International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 04 | July-2015 www.irjet.net p-ISSN: 2395-0072

SECURE DATA SHARING USING AGGREGATE KEY FOR SENSITIVE DATA

M.R.Sumalatha[1], M.B.Rizvana Begam[2], E.Divya Priya[3], J.Bejin Joe[4]

[1] *Associate Professor,Information Technology Department, Anna University,Tamil Nadu,India*
[2,3,4] *Students, Information Technology Department, Anna University,Tamil Nadu,India*

Abstract - ***Security is essential for sharing sensitive data in the cloud. Using the aggregate key, makes the system share the sensitive data without transferring keys for each and every file. This system uses asymmetric encryption standard for encrypting all the data followed by public key encryption. The end user can access their data using their private key and the master secret key which is transferred during or after authentication process. Even though the Master secret key is hacked during transmission, the malicious attacker cannot get the data since it can be decrypted only by using a private key. There is no need to transfer key for each and every file. All data will be encrypted by the Master Secret Key. So data will be safe at a remote place. Users who need sensitive data will access the data using their private key so there is no need to transfer key for each and every file.***

Key Words: *Aggregate Key, Cassandra, Data Sharing, Security, Sensitive data.*

1. INTRODUCTION

Data security involves digital privacy measures that protect sensitive digital data from corruption. These measures are applied to protect computers and databases from unauthorized access. Organizations irrespective of the size or genre mainly prioritize Data Security. Data security has different nomenclatures like information security or computer security. User's outsource confidential data in cloud servers and hence it requires protection from unauthorized access by malicious attackers. The impact of data confidentiality is not only restricted to security and privacy issues but also to juristic concerns.

In cloud computing, Data sharing is an essential aspect for secure, efficient and flexible sharing of data with the other authorized users. New public-key cryptosystems produce cipher texts which are of constant size so that decryption rights for sets of cipher texts can be efficiently delegated [3]. The idea proposed is that the user can gather any set of secret keys and compact them into a single key. This single key comprises the power of all the secret keys which are aggregated.The user who possesses the secret key is allowed to release a constant-size aggregate key so that cipher text set can be flexibly chosen while ensuring that the other encrypted files out of the set stay confidential. The constant-size aggregate key which is released by the user can be easily directed to other users or it can be saved in a smart card. It can perform security analysis of the schemes which are in the standard model.

The data to be shared is mostly sensitive, which is accessible only to a certain level. For example, the data used in business intelligence, health system, bank transactions are highly sensitive. These sensitive data must be shared in a highly secured manner. To maintain confidentiality of user's sensitive data, existing techniques employ cryptographic methods by exposing decryption keys only to the authorized data owners and users [10].

In this paper, a method to share 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.

2. RELATED WORK

The survey has been carried out on data sharing issues in a confidential manner, Privacy-Preserving Public Auditing is being analyzed [1]. In their system a secure 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 users in a parallel and efficient manner. Users can access the cloud infrastructure as if it is in their own local domain without bothering to check its integrity. Service providers 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 security of the outsourced data. The system uses homomorphic linear authenticator along with random masking. This can guarantee that the TPA (third party auditor) is restricted from learning any knowledge about the outsourced data. Data security of systems with multiuser setting is threatened as the privacy-preserving public auditing protocol cannot be extended to future extensive cloud storage as it lacks efficiency.

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. FIDMS (Federated Identity Management Systems) vouches for security and privacy by delegating the authentication functionality to a trusted third party called identity providers. FIDMS misses out on other aspects like platform integrity. These two concepts are coupled to form extended FDMS which has an added functionality of addressing the integrity concerns. 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 paradigm is used widely as it provides software and hardware as services to the users. Therefore a variety of security and privacy concerns arise as the data is not within the domain of the user. One of the security concerns arises due to the handling of data [5]. The users 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 users to set these requirements and hence the user or the company cannot be convinced about the security of their data. In the proposed system for security, distributed data storage service cassandra is extended in such a way that it satisfies the data-handling requirements. The data-handling requirements include the location where the outsourced 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, security and privacy concerns still exist.

