Current Trends in Signal Processing

ISSN-2277–6176

Volume -14

Issue-03

Year-2024

Research Article

Date of Receiving- 28th September 2024

Date of Acceptance- 14th October 2024

Date of Publication- 24th October 2024

A Novel Secure Cloud Storage Solution: Combining AES-OTP, RSA, and Time-Limited Access Control with Adaptive Key Management

[1*]Mansi Ahirwar, [2]Dr. Manoj Tyagi

[1] M.Tech scholar, Department of Computer Science and Engineering, Technocrats Institute of Technology, Anandnagar, Bhopal, Madhya Pradesh 462021

[2]Professor, Department of Computer Science and Engineering, Technocrats Institute of Technology, Anandnagar, Bhopal, Madhya Pradesh 462021

Corresponding author: mansiahirwar253@gmail.com

Abstract: The reliance of cloud computing on the data processing and storage structure creates serious security risks. Ensuring availability, security, and integrity of data in cloud settings becomes a challenge for both the individual and the enterprise. This paper will, therefore, introduce a novel Hybrid Cryptographic Framework that combines RSA, One-Time Pad (OTP), and AES as a means of enhancing data security in cloud storage. It employs RSA for secure key exchange, OTP for added entropy and security, and AES for efficient symmetric encryption. Another technique that enhances the security of data is a time-limited access control, limiting the access within some temporal bound. Such an approach is challenged with other cryptographic methods, such as RSA and AES, to prove how much more secure and performant it is, especially working with big information. All in all, a thorough analysis of encryption timings for various file sizes is provided, and utility of the framework in a cloud environment is proven. This hybrid approach is very suitable for secure storage of cloud data because the experimental results have established how much it reduces the execution time while enhancing security.

Keywords: Hybrid Cryptography, AES-OTP, RSA, Cloud Data Security, Temporal Access Control

I. INTRODUCTION

The way that businesses and individuals handled data has drastically changed as a result of the development of cloud computing. Cloud-based options are gradually replacing traditional on-premise data storage solutions because of their unparalleled scalability, accessibility, and cost-effectiveness. One of the main security concerns

this move raises is protecting personal information in the cloud settings. Secure storage of data in cloud computing has become a significant issue as businesses strive to ensure the security, integrity, and accessibility of their intellectual property while committing them to outside cloud service providers [1]. IT technology connects physical objects with sensors and networking capabilities, transforming various industries. Cryptography plays a crucial role in securing the vast amount of data exchanged in IT. This survey paper reviews IT technology and cryptographic algorithms, highlighting security challenges and mitigation techniques [2].

Numerous investigations have been carried out in this field to enhance cryptography through the reduction of computations and the enhancement of memory efficiency. Building a barrier to protect a paradigm as a whole is the primary goal of the present tactics. Moreover, some cryptosystems are more difficult to build and need longer deployment times. In light of current trends and technology, it is anticipated that a program would be integrated into the system in a way that maximizes its efficiency while requiring no additional memory or processing time. The evolution of cryptography has witnessed enormous advancements in algorithms to guarantee knowledge security over the years. The first inventions were block ciphers, ciphers for transposition, encryption algorithms for polyalphabetic replacement, and monoalphabetic ciphers. Then came the use of more advanced encryption algorithms like AES, DES, RSA, and SHA [3].

Robust encryption algorithms are crucial for safeguarding the privacy of sensitive medical data during transmission and storage. State-of-the-art cryptographic methods like AES and RSA are utilised to secure medical images and their accompanying text data. By integrating these algorithms together, this channel guarantees secure communication in the transfer of this data to prevent access in any form to the patient data at any cost. The confidentiality of medical image data transferred from imaging devices to LLM processing centers will not allow enemy entities to intercept it. Furthermore, such data is also encrypted in the process end-to-end for its protection. It has highly been successful for the risk minimization on unauthorized access while data is being transmitted in a medical imaging platform in integrating AES encryption. [4].

