Internet of Things and Cloud Computing
2024; 12(1): 1-9
http://www.sciencepublishinggroup.com/j/iotcc
doi: 10.11648/j.iotcc.20241201.11
ISSN: 2376-7715 (Print); ISSN: 2376-7731 (Online)

File Storage Security in Cloud Computing Using Hybrid Encryption

Chidi Ukamaka Betrand[*], Chinwe Gilean Onukwugha, Mercy Eberechi Benson-Emenike, Christopher ifeanyi Ofoegbu, Nneka Martina Awaji

Department of Computer Science, School of Information and Communication Technology, Federal University of Technology, Owerri, Nigeria

Email address:
chidi.betrand@futo.edu.ng (Chidi Ukamaka Betrand)
[*]Corresponding author

To cite this article:
Chidi Ukamaka Betrand, Chinwe Gilean Onukwugha, Mercy Eberechi Benson-Emenike, Christopher ifeanyi Ofoegbu, Nneka Martina Awaji. (2024). File Storage Security in Cloud Computing Using Hybrid Encryption. *Internet of Things and Cloud Computing*, *12*(1), 1-9. https://doi.org/10.11648/j.iotcc.20241201.11

Received: December 16, 2023; **Accepted**: December 28, 2023; **Published**: January 11, 2024

Abstract: Cloud is used in various fields for storage of big data with the major challenge regarding this storage being security. Existing conventional encryption systems can be vulnerable to brute-force attacks. The goal of this project was to develop a hybrid encryption scheme that will only allow authorized users to access and download files stored online, thus enhancing file storage security in the cloud. Rapid Application Development (RAD) methodology was used to create the proposed system, allowing for modifications to be made to the system as it was being developed. The hybrid encryption scheme employs both symmetric and asymmetric encryption. The AES (Advanced Encryption Standard) algorithm and RSA (Rivest–Shamir–Adleman) algorithm were combined to develop the proposed hybrid encryption system. PHP, JavaScript and Laravel were the programming languages and web framework used to implement the system. The proposed system was tested and evaluated by users. The experimental results show that the proposed hybrid encryption scheme was fast and provided a high level of security but had some drawbacks which include increase in file size after it was encrypted and inability to sort files in the web app. Overall, the proposed system enhances confidentiality and data protection in cloud environments, guarding against potential breaches and unauthorized access.

Keywords: Cloud Computing, Cryptography, Hybrid Encryption, Data Integrity, Access Control

1. Introduction

Cloud computing has emerged as a dominant paradigm in the field of information technology, offering immense storage capacity and as well as offer great deal of access to online resources and services. With the increasing adoption of cloud-based solutions, concerns regarding data security have become more critical than ever. As organizations and individuals entrust their valuable data to cloud service providers, it is important to ensure adequate stategies are put up in order to make sure there's no unauthorized data access. [1] One of the primary challenges in cloud computing is file storage security. Traditional methods of data encryption, such as symmetric and asymmetric encryption, limitations when it comes to cloud storage. Asymmetric encryption employs different keys for encryption and decryption than symmetric encryption, which depends on a same key for both processes. But there are issues with computational overhead, scalability, and key management with both approaches [2].

Harnal *et al.* [3] presented an approach for maintaining integrity and confidentiality of multimedia in a cloud computing environment. This work has provided a secure storage and transmission of multimedia files across the internet with high level of integrity and confidentiality. Orobosade *et al* [4] on their work, proposed a hybrid encryption model based on Advanced Encryption Standard (AES) and Elliptic Curve Cryptography (ECC) to preserve the privacy and safety of users in the cloud. Their study explored more at the use of hybrid encryption for file storage security in cloud computing. The research will involve a detailed analysis of different hybrid encryption algorithms, considering factors such as security, performance, key management, and scalability. Additionally, the study will investigate the practical implementation of hybrid encryption in a cloud storage environment, examining the impact on data confidentiality, integrity, and availability.

2. Literature Review

The art of cryptography involves securing data by transforming it into several formats that are unknown to an attacker, even in the event that the data is compromised. It causes the data to become jumbled for misuse.[5]. Poduval et al [6] proposed a novel security technique that combines a number of cryptographic algorithms, including steganography and symmetric key cryptography. The algorithms RC6, AES, and 3DES are employed to protect data. All the algorithms utilized made use of 128-bit keys. The LSB steganography was used to store delicate data.. Encryption splits a file into three parts. Thanks to multithreading, these distinct file segments will all be encrypted concurrently using various encryption techniques. The LSB method adds the important information to a picture. The AES, DES, and RC6 algorithms were used to store the encrypted data on a single cloud server. Multiple layers of multi-coding that combine multiple techniques to increase privacy, such as adding Advanced Encryption Standard (AES) and loading the data into cloud storage, as well as applying DNA encryption to gain an even higher level of anonymity. They implemented and examined the suggested method using MATLAB. It has demonstrated that the security, efficiency, complexity, and speed of the suggested approach are realistic [7].

In order to prevent insider attacks, an encryption technique for large data storage in multi-cloud storage was created [9]. The data uploading, slicing, indexing, encryption, dissemination, decryption, retrieval, and merging processes are all included in the suggested framework. Using real-time cloud storage environments, the simulation study records the encryption process at about 2630 KB/S. The outcomes demonstrate the suggested algorithm's superiority over the relevant mark algorithms. For cloud storage, an improved RDPC protocol that verifies data integrity was created [10]. Their homomorphic hash algorithm-based research allows for dynamic operations at the block level, including insert, update, delete, and modify. The Merkle Hash Tree is utilized to assist in locating each dynamic operation, and an independent auditor verifies the accuracy of the data provided by the user and stores it on a cloud server.

