IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 18, Issue 1, Ver. III (Jan – Feb. 2016), PP 36-40
www.iosrjournals.org

Input Cumulative Cryptosystem for Scalable Information Contribution in Cloud

Kamma Madhavi[1], Venkatasivasankara Reddy[2]
[1]*Student of M.Tech (CSE)*
[2]*Asst. Prof, Department of Computer Science and Engineering, QIS Institute of technology, Ongole. A.P, INDIA*

Abstract: *In Cloud computing, data storing and sharing is a capable methodology. This study illuminates secure, powerful, and versatile system to give data to different people in Cloud storage structure. This review, depict novel open key cryptosystems. This structure create relentless size figure messages such that compelling assignment of unraveling rights for any course of action of figure works are possible. This improvement arrangement can add up to any course of action of riddle keys and make them as a decreased single key .The power of the significant number of keys being gathered in a lone key. Toward the day's end, holder of the key can release a predictable size aggregate key for versatile choices of figure substance set in Cloud storing .In this arrangement other encoded records outside the figure substance set stay mystery. The aggregate key can be suitably sent to others or be secured in a smart card with incredibly obliged secure storage. Distributed computing advancement is for the most part used so that the data can be outsourced on cloud can got to easily. Particular people can share that data through various virtual machines yet appear on single physical machine.*

Keywords: *Cloud Computing, Key-aggregate encryption, Attribute based Encryption, Aggregate keys,*

I. Introduction

Cloud computing is creating as a gigantic stage for capacity, Maintaining and sharing data. Enthusiasm for data upkeep and limit is extending in all fields, whether customers are from corporate, military, IT affiliations et cetera. Data assurance has transformed into a basic sensitivity toward cloud customers. Customers don't trust fogs similarly as mystery. Fogs are unequivocally used for sharing data. Cloud hosts can grant subset of their information to their sidekicks and accomplices. While sharing data, security is an essential concern. For the most part we trust untouchable server for giving security. Requesting is sent to the server for check, hosts getting to fogs are constrained to trust the outcast for their security. Nevertheless, there are potential outcomes of tricking, hacking and intrusion strikes [1]. Expect that in a mending office organization structure masters and patients are getting to the cloud for sharing information about ailment and meds. Authority exchanges information about his patient on the cloud, yet he is not content with the security principles of cloud. Along these lines, pro mixed all his data and after that exchange records on the cloud. Taking after two days one of the patients requested the information material to him. As authority has starting now mixed data, the unscrambling key will be allocated to the patient. If standard approach is considered for selecting interpreting key, three conditions are developing: 1. all archives are mixed with similar encryption key. Here, Doctor needs to send one unraveling key which will uncover riddle of all the data. 2. All records are encoded with specific or distinctive key. For this circumstance specific interpreting keys will be sent, which is basically inefficient, as data proprietor needs to send a no. Of translating keys. 3. While assigning the secret keys there are shots of intruder's attack. Some untouchable may endeavor to get crucial information. To overcome above impediments while sharing the data, an answer is proposed in this paper. The proposed course of action is to "Scramble all data with divergent encryption key and send simply single unscrambling key. This single disentangling key should have the ability to unscramble various figure content. The promising segment of unscrambling key is that, it is aggregate of the entire interpreting key yet it stays littler in size as a single key [1]. The hosts incorporated into correspondence should have the ability to screen the security bursts, hence an intrusion acknowledgment structure should be given". The unscrambling key is allocated securely on a secured channel. Minimal size of unscrambling key is pined for, as we can use it for cutting edge cellular telephones, remote sensors, and splendid cards etc. This paper finds its application for facility organization, military organizations et cetera.

II. Related Work

An encryption course of action which is at initially proposed for rapidly transmitting colossal number of keys in telecast situation. The progression is essential and we quickly ponder its key reasoning handle here for a bond depiction of what are the beguiling properties we need to accomplish. The thinking of the key for a game-plan of classes (which is a subset of all conceivable ciphertext classes) is as takes after. A composite modulus is picked where p and q are two unfathomable sporadic primes. A pro question key is picked at optional. Every class is connected with a particular prime. All these prime numbers can be set in the broad group