With symmetric encryption, the same secret key is used for both encryption and decryption. Another significant symmetric encryption algorithm is AES. It possesses the higher efficiency of symmetric encryption algorithms, realizing quick data processing. This encryption algorithm does not fully compromise the data security with performance. However, symmetric encryption has an issue with safely transferring the encryption key, considering some intricate key exchange conditions. Asymmetric encryption utilizes two separate keys for purposes of encryption and decryption: a public key for encryption, and a private key for decryption. Another example of an asymmetric encryption is the Rivest-Shamir Adleman algorithm, or RSA. . Its main advantage is the secure transfer of data without the necessity of sharing private keys, simplifying key management. However, asymmetric encryption typically requires more computational power and is slower compared to symmetric methods [5].

Advanced Encryption Standard (AES) is one of the most used block cipher-based symmetric cryptography encryption techniques. In 2001, Rijndael of the US National Institutes of Standards and Technology Computation introduced it. It takes the place of DES. The most used symmetric cipher encryption algorithm for data protection is AES. Every round of a secret writing method consists of the following four operations: Add Round Key, Mix Column, Sub Byte, and Shift Rows. The algorithm can employ multiple rounds of 10, 12, and 14 as well as lengths of keys of 128, 192, and 256 bits [6]. Sub Bytes, Shift Rows, Mix Columns, and Add Round Key are the four steps that AES uses to encrypt and decrypt messages. Shift Rows stage is one of the four encryption and decryption stages in AES, and because it uses simple, linear operations, it contributes to AES's lower security level [7].

A One-Time Pad is like a stream cipher only it doesn't employ a random key generation (OTP). It's a secure way to encrypt data so that the cryptanalyst can't figure out what the message is (25). A random key used for encryption and decryption must be at least as long as the message length generated by an authentic random generator. After then, it will be removed so that the next encryption and decryption processes can utilize a brand-new random key [8].

Any of the abovementioned building security systems can benefit from the automated one-time password (OTP) technologies as a security enhancement tactic. As an additional security measure, the OTP system can serve as a stand-alone security tool for access to buildings and other uses. An OTP is an automatically produced string of character or numbers that is used for a single login attempt. OTP is used to secure data, private credentials, and web-based services. It can be transmitted to the user's phone via push messaging or SMS (short message service). OTPs reduce the possibility of fraudulent login attempts and, thus, the possibility of data theft. OTPs can take many different forms, but they always provide an additional degree of security. If the authentication system is

integrated with the other security measures stated above or is utilized alone, it will be nearly impossible for unauthorized people to obtain access whenever used for building security. OTPs have a lot over static passwords and are impervious to reply attacks [9].

The one time pad and AES hybrid system is implemented using Java programming language. To encrypt the message, the user must type the inputs (message and key) in the respective text boxes with the use of corresponding buttons. Under OTP-AES encryption process, the message is encrypted with one time pad and the onetime pad ciphertext is obtained. And then, this ciphertext is encrypted again by AES with the use of 128 bit key length. The desired OTP-AES cipxertext is gained as shown in Figure 1 [10].

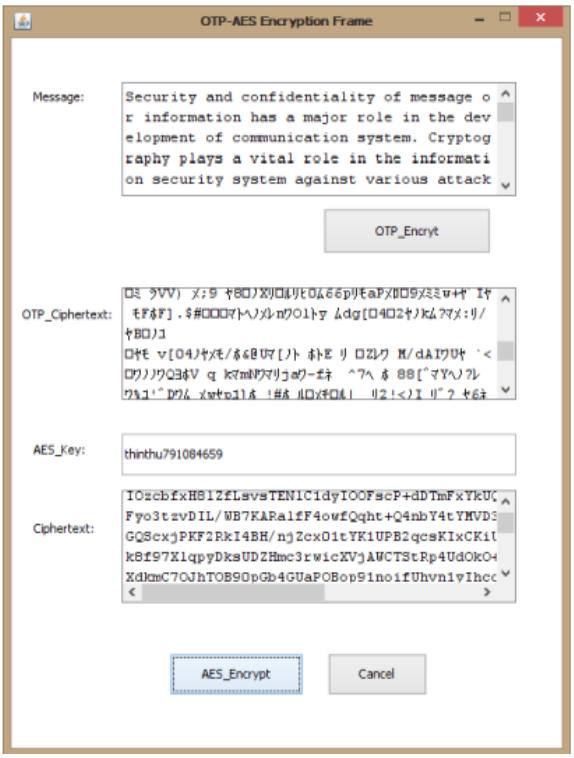


Figure 1: OTP-AES Encryption Process [10]