An efficient and inherently secure dynamic auditing protocol [6] uses cryptographic methods in ensuring data privacy. The technique involves the combination of cryptographic techniques and the bilinearity property of bilinear 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, different security concerns arise. The data owners require an independent and reliable auditing service to ensure the integrity of their outsourced data and convince data owners about the integrity of their data. Integrity checking methods which are already existing can check only static archive data. This cannot be

extended to auditing services because cloud can be updated dynamically. The proposed system carries out both dynamic auditing and batch auditing.. Limitation of this paper is that computing a certain number of updates and challenges are limited and fixed beforehand. This cannot perform block 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 accessible from multiple and connected resources. Cloud storage can provide better accessibility and reliability, strong protection, 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 cipher texts such that the efficient delegation of decryption rights for any set of cipher text are possible. Any set of secret keys can be aggregated and make them as single key, which encompasses power of all the keys being aggregated. This aggregate key can be sent to the others for decryption of cipher text set and remaining encrypted files outside the set are remains confidential.

NoSQL databases try to offer certain functionality that more traditional relational database management systems do not. Whether it is for holding simple key-value pairs for shorter lengths of time for caching purposes, or keeping unstructured collections (e.g. collections) of data that could not be easily dealt with using relational databases and the *structured query language* (SQL) – NoSql databases will help. By design, NoSQL databases and management systems are relation-less (or schema-less). They are not based on a single model (e.g. R*elational model* of RDBMSs) and each database, depending on their target-functionality, adopts a different one.

3. IMPROVED KEY AGREGATE CRYPTOSYSTEM

A. Proposed Work

Secure data sharing in the cloud using the aggregate key for drug addict victim proposed work aims in sharing the data without transferring keys for each and every 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 master secret key which is transferred during the authentication

process. Even though the Master 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 and every file, data will be encrypted using a master secret key. So the data will be safe at remote place. The users who need the data will access the data using their private key.

B. Architecture

Data security is aimed at secure sharing of data using a asymmetric encryption standard followed by public key

cryptosystem.. So withn this method two keys are used for encrypting the data and the keys are master secret key followed by the public key of the user. The intended user will get authenticated and use master secret key followed by private key of the user to decrypt the data.

The backend database used here is Datastax enterprise Cassandra. Apache Cassandra 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 Cassandra cluster. Cassandra caters to clusters present across numerous data centers using asynchronous masterless replication, which decreases the latency of operations of all clients. The data model of cassandra involves a partitioned row store with tunable consistency. In cassandra each row has a 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 complete architecture diagram is shown. The system uses the Amazon cloud for storing their data. Amazon Elastic Compute Cloud (Amazon EC2) maintains resizable computing capacity in the cloud. It is designed to limit the difficulties of the developers in using web-scale cloud computing. Amazon EC2 has an uncomplicated web service interface which allows us to obtain and configure capacity with minimum overhead. It provides us with complete 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 Amazon EC2. Amazon EC2 protects the developers from various failure scenarios. It helps them build applications which are resilient to failure. With changing computing requirements, Amazon EC2 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. Amazon EC2 can also provide dedicated instances for users who need dedicated hardware to run their instances thereby data security is preserved.

4. IMPLEMENTATION DETAILS

CloudSim, a simulation toolkit model and simulates the cloud infrastructure such as data center,virtual machines and application provisioning environments. CloudSim provides custom interfaces, which can be used for the implementation of different provisioning techniques for allocating virtual machines which is used in drug addict victims database.

