2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET)
Loughborough University, London, United Kingdom. May 19-21, 2023

A Hybrid Cryptographic Approach for Secure Cloud-Based File Storage

Kanishka Negi
Computer Science and Engineering,
School of Engineering and Technology
Sharda University
Greater Noida, India
2019600654.kanishka@ug.sharda.ac.in
0000-0002-4620-9622

Ronika Shrestha
Computer Science and Engineering,
School of Engineering and Technology
Sharda University
Greater Noida, India
2019003290.ronika@ug.sharda.ac.in

Tanya Lillian Borges
Computer Science and Engineering,
School of Engineering and Technology
Sharda University
Greater Noida, India
2019639262.tanya@ug.sharda.ac.in,
0000-0001-8910-4276

Subrata Sahana
Computer Science and Engineering,
School of Engineering and Technology
Sharda University
Greater Noida, India
subrata.sahana@gmail.com

Sanjoy Das
Department of Computer Science
Indira Gandhi National Tribal
University
RCM, India
sdas.jnu@gmail.com

2023 IEEE IAS Global Conference on Emerging Technologies (GlobConET) | 979-8-3503-3179-0/23/$31.00 ©2023 IEEE | DOI: 10.1109/GlobConET56651.2023.10150148

***Abstract*— The tremendous increase of sensitive data on the cloud has rendered it more vulnerable. Thus, undoubtedly, the risk stems from an expanding number of individuals with harmful motives. Because the cloud is handled by a third party, guaranteeing cloud security services is critical. The security of the data may be strengthened by combining different symmetric key techniques to store the information on the server so that even if someone gains access, they are unable to access the original data. It must be decrypted, thus. A hybrid model that combines Advanced Encryption Standard (AES) algorithm and Elliptical Curve Cryptography (ECC) is put forth. ECC is employed to complete the user verification process and maintain the security of private data. Data is encrypted on the client side and then decrypted after being downloaded from the cloud, the AES technique is used to securely store and retrieve users' data in the cloud. Data is protected by encryption on the client's side and then decoded after it has been downloaded from the cloud. Enabling a suitable access mechanism to stop unlicensed access to information in the system and safe storage to allow access to information over a cloud-based network would be advantageous for the complete prototype of the suggested solution. The proposed method is efficient for encrypting small to moderate amounts of data as 1.02 milliseconds is a relatively fast encryption time.**

Keywords*— *Cloud, Cryptography Techniques, Elliptic Curve Cryptography, Advanced Encryption Standard, Elliptic Curve Diffie Hellman Key Exchange.

I. Introduction

A strategy for offering information technology services known as cloud computing relies on the provision of resources through the Net as a metered service [1]. It is a form of computing in which several remote servers are interconnected to enable the centralized processing, distribution, and storage of data and software across the Internet. Because of this, businesses and people may obtain computer resources (including networking, servers, intelligence, software, databases, analytics, and storage) whenever they need them without having to develop and manage their infrastructure. Amazon Web Services (AWS) is one example of a cloud computing provider, Microsoft Azure, and Google Cloud Platform. Cloud-based file storage is a service that enables users to store, access, and share digital files over the Internet using remote servers provided by a cloud storage provider. In cloud-based file storage, files are stored on servers that are maintained and managed by the cloud storage. Cloud computing has become popular due to its numerous benefits, such as cost savings, expandability, agility, reliability, and accessibility. However, it also raises concerns over security, privacy, and compliance, which need to be addressed [2].

Cloud-based file storage security refers to the measures that are taken to protect digital files that are stored on remote servers in the cloud [3, 6]. Security is a major concern for organisations and individuals who store sensitive information in the cloud, as the data is stored on remote servers and accessed through the internet, making it vulnerable to cyber threats such as hacking, theft, and unauthorised access. To ensure the security of data and applications in the cloud, organisations must take a multi-layered approach that includes both technical and operational measures. Some of the key security measures that organisations can implement in cloud computing include:

1. Encryption: Sensitive information can be protected from unauthorized access by being encrypted both in transit and at rest.
2. Multi-Factor Authentication: By necessitating a password, it provides an additional layer of protection to user profiles and a secondary form of authentication.
3. Access controls: Organisations can use access controls to restrict who can access sensitive data and applications in the cloud. This can include setting up roles and permissions for users and using network security policies to enforce access restrictions.
4. Data backup and disaster recovery: ensures that sensitive data is not lost in the event of a breach or system failure.
5. Compliance with regulations: Organisations must ensure that their use of cloud computing complies with relevant regulations.