framework parameter. An anticipated size key for set can be conveyed. For the general population who have been distributed the section rights for S' can be made. Notwithstanding, it is made game plans for the symmetric-key setting. The substance supplier needs to get the relating riddle keys to encode information, which is not suitable for a couple of uses. Since strategy is utilized to convey riddle respect rather than a few open/mystery keys, it is questionable how to apply this thought for open key encryption course of action. At long last, we watch that there are game plans which attempt to lessen the key size for accomplishing certification in symmetric-key encryption, On the other hand, offering of deciphering force is not an anxiety in these course of action. Identity based encryption (IBE) is an open key encryption in which the general population key of a client can be set as an personality string of the client (e.g., an email address, portable number). There is a private key generator (PKG) in IBE which holds a expert mystery key and issues a mystery key to every client with deference to the client personality. The content supplier can take general society parameter and a client personality to scramble a message. The beneficiary can decode this ciphertext by his mystery key. To manufacture IBE with key conglomeration. In their plans, key accumulation is obliged as in all keys to be accumulated must originated from diverse ―identity divisions. While there are an exponential number of characters and subsequently mystery keys, just a polynomial number of them can be aggregated This altogether builds the expenses of putting away and transmitting cipher texts, which is illogical by and large, for example, imparted distributed storage. As another approach to do this is to apply hash capacity to the string indicating the class, and continue hashing over and again until a prime is acquired as the yield of the hash function we specified, our plans highlight steady ciphertext size, and their security holds in the standard model. In fluffy IBE one single minimized mystery key can decode ciphertexts encoded under numerous personalities which are shut in a certain metric space, however not for a discretionary arrangement of characters furthermore, subsequently it doesn't coordinate with our concept of key to key.

III. Data Sharing:

KAC in meant for the data sharing. The data owner can share the data in desired amount with confidentiality. KCA is easy and secure way to transfer the delegation authority. For sharing selected data on the server Alice first performs the Setup. Later the public/master key pair (pk, mk) is generated by executing the KeyGen. The msk master key is kept secret and the public key pk and param are made public.Anyone can encrypt the data m and this data is uploaded on server. With the decrypting authority the other users can access those data. If Alice is wants to share a set S of her data with a friend Bob then she can perform the aggregate key KS for Bob by executing Extract (mk, S). As kS is a constant size key and the key can be shared through secure e-mail. When the aggregate key has got Bob can download the data and access it.

IV. Security Of Cloud Data Storage

Many cloud service providers provide storage as a form of service. They take the data from the users and store them on large data centers, hence providing users a means of storage. Although these cloud service providers say that the data stored in the cloud is utmost safe but there have been cases when the data stored in these clouds have been modified or lost may be due to some security breach or some human error. Various cloud service providers adopt different technologies to safeguard the data stored in their cloud. But the question is: Whether the data stored in these clouds is secure enough against any sort of security breach? The virtualized nature of cloud storage makes the traditional mechanisms unsuitable for handling the security issues. These service providers use different encryption techniques like public key encryption and private key encryption to secure the data resting in the cloud. Another major issue that is mostly neglected is of Data-Remanence. It refers to the data left out in case of data removal. It causes minimal security threats in private cloud computing offerings, however severe security issues may emerge out in case of public cloud offerings as a result of dataremanence. Various cases of cloud security breach came into light in the last few years. Cloud based email marketing services company, Epsilon suffered the data breach, due to which a large section of its customers including JP Morgan Chase, Citibank, Barclays Bank, hotel chains such as Marriott and Hilton, and big retailers such as Best Buy and Walgreens were affected heavily and huge chunk of customer data was exposed to the hackers which includes customer email ids and bank account details. Another similar incident happened with Amazon causing the disruption of its EC2 service. The damage caused had proved to be quite costly for both the users and the system administrators. The above mentioned events depict the vulnerability of the cloud services. Another important aspect is that the known and popular domains have been used to launch malicious software or hack into the companies' secured database. It is proved that Amazon is prone to side-channel attacks, and a malicious virtual machine, occupying the same server as the target, can easily gain access to confidential data [10]. The question is: whether any such security policy should be in place for these trusted users as well? An incident relating to the data loss occurred last year with the online storage service provider "Media max" also known as "The Linkup" when due to system administration error, active customer data was deleted, leading to the data loss. SLA's with the Cloud Service providers should contain all the points that may cause data loss

either due to some human or system generated error. Virtualization in general increases the security of a cloud environment. With virtualization, a single machine can be divided into many virtual machines, thus providing better data isolation and safety against denial of service attacks [10]. The VMs provide a security test-bed for execution of untested code from un-trusted users.