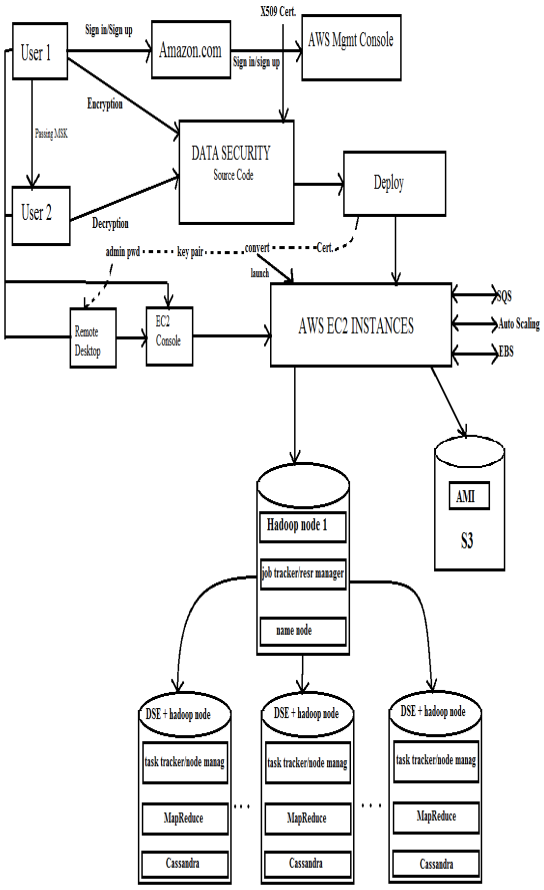

Fig -1. Improved KAC Architecture

A. Improved KAC algorithm

1.Team :

1.enc0x = Ekmsk[filea , filed , filex .....].

2.enc1x= Eku[enc0x].

3.Do authenticate.

4.Transfer file.

2.User:

1.Get authenticated.

2.Receive file.

3.Get MSK.

3.dec0x = Dkr[enc1x].

4.dec1x = Dkmsk[dec0x].

Where,

enc0x -Encrypted file using master secret key.

Ekmsk - Encryption using master secret key.

enc1x - Encrypted file using public key of the intended user.

Eku - Encryption using public key of the user.

Dkr - Decryption using private key of the user.

Dkmsk - Decryption using master secret key.

dec1x - Decrypted file using master secret key.

The Fig.2 shows the implementation scenario of a drug addict victim data and the communications between the Team and the user. The sensitive data of the drug addict victims are stored by the Team after encryption using Improved KAC algorithm in the cloud storage. The user gets authenticated, acquires the Master Secret Key from the Team and decrypt the data using both the private and Master secret key.

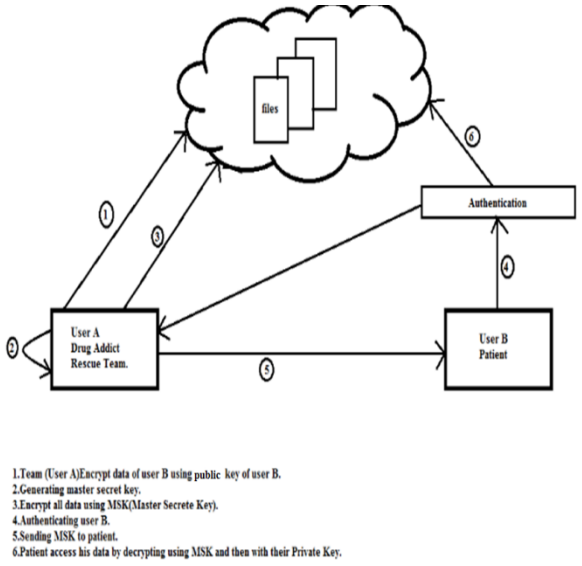

Fig-2. Drug addict victim Data Transfer Scenario

5. PERFORMANCE ANALYSIS

The proposed work is more extensible than existing hierarchical key assignment techniques which is limited to saving spaces if all key-holders share a similar set of privileges. This work uses hybrid crypto scheme which involves both RSA and AES algorithm. This system focuses on encrypting the data two times, initially by using Master secret key and then by using public key of the user. Hence the inter communication involving more transfer of keys for data sharing is reduced. Encryption with public key of any user which is known to all and making only the intended person to read it by using their private key which is known to them alone makes the system more secure. Encryption is made such a way that decryption cannot be done using public key [1].