Encryption is a key aspect of information security that protects sensitive information by transforming it into a code that can only be deciphered with a corresponding key.

Encryption is used in various contexts, including cloud computing, to protect sensitive data both in transit and at rest [3, 14]. Cryptographic algorithms are used during encryption, and cryptosystems are the systems that use such algorithms. To encrypt and decode data, cryptographic methods are implemented by two different types of cryptosystems. These are cryptosystems that use both public and private keys. The primary distinction between the two systems is that a public-key cryptosystem, sometimes referred to as an asymmetric cryptosystem, encrypts data using a public key and decrypts it using a secret key. The private key is kept hidden, but the public key is released widely as its name suggests. One secret key encodes and decodes the files in a private-key cryptosystem, sometimes referred to as a symmetric cryptosystem. Both users participating in encrypting and decrypting the data in such a system must be aware of the secret key in advance [15]. Implementation of various symmetric key algorithms can improve data security by storing data on the server in such a way that even if a person gains access, he cannot open the original data as it must be decrypted [20].

The primary objective of this study is to protect the confidentiality, integrity, and availability of data stored in the cloud. The following objectives have been established to achieve through this research:

1. Protect Sensitive Information
2. Ensure Data Privacy
3. Ensure Data Availability
4. Reduce Information Loss

The rest of the document is structured as follows. Section 2 explores a detailed overview of previous works in the field, Section 3 focuses on the methodology. Section 4 provides a brief on the proposed approach. Section 5 presents the result. Section 6 concludes the paper and presents the future scope

II. LITERATURE REVIEW

Stergiou et al. [1] provided an overview of IoT technology as well as an explanation of how to use it. Furthermore, they highlighted the primary characteristics and trade-offs of cloud computing. They examined both IoT and cloud technologies, focusing on their respective security challenges. They put up an algorithm model to look at the security concerns brought by the interconnection between IoT and cloud computing. The security problems addressed in this work could serve as a practical model for future research aimed at reducing them. To demonstrate all the data protection strategies used to safeguard sensitive data transferred to cloud storage, Hassan et al. [2] carried out a thorough literature analysis. Consequently, the primary goal of their research was to compile, organize, and identify key papers about study. Albugmi et al. [3] covered the topic of cloud computing data security. MS et al. [4] presented partial homomorphic-based encryptions to safeguard cloud request traffic. To demonstrate their advantages, completely homomorphic and partly homomorphic are compared. The suggested approach provides an explanation of partial homomorphic encryption. Li et al. [5] created a cutting-edge methodology for assessing trust in cloud services that integrates reputation and security features. To include privacy metrics in the trust assessment, they proposed a security-based trust evaluation method. Second, to improve the precision and dependability of the feedback rating-based reputation assessment model, they have introduced a reputation-based trust evaluation approach. Bharathi et al. [7] utilised DES, RSA, and AES, to separate data and encrypt it. Files and keys are kept in the cloud and LSB steganography, respectively. The combination of this hybrid approach results in enhanced security. They mentioned that the one drawback of their approach is that an active internet connection is necessary to connect to the cloud server. Hodowo et al. [8] provided a paradigm and a hybrid cryptographic approach to improve data security in cloud computing. In order to improve data security against outsiders or hackers, protect them from accessing the data's actual contents and integrity, reduce the amount of time needed to perform cryptographic operations, enable confidentiality, and accelerate the speed of the cryptographic process by using smaller keys of ECC, their model used both AES and ECC. Goyal et al. [9] proposed an effort to identify secure approaches to data sharing and communication in the cloud. In light of this, a DNA-based cryptographic method for use in a cloud-based cryptographic system is proposed. The current method necessitates the use of a single key that is produced at random for both encryption and decryption. A mapping for decryption is thus required for subsequent decryption. This procedure is required to improve the security of the system. Rehman et al. [10] outlined a safe and efficient plan for cloud-based data sharing while preserving data security and integrity. The ECC and AES methods are primarily combined in the proposed system to provide authentication and data integrity. When compared to existing approaches, the experimental results displayed that the proposed approach is more efficient and produces better results. Pavithra et al. [11] developed a secure cloud-based data evaluation tool. The blinding process begins after the user uploads the data. The current python module is used to encrypt data. While the files are being cleaned up, RSA encryption generates a key and password. Anuj et al. [12] recommended using a sample plaintext text string for the implementation and simulation of the suggested approach. The RSA algorithm is used to implement the first security step, that is, data encryption. For the second layer of security, data from the first stage is encrypted using the DES Algorithm. RSA and DES are combined to provide a highly secure method for storing data in the cloud [13, 14].

