ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

A Secure Data Storage and Sharing Framework Using Specialized Key in Cloud

Pooja Shashank Pol, Prof. Amrit Priydarshi

PG Scholar, Department of Computer Engineering, Dattakala Faculty of Engineering, Pune, Maharashtra, India

Professor, Department of Computer Engineering, Dattakala Faculty of Engineering, Pune, Maharashtra, India

ABSTRACT: Privacy plays vital role in cloud computing and integrity. Key Aggregation, offer integrity and privacy to user without involving key information to be stored and used for every file. Paper follows aggregate key generation and management, The valid user can access their information using their private key and the global secret key information which is regulate during or after authentication process. Hash of keywords are calculated and stored on cloud with these keywords user can search for required document on cloud. System does not bother if secrete key either hack, the intruder cannot access the data while it can be decrypted only by using a private key and as we are passing hash keywords along with encrypted file both network path as well as cloud storage maintains its security. Here is no importance to send associate key and its file. All data will be encrypted by the global Secret Key. So data will be protected at a cloud place. Consumers who need file or information will access the data using their private key so file and its key not required.

KEYWORDS: Aggregate Key management, Cloud Computing, Data Sharing, Integrity and privacy, Data integrity.

I. INTRODUCTION

Cloud computing is being used extensively for data sharing therefore it is an essential aspect for security, efficient and flexible sharing of data with the other authorized users is very important in case of cloud data sharing. New public-key cryptosystems produce Coded texts which are of constant size so that decryption rights for sets of Coded texts can be efficiently Cloud privacy needs to deal with data integrity and privacy to ensure that data is not corrupted. Secret keys which are aggregated are very useful for ensuring data integrity and privacy. The user who is having secret key is allowed to release a constant-size aggregate key so that Coded text set can be flexibly chosen while ensuring that the other encrypted files out of the set stay confidential. The constant-size aggregate key(size of aggregate key does not depend on size of file) which is released by the user can be easily propagated to other consumers or it can be saved in a smart card. It can perform integrity and privacy analysis of the schemes which are in the standard model.

The data to be propagated is mostly crucial, which is avail-able only to a certain stage. For example, the data used in commercial intelligence, hospital system, bank transactions are highly crucial. These crucial data must be mitigated in a highly secured manner. To maintain confidentiality of user's crucial data, existing techniques employ cryptographic methods by exposing decryption keys only to the authorized data owners and consumers [10].

In this paper, a method to mitigate data in a highly secured manner is proposed, using an aggregate key instead of using the separate keys of each file. This reduces the time for transferring the keys and improves performance of sharing data.

II. RELATED WORK

The survey has been carried out on data sharing issues in a confidential manner; Privacy-Preserving Public Auditing is being analysed [1]. In their system a protected cloud storage method is proposed which supports privacy- preserving public auditing. Existing systems make use of a TPA (third party auditor) to satisfy auditing requirements for any number of consumers in a parallel and efficient manner. Consumers can access the cloud infrastructure as if it is in their own local area without bothering to check its integrity. Maintaining the Integrity of the Specifications Service providers

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

cannot attend auditing requests of all its users. Hence, SP relies on TPA, which performs batch- auditing to deal with the auditing requirements of the users. However, the third party auditors are susceptible to compromise in the integrity and privacy of the outer data. The system uses same linear authenticator along with random Tagging. This can guarantee that the TPA (third party auditor) is restricted from learning any knowledge about the outer data. Data integrity and privacy of systems with multiuser setting is threatened as the privacy-preserving public auditing protocol cannot be Carry to future extensive cloud storage as it less Productive. Trusted computing aims to address the problem of trustworthy online computing through the use of remote attestation [2]. Remote attestation leads to denial of service attacks because the challenger should be appraised by the target platform about its software and hardware configurations. Federated Identity Management Systems meant for integrity and privacy by delegating the authentication

