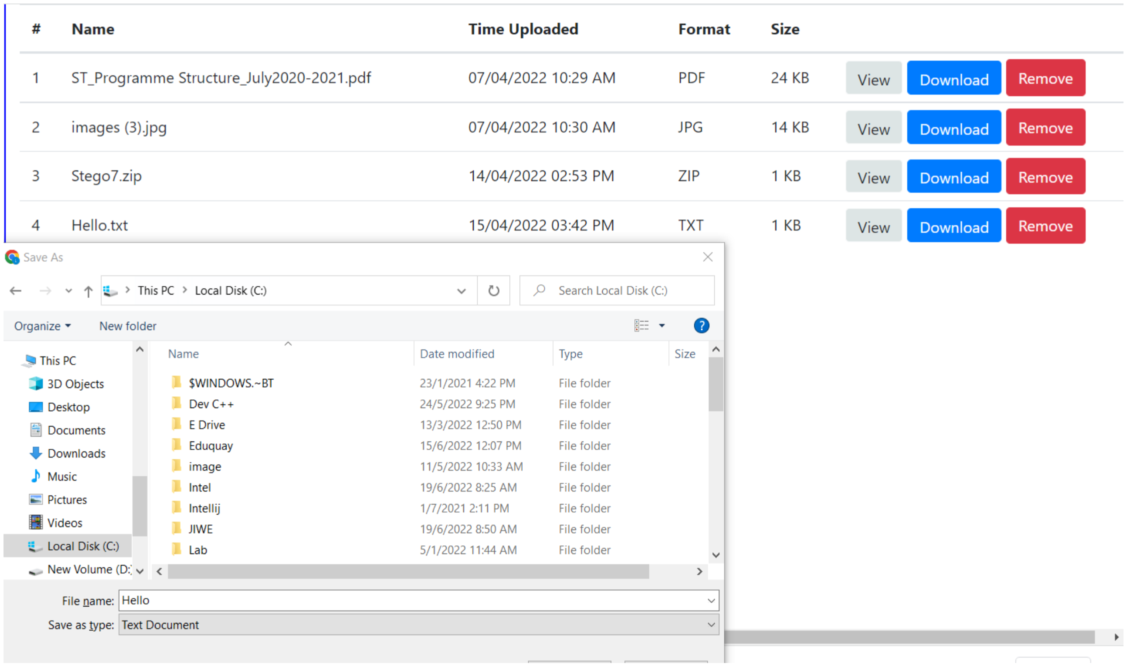
Figure 13. User's File List



```
Hello!
My name is Alice, nice to meet you.
```

Figure 14. Decrypted File Content

The text file named "Hello.txt" is selected to download as shown in Figure 15. After that, the text file is successfully downloaded to the user's C drive with its original content.



#	Name	Time Uploaded	Format	Size	
1	ST_Programme Structure_July2020-2021.pdf	07/04/2022 10:29 AM	PDF	24 KB	View Download Remove
2	images (3).jpg	07/04/2022 10:30 AM	JPG	14 KB	View Download Remove
3	Stego7.zip	14/04/2022 02:53 PM	ZIP	1 KB	View Download Remove
4	Hello.txt	15/04/2022 03:42 PM	TXT	1 KB	View Download Remove

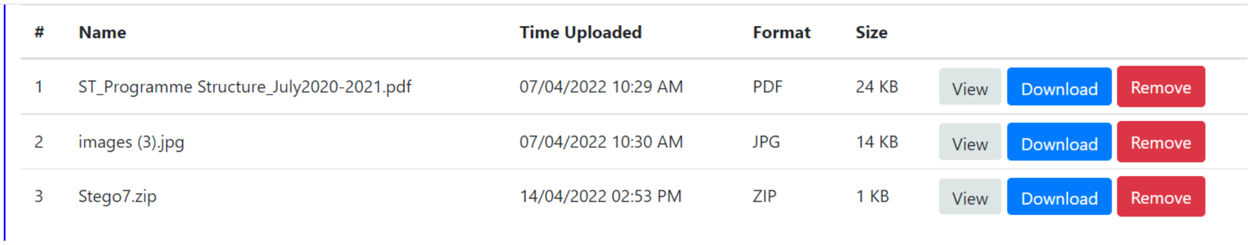
Name	Date modified	Type	Size
$WINDOWS.~BT	23/1/2021 4:22 PM	File folder	
Dev C++	24/5/2022 9:25 PM	File folder	
E Drive	13/3/2022 12:50 PM	File folder	
Eduquay	15/6/2022 12:07 PM	File folder	
image	11/5/2022 10:33 AM	File folder	
Intel	19/6/2022 8:25 AM	File folder	
Intellij	1/7/2021 2:11 PM	File folder	
JIWE	19/6/2022 8:50 AM	File folder	
Lab	5/1/2022 11:44 AM	File folder	