The table I given below summarizes the research papers reviewed to help in the implementation of this project.

III. RESEARCH METHODOLOGY

A. *Advanced Encryption Standard Encryption Algorithm:*

Belgian cryptographers created the Rijndael-based cipher known as the Advanced Encryption Standard (AES) [16, 17]. As a symmetric key algorithm, The AES algorithm uses the same secret to both encrypt and decrypt information.

Around the world, sensitive data is encrypted using AES in both hardware and software. AES is essential for cyber security, the computer security of official agencies, and the safeguarding of private information. AES was developed after the NIST stated that the Data Encryption Standard was becoming susceptible to brute-force assaults, and needed to be replaced for security purposes.

TABLE I. COMPARATIVE ANALYSIS OF VARIOUS CLOUD BASED SECURITY APPROACH

S.No.	Author	Year	Major Contributions
1.	Stergiou et al. [1]	2016	1. Showcase the primary attributes and trade-offs of cloud computing. 2. Conduct an analysis of both IoT and cloud technologies with an emphasis on their respective security challenges. 3. Through the proposed algorithm model, the security implications of combining IoT with Cloud Computing were investigated.
2.	Lee et al. [2]	2017	1. Investigated the IoT network's technical specifications. 2. Give details about the prospective IoT network and the essential elements that will make it possible. 3. Since the IoT connectivity is the essential element that permits IoT services, attention to the IoT's network layer concentrates primarily on network problems in the IoT.
3.	Albugmi et al. [3]	2016	1. The problem of cloud computing data security is addressed. It is an examination of reliability problems related to cloud data and related subjects. 2. The study will also provide some understanding about data vulnerabilities for both in-transit and at-rest data.
4.	Swamy et al. [4]	2020	1. Include a summary of IoT architectures' current status and IoT architectures' overall design. 2. A full review of edge computing in the Internet of Things, including several edge computing architectures, is considered. As a result of the widespread use of IoT in society, security and privacy problems have developed. 3. Focuses on assessing security issues, privacy concerns, security threats, traditional mitigation strategies, and potential future applications for IoT security.
5.	Li et al. [5]	2018	1. A cutting-edge methodology for evaluating trust in cloud services that combines reputation and security features. 2. Through reliable cloud services, the framework can increase the cloud-based IoT context's security 3. To improve the precision as well as dependability of the response rating-based reputation evaluation method, they created a reputation-based trust evaluation technique.
6.	Sethi et al. [6]	2017	1. The architecture of IoT, cloud, and fog-based architectures has been covered in this research study. 2. A thorough examination of the smart gateway for preprocessing. In order to analyses, store, and process the massive volumes of sensor data that smart devices collect, computer and storage resources are needed.
7.	Bharathi et al. [7]	2021	1. Utilizing DES, RSA, and AES, data is separated and encrypted. Files and keys are kept in the cloud and LSB steganography, respectively. Enhanced security as a result of the hybrid approach. 2. Active internet connection is necessary in order to connect to the cloud server.
8.	Hodowu et al. [8]	2019	1. This research provided a paradigm and a two-level cryptographic approach for improving data security concern in cloud computing. 2. Both symmetric and asymmetric cryptography techniques are used in the model to increase data security against hackers, preventing them from accessing the actual information.
9.	Goyal et al. [9]	2019	1. The goal of the proposed effort is to identify approaches for data sharing and communication that are secure in a cloud context. Hence, a DNA-based cryptographic method is suggested for use in a cloud-based cryptographic system. 2. Here, the randomly generated key is needed for encoding and decoding. A mapping for the decryption is therefore necessary for subsequent decryption.
10	Rehman et al. [10]	2015	1. Proposed a safe and effective method for cloud data exchange while preserving data security and integrity. 2. In order to assure authentication and data integrity, the suggested system primarily works by fusing the ECC and the AES approach. 3. The experimental findings demonstrate that the proposed approach works and is successful and generates better outcomes than current approaches.
11.	Pavithra et al. [11]	2021	1. Created a safe cloud-based data evaluation tool. 2. After the user uploads the data, the process of blinding begins. 3. A key and password are generated by RSA encryption while the files are being cleaned up.
12.	Anuj et al. [12]	2020	1. The RSA algorithm is used to apply the 1st step of security. 2. For the next layer of security, data that was the result of the first step is encrypted using the DES Algorithm. 3. RSA and DES are combined to provide a highly secure method that can be used to store data on the cloud.

