ISSN:1991-8178

Australian Journal of Basic and Applied Sciences

Journal home page: www.ajbasweb.com

OTP: RSA Based Key Aggregation Mechanism for Secured Data Sharing in Cloud

[1]Babeetha S and [2]Muruganantham B

[1]Assistant Professor, Department of Computer Science, Saveetha School Of Engineering, Saveetha University, Thandalam, Chennai.
[2]Assistant Professor, Department of Computer Science, SRM University, Chennai.

ARTICLE INFO

Article history:
Received 12 March 2015
Accepted 28 April 2015
Available online 1 June 2015

Keywords:
Cloud storage, RSA Algorithm, Triple Data Encryption Standard, Cipher text, Encryption, Decryption.

ABSTRACT

Cloud computing domain provides the standard architecture to share the application same as the other networks sharing. It provides the online storing facility on several servers which is organized by third party rather than the service provider. Sharing the key and managing is playing a vital role, the sharing process over cloud computing. Usual processes of the key cryptosystem lacking with the efficient safety method those keys are produced by the existing system. Existing system proposing the thoughts about aggregate cryptosystem in that key getting generate by the several origin of the cipher text class which properly ties the data and related keys. The aggregate key was producing once after losing the key it was not possible to access the data. To overcome the abovementioned problems, we propose a novel technique named RSA based Key aggregation Mechanism which consists of two algorithms such as Triple-DES (Triple-Data Encryption Standard) algorithm and RSA (Rivest, Shamir and Adleman) Algorithm. The Triple-DES algorithm is used for file encryption and decryption. The RSA Algorithm is used for key generation and the generated key is utilized as OTP (one time password). This OTP is considered as the authentication or confirmation from the new user. Moreover, this One Time Password should be act as an aggregation key. The experimental results are proved the proposed approach aggregates the keys very effectively rather than other approaches.

© 2015 AENSI Publisher All rights reserved.

To Cite This Article: Babeetha S and Muruganantham B, OTP: RSA Based Key Aggregation Mechanism for Secured Data Sharing in Cloud. ***Aust. J. Basic & Appl. Sci.,*** *9(11): 651-656, 2015*

INTRODUCTION

Nowadays, the cloud storage is too much famous storing process. The third party is doing maintain for storing the data in physical storage. The third party or authenticator is saving the digital data in physical storage or logical hub by using several servers. The data availability and accessibility in physical environment and protection is the third party responsibility (Wang, 2013). Better to save the data in cloud storage, which can be accessible from anywhere and anytime, instead of saving in local storage disk or any hard drive. It decreases the labor to carry the data. We can access the data which is store in the cloud storage from any computer by using internet and it provide a facility to access the information from any computer not only with same computer where it stored.

While consider about data security, here is tough to trust on the traditional method for authentication, because of unpredicted dispensation appreciation will explore all the data.

The solution is that have to encrypt the data with user key. The sharing of data is the important functionality in cloud storage, as the user could do share the data to anyone from anywhere at any time. For instance, organization could allow for accessing the sensitive part of any data for their employees. But the difficult part is that how to do share the encrypted data (Wang, 2013). By the traditional way it was easy to share the data by downloading and decrypting the data but there use of cloud storage wasn't remaining anymore. Cryptography technique could be applied in two major ways:

1. Symmetric key encryption.
2. Asymmetric key encryption.

In symmetric key encryption, for decryption and encryption both process only one key was getting use. By compare, in asymmetric key encryption different keys are getting use, private key and public key for decryption and encryption respectively. In our approach asymmetric key encryption is flexible. This can be discussed by the given example. Assume that Alice saved all her data on Box.com and doesn't want to explore any data with anyone. Because of the possibilities of data leakage she doesn't believe on the facility which is provided by the Box.com, so she encrypting all the data previous to uploading. If she want to share some data with Bob, so simply she can share the data with Bob by sing share function of

Corresponding Author: Babeetha S., Assistant Professor, Department of Computer Science, Saveetha School of Engineering, Saveetha University, Thandalam, Chennai.

Box.com. But the main issue is that how could she share the encrypted data (Chow, 2012). There are two techniques to share the encrypted data:

Alice can encrypt data with a single secret key and she can share the secret key with bob for decrypting the encrypted data.

Alice can encrypt the data with different key and send the related key to the Bob via channel after that Bob can access the data.

These both the technique is insufficient for the users. In second technique the number of secret key is as much as the shared files, which may be not possible to decrypt the data easily.