File name: Hello

Save as type: Text Document

Figure 15. Downloading File

Since removing or deleting the data is a sensitive action, a confirmation box will be prompted to the user before performing any deletion. If the "OK" button is clicked, the selected file will be removed and no longer be reverted. Figure 16 shows the updated file list upon performing file deletion.



#	Name	Time Uploaded	Format	Size	
1	ST_Programme Structure_July2020-2021.pdf	07/04/2022 10:29 AM	PDF	24 KB	View Download Remove
2	images (3).jpg	07/04/2022 10:30 AM	JPG	14 KB	View Download Remove
3	Stego7.zip	14/04/2022 02:53 PM	ZIP	1 KB	View Download Remove

Figure 16. Updated File List

B. SECURITY OF THE PROPOSED SOLUTION

The proposed hybrid encryption scheme aims to resolve the security concerns in cloud storage: data confidentiality, integrity, and authentication. Although the proposed scheme is similar to the existing technologies such as HTTPS and PGP, however, the implementation process is slightly different. In this proposed scheme, any authorised user could safely save their data or file in cloud storage since the data is encrypted and can only be viewed and downloaded by the authorised user, which eliminates privacy concerns. With the help of AES encryption, the original data is transformed into an unintelligible form with respective secret key information before uploading to the cloud. In addition, the respective secret key is encrypted using the owner's ElGamal public key, which is generated in the registration process. To retrieve the original data, the user must select the file that appears in the list and provide his own ElGamal private key. Then a decryption process will be performed automatically to convert back the original data and match its hash value with the hash value computed that stored in the database during the upload process. The file will only be available for download where it is successfully decrypted and validated by SHA-2 algorithm. It is to ensure that data has not been modified or corrupted during transmission. Furthermore, the data stored in the cloud may be extensively sensitive and confidential for individuals. Hence, the 2FA method is also implemented in this proposed scheme against any unauthorised access from third-parties. Whenever users want to sign in, they have to insert a six-digit OTP, which lasts for three minutes after the system has verified the identity they claimed to be.

V. RESULTS AND DISCUSSIONS

The experimental evaluation and performance are computed and compared among different schemes proposed by other researchers. The assessed criteria in this experiment include the encryption and decryption speed, memory consumption and security. Hence, the JAVA programmes for each hybrid scheme have been developed for analysis purposes. The evaluated hybrid schemes are Blowfish & RSA [3], AES & RSA [5], 3DES & ElGamal [6], and AES & ElGamal (Proposed Scheme). Table 3 depicts the encryption time result for the various schemes.

Table 3. Encryption Time Result (ms)

File Size/Scheme	Blowfish & RSA	AES & RSA	3DES & ElGamal	Proposed Scheme
1 KB	15.8	23.8	21	24.2
19 KB	80.4	93.6	86	83.8
36 KB	153.0	165.2	147.6	135.0
99 KB	387.8	411.8	415.0	377.8
465 KB	1580.0	1629.8	1764.2	1525.2
2305 KB	8104.2	7517.2	8008.4	7019.6
5799 KB	20707.6	17620.6	19448.6	17016.6

Figure 17 contains data of different file sizes in the X-axis; Y-axis includes the time taken in milliseconds. There are four colors of the line representing various hybrid schemes, as shown in the legend. The amount of time depends on the data. To approximate comparative performance, Figure 18 provides the mean performance of algorithms. According to mean performance, the encryption results show that the AES & ElGamal which is the proposed scheme consumes less time during encryption than others.

