MJRET
Open Access
Multidisciplinary Journal of Research in Engineering and Technology
Journal homepage: www.mjret.in
ISSN:2348 – 6953

KEY AGGREGATE CRYPTOSYSTEM & INTRUSION DETECTION FOR DATA SHARING IN CLOUD

Garima Kumari, Professor. Lakshmi madhuri
Computer Engineering Dept.
Dr. D.Y.Patil School Of Engineering (affiliated to Savitribai Phule Pune University)
Pune,India
garikrpm@gmail.com, lakshmimadhuri1983@gmail.com

Abstract : *A cryptosystem is a system that provides encryption and decryption schemes for sharing data. An intrusion detection system monitors the security breaches while sharing of data. Cryptosystem integrated with Intrusion detection system makes a leakage resilient system. New public key cryptosystems which produces constant size cipher texts such that decryption rights for any set of cipher text can be delegated efficiently. In this article we describe a cryptographic technique in which of secret keys are combined to make them compact as a single key. Here, power of all keys being aggregated. In other words the key owner will share an aggregate key with the second party on demand. The receivers will able to decrypt the files on cipher text choices but other encrypted files outside the cipher text set remains confidential. To make system leakage resilient, provision for intrusion detection and prevention is also made for secure data sharing between participants.*

1. INTRODUCTION

Cloud storage is gaining popularity recently. In enterprises, the rise in need for data outsourcing demands the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Nowadays, it is easy to apply for free email accounts, social networking sites accounts; file sharing and/or remote access, with storage size more than 25GB. Users can access almost all of their files and emails by a mobile phone in any corner of the world.

There are many cryptographic schemes in which third party will check the files availability on behalf of data owner without leaking anything about data. Cloud users do not belief strongly on these third parties in terms of confidentiality. User is not perfectly trusting the security of virtual machines VM or the honesty of technical staff. To overcome such situations research is going on. One of the study motivated users to encrypt their data themselves. Data owner uses their own keys and encrypt the data. Then uploads them to the server, also an intrusion detection system is provided to prevent the third party interruption.

Sharing and securing data, files, photos is an important function of cloud storage. Below example illustrates the situation. Fig1 [1]. Assume that Alice uploads all her private photos on a picasa, and she does not want to disclose her photos to others. Alice do not trust the

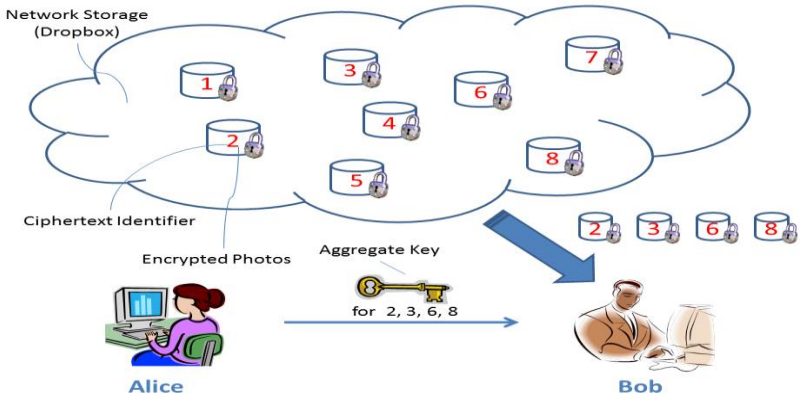

Fig 1. showing data sharing scenario

Privacy protection options provided by picasa. For better privacy she decides her own encryption key to encrypt photos and uploads photos. One day, Bob informed that he needs photos in which Bob appeared. Alice uses the share function of picasa, along with that she has to share the decryption key. But the trouble is how to pass on the decryption rights to Bob. Alice is also concerned if some third party is cheating her pretending that he is Bob. Possible option Alice choose is to securely send Bob the secret keys involved. Naturally, Alice encrypts all files with same encryption key and gives Bob the corresponding secret key directly. But in this case all unwanted data may also leak to Bob. Second method is that Alice encrypts files with dissimilar keys and sends Bob the secret keys consequently. The problem with the second method is efficiency. The number of decryption keys is equal to the number of shared photos. This is very inefficient to share such a large no. of keys.

While sharing the Key there are chances of cheating Alice by third party. Third party is pretending that he is Bob. Intrusion Detection System should monitor the cheating. A third party may exits who is cheating Alice in the name of Bob. Alice should have a collaborative intrusion detection system to recognize such attackers.

Therefore, the best solution for the above scenario is that Alice encrypts files with distinct public-keys, but shares single decryption key or an aggregate key with Bob. The decryption key should be sent through a secure network path having intrusion detection provisions. Desirable property of aggregate key is small size.

