Educational Administration: Theory and Practice
2024, 30(5), 11645-11650
ISSN: 2148-2403
https://kuey.net/

Research Article

A New Hybrid Approach For Key And Data Exchange In Cloud Computing

Priya Singh[1*], Gaurav Tyagi[2]

[1*, 2]Department of Computer Science and Engineering, SCRIET, Chaudhary Charan Singh University, Meerut, India.
Email: priya976096@gmail.com@gmail.com, gauravtyagi.ccsu@gmail.com

Citation: Priya Singh, Gaurav Tyagi (2024), A New Hybrid Approach For Key And Data Exchange In Cloud Computing, Educational Administration: Theory and Practice, 30(5), 11645-11650
Doi: 10.53555/kuey.v30i5.4989

ARTICLEINFO

ABSTRACT

The rapid adoption of cloud computing has led to a need for robust security measures. Traditional encryption methods may not be sufficient for secure data transfer in the cloud. This paper introduces a hybrid approach for key and data transfer in cloud computing, combining the strengths of Advanced Encryption Standard (AES), RSA, Diffie-Hellman key exchange, and one-time password (OTP) authentication. The method uses AES for efficient data encryption, RSA for secure key exchange, and Diffie-Hellman for generating shared keys. OTP adds an additional layer of authentication during user access. The approach consists of a three-phase process: registration, login, and data transfer, ensuring secure sharing of keys and data between authorized parties. This hybrid approach aims to provide data protection and user authentication, mitigating risks of unauthorized access and data breaches.

Keywords— AES, Diffie Hellman, OTP, RSA.

Introduction

In earlier time, we used to manage and store data by ourself limiting us on the aspect of storage and data management. As the size of data increases these limitations developed into serious issues in the business world. And then storing data at a remote location where you do not need to worry about its management and storage called cloud storage was introduced. This technology developed rapidly as this provided user with convenience in storing their data. But with allowing others the access to manage and store our data there arises issue of potential threat to data security. The remote data centers that are accessed via the internet and managed by cloud service providers have replaced the on-site infrastructure of the personal and corporate computing models. However, this paradigm shift in computers has led to security concerns for both corporations and consumers. These security issues need to be carefully considered and resolved in order to boost cloud implementation (Mohammed et.al, 2021). To tackle these threats many cryptographic algorithms were introduced, which encrypt the data so no other than the one who had ethical access could read the data. There are 3 types of cryptography algorithms based on the type of key used- symmetric, asymmetric algorithms and hash functions (Kaushik et.al, 2023). Asymmetric algorithms operate on the principle of a public and private key, where the public key is shared with another individual while the private key is kept to oneself. Thus, if another party encrypts data using its public key, only the first party will be able to decode it since he only has the private key (Pansotra & Singh, 2015). Symmetric algorithms utilize the same key for both data encryption and decryption.

Though there are many strong symmetric key algorithms like AES (advanced encryption standard) but the main security threat is in sharing the symmetric key to the other party. In this paper, we tried to introduce an approach to share the key securely so data can be transmitted between two parties securely. When Data Encryption Standard (DES) algorithm was breached, there rises the need of a more secure algorithm and thus came Advanced Encryption Standard (AES) algorithm. AES is a symmetric key encryption algorithm, where data is encrypted and decrypted using the same key thus making the processing faster. But issue of security attack rises on sharing the shared key to the receiver so it can decrypt the data. This issue is not present in asymmetric key encryption algorithm like RSA where a key pair of public and private key are generated on both sender and receiver side.

In this paper our objective is to provide AES level encryption to data without sharing actual key on the connection between user and cloud service provider. It is to introduce an approach to provide a secure key and data exchange in symmetric key encryption in Cloud Computing. We are proposing a hybrid paradigm where we will use Diffie-Hellman algorithm (Verma et.al, 2017), RSA and AES algorithm. This is to maintain security in sharing key and data in symmetric key algorithm AES (Advanced Encryption Standard).