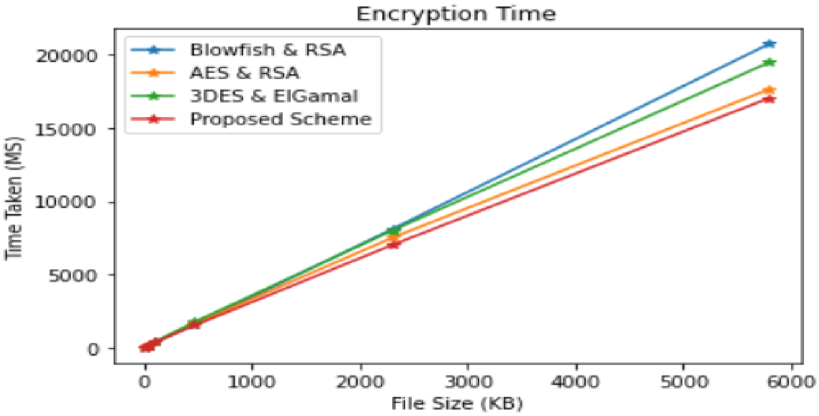


Figure 17. Encryption Time Taken

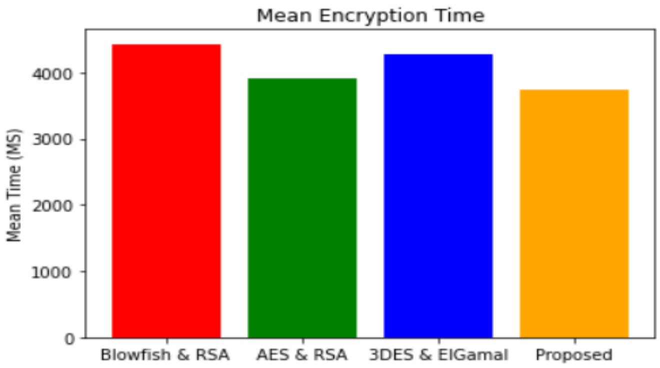


Figure 18. Mean Encryption Time

Meanwhile, Table 4 depicts the decryption time for various schemes. The results as defined in Figure 19 show that the decryption time of the proposed scheme is more efficient than others. According to the mean decryption time in Figure 20, the proposed scheme achieved the least time (3663 ms), which is slightly faster than AES & RSA (3788 ms), it also outperformed Blowfish & RSA (4155 ms) and 3DES & ElGamal (4165 ms).

Table 4. Decryption Time Result (ms)

File Size/Scheme	Blowfish & RSA	AES & RSA	3DES & ElGamal	Proposed Scheme
1 KB	77.4	78.8	6.6	6.8
19 KB	170.4	152.4	78.6	78.4
36 KB	197.4	186	160.0	122.0
99 KB	410.0	390.6	363.8	328.4
465 KB	1652.0	1532.6	1569.4	1419.8
2305 KB	7459.4	7078.0	7519.6	6942.6
5799 KB	19120.0	17099.0	19457.0	16746.0