II. LITERATURE REVIEW

[11] suggested a new security architecture for cloud data storage using modified blowfish encryption to rectify these kinds of problems. Secure storage, outsourcing, and the proxy-re-encryption phase are the three stages. During the secure storage phase, a modified version of the blowfish technique is suggested for data encryption. The Kookaburra optimization algorithm is utilized for obtaining the optimal key. Auditing, proof, verification, File Access request, and file download are the five processes that occur throughout the outsourcing phase. Ultimately, the final stage of the suggested work is proxy re-encryption. In this case, the data user requests a key for a proxy, where the data owner has already transmitted the encrypted data key and the proxy provides the data user access to the encoded key. Lastly, a comparison of the suggested approach with various security methods validates its superiority.

[12] offered a fresh approach to encrypting and decrypting confidential text files using the Kurdish alphabet by utilizing the one-time pad (OTP) encryption-based Cipher Block Chaining (CBC) phase of the updated the Advanced Encryption Standard, cipher system. Additionally, the supplied work transmits at random secret codes used by AES and OTP over dubious channels by means of the customized RSA cipher scheme. Instead of using two huge prime numbers, the altered RSA cipher system randomly selects two large co-prime numbers, with the requirement that each factor have no more than two.

[13], which reviews the use of cloud computing and its impacts on data management, underlines the provision of scalable solutions for storage and processing. The research approach employed here is based on a comprehensive review of literature and secondary sources of data analysis that give a comprehensive description of the subject

matter. A systematic literature review was carried out to gather the findings and information related to how such a phenomenon of cloud computing affects data management. This method is essentially an aggregation of existing knowledge based on insights from a variety of sources. The observations here indicate that cloud computing has been transformative in changing data management structures by providing scalable and adaptable choices for storage and processing. It has enabled the processes of large sums of data and utilization of computing resources based on demand, but the elasticity and ubiquity of cloud computing can also bring about issues related to information confidentiality and protection. Therefore, the enforcement of encryption, access control, and application rules protecting data are indispensable. The technology has profoundly changed data management since it provides companies with strong tools that enable proper management of data but with the flavor of necessitating reliable data privacy and security safeguards. Organizations should focus on encryption and access control for sensitive information and have the same excellence in these aspects of security. Above all, it's about the law; doing what's necessary to become GDPR compliant.

[14] Maintaining the privacy and safety of data while using networks and communication technologies is crucial. The art of cryptography is employed to address the single issue of data security. Among the most often used cryptography algorithms is encryption. To improve data security, a new encryption technique is defined and put into practice in this study. The data has been protected using an enhanced one-time password (OTP) encryption method in conjunction with the latest encryption technology (AES) of 128-bit encryption, which can be increased to 256-bit encryption if necessary. To further strengthen system security, an encrypted key generator (SKG), a recently developed technique for random key creation, is also suggested. As a result, the study thoroughly examines the AES then OTP methods of encryption and contrasts the outcomes of the improved OTP method with those of other encryption methods already in use.