2. RELATED WORK

In this section we will study various cryptographic techniques which are used for KAC generation. Behaviour of Intrusion Detection system having collaborative hosts is studied for cloud data sharing.

2.1 Cryptographic Keys for a Predefined Hierarchy

Aim of cryptographic key assignment schemes (e.g. [7], [8]) is to reduce the cost in storing secret keys. Management of secret keys is also a major concern of cryptosystems.
Here tree structure is used for, deriving aggregate secret keys. In a tree hierarchy, key for a given node is used in deriving the keys of its child nodes. Just giving the parent key unconditionally grants all the keys of its child nodes. Sandhu [10] proposed how to create a tree hierarchy of symmetric keys. Repeated evaluations of pseudorandom function was used on a fixed secret.

Parent node in the tree represents a secret key, whereas the child nodes represent the keys for cipher text classes. Small circles in the Fig2 represent the keys for the classes to be delegated and Rectangle represents the keys to be granted. Note that every key of the non-leaf node can derive the keys of its child node. It is not always easy to decide key for a fixe hierarchy. Suppose Alice wants to upload some files in professor category, child nodes will come under class “professor”. Now two classes under “professor” are “department” and “classes”.

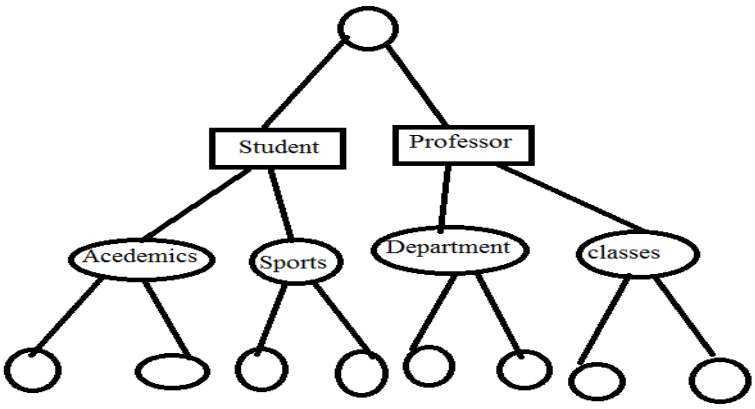

Fig 2. Tree heirarchy

In Figure2, if Alice wants to share all the files in the “professor” category, she only needs to grant the key for the node “professor”, which automatically grants the Bob who is delegatee the keys of all the descendant nodes (“Department”, “classes”). This is the ideal case, where most classes to be shared belong to the same branch and so a parent key of them is sufficient. However, it is difficult for general cases.

As shown in Figure 2, if Alice shares her demo class at student (“student”!“academics”!“demo” and “student”!“sports”!“demo”) with a colleague who also has the rights to see some of her professor’s data, she will give more keys, which results increase in the total key size. It is seen that this technique is not flexible when the classifications are getting more complex. people. Hierarchical approaches partially solve the problem.

2.2 Public -Key cryptography

Public-key cryptography [11], also known as **asymmetric cryptography**. It requires two distinct keys one of which is private and other one is public. Two parts of this key pair mathematically linked with each other. The public key is used for encryption and private key is used for decryption. Public Key encrypts the plain text to generate an encrypted data, while the private key is used to decrypt cipher text or to create original data. The term "asymmetric" arises from the use of different keys, each key is the inverse of the other.

Users can create their own public and private key-pair by doing mathematical computations and to use them for encryption as well as decryption. The strength of public-key cryptography is that it is "impracticable" to determine public key corresponding to a properly generated private key.Thus the public key may be available without compromise in security, but the private key must not be disclosed to unauthorized person to read messages.

Public-key algorithms are primary security methods in cryptographic applications and protocols. They support various networking standards, such as (TLS) Transport layer Security, PGP. Some public key algorithms provide key distribution and secrecy (e.g., Diffie-Hellman key exchange), some provide digital signature (e.g., Digital signature), and some provide both (e.g., RSA).

2.3 Identity-Based Encryption

Identity-based encryption (IBE) (e.g., [13] [14]) is a kind of public-key encryption technique. In IBE the public-key of a user is set as an identity-string of the user (e.g., an email id). There is a private key generator (PKG) in IBE which posses a master-secret key. PKG issues a secret key to each user and grants these keys with respect to the user’s identity.

Encryption of message is done using the public parameter and a user's identity. The recipient decrypts this ciphertext by using his secret key. Guo et al. [15], [6] tried to produce IBE with key aggregation. One scheme [15] assumes random oracles and another [6] does not. In this key aggregation, all keys to be aggregated should come from different "identity divisions". There are identities in exponential numbers and secret keys can be aggregated only in polynomial number of these identities. Key-aggregation [15], [6] has the expense of size O(n) for both ciphertexts and the public parameter. The number of secret keys is n which can be aggregated into size one. This increases the expense of storing and transmitting ciphertexts. Second way to use hash function to the string representing the class, and repeatedly hashing until a prime is obtained as output. Feature of these schemes are constant ciphertext size. In fuzzy IBE [14], one single compact secret key can decrypt ciphertexts encrypted under many identities.