V. Data Privacy In Cloud Computing Environment

Considering data privacy in cloud computing environment, a traditional way to ensure data privacy is to rely on the server to enforce the access control after authentication, which means any unexpected privilege increase will expose all data. In a shared-lease cloud computing environment, things become even bad. Data from different users can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VirtualMachine could be stolen by instantiating another Virtual Machine cooccupant with the target one

VI. Problem Statement

- Constant-size decryption key require pre-defined hierarchical relationship.
- The fixed hierarchy is used. In that there is only one way in which we can partition the record. If we want to give out access rights based on something else (e.g. based on document type or sensitivity of data) we will have to look at all the low level categories involved, and give a separate decryption key for each [2].
- More number of decryption key was used [1]

VII. System Architecture

A key-aggregate encryption scheme consists of five polynomial-time algorithms [1] as shown in Figure. The data owner establishes the public system parameter via Setup and generates a public/master-secret3 key pair via KeyGen. Messages can be encrypted via Encrypt by anyone who also decides what cipher text class is associated with the plaintext message to be encrypted. The data owner can use the master-secret to generate an aggregate decryption key for a set of cipher text classes via Extract. The generated keys can be passed to receivers securely (via secure e-mails or secure devices). Finally, any user with an aggregate key can decrypt any cipher text provided that the cipher text's class is contained in the aggregate key via Decrypt4.

A. Setup (1;n):

Executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1 and the number of cipher text classes n (i.e., class index should be an integer bounded by 1 and n), it outputs the public system parameter param, which is omitted from the input of the other algorithms for brevity.

B. KeyGen():

Executed by the data owner and randomly generates a master-secret key (msk).

C. Encrypt(pk; i;m):

Executed by data owner to encrypt data. On input msk, an index i denoting the cipher text class, and a message m, it outputs a cipher text C..

D. Extract(msk; S):

Executed by the data owner for delegating the decrypting power for a certain set of cipher text classes to the receiver. On input the master secret key msk and a set S of indices corresponding to different classes, it outputs the aggregate key for set S denoted by KS.

E. Decrypt(KS; S; i; C):

Executed by a receiver who received an aggregate key KS generated by Extract. On input KS, the set S, an index i denoting the cipher text class the cipher text C belongs to, and C, it outputs the decrypted result m if i in S.

VIII. Proposed Work

In this paper we propose a technique to make data sharing secure and leak resilient. The purpose of this article is to provide a way for secure data sharing on cloud using key aggregate encryption and Intrusion Detection (KAEID). In KAEID Decryption key is made more and more powerful so that it can decrypt multiple cipher texts. At the same time Intrusion detection system (IDS) monitors data exchange between two hosts and ensures if these are trusted hosts [2]. Specifically, the problem statement is "To generate a constant size aggregate decryption key by data owner which can decrypt multiple cipher text. The decryption key is aggregate key which encompasses the power of all secret keys. This data sharing system also supports intrusion detection to find out the suspicious activities of hosts. If hosts involved in communication are trusted hosts data sharing will take place else rejected." In KAEID user encrypts message under public key cryptosystem. Messages are encrypted by one who decides public key as well as cipher text category. Cipher text is categorized under different "classes". Plain messages which are subset of cipher text class possess few common features. Here all the hosts set up an account on the cloud server. Hosts can login to the cloud server; they can perform their task

and logout of the server. The data owner generates public key/ master key pair. Public key is used for encryption while master key is kept secret. Master key is used for aggregating all the decryption keys. The aggregate key is extracted out of master key and corresponding cipher text class identifier. This aggregate key is delegated to data recipient. The data recipient compares the set of cipher text classes and decrypts the message. Hence, it also prevents the downloading of unwanted data. Each host in the data sharing system works as IDS. An IDS collects IP address of all hosts in its sub network, and keep eyes on suspicious activities in the network. If any suspicious host is found it is blacklisted. Data sharing with suspicious host is rejected. As shown in Fig-1. Two hosts data owner and data recipient are accessing the cloud network. Data owner encrypts the data and uploads data on cloud server. Aggregate key is delegated to Data recipient for decryption of requested messages. Hosts involved in communication are also working as IDS. IDS collects and lists IP addresses of corresponding sub network. Monitors the suspicious activities and reject data sharing with the hosts found blacklisted.