[15] Big data processing and storing with reliability can now be achieved with Hadoop. Huge data is made accessible with flexible and affordable services via the Hadoop Distributed File System, also known as HDFS, storage. Regretfully, the lack of any built-in security measures in Hadoop makes it more likely that unwanted attacks would target the data that is processed or stored using Hadoop. In this case, protecting the data kept in HDFS turns into a difficult undertaking. As a result, scientists and industry professionals are working harder to develop safeguards for user data compiled in HDFS. Many encryption-decryption techniques have been developed as a result, but as file sizes grow, so does their performance. The authors of this work have included a way to address the problem of security of data in Hadoop storage. The authors have created Attributes Based Honey Encryption (ABHE), which combines attribute-based encryption with honey encryption on Hadoop. Files that have been encoded within the High Definition File System (HDFS) and decrypted inside the Mapper can be processed using this method. Furthermore, the authors assessed the suggested ABHE method through encryption-decryption on various file sizes and contrasted it with the current ones, such as AES and AES with OTP methods. The ABHE method performs noticeably better when files are encrypted and then decrypted.

[16] Digital services commonly employ one-time password algorithms as a means of enhancing security. Nevertheless, a lot of these systems encrypted (process) one-time plaintexts using a constant secret key. A change in thinking from one-time to constant keys could have a positive impact on application security. The Rivest-Shamir-Adleman algorithm's one-time password notion, wherein each key element is hidden and the modulus value is altered after each encryption attempt, is examined in this study. An insecure channel is used to transfer differences between successive moduli between the communication parties. Research demonstrates the insecurity of this strategy. Furthermore, establishing the one-time password component (Rivest–Shamir–Adleman exponent) can be trivial. For the examined algorithm, a countermeasure is suggested.

The given table 1. provides a comparative analysis of various cryptographic techniques, highlighting the encryption methods used, key management or optimization techniques, application areas, advantages, and limitations. It encompasses studies related to cloud data storage, big data processing, network security, and specialized encryption for classified texts. The comparison emphasizes the importance of encryption performance, security, and scalability across different platforms.

Table 1: Comparative Analysis of Various Cryptographic Techniques and Their Applications

Reference	**Encryption Technique**	**Optimization/Key Management**	**Security Phase/Process**	**Applications**	**Advantages**	**Limitations**

[11]	Modified Blowfish	Kookaburra Optimization Algorithm	Secure Storage, Outsourcing, Proxy Re-encryption	Cloud Data Storage	Improved encryption strength, optimal key generation	Complex key management in the proxy phase
[12]	Modified AES with CBC Mode + OTP	Modified RSA for Key Transmission	Encryption and Decryption of Classified Text in Kurdish Alphabet	Classified Data Protection	Enhanced security with OTP, secure key exchange	Restricted to specific alphabets (Kurdish), large RSA keys
[13]	General Cloud Computing Techniques	Encryption and Access Control Mechanisms	Secure Storage and Processing	Cloud Computing, Data Management	Scalable, adaptable cloud storage	Data confidentiality and privacy concerns
[14]	AES (128-bit to 256-bit) + Modified OTP	Secure Key Generator (SKG)	Enhanced Data Encryption	Network Communication	Increased security, flexible encryption key lengths	Performance trade-offs with higher encryption lengths
[15]	Attribute-Based Honey Encryption (ABHE)	Attribute-Based Encryption (ABE)	Hadoop Distributed File System (HDFS)	Big Data Storage and Processing	Efficient for large datasets, improved performance with ABHE	Performance decreases with traditional methods like AES and OTP
[16]	OTP Algorithms with One-Time Keys in RSA	One-Time Key Management	Key Modulus Change with Each Encryption Attempt	Application Security	Potential for improved security	Insufficient security for RSA modulus manipulation

III. PROPOSED METHODOLOGY

The key issue related to data security within cloud computing environments is selected to be addressed in this research. Various issues have been observed in previous studies about the techniques in use for data encryption and authentication mechanisms. The traditional techniques of encryption, like the RSA algorithm, and authentication techniques that use MD5 hashing have been proven to be susceptible to attacks. Actually, using username- and ID-based authentication enhances the possibility of unauthorized access. Also, since RSA encryption is based on low-strength prime numbers, the attackers can easily deduce what the prime numbers may be, just by seeing the encrypted data. Also, the application of the MD5 algorithm, which has a rather low level of security, further nullifies the protection that would be offered by these combined systems against any unwanted data access. The traditional application of the RSA algorithm is based on an RNG model for generating the prime numbers; this also poses a threat to security. The aim of this research is to design a more secure hybrid cryptographic framework that will address these problems.