To securely access and store data on the cloud, an effective two-stage cryptography system was presented [8]. It is made up of encryption and user authentication procedures. To overcome the shortcomings in the current authentication techniques, a one-time password, two-factor authentication scheme is suggested. The strategy applied by the researchrers has no difficulty in identifying users. They divided their plaintext into two parts, each of which is keyed differently for encryption. The logistic chaos model's assumption wass used to develop the keys. Their plan introduces several security procedures with varying stages, guaranteeing a high degree of security. Their simulation findings demonstrated that compared to earlier techniques, the suggested scheme minimizes the amount of the ciphertext and the times required for encryption and decryption.

The file that the user needs to store in cloud storage is checked to see whether it already exists on the cloud server using the de-duplication method. Their plan works well and can withstand a rogue server's replace assault.

2.1. File Storage in Cloud

Files stored in cloud gives mulltiple users access to the same file. With the data stored in data centers(cloud), users can access them online, as an alternative to locally storing them on a network-attached storage (NAS) device. Cloud-based file storage also eliminates the need for people and organizations to budget for the installation, upkeep, and staff required to operate it, as well as to replace their storage hardware every three to five years.

Rather, customers merely pay a fixed monthly or yearly subscription cost to a cloud storage service. Companies can reassign these technical resources to more revenue-generating divisions of their company or cut back on their IT staff.

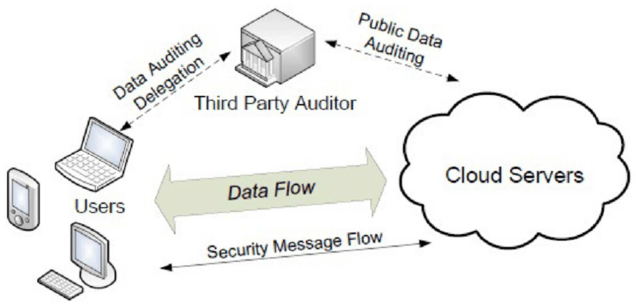


Figure 1. *The architecture of cloud data storage by Hualong Wu, 2016.*

By storing file data on the cloud, users can increase capacity on demand. Users can dynamically extend data quantities and change IOPS for a fixed monthly charge, only paying for what they use.(IBM, 2023)

Since the cloud platform lacks a defined infrastructure and security limits, some of the security difficulties in cloud computing were investigated. Isolating a specific compromised resource in the event of a security incident is challenging. Furthermore, the study found that user data might be accessed by unauthorized users because of the openness of the cloud and multitenant sharing of virtualized resources [11].

In cloud computing, data is spread and segmented while it is in transit. The computer technology's weakness and vulnerability—in particular, its susceptibility to reaction assaults, third-party attacks, sniffing, and spoofing—present additional hazards. Launching a malicious service instance or virtual machine instance into the cloud system by an intruder is another potential attack vector [12] The adversary then deceives the system into believing that the instance is legitimate. As a result, when a user request is received, the cloud service provider might automatically route it to a malicious service instance, which when used maliciously contaminates the entire system [13]. An attribute-based encryption (ABE) based on hybrid cryptography. This system was designed to offer more flexible and secure methods for access control and data exchange, as well as fine-grained access control to encrypted data [14].

2.2. Review of Related Works

In order to create a hybrid algorithm with strong data security and confidentiality in a cloud computing environment, a new hybrid approach merged Blowfish with ECC. A comparison of this system output to others gave a clear understanding that strategy guarantees the highest level of user data security and secrecy [15]. Hybrid system as an encryption method that makes use of the benefits of both symmetric and asymmetric key encryption schemes [16] utilized the digital signature technique (DSA) and the Blowfish symmetric algorithm to encrypt the secret key and encapsulate the data, respectively. This system used the blowfish method to encrypt the data block and the public key of RSA to encrypt the symmetric key prior to executing the key exchange.

AES, Twofish, RSA, and ElGamal are the four well-known algorithms that the researchers [17] used in their hybrid implementation. To investigate the performance of each hybrid strategy, a Java program was developed. The combination of AES and ElGamal produced faster encryption times when encrypting larger files, according to their experimental results.

Improving security with the integration of AES and ECC in a scenario where there exist no trusted center, Distribution as well as management was done using Shamir Secret-sharing (SSS)[18]. The recommended combination approach still needs a lot of time and processing power, even though it increases system security. Yahia et al. (2021) [19] devised a hybrid encryption approach that made use of the AES, DES, and Blowfish algorithms. These algorithms offer dependable and effective data storage, avoiding disputes between multiple users and safeguarding the private information of each individual user. Access to data was managed by the service provider even as the impact off plain text and data block size was measured.

ECC and RSA algorithms are combined in the hybrid encryption technique that Madhaviet al. (2020), [20] suggested. The data over 264 bits is used to combine the security strength of RSA and ECC, while the remaining 256 bits of data adhere to NIST rules. The oprations of the algorithms show a better performance of ECC to RSA method offering a more secured services on lesser datasets. A hybrid encryption method, [21] was proposed, This method guarantees that clients/users will use the service and will follow through in a helpful manner. The hybrid cloud method, which offers excellent security and is appropriate for the current circumstances, can be advantageous for cloud data security. The elliptical curve that follows the hybrid algorithm and polynomial-based hashing have given the system and its users vital security measures for improved and effective services.

Kaur (2017), [22] used blowfish and MD5 techniques to create a hybrid encryption scheme. Sharma et al. (2020), [23], introduced a hybrid method that integrates the AES, AES, DES, and RC4 algorithms. In order to encrypt a file, it must first be downloaded and then combined with another file before being submitted.