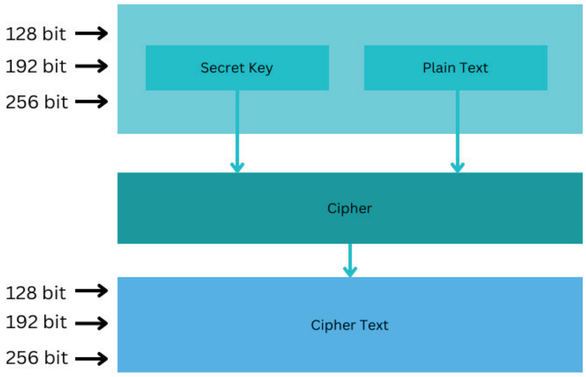

Fig. 1. AES Working Design

Using encryption keys of lengths of 128, 192, or 256 bits, AES protects and decrypts data in 128-bit chunks using three-block encryption. For 128-bit keys, there are 10 rounds; for 192-bit keys, there are 12 rounds; and for 256-bit keys, there are 14 rounds [18]. To generate the final product of ciphertext, the original plaintext is processed through a number of stages that include transposition, replacement, and mixing.

The AES encryption method specifies a large number of alterations that must be done to the data contained in an array. Following the initial organisation of the data into an array, the cipher alterations are replicated throughout several encryption cycles. The first update of the AES encryption technique involves the substitution of data by utilising a substitution table. The second round involves moving data rows. The third column combines the columns. A separate chunk of the encryption key is used for the final transformation of each column. Longer keys require more rounds to complete.

B. *Diffie-Hellman key exchange:*

The Diffie-Hellman key exchange (DHKE), named after Whitfield Diffie and Martin Hellman [19], was one of the first public-key protocols. It was among the first instances of public key exchange being used in a cryptographic setting. This is the first known instance where the idea of a private key and an associated public key was put out. Diffie and Hellman issued it as a publication in 1976 .

The Diffie-Hellman key exchange method, which uses cryptography, enables two parties with no prior knowledge of one another to construct a shared secret key across an unreliable communication medium [20]. Subsequent communications can subsequently be encrypted with a symmetric key cypher using this key. It makes use of two keys: a secret key and a private key. The sender must encrypt the message with both his private key and the sender's public key to interact with the receiver. Combining his private key and the sender's public key, the receiver decrypts the communication at the other end. The complexity of constructing logarithmic functions for prime exponents is the basis for this technique. The Discrete Logarithm Problem (DLP) is what is referred to as here [21].

The Diffie-Hellman key exchange cryptographic technique can be used by two parties without any prior knowledge of one another to produce a shared secret key over an unprotected communication channel [22]. They use this key to encode successive messages using a symmetric-key cipher.

C. *Elliptic Curve Cryptography:*

Around 20 years ago, IBM's Victor Miller and the University of Washington's Neal Koblitz separately proposed elliptic curves as the cornerstone for discrete logarithm-based cryptosystems.[23]. Numerous cryptographic tasks, like integer factorization and primality proving, already used elliptic curves at the time.

Due to its ability to preserve security and smaller key size, ECC has recently grown in favour as longer keys require more room, bandwidth, and computational power. It will also take time to generate a key, encode data, and decode data. This is why understanding ECC in context is critical.[26]

ECC builds its methodology to asymmetric cryptography systems on how elliptic curves are mathematically built over defined areas. As a result, ECC creates keys that are statistically more difficult to crack. Hence, it is popularly known as the next-generation execution of public key cryptography, as well as being more protected than other asymmetric algorithms.In cryptography, an elliptic curve can be represented by an affine equation of the form [27]

$$y = x^3 + ax + b \qquad (1)$$

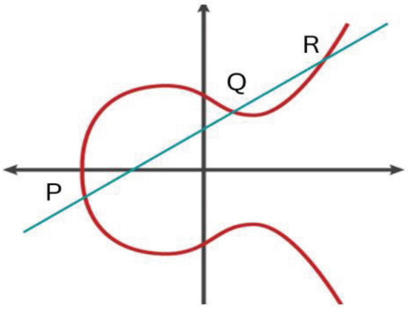

Fig. 2. Elliptic curve cryptography [23]

Where a and b are considered finite field elements with P^n elements (where **p** is prime and **p** > 3)

ECC generally produces cipher texts, keys, and signatures, in addition to faster key and signature generation [24]. It is capable of relatively quick decryption and encryption. ECC consistently produces better connectivity and faster responses than the opposite by computing identifiers in two steps. For approved key exchange, ECC provides solid protocols, and the technique is widely used [32, 33].