Hence, the suitable solution for this problem is that Alice has to encrypt all the data with different keys, but send only decryption key to bob, for using the data, at constant size. While the decryption key is sent via the secure channel and it's difficult to keep secure that small size. For designing an enhanced public key encryption system that suitable for the allocation, if any partition of the cipher text is decrypt able by constant size.

The rest of the paper is organized as follows. Section II which overviews the related work. Then we provide the detailed description of our proposed scheme in Section III. Section IV gives the results and discussions of our proposed scheme. Finally, Section V concludes the whole research work.

Related work:

In the research work (Benaloh, 2009), the Author Benaloh *et al*. has proposed a scheme which is really presented for concisely transferring huge number of secret keys in scenario (Benaloh, 2009). The creation is easy and we evaluate the key origin method for actual definition of what are the popular characteristics that we want to gain. The key derivation for the set of classes is as follows. A complex modulus is selected where p and q are large prime. A master secret key is selected at random. Every class is connected with different key. All the prime keys could be kept in the system parameter. A constant key for the set could be generated for the assigned accessing right. Though, it is designed for the symmetric-key setting instead. The data providers need to get the keys for encrypting the data, which will not suitable for the other applications, because this technique is used to produce a value rather than a pair of key, but it is not clear that how to apply this method for the public key encryption scheme. Finally, we note that there are schemes which try to reduce the key size for achieving authentication in symmetric-key encryption (Boneh, 2001). However, sharing of decryption power is not a concern in these schemes.

IBE (Identity-based encryption) (Alomair, 2009; Sahai, 2005; Guo, 2007), is a public key encryption that user could make a identity-string. The PKG (private key generator) in identity-based encryption holds a secret key and generate a secret key for the every user with the proper identity. The author Guo *et al* (2010; Guo, 2007) had tried to make Identity-based encryption with the secret key aggregation. In this method, the key aggregation controlled the sense that every keys to be aggregated must have to validate from the distinct location. When there is an exponential number of identities and respective secret keys, then only a polynomial number can be aggregated. These particularly increment the costs of storing and transferring the cipher text that is unfeasible with more condition like sharing the data from cloud storage. The way to apply Hash functions in the string for denoting the class, and continue with the function until obtaining the output for the hash function. We stated our methods characteristics constant for the cipher text size, and security in the model. In research work (Chow, 2010), the fuzzy identity-based encryption (IBE), a single secret key can decrypt cipher text encryption under many more identities which are near to the assured metric gap or space, but not for the identities arbitrary and then doesn't match with the key aggregation idea. ABE (Attribute-based encryption) (Chase, 2009; Goyal, 2006) allows every cipher text for associating the attribute, and the secret key holder could easily explore the information from the decrypted cipher text policy. For instance, with the secret key for the policy ($1 \vee 3 \vee 6 \vee 8$), one can decrypt cipher text tagged with class 1, 3, 6 or 8. Though, the aim issue is that Attribute-Based encryption is collusion-resistance but not the exact secret keys. Indeed, the size of the key often increases linearly with the number of attributes it encompasses, or the cipher text-size is not constant (Okamoto, 2011).

Proposed work:

Overview:

The preliminary functions of our proposed framework are shown in the Figure 1. Our proposed System implements RSA Based Key Aggregation Mechanism for Secured Data Sharing in Cloud by using RSA algorithm and Triple Data Encryption Standard Algorithm. The system model in this research work involves three parties: the cloud service provider, data owner and new user. Initially, the original user or data owner stores his/her own data file in the cloud store. Before storing the data, the user must encrypt the data file using Triple Data Encryption Standard (Triple-DES). This algorithm improves the privacy of stored data and it is used for secured data sharing. After that, a new user sends request to the data owner for download or access the data. Then, the data owner gives access permission for particular files and provide aggregate key for that. By using RSA algorithm, the user can generate the aggregation key for data sharing. This aggregate is utilized as One Time Password which is generated by authenticator that is sending to the requested users which keep the secret key secure from malicious users. Moreover, this One Time Password (OTP)

prevents the data leakage. After completing the aggregation process the requested user can decrypt the respected data from cloud storage.

Working Methodology:

Our proposed Framework consists of three major stages. Such as

Cloud Deployment and Data Encryption
User Registration and Key Distribution
Data Retrieval and Key sharing

1. Cloud deployment and Data Encryption:

This stage describes the details about the cloud deployment. The cloud contains cloud service provider, data owners and users. The data owner is connected with cloud service provider. Here the data owner directly handles the data. The Data owner provides secret key to the requested user. Initially the Data owner user makes register by entering the required details and login to the cloud. Data owner is an entity who owns his messages or data and wishes to store them into the external data storage node for ease of sharing or for reliable delivery to users in the extreme cloud environments. The Data owner encrypts the data using Triple DES (Data Encryption Standard) algorithm. This algorithm is used for secured data sharing.