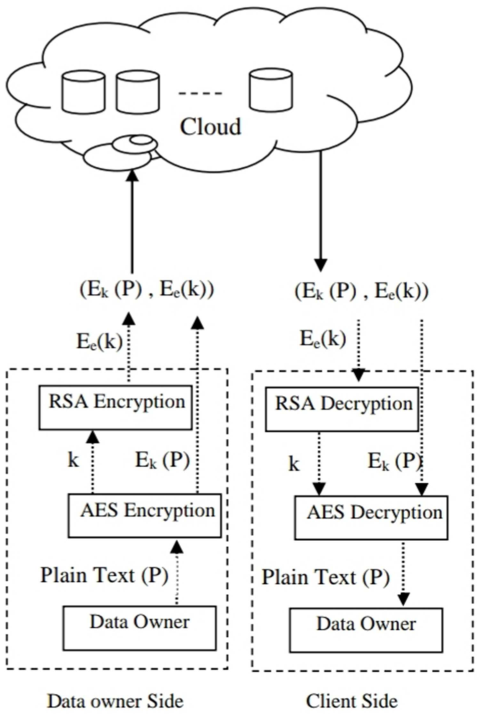


Figure 2. *Physical design of the hybrid model.*

A two-level cryptographic approach and a methodology for strengthening information security in cloud computing were presented by Hodowuet al. (2020) [24]. The methodology fortifies information security against hackers and keeps them from obtaining valid data by utilizing both symmetric and asymmetrical encryption calculations (AES and ECC). This improves confidentiality, data trustworthiness, and the processing time of cryptographic operations. It also increases customer confidence in cloud computing and speeds up the adoption of smaller ECC keys in cryptographic transactions.

3. System Model and Framework

Hybrid encryption process

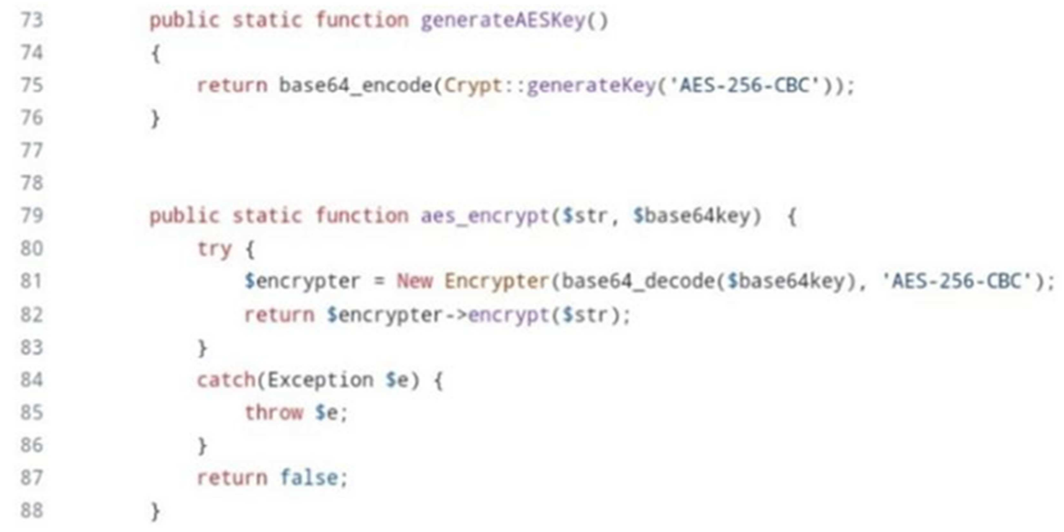
The hybrid encryption process of the proposed system involves four steps: generating the symmetric key, encrypting the file with the symmetric key, encrypting the symmetric key with the RSA private key, and decrypting the file with the RSA public key. AES and RSA encryption and decryption typically requires a library or tool that provides cryptographic transformations and handles the details of the encryption process. PHP libraries are used to simplify AES and RSA encryption and decryption operations. An RSA key pair, made up of a public key and a private key, is produced by the RSA algorithm. Those who are authorized can share the public key. The file uploader is the only one who knows the private key, which is kept private. The user uses the valid RSA public key to decrypt the encrypted AES key in order to access stored files in the suggested system. The primary encrypted file will be decrypted using this recovered AES key. The proposed system was built using PHP, Laravel, and JavaScript. A sample file was encrypted and successfully uploaded and downloaded from the cloud after providing the necessary keys including the AES symmetric key and RSA private and public keys.

The proposed system stores files on the cloud. In the case of data leakage from the cloud, the fragments from the file cannot be restored as it encrypted with AES. It is useless to try to decrypt these bits without the key. As a result, the keys' security determines the system's security.

The proposed system is not susceptible to unauthorized access as it stores only encrypted files. Even the database administrator cannot access or decrypt the file as far as he does not have the user's RSA public key. The fact that RSA is used to encrypt the AES symmetric key, even if a hacker manages to get access to the proposed system by stealing a user's login credentials, he will be unable to access the data. Before attempting to brute-force the AES key that is used to encrypt the file, the hacker would need to crack the RSA encryption.