D. *Elliptical Curve Diffie-Hellman*

Two parties, A and B, who each have an elliptic curve public-private shared key, can create a shared secret key using the protected key agreement technique known as ECDH. It is an elliptic-curve cryptographic variant of the Diffie-Hellman protocol [25].

ECDH is constructed around the following EC point property:

$$m(m * P) * n = (n * P) * m \qquad (2)$$

If we have two encryption integers m and n (two private keys belonging to A and B) and an ECC with generator point P, we may interchange the values (m * P) and (n * P) (the public keys of A and B) across an unsecured channel and deduce a shared secret: hidden = (m * P) * n = (n * P) * m. It is really that simple. The equation is written as follows:

$$nAPubKey * BPrivKey = BPubKey * APrivKey = Secret \quad (3)$$

IV. Proposed Model

Cloud hosting enables users to access and save their files from any location. Additionally, it guarantees optimum resource use that is accessible. Given this cutting-edge technology, users' privacy is undoubtedly compromised, posing fresh security risks to the reliability of cloud-based data. When the topic of cloud security comes up, security issues including maintaining data dependability, data concealing, and data security take centre stage. Numerous studies have concentrated on the fact that users must often access massive amounts of cloud data in a secure way. But little attention has been paid to the cryptographic algorithm's complexity. The algorithm's complexity has a direct impact on how quickly data is accessed. We require an algorithm that will facilitate effective and quick secure data access [28].

AES is also very efficient, meaning that it can encrypt and decrypt data quickly and with minimal processing power. This makes it well-suited for use in a variety of applications, including large-scale data encryption and real-time communication [29].

This makes ECDH a popular choice for key exchange in secure communication protocols, as it provides a way to securely establish a shared secret key without the risk of key exposure during transmission. Additionally, ECDH is often faster and more efficient than other key exchange methods, it is therefore suitable for use in environments with limited resources, such as mobile or Internet of Things devices. ECDH can be more efficient than other key exchange algorithms, particularly when using smaller key sizes.

Combining AES with ECDH (Elliptic Curve Diffie-Hellman) provides better security than using AES alone because ECDH makes it possible for two parties to securely exchange encryption keys.

In this method, a shared private key between the sender and receiver is created using ECDH., which is then used to encode the data with AES. This provides two layers of security: the ECDH key exchange ensures that only the two parties have access to the key, while the AES encryption provides secure communication once the key is established.

This method is considered more secure than using AES alone because it provides Perfect Forward Secrecy implies that past communications cannot be decrypted even if one party's long-term private key is compromised.

AES encryption and ECDH key exchange can be implemented in Java using the Java Cryptography Architecture (JCA). JCA provides a set of APIs that enable developers to easily implement cryptography in Java applications [30, 31].

We have suggested a model that encrypts text files using AES and ECDH. AES encryption is a fast and efficient encryption algorithm that is well-suited for a wide range of applications. The inputted file is converted into an encrypted database using the AES technique with an ECC key, and Diffie-Hellman will assist in producing a shared secret that is then coupled with the ECC key and uploaded to the server. Using ECDH and AES, the most advanced and reliable cryptographic approach for online storage is produced. Single AES has a larger key size than the hybrid (ECDH-AES) technique, which makes it significantly slower. The hybrid technique, however, has a smaller key size and a speedier security mechanism for safeguarding the data. The performance is increased when advanced encryption standard uses elliptic curve cryptography for encryption since ECDH has a small key size as its main distinguishing feature. ECC minimises key size and creates a protected key system by utilising encryption and decryption key standards. ECC is the ideal methodology to use in combination with AES to secure the information and protect it from unapproved access. Once the key size has been established, the cipher text will result in the encoding and decoding of the data. The combination of ECC and AES is sufficient for the suggested approach of cloud storage to produce a system with higher security. Secure data storage size can be reduced in this way. Figure 3 gives a brief explanation of the working of AES and ECC algorithms.