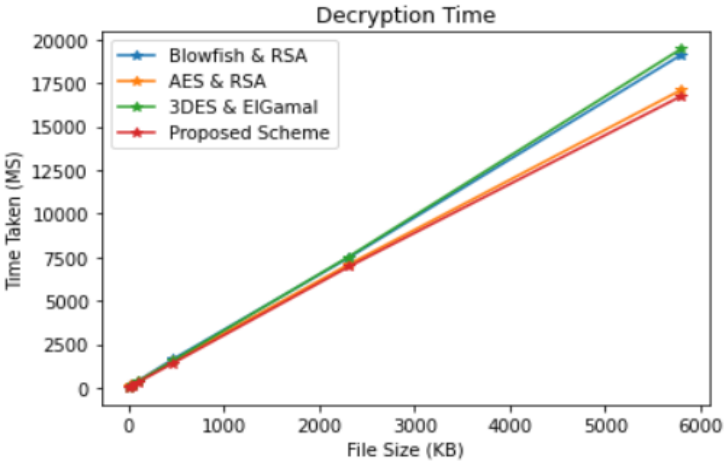


Figure 19. Decryption Time Taken

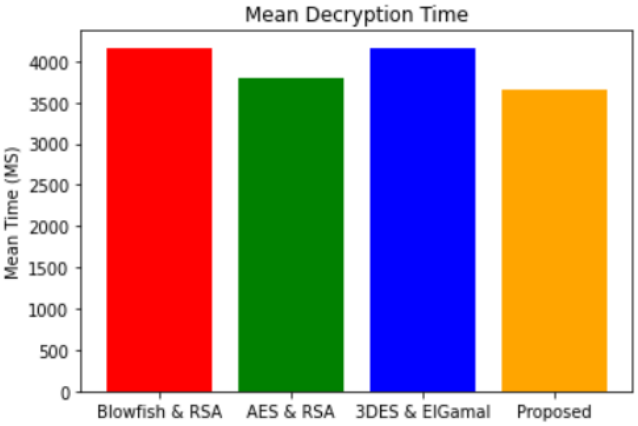


Figure 20. Mean Decryption Time

Figure 21 depicts the mean result of the memory consumption in kilobytes from the different combinations of algorithms. Based on our observation, the memory consumed from the discrete logarithm (ElGamal) is relatively less than the factorisation (RSA). Furthermore, the AES algorithm also consumes less memory as compared to the 3DES algorithm.

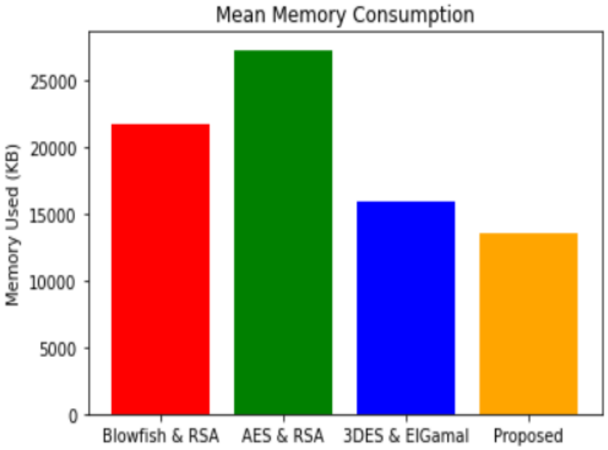


Figure 21. Mean Memory Consumption (KB)

Table 5 concludes the performance and evaluation of the analysis. This analysis has been assessed among multiple hybrid schemes proposed by other researchers and a proposed scheme in this paper. Each scheme is evaluated in the same environment regardless of CPU, RAM, and File size. From the obtained results, 3DES & ElGamal and our proposed scheme achieved better efficiency than Blowfish & RSA and AES & RSA as factorisation is a more consuming process compared to discrete logarithm. However, the overall criteria of the proposed scheme (AES & ElGamal) outperformed other schemes in terms of encryption speed, decryption speed, memory consumption and security.

Table 5**.** Performance Summary

File Size/Scheme	**Blowfish & RSA**	**AES & RSA**	**3DES & ElGamal**	**Proposed Scheme**
Speed	Slow	Moderate	Slow	Fast
Memory Consumption	Moderate	High	Low	Low
Key Size	448/2048 bits	128/2048 bits	192/2048 bits	128/2048 bits
Security	High	High	Moderate	High

VI. CONCLUSION

The security issues on the cloud are worth paying attention to due to the significance of information and the variety of services provided to the users. Hence, this paper has proposed a hybrid encryption approach to secure user data in a cloud environment. The proposed hybrid encryption scheme aims to resolve the security concerns in cloud storage: data confidentiality, integrity, and authentication.