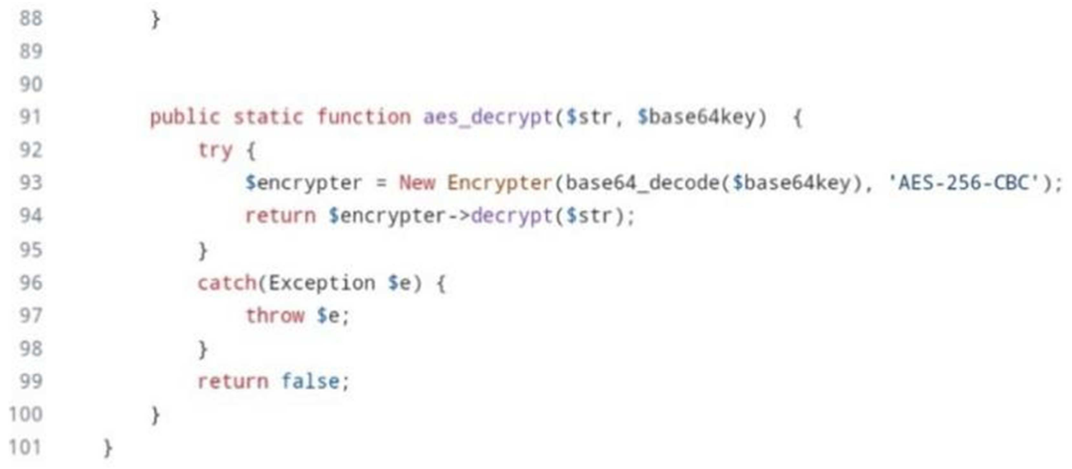
The symmetric block cipher utilized in the suggested system for both file encryption and decryption is AES-256.

4. Result Discussion and Performance Evaluation



```
        public static function generateAESKey()
        {
            return base64_encode(Crypt::generateKey('AES-256-CBC'));
        }


        public static function aes_encrypt($str, $base64key)  {
            try {
                $encrypter = New Encrypter(base64_decode($base64key), 'AES-256-CBC');
                return $encrypter->encrypt($str);
            }
            catch(Exception $e) {
                throw $e;
            }
            return false;
        }
```

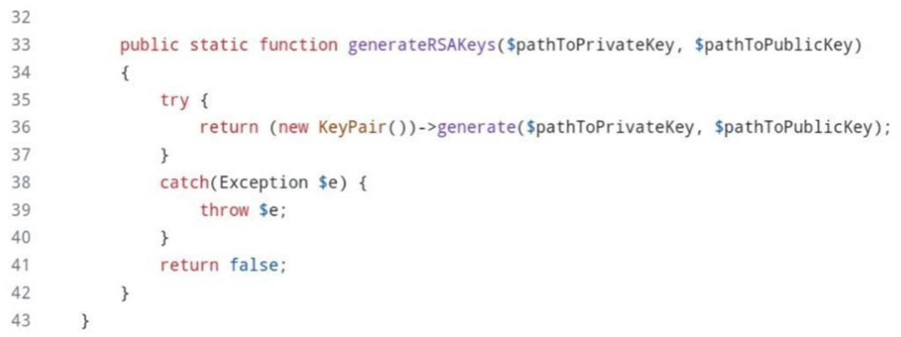
Figure 3. *AES encryption function in PHP.*



```
        }


        public static function aes_decrypt($str, $base64key)  {
            try {
                $encrypter = New Encrypter(base64_decode($base64key), 'AES-256-CBC');
                return $encrypter->decrypt($str);
            }
            catch(Exception $e) {
                throw $e;
            }
            return false;
        }
    }
```

Figure 4. *AES decryption function in PHP.*

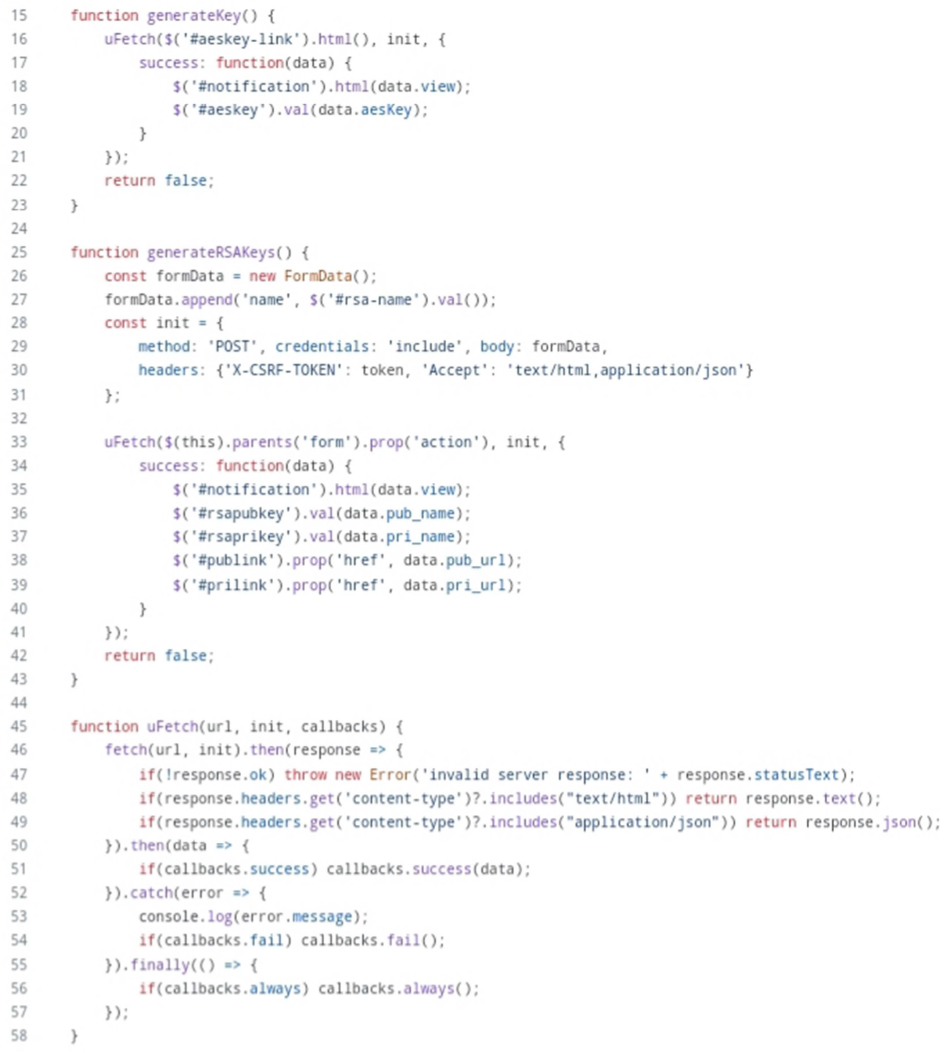
Space\Crypto\Rsa\RSA private/public key pairs were simply generated using the KeyPair package, and files could then be encrypted and decrypted using those keys. Using the generate function on the KeyPair class, a key pair was created. The AES key is encrypted using the produced private key, and the file is decrypted using the public RSA key.



```php

        public static function generateRSAKeys($pathToPrivateKey, $pathToPublicKey)
        {
            try {
                return (new KeyPair())->generate($pathToPrivateKey, $pathToPublicKey);
            }
            catch(Exception $e) {
                throw $e;
            }
            return false;
        }
    }
```