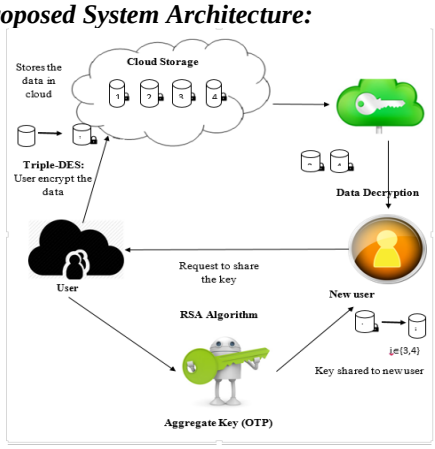
2. User Registration and Key Distribution:

This stage describes the details of user registration and key distribution process. Initially, user gets registered with cloud storage then only the user can access the data of respected data owner with his/ her permission. The user gives request to the data owner for accessing the stored data of data owner. The user must provide the access rights to the other user as the data is encrypted and the decryption key should be send securely.

3. Data Retrieval and Key sharing:

This stage describes the details about secured key sharing technique or RSA algorithm. Initially, user gives a request to the data owner then the data owner send aggregated key for requested user using RSA (Rivest, Shamir and Adleman) Algorithm. Aggregate key is used for the secured data sharing over the distributed data sharing in cloud environment. Aggregate key consist of various derivation of identity and attribute based classes of respective data owner in the cloud. Aggregation key is used to sharing the data between one to other. The key aggregation property is especially useful when we expect the delegation to be efficient and flexible. Key aggregation enables content provider to share other's data in a confidential and selective way, with a fixed and small cipher text expansion, by distributing to each authorized user a single and small aggregate key. By using this aggregate key, the requested user can able to download and view the requested data only and remaining data were not accessible. This algorithm improves the key sharing strategy. The key sharing is done by the One Time password (OTP) method, which Restricting the other users or malicious behavior from stolen the secret key. The key is transferring by the secure method where the owner sending one password to the related user for confirmation of the sharing the key with the users to access the database. Without matching of the confirmation password authenticator does not generating any key for the specific request. Moreover, for improving the security of our proposed mechanism utilize 2048 bits key for key aggregation. Triple – Data Encryption Standard Algorithm is used to secure data encryption and decryption process.

3.3 Proposed System Architecture:



3.4 Algorithms Used:

There are two algorithms used in our proposed RSA Based Key Aggregation Mechanism. These are

3.4.1 Triple DES:
3.4.2 RSA Algorithm:
1. Triple DES:

In our proposed work, we are using Triple DES (Data Encryption Standard) algorithm for both the data encryption and decryption process. General steps are given below:

Triple-DES is used as a "Key Collection" that consist three DES keys such as Key1, Key2 and Key3. Each and every key size is 56 bits length without the parity bits.

The triple-DES encryption algorithm process is below:

Cipher text = EKey3 (DKey2 (EKey1 (plain text)))

The Data Encryption Standard (DES) method encrypts the normal text with Key1 then decrypt with Key2 then again DES method encrypt the text with Key3

The Triple-DES Decryption is the reverse of encryption process which is shown in below

Plain text = DKey1 (EKey2 (DKey3 (cipher text)))

The DES method decrypts cipher text with Key3 initially then it encrypt with Key2 again it decrypts with Key1

Each time 64 bit of data is encrypted by triple encryption method.

The center process overwrites process of first and last in every test case.

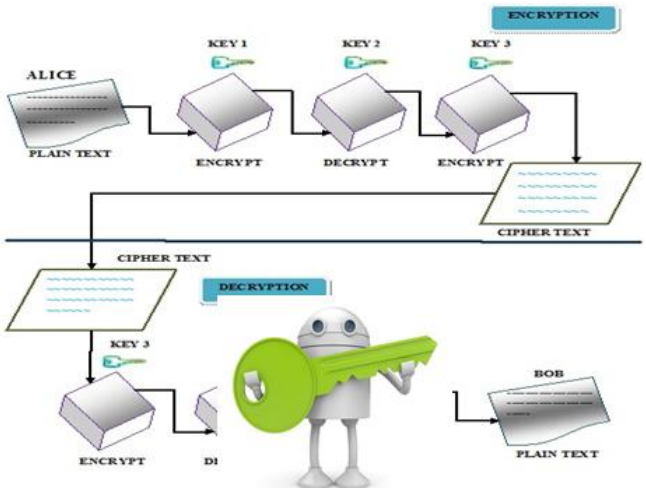


Fig. 1: Symmetric Key Triple DES Encryption.