Cryptography Algorithms

Cryptography is the art of ensuring security and safety of data by transforming it into form that unauthorised receivers cannot understand. It is the technique of encrypting or coding data such that a communication is only read by the intended recipient ("What is Cryptography?", n.d.). Asymmetric and symmetric cryptography algorithms make up the two main categories of cryptography algorithms. The same key, or symmetric cryptography algorithm, uses the same key for both data encoding and decoding. Symmetric key algorithms abound and include DES, AES, and others. Additionally, two distinct keys were employed by asymmetric cryptography techniques to encode and decode data. Here one key is kept hidden while other is made public thus called private and public keys respectively. Asymmetric Key Encryption is having more benefit over the Symmetric Key Encryption as they can be used easily and effectively for transferring of the encryption keys or other information securely even when both the users are not there, also since Asymmetric Key algorithms uses longer keys than Symmetric Key its data is more securely transferred. Asymmetric Key have a drawback that it works on low speed i.e. it is slow and not feasible to encrypt massive data (Chopra, 2015). The asymmetric enciphering strategies are roughly 1,000 times slower than symmetric encoding, which makes it unfeasible upon encoding big amounts of information (Abood & Guirguis, 2018).
In this paper we are going to use asymmetric key algorithm RSA and symmetric key algorithm AES (Advanced Encryption Standard) and Diffie Hellman key exchange to enhance the security in key and data exchange in cloud computing.

Related Work

As cloud computing becomes more and more popular, people are storing an increasing amount of private and sensitive data on the cloud. One of the key security concerns in cloud computing is cloud storage security (Min & Yang, 2019). For a particular cloud service, R. K. Seth, Rimmy Chuchra (2014), and Simran suggested a system that uses digital signatures with automatically issued TOKEN IDs to offer security. A completely homomorphic encryption method, Amit Verma, Ramandeep Brar, and Amandeep Ummat (2017) created novel modalities for key management and sharing. A symmetric key encryption system was created by Md. Abu Musa and Md. Ashiq Mahmood (2021). A file would be encrypted locally at the client-side before being uploaded to the cloud, and it would use the key obtained during encryption to decode the file once it was downloaded on the client-side. In 2023, Bharti Kaushik, Vikas Malik, Vinod Saroha provide review of various types of cryptography techniques. In 2014, Richa Singh and Amit Kumar Sharma provide review of RSA, Random mask, Hash and MAC for cloud data security and their limitations. The privateDH algorithm was invented by Ripon Patgiri (2021). It encrypts all shareable data during key exchange using the AES algorithm and utilizes the RSA technique to transfer the AES symmetric key. This prevents any data from being shared publicly with the intended party. In 2023, Aviral Srivast and Aryaman Kumar introduces a hybrid approach by employing AES, With a symmetric block cipher, files may be encrypted using a 128-bit key that is created at random for every encryption instance. The AES key is then encrypted using a strong RSA 2048-bit public key to provide an additional degree of protection. Three variations of the Diffie-Hellman protocol were studied by Manoj Ranjan Mishra and Jayaprakash Kar (2017) the authenticated key exchange protocol, the one-pass key exchange protocol, and the Diffie-Hellman protocol itself. ECC and Blowfish were merged by Chinnasamy Ponnusamy (2020) to create a hybrid algorithm. When the hybrid system's performance is compared to the current hybrid approach, it becomes clear that the suggested approach offers the highest level of patient data security and confidentiality. By using hybrid cryptography, the drawbacks of symmetric and asymmetric algorithms are overcome.

Methodology

There will be two parties involved in this methodology: the cloud client and the cloud service provider (Seth et.al, 2014). We may refer to them as sender and recipient, user and supplier, etc., for the sake of convenience and clarity.
Three steps comprise the methodology: the registration phase, the login phase, and the data transmission phase.

1. Registration Phase

This stage involves the user registering for the first time to utilize the cloud service provider's services. We will utilize the Diffie-Hellman method with RSA for sharing to produce a shared key when the user initiates the request to register on the Cloud Service Provider Interface. The sender and the recipient will each produce variables a and b according to the Diffie-Hellman algorithm (Verma et.al, 2017). Before creating a symmetric key, they will both need to select two numbers, n and p, where n is a prime number and p is a primitive root

mod n that will be made public. Now both sender and receiver will have the values of n and p. The Sender will calculate value of variable M where M= (p^a mod n) and Receiver will calculate value of variable S where S= (p^b mod n).

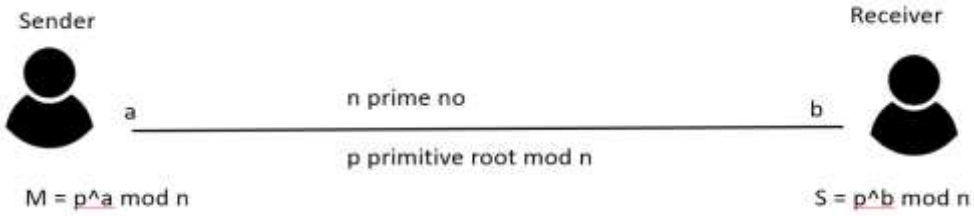

Figure 1. Sender and Receiver calculating values of M and S using n, p.