Figure 5. *RSA key pair generation function in PHP.*

The URL request is sent via the JavaScript fetch API. The response is shown if the request is sent. There will be an error if the request is not sent. An alternate way for submitting forms without refreshing the page is to use fetch API methods. When the button is pushed, JavaScript is also utilized to send a POST request to a PHP script running on the server. The value of the button is sent in the request as data, and the superglobal variable $_POST allows the PHP script to access it.

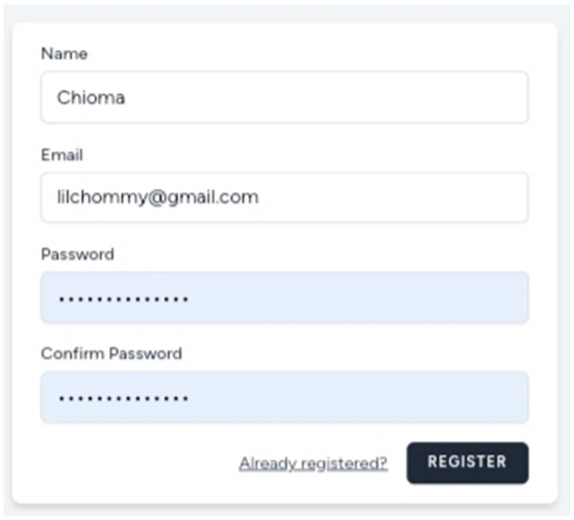


```javascript
function generateKey() {
    uFetch($('#aeskey-link').html(), init, {
        success: function(data) {
            $('#notification').html(data.view);
            $('#aeskey').val(data.aesKey);
        }
    });
    return false;
}

function generateRSAKeys() {
    const formData = new FormData();
    formData.append('name', $('#rsa-name').val());
    const init = {
        method: 'POST', credentials: 'include', body: formData,
        headers: {'X-CSRF-TOKEN': token, 'Accept': 'text/html,application/json'}
    };

    uFetch($(this).parents('form').prop('action'), init, {
        success: function(data) {
            $('#notification').html(data.view);
            $('#rsapubkey').val(data.pub_name);
            $('#rsaprikey').val(data.pri_name);
            $('#publink').prop('href', data.pub_url);
            $('#prilink').prop('href', data.pri_url);
        }
    });
    return false;
}

function uFetch(url, init, callbacks) {
    fetch(url, init).then(response => {
        if(!response.ok) throw new Error('invalid server response: ' + response.statusText);
        if(response.headers.get('content-type')?.includes("text/html")) return response.text();
        if(response.headers.get('content-type')?.includes("application/json")) return response.json();
    }).then(data => {
        if(callbacks.success) callbacks.success(data);
    }).catch(error => {
        console.log(error.message);
        if(callbacks.fail) callbacks.fail();
    }).finally(() => {
        if(callbacks.always) callbacks.always();
    });
}
```

Figure 6. *JavaScript fetch API, AES/RSA encrypt and decrypt functions.*

User interface and functionality

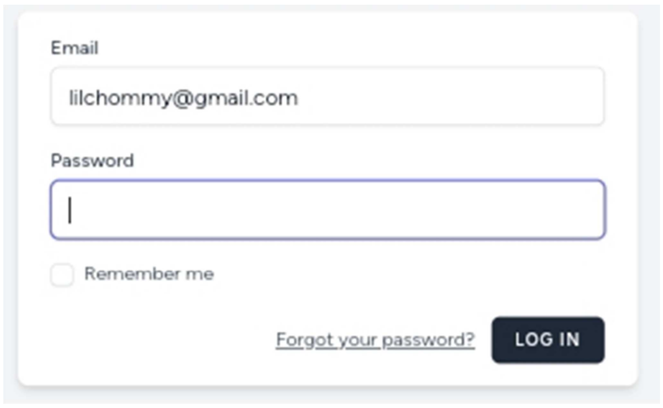
Registration page: Users must provide information such as name, email address and password to proceed with the registration.