Methodology—Hybrid Cryptographic Framework: In this section, a concept for a hybrid cryptographic framework is presented to provide enhanced security of data in the cloud environment. Input to the proposed research will be provided by a health dataset, previously retrieved from a dataset repository, normally in '.csv' or '.xlsx' format. There are a few important stages in the process. In the dataset selection stage, the dataset used is chosen with the help of the Panda's Package that helps deal with and process data in Python. Afterwards, pre-processing is done on the input data that involves handling missing data, doing label encoding when necessary, and dropping columns irrelevant for these purposes.

After the registration, the systems are logged in by the users. They may then upload the message or datasets they want to have encrypted. In the process of encryption, the uploaded data gets secured with a combination of AES encryption, One-Time Pad, and RSA algorithms with the Blowfish algorithm for added security. The encryption process involves key generation using the RSA algorithm: RSA we shall study is an asymmetric cryptographic algorithm based on prime factorization. Practically, the RSA key pair is generated for usage by the following two-step process whereby: p is the result calculated on multiplying two large random prime numbers m, n, and calculated as, $p = m\,x\,n$. First, the evaluation of the golden ratio of $\phi\,(p) = (m-1)(n-1)$. We then select an integer e, such that $\gcd(\phi(p), e) = 1$, and $1 < e < \phi\,(n)$. The decryption key, d, is calculated as $d = e^{-1}|\phi\,(n)|$. The public key is (e, p) and the private key is (d, p).The P is encrypted using the public key to produce $C = P^e|p|$. The C is decrypted using the private key to retrieve the original $P = C^d|p|$.

In the decryption phase, if the key is correct, the data decrypts the encrypted data; otherwise, the request is denied. The encrypted data is stored in a cloud environment, for example, in CloudMe or Box Cloud, and the data is stored in different zones for good security. The proposed model will help to cover the key requirements which are necessary for security in a cloud computing environment: confidentiality, privacy, integrity, authentication, verification, non-repudiation, and acceptable execution time. The model has been developed to make use of the strengths of the RSA and AES algorithms merged together for cloud data security.

The proposed algorithm will operate in the following steps: It receives input of any type of file, which has different kinds of text file extensions like txt, doc, excel, ppt, pdf. Its output, if not tampered with or corrupt, will be a downloadable file. As mentioned, the process is first initiated with the processing of the input file, and in the case of selecting the number of cloud storage locations to which the file should be distributed. In the next step, a key is generated for each user. The file is divided into many blocks and for every block the AES algorithm is used for encryption. A hash of the encrypted block is further generated and stored with RSA. It creates a digital signature, and metadata in a cloud server. When the audit request and download query are made, RSA challenge the cloud server to ensure the disability is successfully verified. At RSA verification of hash occurs. If it matches, then the file downloads successfully; otherwise, the download fails, and an alert for corruption is raised.

The proposed algorithm has several advantages. It is therefore efficient for huge datasets and has proven high performance compared to existing systems. The time consumed for execution is relatively low. Security of data forms a major responsibility of a cloud service provider. Ensured efficient mechanisms need to be provided for the secure means of encrypting data and preventing its theft. Earlier studies have focused on how a secure and fast access of huge data in the cloud needs to be ensured. The present work deals with these issues by proposing a hybrid cryptographic framework based on the strengths of RSA and AES algorithms. This paper presents a mechanism that could provide data security in the cloud environment effectively and provides a solution for addressing the core security requirements of cloud computing.

IV. EXPERIMENTAL AND RESULT ANALYSIS

This chapter provides an explanation of the recommended methodology. A completely open-source and free laboratory environment for Python, Spyder/Jupyter combines data exploration, sophisticated analysis, debugging tasks, editing, and profiling. A multilingual editor window is available in Spyder/Jupyter for the creation, opening, and editing of source files.