In this proposed scheme, any authorised user could safely save their data or file in cloud storage since the data is encrypted and can only be viewed and downloaded by the authorised user, which eliminates privacy concerns. With the help of AES encryption, the original data is transformed into an unintelligible form with respective secret key information before uploading to the cloud. In addition, the respective secret key is encrypted using the owner's ElGamal public key, which is generated in the registration process. To retrieve the original data, the user must select the file that appears in the list and provide his own ElGamal private key. Then a decryption process will be performed automatically to convert back the original data and match its hash value with the hash value computed that stored in the database during the upload process. The file will only be available for download where it is successfully decrypted and validated by SHA-2 algorithm. It is to ensure that data has not been modified or corrupted during transmission. Furthermore, the data stored in the cloud may be extensively sensitive and confidential for individuals. Hence, the 2FA method is also implemented in this proposed scheme against any unauthorised access from third-parties. Whenever users want to sign in, they have to insert a six-digit OTP, which lasts for three minutes after the system has verified the identity they claimed to be.

In a nutshell, efficient and secure file storage on the cloud using AES & ElGamal algorithms has been proposed and developed. Based on the result of the analysis conducted, the proposed scheme has achieved better efficiency and security compared to other schemes. Furthermore, various types of file such as .txt, .pdf, .jpeg, .docx, .pptx, and .mp4 have been tested, reconstructed and retrieved successfully during the testing process. Hence, the hybrid technique for file encryption and decryption will adapt well in applications where security demand is a requirement. This hybrid scheme will contribute to improving the security of cloud storage over the internet.

ACKNOWLEDGEMENT

This work was supported by the Ministry of Higher Education of Malaysia's Fundamental Research Grant Scheme (FRGS/1/2018/ICT04/MMU/01/01).

REFERENCES

[1] R. Kaur and J. Kaur, "Cloud computing security issues and its solution: A review", 2nd International Conference on Computing for Sustainable Global Development (INDIACom), pp. 1198-1200, 2015.

[2] A. J. Nathan and A. Scobell, "2020 Data Breach Investigations Report", Verizon, 2020.

[3] D. P. Timothy and A. K. Santra, "A hybrid cryptography algorithm for cloud computing security", 2017 International Conference on Microelectronic Devices, Circuits and Systems (ICMDCS), pp. 1-5, 2017.

[4] P. Loshin, "Selected FAQs on using GnuPG", Simple Steps to Data Encryption: A Practical Guide to Secure Computing, pp. 11-21, 2013.

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

[6] E. Jintcharadze and M. Iavich, "Hybrid implementation of Twofish, AES, ElGamal and RSA Cryptosystems", 2020 IEEE East-West Design and Test Symposium (EWDTS), pp. 1-5, 2020.

[7] O. Alabi, A. Thompson, B. K. Alese and A. J. Gabriel, "Cloud application security using hybrid encryption", Communications on Applied Electronics, vol. 7, no. 33, pp.25-31, 2020.

[8] J. Daemen and V. Rijmen, “The Block Cipher Rijndael”, Lecture Notes in Computer Science, vol 1820. Springer, Berlin, Heidelberg, 1998. https://doi.org/10.1007/10721064_26

[9] J. Nechvatal, E. Barker, L. Bassham, W. Burr, M. Dworkin, J. Foti and E. Roback, “Report on the development of the Advanced Encryption Standard (AES)”, Journal of Research of the National Institute of Standards and Technology, vol. 106, no.3, pp. 511-577, 2001.

[10] T. Elgamal, “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469-472, 1985.

[11] W. Penard and T. V. Werkhoven, “On the secure hash algorithm family”, Cryptography in Context, pp. 1-17, 2008.

[12] P. P. Pittalia, “A comparative study of hash algorithms in cryptography”, International Journal of Computer Science and Mobile Computing, vol. 8, no. 6, pp. 147-152, 2019.

[13] M. Stevens, E. Bursztein, P. Karpman, A. Albertini and Y. Markov, “The first collision for full SHA-1”, Lecture Notes in Computer Science, vol. 10401, pp 570–596, 2017.

[14] C. R. Severance, “Inventing PHP: Rasmus lerdorf”, Computer, vol. 45, no. 11, pp. 6-7, 2012.

[15] L. Moroney, “The firebase realtime database”, The Definitive Guide to Firebase, pp. 51-71, 2017.