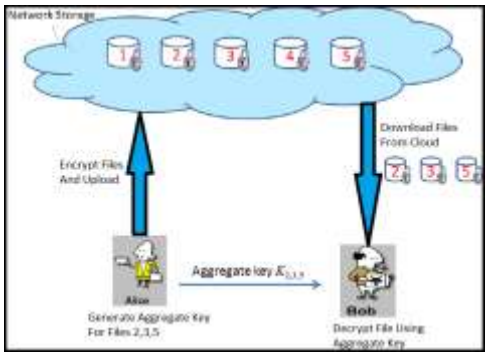

Fig. Proposed Architecture diagram

IX. Results And Discussion

In this experiment it is assumed that we have n number of cipher text classes denoted as CI. S is a set of cipher text class identifier CI, represented as S ={CI| CI=1,2,3...n}. The encryption phase is independent of the account setup. Encryption time does not depends upon the no of message to be encrypted. Encryption is done in constant time. r is the portion of cipher text classes to which data recipient is concern. r represents the ratio of delegated cipher text classes to the total no. Of cipher text classes. Decryption is done in group; decryption key matches keys for cipher text classes, with pairing operations where S is the set of cipher text classes. Each host in communication collects the IP addresses of neighbours in Δt time interval. IDS blacklist the suspicious IP addresses. Blacklist's size increases for fixed no. peers. As no. of peers increases detection delay increases. Here routing time is not much significant, it is less than the time taken to handle increased load.

X. Conclusion

As we all know data security is a major concern for cloud users. This paper comes with a technique, which helps to achieve a secured and leak proof system. Here modern cryptographic algorithms and intrusion detection algorithms are used in order to achieve a secured way of data sharing. In this system data owner uses distinct encryption keys and encrypts messages before uploading it on cloud and sends a single decryption key to other host. This single decryption key decrypts multiple cipher text at a time thereby saving the time as well as storage space. Unwanted data will not be downloaded at data recipient's side. Intrusion detection systems monitor the security breakdown in the network. Data sharing is stopped if any un-trusted party comes in the network. Obtaining an ideal system without data any leakage is practically is not possible, but this research work helps to solve certain problems very efficiently. It saves the storage space; it also saves time spent in key exchange. Key sizes remains constant and compact.

References

[1] "Key-Aggregate Cryptosystem for Scalable Data sharing in Cloud Storage" Cheng-Kang Chu, Sherman S.M Chow, Wen-Guey Tzen, Jianying Zhou, Robert H. Deng IEEE, 2014

[2] "A Peer-to-Peer Collaborative Intrusion Detection System" Chenfeng Vincent Zhou, Shanika Karunasekera and Christopher Leckie National ICT Australia Department of Computer Science and Software Engineering.

[3] University of Melbourne, Australia 2005 [3] Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM 04). IEEE, 2004, pp. 20672071.

[4] M. J. Atallah, M. Blanton, N. Fazio, and K. B. Frikken, "Dynamic and Efficient Key Management for Access Hierarchies," ACM Transac ormation and System Security (TISSEC), vol. 12, no. 3, 2009tions on Inf.

[5] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Proceedings of Advances in Cryptology - EUROCRYPT 05, ser. LNCS, vol. 3494. Springer, 2005, pp. 457473.

[6] S. S. M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," in ACM Conference on Computer and Communications Security, 2010, pp. 152161.

[7] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "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, 2006, pp. 8998.
[8]" Multi-Authority Attribute-Based Encryption," in ACM Conference on Computer and Communications Security, 2009, pp. 121130
[9] CERT Coordination Center, "Module 2 - Internet Security Overview", 2003. [9] M. E. Locasto, J. J. Parekh, A. D. Keromytis, S. J. Stolfo, "Towards Collaborative Security and P2P Intrusion Detection", 2005 IEEE Workshop on IAS, June 2005.
[10] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing", Technical Report CSD-01-1141, University of California, Berkeley, 2000.

AUTHORS PROFILE

Author 1

KAMMA MADHAVIPursuing M.Tech (Computer Science and Engineering) in QIS Institute of Technology, PrakasamDist, Andhra Pradesh, India.

Author 2

VENKATASIVASANKARA REDDY currently working as Asst. Professor in QIS Institute of technology, in the Department of Computer Science and Engineering, Ongole, PrakasamDist, Andhra Pradesh, India.