Development Environment

A server-client program called the Jupyter Notebook App provides web browser editing and execution of notebook pages. The Jupyter Notebook application can be utilized locally on the desktop without an internet connection, or it can be installed on a remote computer and viewed online. A JSON file that follows a customizable format and frequently has the ".ipynb" extension is a Jupyter Notebook document. Jupyter Notebooks' three main parts are metadata, the Notebook format, and a list of cells. A dictionary of data definitions called information is used to set up and display the notebook. The version number of the computer is Notebook Format. Cells in the list are of many types, including Markdown (display), Code (to execute), and code output cells. While ".ipynb" and JSON are the most often used and default formats, it is possible to forego some functionality (such as collecting photos and metadata) and save notebooks as markdown documents using an extension such as JupyText. Jupytext is frequently used in combination with configuration files to facilitate notebook diffing and combining.

Software Requirements

Operating System : Windows 7 / 8 / 10

Language Used : Python

Database : My SQL

User Interface Design : JFrame

Server : Wampserver

Hardware Requirements

Processor : Intel Core i5

Hard Disk : 200 GB

Monitor : 18’ LED color

Mouse : DELL.

Keyboard : 110 keys enhanced

RAM : 3GB

Performance Parameters

The experimental findings show that our IDS has a reasonable degree of precision and detection rate when it comes to finding anomalies. The experiment's optional result is found by comparing the results with the traditional approach. When compared to the suggested approach, the suggested task is more efficient. The observed result and the process of computation parameters are detailed below.

1. Accuracy Analysis

Correct indication of an intrusion results in a True Positive fact. If we have not discovered any attacks, a True Negative. A misleading positive alarm is set off if an incursion is detected by IDS, but this claim turns out to be false. At last, the non-intrusive is found, the intrusion is real, and a false-negative (FN) event has occurred. A false negative is the worst-case situation, which happens when a detection process raises the wrong alarm. Taking these factors into consideration, we assessed our IDS using its resultant number and detection rate. Accuracy (ACC) is calculated by dividing the whole number of outcomes by the total count of invaders.

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

2. Detection Rate

Conversely, a detection rate (DR) represents the likelihood of identifying the actual incursions from the given warning.

$$DR = \frac{TP}{TP + FN}$$

Accuracy, precision, and F1-score are the parameters that were used to assess the performance of the presented models. The model's overall accuracy was calculated using these standards. Classification results are computed using precision, recall, F1-score, accuracy of each class, and overall accuracy for multi-class and binary classification.

3. Precision

Based on those optimism assumptions, it tells us how accurate the model is and how numerous of them are incorrect. It is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \times 100\ (\%)$$

4. Recall

It determines whether any among the true positives found by the suggested model are really gathered by categorizing them as positive:

$$Recall = \frac{TP}{TP+FN} \times 100\ (\%)$$

5. F1-Score

It provides as an illustration of how accurate a test was. The F1-score, which has a maximum value of 1, is the harmonic mean (HM) of accuracy and recall. The F1 rating is determined by:

$$F1score = 2 \times \frac{Precision \times Recall}{(Precision + Recall)} \times 100\ (\%)$$

Here,

TP: True Positive,

TN: True Negative,

FN: False Negative, and

FP: False Positive.