3.4.2 RSA Algorithm:

Currently, the Rivest, Shamir and Adleman algorithm is one of the most widely used algorithm. Basically, message encryption, m, entails exponential function, $c = m^e mod\ n$ a mathematical process that needs a couple of calculation, which making an unfeasible path to gain private key system like DES, an occurrence that is exact for public key system. To establish RSA cryptosystem, Alice (a user) choose two prime i and j then computes the product n=ij. The multiplicative group is used by the group (G=Z) units in integer module n. It is known as the order of G is

φ =(i-1)(j-1), in which Euler phi function is denoted by φ . Clearly, Alice's public key is the pair of integers {n,c} and her private key is k.

3.4.3 RSA Key Generation:

The RSA key could be generating by the algorithm 1. The public key contains a integers pair (n,c) where n is the RSA Module that is generated by two prime i and j of the same length. The exponent e is satisfying integer 1<c< φ and gcd(c, φ)=1,

Where φ = (i−1)(j −1). The private key k, also called the decryption exponent, is the integer satisfying 1 < k < φ and cd ≡ 1 (mod φ). It has been proven that the problem of determining the private key k from the public key (n, c) is computationally equivalent to the problem of determining the factors i and j of n; the latter is the integer factorization problem (IFP).

Algorithm1 RSA key generation

INPUT: Security parameter s.

OUTPUT: RSA public key (n,ce) and private key k.

1. Randomly select two primes i and j of the same bitlength s/2.
2. Compute n = ij and φ = (i−1)(j −1).
3. Select an arbitrary integer c with 1 < c <φ and gcd (c, φ) = 1.
4. Compute the integer k satisfying 1 < k <φ and cd ≡ 1 (mod φ).
5. Return (n, c, k).

RESULTS AND DISCUSSIONS

The study so far shows the advantage of our proposed RSA Based Key Aggregation Mechanism over other algorithms in terms of the providing more security. To evaluate the performance of our proposed approach, a series of experiments on stored dataset were conducted. In these experiments, we implemented and evaluated the proposed methods following configuration Windows 7, Intel Pentium(R), CPU G2020 and processer speed 2.90 GHz.

Encryption Time:

Here, we described about existing algorithms have been executed on Java platform on a number of data files differing content types and sizes of a wide range of files and proposed algorithm will be implementing in future on the same platform but, at present we are showing expected results of the proposed algorithm as compared to the existing algorithms. Encryption time (approximately) of different file comparisons shown in table 1.

Table 1: Comparison of File Encryption between Existing and Proposed Algorithm.

S.NO	FILE SIZE	DJSA	NJJSAA	PROPOSED ALGORITHM
	In KB	Execution Time in Second (Approximately)		
1.	189	18-20	16-19	8-10 (Expected)
2.	566	37-40	34-38	25-30 (Expected)
3.	1136.64	70-73	65-67	50-55 (Expected)
4.	1495.04	97-99	70-72	60-65 (Expected)

4.2 Results:

4.2.1 Accuracy of Proposed System:

The figure 4 illustrates the key aggregation accuracy of our proposed algorithm and other traditional algorithms. Furthermore, RSA Algorithm has a better Accuracy than other common encryption algorithms which are shown in figure 3.

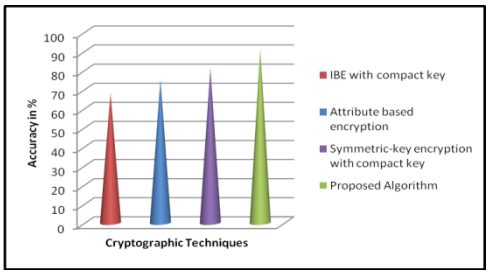


Fig. 3: Accuracy Vs Number of cryptographic Techniques.

4.2.2 Efficiency of Proposed System:

Here proposed algorithm proposing expected efficiency in terms of key aggregation time. So following figure are showing results in terms of key aggregation time. Proceeding more efficiency will also be looked after in terms of the total CPU clock & process time for all the three considering algorithms. Our proposed Key aggregation algorithm utilizes minimum time duration for key aggregation process while comparing other traditional algorithms like IBE with compact key and Attribute Based Encryption Algorithm.