2.4 Intrusion Detection System for Collaborative Hosts

Intrusion detection System is a type of security management scheme for computer and networks. When some untrusted party tries to gain authorization over communication channel, it is called intrusion. An ID system examines and gathers information from various areas within a computer network to identify security breaches. Intrusion detection include both third party attack for example hackers and misuse done by technical staff inside organization.

Intrusion detection functions include:

- Monitoring activities of users and outsiders and analyses system activity.
- Analyzing system configurations and vulnerabilities
- Assessing file integrity
- Ability to recognize attacks, type of attacks and pattern of attacks
- Watching of abnormal activity patterns
- Tracking user policy violation

Intrusion Detection in Cloud needs the distributed collaborative intrusion detection system in short termed as CIDS. In this collaborative intrusion detection system[5] each system becomes a peer(participant) of a distributed network. Each peer submits its own suspicious IP address list, which is detected from its own subnetwork, to the collaborative (peer-to-peer) system, and the participants are notified if other nodes are infected by the same source. . All the information sent to the network is only visible to the peers experiencing stealthy scans from the same source.

3. PROPOSED WORK

In this paper it is discussed how to use modern cryptographic techniques along with intrusion detection system so that data sharing in clouds becomes secure and intrusion free. Generally we study about how to use small piece of knowledge into cryptographic functions (e.g. encryption, authentication). However, in this paper we study how to make decryption key more powerful. Decryption Key should be powerful in the sense that it should allow decryption of multiple ciphertext without increasing its size. When decryption key is shared between two parties an Intrusion Detection System checks if any party involved in data sharing is cheating over the other.

Key Aggregate Cryptosystem is a special type of public-key encryption known as KAC. In KAC users encrypts a message not only under a public-key. These encrypted messages are classified under ciphertext classes. The key owner possesses a master-secret key, which is used to extract secret keys for different ciphertext classes. The extracted key is an aggregate key which is as compact as secret key for a single class. This proposed solution is shown in fig3.below, Alice can send Bob a single aggregate Key simply through email, Bob can download the encrypted photos from Alice's picasa space and then use this aggregate key to decrypt these encrypted photos. The sizes of ciphertext, public key, master secret key and aggregate key are constant in KAC.

Intrusion Detection System is also integrated with KAC. IDS is an independent system that monitors the network traffic and analyzes them if they are free from attack or not. IDS attempts to discover unauthorized access to a computer network by analyzing traffic on the network for malicious activity. Fig 3. Shows Intrusion Detection System that monitors KAC.

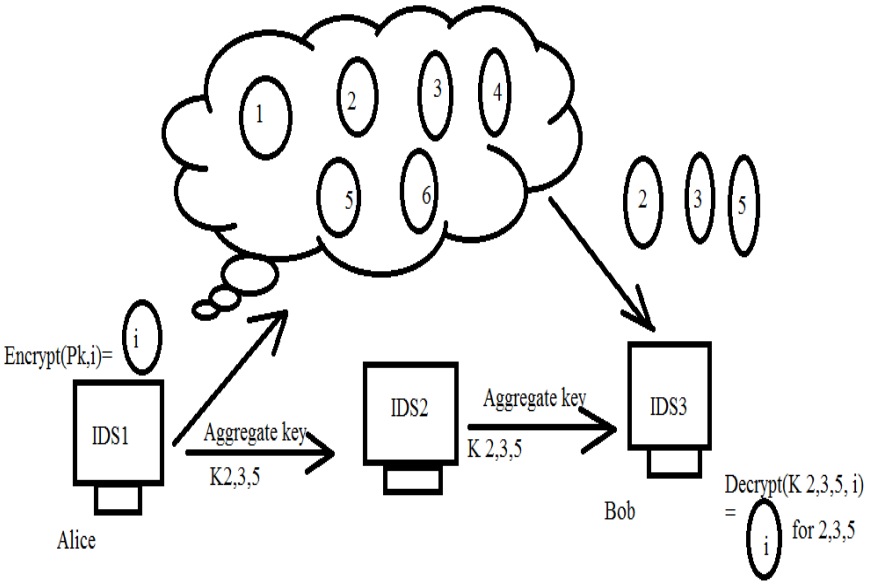

Fig 3. Proposed architecture

In the proposed architecture as shown in fig. 3. We have three system in clouds. Each system is acting as an intrusion detection system IDS1, IDS2, IDS3. Alice Encrypts the file and uploads on the picasa. Bob demands the decryption right of photos . Alice will share Aggregate secret key . At the same time three system monitors if any of the party is cheating over the other using CIDS.