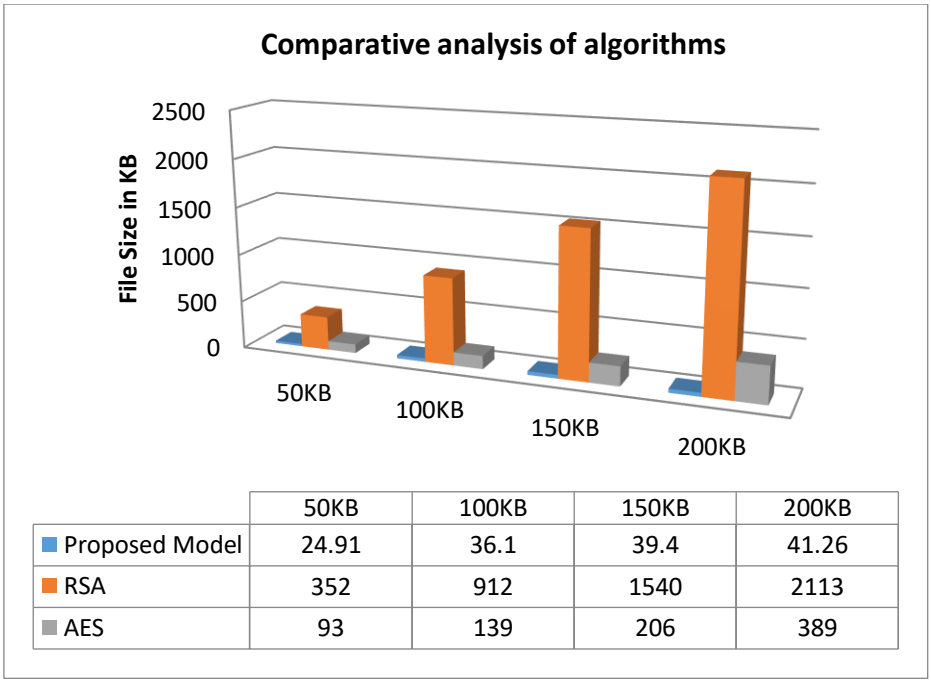
Table 2: Result Analysis Algorithms.

File size (KB)	The execution time of the proposed model in the transmitter's side (ms)	The execution time of the RSA algorithm in the transmitter's side (ms)	The execution time of the AES algorithm in the transmitter's side (ms)
50KB	24.91	352	93
100KB	36.1	912	139
150KB	39.4	1540	206
200KB	41.26	2113	389

The RSA algorithm is used to encrypt text with a length of 16 Byte twice only and is not used to encrypt a relatively large message whether that compared it with this length. Also, the AES algorithm has been used only to encrypt the first block m1 which is also 128bit long. This ensures the reduction of the other times necessary to encrypt the rest of the message blocks, as they are encrypted by relying on the XOR factor. This process takes place in the transmission and reception, which ensures shortening the execution time, whether on the transmitter or receiver's side. Table 2. shows the execution times for the proposed model and each of RSA and AES algorithms separately on the transmitter's side for various file sizes. The difference in analysis among the algorithms is displayed in the table above.

Graphical Comparison Analysis

This part provides an overview of statistical and graphical analysis and presents an analysis of the results.



	50KB	100KB	150KB	200KB
Proposed Model	24.91	36.1	39.4	41.26
RSA	352	912	1540	2113
AES	93	139	206	389

Figure 2: Comparing the results of the algorithmic analyses.

This section compares the results of the prior strategy, and the suggested approach based on the observed and obtained implementation results shown in figure 2.

We have explained about our work done by taking several inputs and obtaining outputs, also we have worked on the encryption of the private files for their security maintenance. The experimental setup and further finding it working module with given computation is discussed by the system. Computational parameter and further finding the result from it is also discussed in the given chapter.

V. CONCLUSION

Because cloud computing offers scalability, cost-efficiency, and flexibility, it has completely changed the way businesses handle and keep data. However, there are serious security concerns raised by the inherent weaknesses in cloud storage systems. The strong hybrid cryptographic system presented in this study combines time-limited access control and the best features of RSA, OTP, and AES encryption methods to provide safe cloud data storage. This framework uses the symmetric and asymmetric encryption advantages to eliminate the traditional cryptographic techniques' weaknesses. For encryption, it employs AES, which is well known for its speed and efficiency, while OTP introduces an element of unpredictability that it is difficult to decrypt. It minimizes the risks associated with key management through secure transfer of encryption keys. Furthermore, the authors ensured that time-based access control limits authorized access to sensitive information based on time boundaries so that only authorized users may access such information within the determined time frames. The proposed framework's effectiveness in protecting large datasets is shown by the experimental evaluation. The encryption and decryption time decreases considerably when compared to traditional approaches like RSA or AES, which is usually used alone, especially for large file volumes. This makes the proposed approach very efficient and safe in cloud contexts, where the performance plays a major role. In summary, basic weaknesses have been the main focus for reinforcing the security of cloud data using hybrid architectural cryptography based on well-balanced solution that combines strong encryption with performance. This framework is perfect for businesses looking for effective and safe ways to handle and safeguard sensitive data in cloud-based systems. After these investigations, further probing could be necessary regarding key management strategy improvements and the integration of machine learning for real-time threats detection.