functionality to a trusted third party called identity providers. Missed out on other aspects or area like platform integrity. The identity providers are entrusted with the duty of ensuring the integrity of the user platform. The remote attestation technique used here is not suitable for real world problems. More practical remote attestation method has to be employed. The cloud computing Flow is used widely as it provides software and hardware as services to the users. Therefore a variety of integrity and privacy and privacy concerns arise as the data is not within the area of the user. One of the integrity and privacy concerns arises due to the handling of data [5]. The consumers and companies that make use of the cloud services have preferences about the treatment of their data. The lawmakers impose requirements and obligations for specific types of data. Existing cloud services do not allow the consumers to set these requirements and hence the user or the company cannot be convinced about the integrity and privacy of their data. In the proposed system for integrity and privacy, distributed data storage service is Carry in such a way that it satisfies the data-handling requirements. The data-handling requirements include the location where the outer data of the user resides and how long the data is about to stay in that location. The user creates a data annotation which has the specifications about the location and duration of data. If it matches the data handling policies, the service provider signs the annotation. However, integrity and privacy and privacy concerns still exist. An efficient and inherently protected dynamic auditing protocol [6] uses cryptographic methods in ensuring data privacy. The technique involves the combination of cryptographic techniques and the orthogonal property of Orthogonal pairing instead of mask technique. Unlike the mask technique, this system does not involve additional trust organizer. This auditing scheme incurs less cost for both communication and computation. As Cloud computing allows data owners to store their data in cloud servers and allows access to the users, various integrity and privacy concerns arise. The data owners require an independent and reliable auditing service to ensure the integrity of their outer data and convince data owners about the integrity of their data. Integrity checking methods which are previously existing can check only static archive data. this paper is that computing a certain number of updates and challenges are limited and fixed beforehand. This cannot perform loop insertion anywhere. Hence this scheme causes heavy computation cost to the server. Cloud storage is a storage of data online in the cloud which is available from multiple and connected resources. Cloud storage can provide better accessibility and reliability, strong protected ion, disaster recovery, and lowest cost. Cloud storage having important functionality, i.e. securely, efficiently, flexibly sharing data with others. A novel public key encryption, which is called as Key-aggregate cryptosystem (KAC) is introduced in this work. Key-aggregate cryptosystem produce constant size Coded texts such that the efficient propagation of decryption rights for any set of Coded text are possible. Any set of secret keys can be aggregated and make them as single key, which power of all the keys being aggregated. This aggregate key can be sent to the others for decryption of Coded text set and remaining encrypted files outside the set are remains confidential.

III. PROPOSED SYSTEM

Protected data sharing in the cloud using the aggregate key aims in sharing the data without transferring keys for each file. The asymmetric encryption standard is used for encrypting all the data followed by public key encryption. The end user can access their data using their private key and the Global secret key which is transferred during the authentication process. Even though the Global secret key is hacked during transmission, malicious attacker cannot get the data since it can be decrypted only by using a private key. Keys need not be transferred for each file, data will be encrypted using a Global secret key. So the data will be safe at remote place. The consumers who need the data will access the data using their private key

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

3.1 Architecture

Data integrity and privacy is motive at protected sharing of data using a asymmetric encryption standard followed by public key cryptosystem. So within this method two keys are used for encrypting the data and the keys are Global secret key followed by the public key of the user. The intended user will get authenticated and use Global secret key followed by private key of the user to decrypt the data.

The backend database used here is MYSQL enterprise Data Store. Apache Data Store is an open source distributed database management system which can operate enormous amount of data stored across commodity servers .The redundant storage provides for an increased availability has no chance for a single point of failure. Innumerable commodity servers can be included in a Data Store cluster. Data Store caters to clusters present across numerous data centres using asynchronous master less replication, which decreases the latency of operations of all clients. The data model of Data Store involves a partitioned row store with tunable consistency. In Data Store each row has unique row key. Each key has a value which corresponds to a column. Then the columns are grouped to form column families which can be considered as a table. The Overall architecture diagram is shown. The system uses either the Amazon cloud or GlassFish server for storing their data. Amazon Elastic Compute Cloud (Any Elastic Cloud) maintains resizable computing capacity in the cloud. It is designed to limit the difficulties of the developers in using web-scale cloud computing. Any Elastic Cloud has an uncomplicated web service interface we are using Java web service for implementation which allows us to obtain and arrange capacity with less overhead. It provides us with Overall control of our computing resources and lets us run on Amazon's cloud computing environment. Obtaining and booting new server instances can be done in a lesser time using Any Elastic Cloud. Any Elastic Cloud protects the developers from various failures Situation. It helps them build applications which are resilient to failure. With changing computing requirements, Any Elastic Cloud helps to scale capacity rapidly. There are cost benefits as well as it is a pay as per usage model. A virtual private cloud can be set up with the desired IP range ensuring data security. Any Elastic Cloud can also provide dedicated instances for consumers who need dedicated hardware to run their instances thereby data security are preserved.