Name
Chioma
Email
lilchommy@gmail.com
Password
••••••••••••••
Confirm Password
••••••••••••••

Figure 7. *Registration page.*

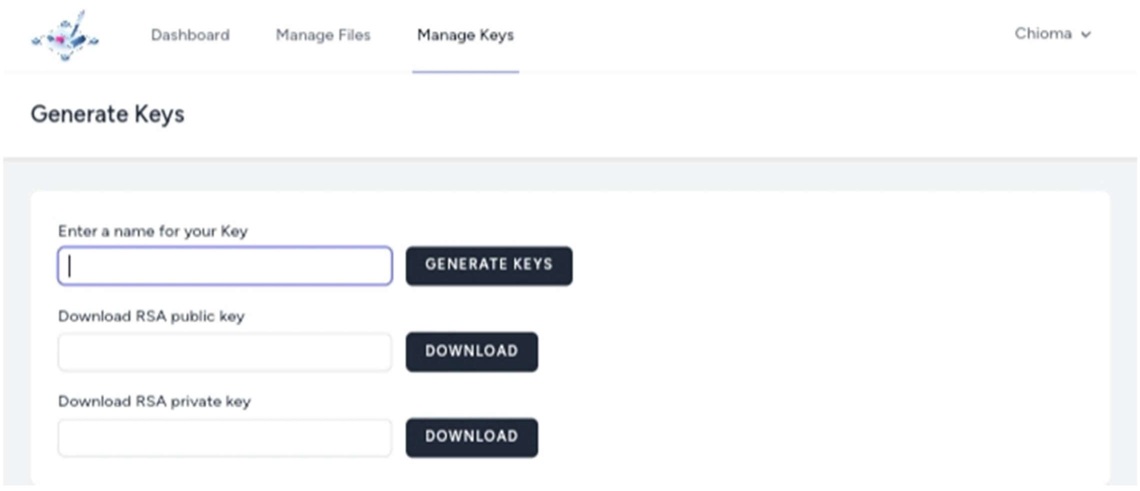
After successful registration, users can log in at any time.



Email
lilchommy@gmail.com
Password

Figure 8. *Log in page.*

Once logged in, users can generate and manage keys.



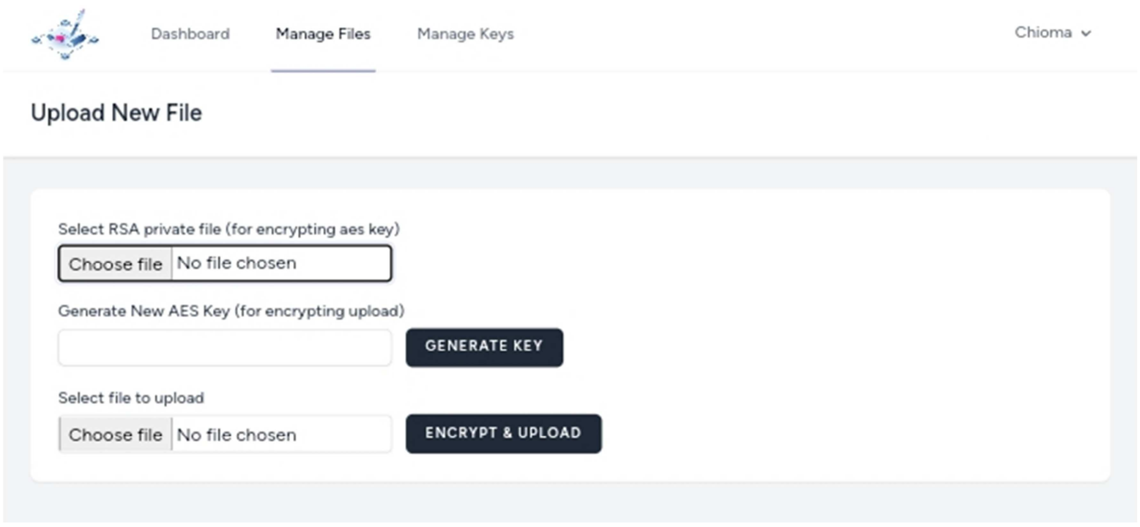
Generate Keys

Enter a name for your Key
Download RSA public key
Download RSA private key

Figure 9. *Key generation page.*

Users can choose the file they want to upload. Since the web application only stores encrypted files, users must provide their

RSA private key in order to generate the AES key that will be used for file encryption. The files are stored in the cloud server and anyone with authorized access to the files can view and download them.



Upload New File

Select RSA private file (for encrypting aes key)

Choose file No file chosen

Generate New AES Key (for encrypting upload)