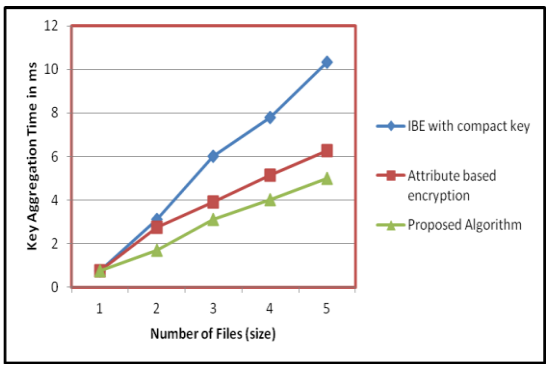


Fig. 4: Processing Time VS Number of Files.

Conclusion:

We are implemented RSA Based Key Aggregation Mechanism for secured data sharing and key aggregation by using Triple-DES Algorithm and RSA Algorithm. Basically, the encrypted data stores in the cloud storage by the users. Then after, we utilized RSA algorithm for the safe access and key aggregation. The smooth data sharing is a vital role in cloud computing. Users go to upload their data on cloud storage and amid of distinct users. The data outsourcing is increasing the susceptibility of data leaking. Encryption is the perfect way to share data securely with the desired users. For securing sharing of the secret key owner using OTP technique that is providing the safest way for the sharing the secret or aggregate key. The sharing of secret key is act an imperative character. The cryptosystems of public-key providing the delegation of secret public key for distinct cipher text in cloud storage. So finally, our proposed work is capable to gain the files and key values. This will make a secure sharing of key and files from the misusing.

REFERENCES

Alomair, B. and R. Poovendran, 2009. Information Theoretically Secure Encryption with Almost Free Authentication,‖ J. UCS, 15(15): 2937–2956.

Benaloh, J., 2009. Key Compression and Its Application to Digital Fingerprinting,‖ Microsoft Research, Tech. Rep.

Benaloh, J., M. Chase, E. Horvitz and K. Lauter, 2009. Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records,‖ in Proceedings of ACM Workshop on Cloud Computing Security (CCSW '09). ACM, pp: 103–114.

Boneh, D. and M.K. Franklin, 2001. Identity-Based Encryption from the Weil Pairing,‖ in Proceedings of Advances in Cryptology – CRYPTO '01, ser. LNCS, 2139: 213–229.

Chase, M. and S.S.M. Chow, 2009. Improving Privacy and Security in Multi-Authority Attribute-Based Encryption,‖ in ACM Conference on Computer and Communications Security, pp: 121–130.

Chow, S.S.M., Y. Dodis, Y. Rouselakis and B. Waters, 2010. Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions,‖ in ACM Conference on Computer and Communications Security, pp: 152–161.

Chow, S.S.M., Y. Dodis, Y. Rouselakis and B. Waters, 2010. Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions,‖ in ACM Conference on Computer and Communications Security, pp: 152–161.

Chow, S.S.M., Y.J. He, L.C.K. Hui and S.M. Yiu, 2012. "SPICE - Simple Privacy-Preserving Identity-Management for Cloud Environment," in Applied Cryptography and Network Security – ACNS 2012, ser. LNCS, 7341: 526–543.

Goyal, V., O. Pandey, A. Sahai and B. Waters, 2006. Attribute-Based Encryption for Fine-Grained Access Control of Encrypted data,‖ in Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06). ACM, pp: 89–98.

Guo, F., Y. Mu and Z. Chen, 2007. Identity-Based Encryption: How to Decrypt Multiple Ciphertexts Using a Single Decryption Key,‖ in Proceedings of Pairing-Based Cryptography (Pairing '07), ser. LNCS, 4575: 392–406.

Guo, F., Y. Mu, Z. Chen and L. Xu, 2007. Multi-Identity Single-Key Decryption without Random Oracles,‖ in Proceedings of Information Security and Cryptology (Inscrypt '07), ser. LNCS, 4990: 384–398.

Okamoto, T. and K. Takashima, 2011. Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption,‖ in Cryptology and Network Security (CANS '11), pp: 138–159.

Sahai, A. and B. Waters, 2005. Fuzzy Identity-Based Encryption,‖ in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS, 3494: 457–473.

Wang, B., S.S.M. Chow, M. Li and H. Li, 2013. "Storing Shared Dataon the Cloud via Security-Mediator," in International Conference on Distributed Computing Systems - ICDCS 2013.

Wang, C., S.S. M. Chow, Q. Wang, K. Ren and W. Lou, 2013. "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans.Computers, 62(2): 362–375.