Cryptanalysis for secure data sharing is made using Cloudsim and Cassandra. This approach will provide scalable data sharing system by generating keys in linear order of time. Execution time for transferring file becomes comparatively effective. There is no need to transfer key used for encryption, which in turn reduces the execution time. Even though it takes more time compared to RSA algorithm to transfer files, the level of security is higher since it does encryption two times.

The Fig.3 shows that execution time vs file size for the aggregate key algorithm.This algorithm maintains their execution time for different sizes of file within a range.

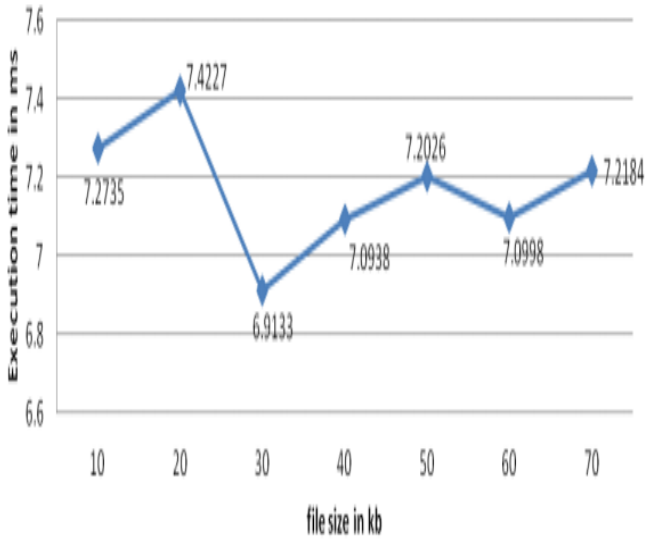

Fig-3. Execution time vs file size for the aggregate key algorithm

The Fig.4 shows that execution time vs file size for RSA algorithm.The RSA algorithm consistently maintains the execution time for different sizes of file within a specified range of values. There is no major deviation in the execution time with respect to file size.

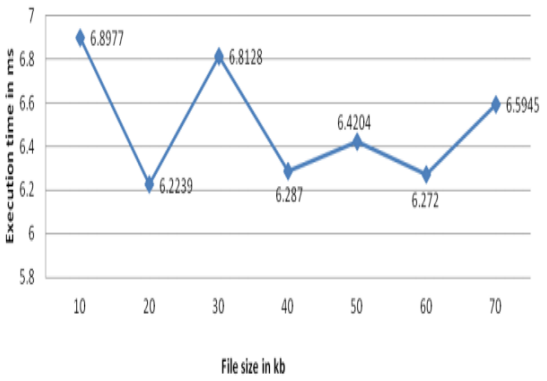

Fig-4. Execution time vs file size for RSA algorithm

The Fig.5 shows that execution time vs file size for AES algorithm. The AES algorithm has increased execution time for increasing file sizes. The graph is plotted for Execution Time against File Size(KB).

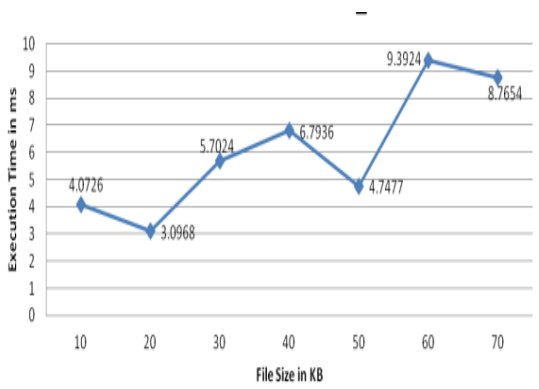

Fig-5. Execution time vs file size for AES algorithm

6.RESULT ANALYSIS

The preliminary results obtained for the proposed security model is analyzed considering different scenarios. Different encryption techniques are compared with their execution time for various file sizes. The graph Fig.6 shows the overall comparison of the encryption techniques. It clearly shows that the proposed algorithm works more efficiently than the other encryption standards being compared.