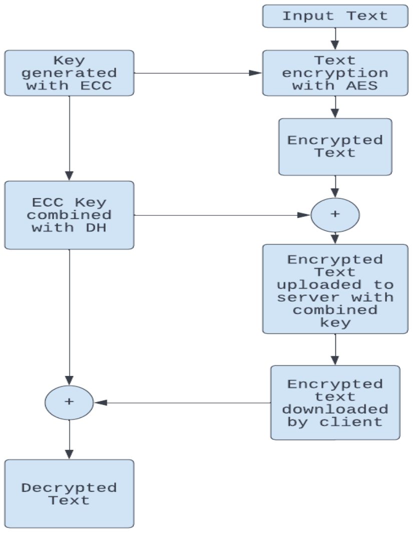

Fig. 3. Execution time of algorithm

Step 1: The client will ask the server for a file, and the server will choose that file for encryption after receiving the client's request.

Step 2: Elliptic curve will produce several private and public key pairs after receiving a file as input. AES will use one of the key pairs produced by an elliptic curve specified over a field to encrypt the text file. AES performs the encryption. One key will be kept secret with the server, say "d," and one key will be kept hidden with the client, say "e," from among the several key pairs.

Step 3: By achieving a successful key agreement between the two communicating parties, ECDH will establish a **shared s**ecret between the client and server. Therefore, the shared secret will only be between the server and client for that specific session, and a third party must discover a way to the discrete logarithm problem to discover the shared secret. Diffie-Hellman will handle the key agreement between the client and server while AES encrypts the input text file on the other side. The user will only be able to decode the encrypted text file if the agreement is successful.

Step 4: The input text file will be encoded with AES using the ECC-generated key. Following encryption, the encrypted content is uploaded to the server using the combined key, or another key that has been received using DHKE as a shared secret. After both the client and the server have successfully established a secret key, the client will obtain the encoded document from the server and decrypt it using the combined key created by ECC and DHKE.

Step 5: Upon successful completion of the decryption, the client will receive the original file.

V. Result And Discussion

Numerous data on cloud storage are protected and get secured connectivity for the encryption and decryption of the data by applying a combination of AES and ECDH, which in a novel method improves the system's distinctiveness and effectiveness. This suggests that by utilising these two techniques, the user will be able to comprehend the original information with ease. Java is used to implement the algorithms (Eclipse Platform Version: 3.3.1.1) A text-based file will serve as the implementation dataset.

The encryption and execution times of our suggested method are used to evaluate it.

The time spent by the system carrying out a process, as well as the time spent by the system performing run-time or system tasks on its behalf, is referred to as the implementation time or computational time of that activity. We have measured the execution duration in this case in milliseconds. The system must be quick and responsive with a shorter execution time. The execution times of AES, ECDH, and AES+ECDH on identical text files are shown in Fig. 4.

Encryption time is the quantity of time needed to convert plaintext to ciphertext. The duration of the encryption is determined by the method, plaintext block size, and key size. We measured the encryption time in our experiment in milliseconds. The system's performance is impacted by the encryption time. Less time must pass for encryption, resulting in a quick and responsive system. Fig. 5 shows the encryption time of AES, ECDH, and AES+ECDH on the same text files. The encryption time of the suggested method is the least

In general, AES is faster than ECDH, as AES is a symmetric encryption algorithm and operates on fixed-length blocks of data, whereas ECDH is an asymmetric encryption algorithm that operates on points on an elliptic curve. However, the speed of AES also depends on the key size, with larger key sizes being slower.

TABLE II. Algorithms with Execution and Encryption time

Algorithms	Execution Time in milliseconds	Encryption Time in milliseconds
AES	145	1.05
ECDH	311	2.32
AES+ECDH	173	1.02

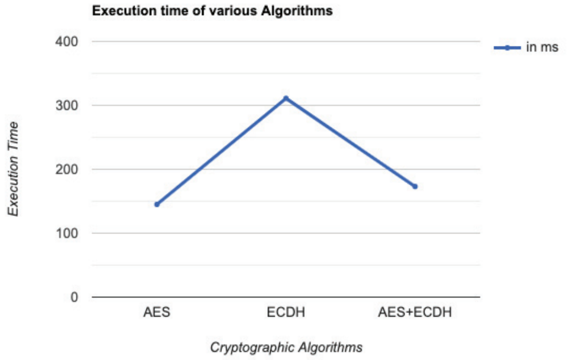

Fig. 4. Execution time of algorithm

ECDH is generally slower than AES, but it has the advantage of being more secure for a given key size. In comparison to RSA, a popular public-key encryption technique, ECDH uses smaller key sizes while offering the same security features as RSA with higher key sizes. The speed of ECDH also depends on the specific implementation and the size of the key being used.

The combination of AES and ECDH is often used to provide both confidentiality and key agreement, and its execution time will depend on the specific implementation and how the two algorithms are combined. It's important to note that the execution time of encryption algorithms is only one factor to consider when choosing an encryption algorithm. Security, compatibility, and other requirements must also be considered when deciding for choosing an encryption algorithm for a specific use case.