Now these values of variable M and S needs to be exchanged between the sender and the receiver so they can calculate the symmetric key. To share value of M and S we will use RSA Algorithm. As per RSA Algorithm both sender and receiver will generate key pair- public and private key. Sender will share its public key with the receiver and receiver will share its public key with the sender.

Figure 2. Sender and Receiver generated key pair.

Sender will encrypt M value by receiver's public key thus it can be decrypted only by receiver's private key. Sender will also generate a hash from M value and will encrypt this hash using its own private key. Sender will send both the encrypted M value as well as the encrypted hash to receiver. On the receiver side, it will decrypt the M value using its own private key and will ensure data integrity by decrypting the hash value and calculate a hash by M value decrypted and check whether both hash matches to ensure that data is not altered.

Similarly, Receiver will also encrypt the value of S using the public key of sender and will also encrypt the hash generated from value of S using its own private key. When sender will receive the data, it will decrypt the value of S using its own private key and will also decrypt the hash value using the public key of receiver and will check whether hash generated from decrypted value of S will match with the decrypted value of hash to ensure data integrity.

Figure 3. Sender and Receiver sharing M and S values.

Now Sender and Receiver can calculate value of shared key K. Sender have the value of a and S and will calculate shared key $K=S^a = (p^{ab})$mod n. Receiver have value of b and M and will calculate the value of shared key $K = M^b = (p^{ab})$ mod n. After the shared key K is generated successfully on both the side, cloud service provider will provide a unique user-id to the user for future login to avail services provided by the cloud service provider. After generating user-id, user will also provide basic details like name, age, personal mobile number or email id etc.

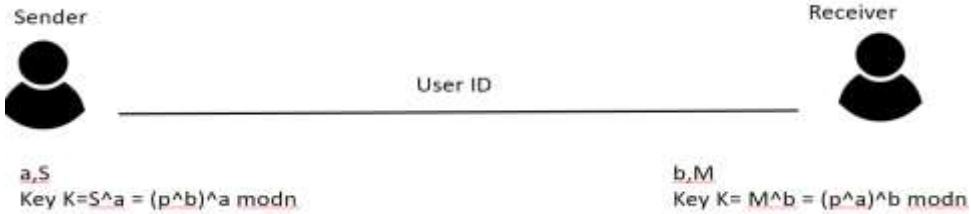

Figure 4. Sender and Receiver generating the shared key K.

At this stage the registration phase is complete and user has successfully registered with the cloud storage provider.

2. Login Phase

Whenever after the first-time registration of the user, if the user tries to avail the services it will login with the interface provided. From this stage the login phase will start. In the login phase, user will first provide its user-id to login with the provider. In response provider will generate an OTP and send it on the registered mobile number.

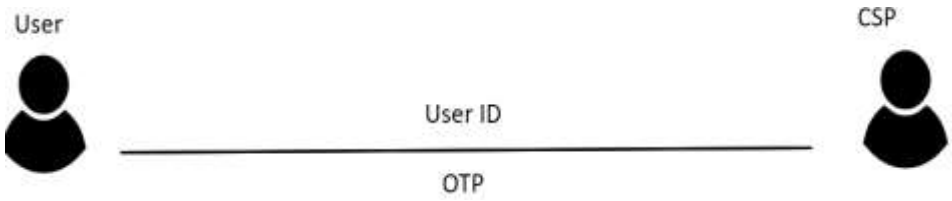

Figure 5. User Login

User will have to enter a password at this stage. Password will be OTP + Key K and sender will generate the hash value from the password and will send it to the provider. Provider also have the shared key K so it will validate the hash by generating the password on its side (OTP +Key K) and generate its hash. If hash provided by user and hash generated by provider matches then the user will be provided access and thus successfully logged in.

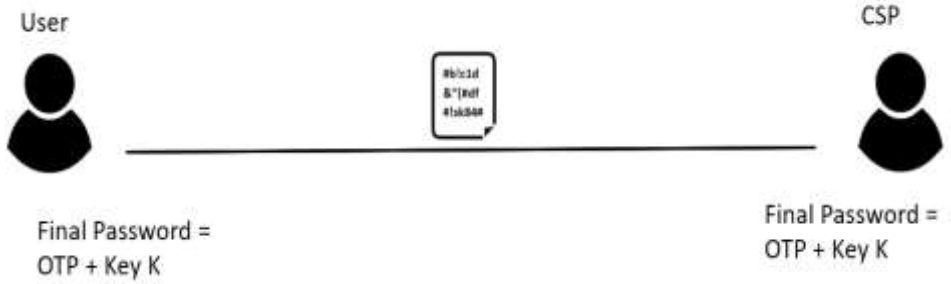

Figure 6. Generating password from OTP and shared key K.