GENERATE KEY

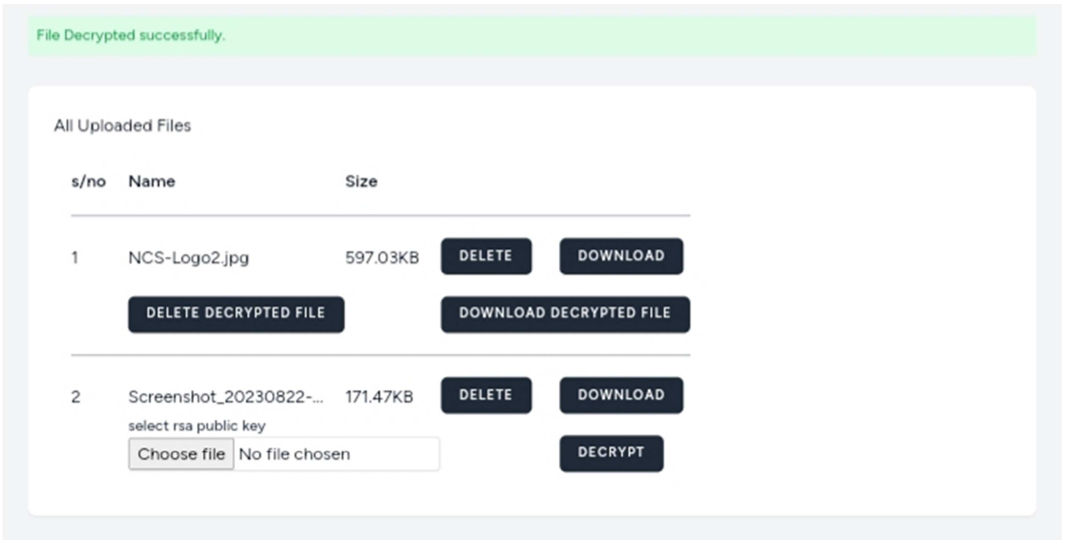
Select file to upload

Choose file No file chosen

ENCRYPT & UPLOAD

Figure 10. *Upload file page.*

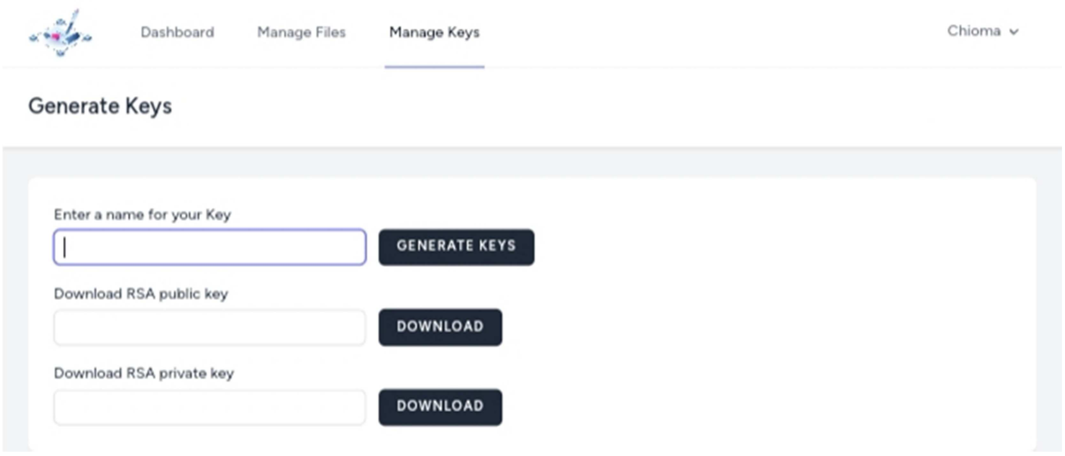
Sharing an RSA public key will give authorized users access to the encrypted file stored in the cloud server.


File Decrypted successfully.

All Uploaded Files

s/no	Name	Size		
1	NCS-Logo2.jpg	597.03KB	DELETE	DOWNLOAD
	DELETE DECRYPTED FILE		DOWNLOAD DECRYPTED FILE	
2	Screenshot_20230822-...	171.47KB	DELETE	DOWNLOAD
	select rsa public key Choose file No file chosen			DECRYPT

Figure 11. *File decryption page.*



Generate Keys

Enter a name for your Key

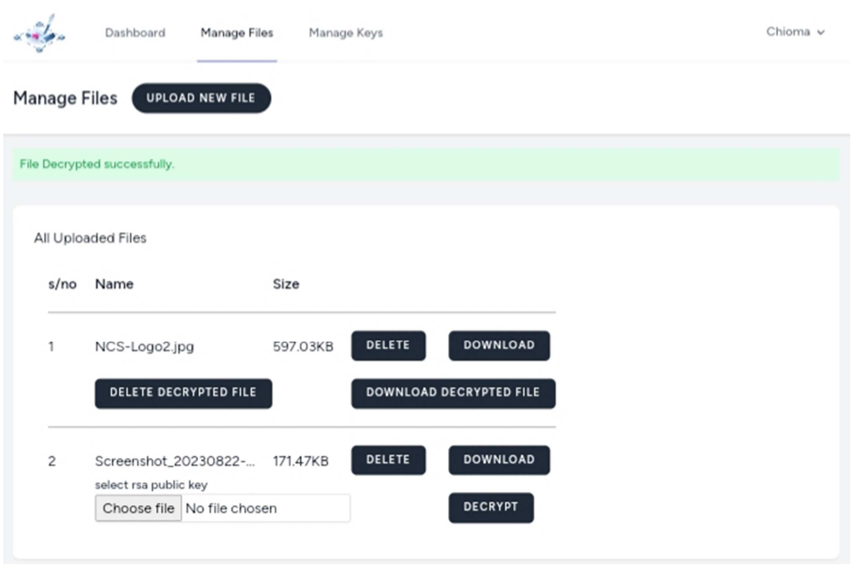
GENERATE KEYS

Download RSA public key

DOWNLOAD

Download RSA private key

DOWNLOAD



File Decrypted successfully.

All Uploaded Files

s/no	Name	Size		
1	NCS-Logo2.jpg	597.03KB	DELETE	DOWNLOAD
	DELETE DECRYPTED FILE		DOWNLOAD DECRYPTED FILE	
2	Screenshot_20230822-... select rsa public key Choose file No file chosen	171.47KB	DELETE	DOWNLOAD DECRYPT

Figure 12. *File Encryption.*

5. Conclusion

The cloud has an advantage over other storage solutions because it can access files from distant locations with just a reliable internet connection. Hybrid encryption is known to provide more security and performance than traditional single encryption techniques for protecting data stored in the cloud. These one-off encryption techniques may be symmetric or asymmetric in nature. Hybrid encryption algorithms, on the other hand, combine the benefits of symmetric and asymmetric encryption techniques to provide speed and security.