The proposed method demonstrates how well AES and ECDH protect data recorded in the cloud. The new proposed model, which depicts secure user information transmission to the server and subsequent secure storage mechanism owing to encrypted data, reveals how imaginative the proposed technique is. Furthermore, computing time and cost may be utilized to assess innovation. To avoid assaults, the following approach can be used: The input file is transformed to encrypted text once the user uploads it using AES encryption, ensuring that the content is entirely encrypted. This is useful if a hacker wants to target the client to obtain private information or for unethical purposes. As a result of this, even if an attack succeeds and the user-uploaded file is retrieved, the data has already been encrypted when the file was submitted. Similarly, if an assault is conducted from the opposite end, the attacker will be unable to decrypt the encrypted file, therefore protecting the data.

VI. Conclusion

If sensitive file data is protected, a cloud-stored file can be transmitted and utilized by others. Cloud storage is rapidly evolving, but it also comes with a slew of negative challenges, including data security concerns, that have greatly hampered the future adoption of cloud storage. When two parties exchange files via an insecure network, data security is essential as it helps protect the information that is being transmitted. Cryptography provides a wide range of techniques to safeguard such communications, allowing information to be securely transferred across the wireless medium while also providing authentication, data integrity, privacy, and non-repudiation. To protect communication from external threats, this research suggests a hybrid model that combines the characteristics of the AES algorithm and ECDH. The suggested prototype is implemented on a client-server model, with the user communicating with the server and securely managing all information using the AES-ECDH hybrid model. The hybrid technique is far more secure, and the combined AES-ECDH security is tough to breach. The obtained findings suggest that this hybrid strategy has a considerable and superior impact over other algorithms due to a reduced encryption time of 1.02 milliseconds.

REFERENCES

[1] Stergiou, C., Psannis, K., & Kim, B. G. i Gupta, B.(2018). Secure integration of IoT and Cloud Computing. *Future Generation Computer Systems*, *78*(3), 964-975.

[2] Hassan, J., Shehzad, D., Habib, U., Aftab, M. U., Ahmad, M., Kuleev, R., & Mazzara, M. (2022). The rise of cloud computing: data protection, privacy, and open research challenges—a systematic literature review (SLR). Computational Intelligence and Neuroscience, 2022.

[3] Albugmi, A., Alassafi, M. O., Walters, R., & Wills, G. (2016, August). Data security in cloud computing. In 2016 Fifth international conference on future generation communication technologies (FGCT) (pp. 55-59). IEEE.

[4] MS, F. (2023). Secure Framework to Enhance Security Using Hybrid Algorithm in Cloud Computing With Ssl.

[5] Li, X., Wang, Q., Lan, X., Chen, X., Zhang, N., & Chen, D. (2019). Enhancing cloud-based IoT security through trustworthy cloud service: An integration of security and reputation approach. IEEE access, 7, 9368-9383.

[6] Tariq, M. I., Tayyaba, S., Jaffar, M. A., Ashraf, M. W., Butt, S. A., & Arif, M. (2022). Information security framework for cloud and virtualization security. In Security and Privacy Trends in Cloud Computing and Big Data (pp. 1-18). CRC Press.

[7] Bharathi, P., Annam, G., Kandi, J. B., Duggana, V. K., & Anjali, T. (2021, July). Secure file storage using hybrid cryptography. In 2021 6th International Conference on Communication and Electronics Systems (ICCES) (pp. 1-6). IEEE.

[8] Hodowu, D. K. M., Korda, D. R., & Ansong, E. D. (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.

[9] Goyat, S., & Jain, S. (2016, August). A secure cryptographic cloud communication using DNA cryptographic technique. In 2016 International Conference on Inventive Computation Technologies (ICICT) (Vol. 3, pp. 1-8). IEEE.

[10] Rehman, S., Talat Bajwa, N., Shah, M. A., Aseeri, A. O., & Anjum, A. (2021). Hybrid AES-ECC model for the security of data over cloud storage. Electronics, 10(21), 2673.

[11] Pavithra, R., Prathiksha, S., Shruthi, S. G., & Bhanumathi, M. (2021). A New Approach for Security in Cloud Data Storage for IOT Applications Using Hybrid Cryptography Technique. In Advances in Parallel Computing Technologies and Applications (pp. 175-182). IOS Press.