After the successful login, user can perform the data transfer operation which will be discussed briefly in the data transfer phase.

3. Data Transfer Phase

This is the phase where actual data transfer takes place. For data transfer we will use AES encryption algorithm which is a symmetric key algorithm. Both parties need to have the symmetric key to perform encryption or decryption. To generate symmetric key, user will generate a string and will share it to the provider in encrypted form. The string s will be encrypted by public key of other party to generate cipher text, which will be shared to the provider. Provider will decrypt the string from the cipher text using its own private key. From this string we will generate the final string used to form the symmetric key for AES encryption. Final String str = s + Key K, here final string will be formed by concatenating the string shared by user with the shared key K generated using Diffie-Hellman algorithm. This final string str will be used to generate the AES symmetric key.

Figure 7. Generating AES symmetric key

Now the symmetric key is generated on both the sides. User will send its data encrypted using the AES symmetric key generated to the provider where the provider can decrypt the data when needed to perform operations on it.

Figure 8. data transfer

Implementation

Refer to Table 1 for nomenclature.

Table 1. Nomenclature for algorithm

CSP	Cloud Service Provider
DH	Diffie Hellman
REQ	Request
RES	Response
ACK	Acknowledgement
REG	Registration
DAPP	Decentralised Application
CSPMS	Cloud Service Provider Microservice
CSPDB	Cloud Service Provider Database
EN	Encryption
DE	Decryption
PUBK	Public Key
PRIK	Private Key
U_ID	User Identification Number

1. When user send REQ through DAPP for the first time, then REG phase will start:
1.1 RSA key pair generation : Event : KEYSHARE
1.1.1 both parties will generate PUBK, PRIK.
1.1.2 PUBK will be shared to each other.
1.2 DH key exchange Event : SHAREDKEYGEN
1.2.1 CSPMS will generate value of n (prime no) and p (primitive root mod n)
1.2.2 CSPMS send n, p values in RES.
1.2.3 User generate a random value a and CSPMS generate a random value b.
1.2.4 Calculation User: M = p^a mod n & CSPMS: S = p^b mod n.
1.2.5 User RES: send M EN by PUBK of CSPMS + hash of M EN by PRIVK of User
1.2.6 CSPMS Side Calculation:
- DE of M and check hash(M_{de})==hash given
- Derive final key K=M^b = (p^a)^b modn

1.2.7 CSPMS ACK :
- send S encrypted by PUBK of User + hash of S EN by PRIVK of CSPMS
- a unique U_ID to user.
- Store key K corresponding to U_ID in CSPDB.
- 1.2.8 User Side Calculation :
- DE of S and check hash(S_{de})==hash given
- Derive final key K=S^a = (p^b)^a modn
- Key K stored in DAPP storage.

2. When existing User send REQ: Login Phase Starts
2.1 User send U_ID in REQ Event: LOGIN
2.2 CSPMS send OTP to User.
2.3 User send RES= hash(OTP+K)
2.4 CSPMS validate RES
3. When User try to upload or download any data: Data Transfer Phase
3.1 User generate string s, send to CSPMS EN by PUBK of CSPMS
3.2 User Side: s + Key K string used to generate AES Key
CSPMS Side : s+ K string used to generate AES key.
3.3 Data EN by AES Key from User side and can be decrypted by CSPMS if req Similar data transfer phase will follow in case of download data as well.

Result

A hybrid technique has been presented to address the security issue of sharing a symmetric key between two parties for AES encryption while exchanging data The adoption of a hybrid cloud security strategy has significantly improved user authentication and data security in cloud settings. This approach uses Advanced Encryption Standard (AES), RSA, Diffie-Hellman key exchange, and one-time password (OTP) approaches to create a secure framework for data transmission and communication. AES data encryption ensures privacy, while RSA and Diffie-Hellman protocols facilitate secure key exchange and shared keys. The incorporation of OTP adds verification to the login process, assisting in user identity verification and guarding against unwanted access. The hybrid solution effectively solves common cloud security issues by providing a smooth and safe channel for data and key transfer, reducing risks of data breaches, illegal access, and phishing attempts. The hybrid method's flexible procedures and layered security architecture make it suitable for various cloud computing scenarios, offering potential for future real-world applications and further research into safe data transmission techniques.