In this research work, some benefits of the proposed hybrid encryption scheme (AES and RSA) were recorded. These benefits include the combination of the speed and efficiency of symmetric encryption, the security and flexibility of asymmetric encryption and the ability to avoid the key distribution problem of symmetric encryption (AES), as the symmetric key can be securely exchanged using asymmetric encryption (RSA). There were also challenges encountered in the project, such as difficulty in identifying the symmetric and asymmetric encryption to use, complications of learning encryption processes and their mathematical structure, and interview participants not having IT background. The combination of the AES and RSA algorithms yields data security. The system achieved the security requirements of cloud computing on higher data confidentiality and gets security requirements in cloud computing with the least possible execution time. This is accomplished by using the RSA and AES algorithms to encrypt a fixed text block of 128bit, additionally, the AES algorithm is using a dynamic cipher key that is generated randomly in each session. Therefore, a system with high security, better performance, data integrity was designed and implemented. Some limitations of the study include implementation challenges, incomprehensive analysis of performance metrics and limited research scope.

ORCID

Chidi Ukamaka Betrand: 0000-0003-0452-375X
Chinwe Gilean Onukwugha: 0000-0001-6462-4662
Mercy Eberechi Benson-Emenike: 0000-0003-1771-5806
Christopher Ifeanyi Ofoegbu: 0000-0004-4144-8132
Nneka Martina Awaji: 0000-0001-6680-6347

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Abiona, O., Islam, R., Patamsetti, V., Gadhi, A., Gondu. R., Bandaru, C., and Kesani, S.,. (2023). The Future of Cloud Computing: Benefits and challenges. *International Journal of Communications, Network and System Sciences, 16*, 53-65.

[2] Seth, B., Dalal, S., & Kumar, R. (2019). Hybrid homomorphic encryption scheme for secure cloud data storage. *Recent Advances in Computational Intelligence*, 71-92.

[3] Harnal, S., and Chauhan, R. K. (2019). Hybrid Cryptography base E2EE for Integrity & Confidentiality in Multimedia Cloud Computing. *International Journal of Innovative Technology and Exploring Engineering*, 918-924.

[4] Orobosade, A., Favour-Bethy, T., Kayode, A., and Gabriel, A. (2020). Cloud Application Security using Hybrid Encryption. *Commun. Appl. Electron, 7*, 25-31.

[5] Sowmiya, M., & Prabavathi, S. (2019). Symmetric and asymmetric encryption algorithms in cryptography. *Int J Recent Technol Eng*, *8*(1S2), 355-7.

[6] Poduval, A., Doke, A., Nemade, H., and Nikam, R. (2019). Secure File Storage on Cloud using Hybrid Cryptography. *International Journal of Computer Sciences and Engineering (IJCSE)*.

[7] Mohammed, N., & Ibrahim, N. (2019). Implementation of new secure encryption technique for cloud computing. *International Conference on Computing and Information Science and Technology and Their Applications (ICCISTA)*, 1-5.

[8] Abdel-Kader, R., El-Sherif, S., & Rizk, R. (2020). Efficient two-stage cryptography scheme for secure distributed data storage in cloud computing. *International Journal of Electrical & Computer Engineering.*

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

[10] Rashmi, R., Gandhi, Y., Sarmalkar, V., Pund, P., and Khetani, V. (2020). Secure Cloud Storage with Deduplication Technique. *Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, 1280-1283.

[11] Kumar, U., and Prakash, J. (2020). Secure File Storage On Cloud Using Hybrid Cryptography Algorithm. *International Journal Of Creative Research Thoughts (IJCRT). Vol. 8, No. 7. ISSN: 2320-2882*, 334-340.

[12] Sirohi, P., and Agarwal, A. (2015). Cloud computing data storage security framework relating to data integrity, privacy and trust. *1st International Conference on Next Generation Computing Technologies (NGCT).*

[13] Basu, S., Bardhan, A., Gupta, K., Saha, P., Pal, M., and Bose, M. (2018). Cloud computing security challenges & solutions - A survey. *IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC).*

[14] Naveetha, K., and Tamilarasan, T. (2021). Secure File Storage on Cloud Using Hybrid Cryptography. *International Journal of Advanced Research in Computer Science Engineering and Information Technology (IJARCSEIT).*

[15] Chinnasamy, P., Padmavathi, S., Swathy, R., & Rakesh, S. (2021). Efficient data security using hybrid cryptography on cloud computing. In *Inventive Communication and Computational Technologies: Proceedings of ICICCT 2020* (pp. 537-547). Springer Singapore.

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

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

[18] Shukla, D. K.; Dwivedi, V. K.; Trivedi, M. C. (2020). Encryption algorithm in cloud computing. *Mater. Today Proc*, 1869-1875.

[19] Yahia, H., Zeebaree, S., Sadeeq, M., Salim, N., Kak, S., Al-Zebari, A., Salih, A., Hussein, H. (2021). Comprehensive survey for cloud computing based nature-inspired algorithms optimization scheduling. *Asian J. Res. Comput. Sci*, 1-16.

[20] Madhavi, G., Samatha, J. (2020). Secure data storage and access of data in cloud using Elliptic curve cryptography. *IEEE J.*

[21] Selvam, J., Srivaramangai, P. (2020). Time complexity analysis of cloud authentications and data security: Polynomial based hashing and elliptic curve cryptography. *Int. J. Anal. Exp. Modal Anal.*

[22] Kaur, H. (2017). A Novel Technique of Data Security in Cloud Computing based on Blowfish with MD5 method. *International Journal of Advance Research, Ideas and Innovations in Technology, Volume 3, Issue 6.*

[23] Sharma, S., Singla, K., Rathee, G., and Saini, H. (2020). A hybrid cryptographic technique for file storage mechanism over cloud. In (pp. 241-256). Springer, Singapor. *First international conference on sustainable technologies for computational intelligence*, 241-256.

[24] Hodowu, D., Korda, D., Ansong, E. (2020). An enhancement of data security in cloud computing with an implementation of a two-level cryptographic technique, using AES and ECC algorithm. *Int. J. Eng. Res. Technol, 9*, 639-650.