REFERENCES

[1] Shivaramakrishna D, Nagaratna M. A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating AES-OTP and RSA with adaptive key management and Time-Limited access control. Alexandria Engineering Journal. 2023 Dec 1; 84:275-84.

[2] Jalal AO. Enhancing Data Security in Fog Computing Using AES Cryptography Technique Based on Argon2 Key.

[3] Vaishali R, Manohar Naik S. A DNA Cryptosystem Using Diffie–Hellman Key Exchange. SN Computer Science. 2024 Feb 20;5(3):274.

[4] Ahir P, Parikh M. SAFEGUARDING AI-MEDIATED DIAGNOSES: A COMPREHENSIVE REVIEW OF CYBERSECURITY CHALLENGES AND SOLUTIONS IN LARGE LANGUAGE MODEL-ASSISTED MEDICAL APPLICATIONS. Towards Excellence. 2023;2(2).

[5] Nanumura UA. In-Depth Analysis of Encryption Techniques for the Protection of Mobile Health Care Applications. International Journal of Research in Engineering, Science and Management. 2023 Nov 26;6(11):139-42.

[6] Abikoye OC, Garba QA, Akande NO. Implementation of textual information encryption using 128-, 192- and 256-bits advanced encryption standard algorithm. Annals. Computer Science Series. 2017;15(2):153-9.

[7] Zinabu NG, Asferaw S. Enhanced efficiency of advanced encryption standard (EE-AES) algorithm. American Journal of Engineering and Technology Management. 2022;7(3):59-65.

[8] Alattas AH, Al-Shareeda MA, Manickam S, Saare MA. Enhancement of NTSA secure communication with one-time pad (OTP) in IoT. Informatica. 2023 Feb 15;47(1).

[9] Sorochi UV, Nasiru A, Inusa IA. Security enhancement for building access, using one time password (OTP) technology. European Journal of Electrical Engineering and Computer Science. 2020 Jun 25;4(3).

[10] Tun T, Myint E, Aung M. Message Security using One Time Pad and AES Hybrid Cryptography.

[11] Singh P, Singh P, Agarwal AK, Kumar A. Modified Blowfish Encryption Standard with Optimization Strategy for Secured Cloud Storage with Data Dynamics. In2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS) 2024 Jun 28 (pp. 1-8). IEEE.

[12] Abdulrazaq NN. A Novel Approach for Safeguarding Kurdish Text Files via Modified AES-OTP and Enhanced RSA Cryptosystem on Unreliable Networks. EURASIAN JOURNAL OF SCIENCE AND ENGINEERING. 2024 Jun 12;10(2):102-19.

[13] Ateeq A, Alaghbari MA, Ateeq RA, Ahmed AY. Understanding and Addressing Data Security and Privacy Concerns in Modern Cloud Computing Systems. In2024 ASU International Conference in Emerging Technologies for Sustainability and Intelligent Systems (ICETSIS) 2024 Jan 28 (pp. 220-224). IEEE.

[14] Gupta S, Jain S, Agarwal M, Nanda N. An encryption approach to improve the security and performance of data by integrating AES with a modified OTP technique. International Journal of Advanced Intelligence Paradigms. 2024;27(2):129-49.

[15] Kapil G, Agrawal A, Attaallah A, Algarni A, Kumar R, Khan RA. Attribute based honey encryption algorithm for securing big data: Hadoop distributed file system perspective. PeerJ Computer Science. 2020 Feb 17;6:e259.

[16] Sarna S, Czerwinski R. Small prime divisors attack and countermeasure against the rsa-otp algorithm. Electronics. 2021 Dec 28;11(1):95.