Figure 1. Proposed system Architecture

3.2 Implementation Details

Figure 1 shows working of proposed system as in proposed cloud environment user first needs to do registration after registration user can login to the system and can select the file. User need to select the class of file and send request for uploading. Once request is received from user server generates server key and send it to user. When user receives server key user inputs user key and generates new key using key of server, class and user key. System Applies keyword ranking and calculate hash of each keyword. System then encrypts file using generated key and send file along with hash of keywords to server. Encrypted data and keywords are also saved in DB. At the receiver end when registered receiver logins to the system he inputs/gives keywords for searching a file on cloud storage then system calculates hash

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

of those keywords and send to the server. After comparing hash of file keywords with query keywords and if match is found relevant file list is sent to the client. User then request for a file to download. Regeneration of key is performed at receiver end and data is then decrypted using this newly generated key and receiver can view original data

Algorithm 1 Creating user, attacker profile and detect intruder

1. Sender:

I. 1.SelFilex = Files[filea , fileb , filec .....]

II 2.InpUserKey = UKey[Ukey0]
III 3.GenAggKey =AggKey[AggKeyx]
IV 4.KeyWordRanking=Rank[Keyword0,Keyword1,Keyword2Keywordn

V 5.HashKeyword=Hash[hshKeyword0,hshKeyword1,hshKeywordn]

VI 6.EncryptFile using generated key = EncFile[AggKeyx]

VII 7.Transfer File and Has of Keywords

2. Cloud Server:

I 1.GenKey = [Key0,Key1,.Keyn]

II 2.EncDtKeywords = Insert encrypted data and keywords into DB

III 3.CmpHshFilewithHshQuery = cmp(hshFile,hshQuery)

IV. 4. If match found = send relevant data to client

V. 5.Decrypt Data using Generated Aggregate Key = Dec[AggKeyx]

VI. 6.View original Data

The Fig.1 shows the implementation overview of a cloud data storage system using aggregate key. The crucial data of sender is sent to the cloud storage by safe network path, searching of data is also easier in this system due to calculation of hash keywords and due to keyword matching implementation. The user gets authenticated by comparing digital signature while authentication.

IV. RESULT ANALYSES

4.2 Performance Evaluation

The proposed work is more extensible than existing hierarchical key assignment techniques which is limited to saving spaces if all key-Person mitigate a same set of privileges. This work uses meta crypto scheme which involves both ARSA and AAES algorithm. This system focuses on encrypting of data as well as calculation of hash of keywords due to which data remains secured on cloud storage server as well as through the network. Hence the inter communication involving more transfer of keys for data sharing is reduced. Encryption with aggregate key which is generated at sender as well at the receiver ends and hash of keywords make system more secure and enhances performance by allowing user to search relevant data more speedily.

Encryption is made such a way that decryption cannot be done using public key [1]. Cryptanalysis for protected data sharing is made using Cloud Framework and Data Store. This approach will provide scalable data sharing system by generating keys in linear order of time. Processing time for transferring file becomes very low. With the fact that there is no need to transfer key used for encryption, which in turn reduces the processing time. Even though it takes

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

more time compared to ARSA algorithm to transfer files, the stage of security is higher since it does encryption by adding class of files and calculates hash of keywords as well. Therefore if some data want to compromise it needs to compromise at three stages while calculating aggregate key, while adding a class of file and while calculating hash of keywords for a particular file. If all these three phases are compromised then only data or file stored on cloud storage can be hacked.