Conclusion

In context of the security issue in sharing symmetric key between the two parties for performing AES encryption in sharing data, a hybrid methodology has been introduced where we are using AES, RSA and Diffie-Hellman algorithm to provide security in sharing key and data in cloud computing. The proposed approach is divided into three phases – registration phase, login phase and data transfer phase. In registration phase, user will register with the provider for the first time. In this, we will generate shared key using Diffie-Hellman and RSA for secure exchange. A user-id will be provided to the user for future login. In login phase, user will provide the user-id and an OTP will be generated. User will share the hash of the final password generated = OTP + Key K. This will be authenticated on provider side as well. After successful login, user can share data in data transfer phase. User will generate a string and will share it in encrypted form by using public key of other party. Final string will be form by adding shared key K to the string. Final string will be used to generate the symmetric key for AES encryption. After the symmetric key is generated, it is used for sending data in encrypted form and for decrypting it on provider side. The proposed approach may provide security in sharing symmetric key and data in cloud computing.

Future Work

Future research will concentrate on putting the hybrid technique for key and data transmission in cloud computing into practice and evaluating it. This entails testing the technique's scalability and performance in actual cloud settings. Its resilience will be confirmed by security testing against a range of attacks and vulnerabilities. Usability will be improved by initiatives to improve user experience, especially with regard to integration and authentication procedures. Strengthening security and privacy will need research into sophisticated authentication mechanisms and privacy-preserving strategies. Working together with business partners will make it easier to get feedback and real-world data, which will lead to focused changes. Finally, the approach's description and distribution will help to further current developments in cloud security by disseminating information to the research community.

References

1. Abood, O. & Guirguis, S. (2018). A Survey on Cryptography Algorithms, International Journal of Scientific and Research Publications, 8. 495-516. 10.29322/IJSRP.8.7.2018.p7978.
2. Chopra, A. (2015). Comparative Analysis of Key Exchange Algorithms in Cryptography and its Implementation, IMS Manthan (The Journal of Innovations). 8.10.18701/imsmanthan.v8i2.5126
3. Kaushik, B., Malik, V. & Saroha, V. (2023). A Review Paper on Data Encryption and Decryption, International Journal for Research in Applied Science & Engineering Technology (IJRASET), ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538 Volume 11 Issue IV
4. Min, Z. E., Yang, G. (2019). Homomorphic Encryption Technology for Cloud Computing, 8th International Congress of Information and Communication
5. Mishra, M. & Kar, J. (2017). A study on diffie-hellman key exchange protocols, International Journal of Pure and Apllied Mathematics. 114. 10.12732/ijpam.v114i2.2.
6. Mohammed, K. K., Abdulrahman, A. A. & Muhammad, A. (2021). Cloud Security. 10.13140/RG.2.2.13876.58242.
7. Musa, Md. & Mahmood, Md. A. (2021). Client-side Cryptography Based Security for Cloud Computing System, 10.1109/ICAIS50930.2021.9395890.
8. Pansotra, Er. & Singh, S. P. (2015). Cloud Security Algorithms, International Journal of Security and Its Applications, 9. 353-360. 10.14257/ijsia.2015.9.10.32.
9. Patgiri, R. (2021). privateDH: An Enhanced Diffie-Hellman Key-Exchange Protocol using RSA and AES Algorithm, 10.13140/RG.2.2.23938.40647.
10. Ponnusamy, C., Padmavathi, S. & Swathy, R. (2020). Efficient Data Security Using Hybrid Cryptography on Cloud Computing, 10.1007/978-981-15-7345-3_46.
11. Seth, R. K., Chuchra, R. & Simran. (2014). TBDS- A New Data Security Algorithm in Cloud Computing, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 2703-2706
12. Singh, R. & Sharma, A. K. (2014). A Comparative Study: Various Approaches for Cloud Data Security, (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2).
13. Srivast, A. & Kumar, A. (2023). A Robust Approach to Secure Data Encryption: AES-RSA Hybrid with Kernel Key Protection, 10.21203/rs.3.rs-3565782/v1.
14. Verma, A., Brar, R. & Ummat, A. (2017). Cloud Computing and Homomorphic Encryption International Journal of Computer Science and Information Security (IJCSIS), Vol. 15, No. 3, March 2017
15. "What is Cryptography?" [Online] Available: https://www.fortinet.com/resources/cyberglossary/what-is-cryptography [Accessed January 1, 2024].