[12] Kumar, A., Jain, V., & Yadav, A. (2020, February). A new approach for security in cloud data storage for IOT applications using hybrid cryptography technique. In 2020 international conference on power electronics & IoT applications in renewable energy and its control (PARC) (pp. 514-517). IEEE.

[13] Ariffin, M. A. M., Rahman, K. A., Darus, M. Y., Awang, N., & Kasiran, Z. (2019). Data leakage detection in cloud computing platform. International Journal of Advanced Trends in Computer Science and Engineering, 8(1.3), S1.

[14] Bhanot, R., & Hans, R. (2015). A review and comparative analysis of various encryption algorithms. International Journal of Security and Its Applications, 9(4), 289-306.

[15] Al-Shabi, M. A. (2019). A survey on symmetric and asymmetric cryptography algorithms in information security. International Journal of Scientific and Research Publications (IJSRP), 9(3), 576-589.

[16] Nechvatal, J., Barker, E., Bassham, L., Burr, W., Dworkin, M., Foti, J., & Roback, E. (2001). Report on the development of the Advanced Encryption Standard (AES). Journal of research of the National Institute of Standards and Technology, 106(3), 511.

[17] Sachdeva, A. (2013). A study of encryption algorithms AES, DES and RSA for security. Global Journal of Computer Science and Technology, 13(E15), 32-40.

[18] Heron, S. (2009). Advanced encryption standard (AES). Network Security, 2009(12), 8-12.

[19] Hellman, M. (1976). New directions in cryptography. IEEE transactions on Information Theory, 22(6), 644-654.

[20] Stallings, W. (2006). Cryptography and network security, 4/E. Pearson Education India.

[21] Khalique, A., Singh, K., & Sood, S. (2010). Implementation of elliptic curve digital signature algorithm. International journal of computer applications, 2(2), 21-27.

[22] Xin, L. (2007). An improvement of Diffie-Hellman protocol. Network & Computer Security, 12, 22-23.

[23] Lauter, K. (2004). The advantages of elliptic curve cryptography for wireless security. IEEE Wireless communications, 11(1), 62-67.

[24] D. L. K. Reddy, D. R. Soumya, S. Sahana, N. Rakesh, and others, “Analysis of Various Security Defense Frameworks in Different Application Areas of Cyber-Physical Systems,” in Advancing Computational Intelligence Techniques for Security Systems Design, CRC Press, 2023, pp. 1–20.

[25] Barker, E., Chen, L., Keller, S., Roginsky, A., Vassilev, A., & Davis, R. (2017). Recommendation for pair-wise key-establishment schemes using discrete logarithm cryptography (No. NIST Special Publication (SP) 800-56A Rev. 3 (Draft)). National Institute of Standards and Technology.

[26] Haakegaard, R., & Lang, J. (2015). The elliptic curve diffie-hellman (ecdh). Online at https://koclab. cs. ucsb. edu/teaching/ecc/project/2015Projects/Haakegaard+ Lang. pdf.

[27] V. Maheshwari, S. Sahana, S. Das, I. Das, and A. Ghosh, “Factors Influencing Security Issues in Cloud Computing,” in Advanced Communication and Intelligent Systems: First International Conference, ICACIS 2022, Virtual Event, October 20-21, 2022, Revised Selected Papers, 2023, pp. 348–358.

[28] Zhou, M., Zhang, R., Xie, W., Qian, W., & Zhou, A. (2010, November). Security and privacy in cloud computing: A survey. In 2010 Sixth International Conference on Semantics, Knowledge and Grids (pp. 105-112). IEEE.

[29] K. Negi, G. P. Kumar, G. Raj, S. Sahana, and V. Jain, “Degree of Accuracy in Credit Card Fraud Detection Using Local Outlier Factor and Isolation Forest Algorithm,” in 2022 12th International Conference on Cloud Computing, Data Science \& Engineering (Confluence), 2022, pp. 240–245.

[30] Ametepe, A. F. X., Ahouandjinou, A. S., & Ezin, E. C. (2022). Robust encryption method based on AES-CBC using elliptic curves Diffie–Hellman to secure data in wireless sensor networks. Wireless Networks, 28(3), 991-1001.

[31] I. Das, S. Das, S. Sahana, and A. Kumar, “A Two layer secure image encryption technique,” in 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), 2019, pp. 176–178.

[32] Mell, P., & Grance, T. (2011). The NIST definition of cloud computing.

[33] Mell, P., & Grance, T. (2009). Draft NIST working definition of cloud computing. Referenced on June. 3rd, 15(32), 2.