The Initial results obtained for the given security model is analyzed considering Various Situation. Various encryption techniques are compared with their processing time for various file sizes. The graph Fig.5 shows the overall evaluation of the encryption techniques. It clearly shows that the proposed algorithm works more efficiently than the other encryption standards being compared.

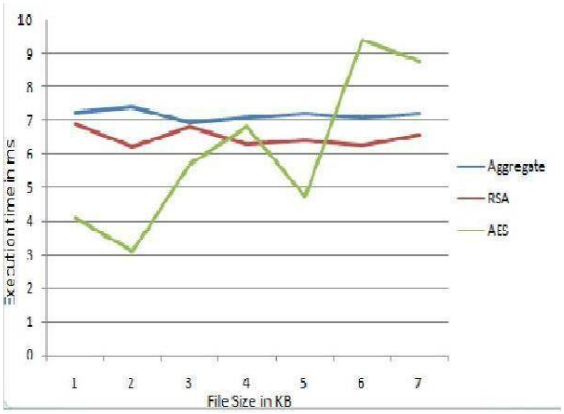

Figure 2. Evaluation of AES, RSA and Improved KAC encryption standards

V. CONCLUSION

Figure 2 shows evaluation of proposed system against AES, RSA and Improved KAC encryption standards which proves that, proposed system secure data storage and sharing framework using an aggregate key is very effective for storing crucial data on cloud. The Data can be securely propagated in cloud storage using this aggregate key technique. Using a single specialized Aggregate key is an important feature of the proposed system. This reduces the usage of multiple keys sharing between the senders and receivers and hence ensures security of the data being mitigated. Despite being encrypted, the data to be mitigated will be safe in the remote place. In this system we are calculating hash of keywords for file being stored in cloud which results in faster searching of file/data on cloud and also ensure secured network transmission path as well data storage.

ACKNOWLEDGMENT

We would like to express our gratitude towards Dattakala Facuty of Engineering for providing encouraging environment which help us for carrying this research work.

REFERENCES

[1] S. Yu, C. Wang, K. Ren, and W. Lou, “Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing”, Proc. IEEE INFOCOM, pp. 534-542, 2010.
[2] R. Lu, X. Lin, X. Liang, and X. Shen, “Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing”, Proc. ACM Symp. Information, Computer and Comm.Security, pp. 282-292, 2010.
[3] X. Liu, Y. Zhang, B. Wang, and J. Yan. “Mona: secure multiowner data sharing for dynamic groups in the cloud”, IEEE Transactions on Parallel and Distributed Systems, 2013, 24(6): 1182- 1191.
[4] C. Chu, S. Chow,W. Tzeng, et al. “Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage”, IEEE Transactions on Parallel and Distributed Systems, 2014, 25(2): 468-477.
[5] X. Song, D.Wagner, A. Perrig. “Practical techniques for searches on encrypted data”, IEEE Symposium on Security and Privacy, IEEE Press, pp. 44C55, 2000.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 7, July 2017

[6] R. Curtmola, J. Garay, S. Kamara, R. Ostrovsky. "Searchable symmetric encryption: improved definitions and efficient constructions", In: Proceedings of the 13th ACM conference on Computer and Communications Security, ACM Press, pp. 79-88, 2006.
[7] S. Kamara, C. Papamanthou, T. Roeder. "Dynamic searchable symmetric encryption", Proceedings of the 2012 ACM conference on Computer and communications security (CCS), ACM, pp. 965-976, 2012.
[8] C. Dong, G. Russello, N. Dulay. "Shared and searchable encrypted data for untrusted servers", Journal of Computer Security,pp. 367-397, 2011.
[9] X.F. Chen, J. Li, X.Y. Huang, J.W. Li, Y. Xiang. "Secure Outsourced Attribute-based Signatures",IEEE Trans. on Parallel and Distributed Systems, DOI.ieeecomputersociety.org/10.1109/TPDS.2013.180, 2013.