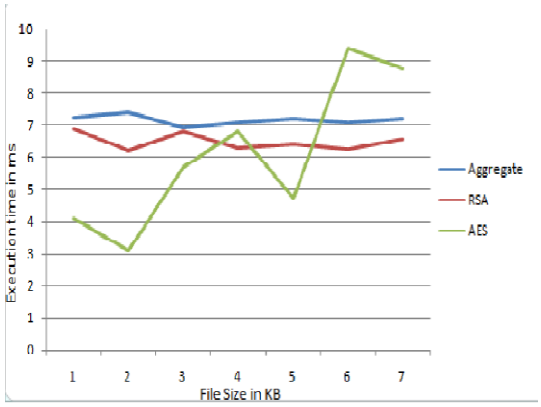

Fig-6. Comparison of AES,RSA, Improved KAC encryption standards

7.CONCLUSION AND FUTURE WORK

In this work secured data sharing using an aggregate key is proposed for handling sensitive data considering drug addict victims data. The Data can be securely shared in Cloud storage using this Aggregate key techniques. Asymmetric Encryption standards are more secure than the symmetric encryption standards which uses a single key on both sides for transmission of data. Using a single Master Secret key is an important feature of the proposed algorithm. This reduces the usage of multiple keys sharing between the users and hence ensures security of the data being shared. Despite being encrypted, the data to be shared will be safe in the remote place, especially in the case of drug addict victim's data. Cassandra, a distributed data storage system is used for storing drug addict victims data with security enhancements for handling Data sharing.

The algorithm can be extended for n number of data on any application specific information. Further work can be done on the current statistics of any real-time data which is highly sensitive and is prone to attack. Highly sensitive data can be encrypted using a highly secured Master secret key. For example, Applications like Health Records, Forensic Data, Criminal Database, etc., requires secured sharing of sensitive data. The Improved KAC algorithm can be extended to these applications to share the sensitive data in a highly secured manner.

REFERENCES

[1] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren and Wenjing Lou,"Privacy-Preserving Public Auditing for Secure Cloud Storage", 2013 IEEE Transactions on Computers.

[2] Tamleek Ali, Mohammad Nauman, Muhammad Amin and Masood Alam,"Scalable, Privacy-preserving Remote Attestation in and through Federated Identity Management Frameworks",2010 IEEE.

[3] Xuyun Zhang, Laurence T. Yang, Chang Liu and Jinjun Chen,"A Scalable Two-Phase Top-Down Specialisation Approach for Data Anonymization Using MapReduce on Cloud",2014 IEEE Transactions on Parallel and Distributed Systems.

[4] Maria Chalkiadaki and Kostas Magoutis,"Managing Service Performance in Cassandra Distributed Storage System",2013 IEEE International Conference on Cloud Computing Technology and Science.

[5] Martin Henze, Marcel Grobfengels, Maik Kaprowski and Klaus Wehrle,"Towards Data Handling Requirements-Aware Cloud Computing", 2013 IEEE International Conference on Cloud Computing Technology and Science.

[6] Kan Yang and Xiaohua Jia,"An Efficient Dynamic Auditing Protocol for Data Storage in Cloud Computing", 2012 IEEE Transactions on Parallel and Distributed Systems.

[7] Huiki Xu, Shumin Guo and Keke Chen,"Building Confidential and Efficient Query Services in The Cloud using Data Perturbation", 2014 IEEE Transactions on Knowledge and Data Engineering, VOL.26,NO.2.

[8] Mahdi Tayarani Najaran and Norman C. Hutchinson,"Innesto: A Searchable Key/Value Store for Highly Dimensional Data", 2013 IEEE International Conference on Cloud Computing Technology and Science.

[9] Satoshi Fukuda, Ryota Kwashima, Shoichi Saito and Hiroshi Matsuo,"Improving Respose Time for Cassandra with Query Scheduling",2013 First International Symposium on Computing and Networking.

[10] Cheng-Kang Chu, Sherman S.M. Chow, Wen-Guey Tzeng, Jianying Zhou and Robert H.Deng,"Key-Aggregate Cryptosystems for Data Sharing in Cloud Sharing", IEEE Transactions.