4. ALGORITHMS USED

Various Algorithms are used for implementing the proposed system. Each phase of this project involves one algorithm. First phase is uses public-key cryptographic algorithm (RSA) for encrypting files at file owner's wish. In second phase Aggregate Secret key is generated. Aggregate secret key generation is completed in four steps , where many cryptographic schemes are applied. Third phase is intrusion detection phase in which CIDS[5] (collaborative intrusion detection system) algorithm is used.

4.1 RSA Algorithm

RSA[6] is one of public key cryptosystem used for secure transmission of data. In such a cryptosystem, the encryption key is public and differs from the decryption key which is kept secret. RSA stands for Ron, Rivest, Adi Shamir and Leonard Adleman, who first publicly proposed the algorithm in 1977.

A user creates a public key based on the two large prime numbers, and then publishes along with an auxiliary value. The prime numbers should be kept secret. Anyone can use the public key to encrypt a message. When key is large enough, someone with knowledge of the prime numbers can possibly decode the message. Breaking RSA encryption is called as the RSA problem.

In our KAC scheme we allow data owner to use RSA algorithm to encrypt files. Files are encrypted under different ciphertext classes on data owner's choice.

4.2 Aggregate Secret Key Generation

The data owner produces public/master-secret key pair in this phase. Aggregate key[1] generation phase is divided in three steps. Messages can be encrypted using Encrypt( ) function by anyone who also decides which ciphertext class is associated with the plain text message. The data owner can use master-secret to generate an aggregate decryption key for ciphertext classes using function Extract( )

- Encrypt(PK,i, m) : In the first phase public key PK is generated. On giving input a public key pk, an index i(increment counter), which denots the ciphertext class, and a message m, it outputs a ciphertext C.
- Extract(Msk, S) : Data owner extracts the Aggregate secret key. Here, On giving input as master-secret key Msk and a set of indices S that corresponds to different classes, it outputs the aggregate key Ks for set S.
- Decrypy(Ks, S, i, C) : Decryption of aggregate secret key is performed by one who received an aggregate key Ks generated in extract phase. On input Ks, the set S, an index i denoting the ciphertext class C belongs to, and C, it outputs the decrypted result m.

4.3 Collaborative intrusion detection System CIDS

In CIDS[5] algorithm each peer (participants) in the distributed network requires an approach to selectively root evidences between peers. In particular to realize publish/subscribe mechanism we require an approach that can scale to a large number of peers exchanging many subscription and notification messages. To satisfy these requirements, we propose Distributed hash table (DHT) approach for sharing evidences between peers.

Each peer have two roles: (1) it is responsible for maintaining the blacklist for its local subnetwork, and (2) it is responsible for correlating subscription messages and generating notification messages about the source addresses that are mapped to the peer

- In the first role, the peer will generate subscribe messages, each peer submits its blacklist that is its own suspicious IP addresses list, which is detected from its own sub-network.
- Then it receives the notify messages, that all other peers are aware of the suspicious sub network.
- While in the second role, the peer will receive subscribe messages.A node receives a subscription message, this means that another node considers the corresponding source s to be suspicious and generate notify messages.
- Then it receives the notify messages, that all other peers are aware of the suspicious sub network.
- All the information sent to the network is only visible to the peers experiencing stealthy scans from the same source.

5. IMPLEMENTATION REQUIREMENTS

The basic KAC was implemented in C. Here for this paper language used is JAVA. Collaborative intrusion detection system is implemented on a peer-to-peer architecture

called OPeN. Three layers in the OPeN architecture are application layer, core services layer and connectivity layer. We built our CIDS service on top of the connectivity layer of OPeN. It currently uses Chord as routing protocol. The system was implemented in JAVA.

6. CONCLUSION AND FUTURE ENHANCEMENT

Protect users' data privacy is a central concern of cloud storage. Cryptographic schemes are getting more and more versatile with the help of mathematical tools. Single application involve multiple keys. In this article, we consider how to "compress" or "aggregate" secret keys in public-key cryptosystems. This supports assignment of secret keys to different ciphertext classes in cloud storage. It does not matter which class is among the power set of classes. The delegate to whom aggregate key is handed over always gets an aggregate key of constant size. Intrusion detection system is integrated with KAC that has the potential to improve detection accuracy, to the problem of how to effectively share information between participants. Distributed hash table architecture for collaborative intrusion detection overcomes the challenges of the collaboration, such as data routing, load balancing, scalability and central points of failure.

In cloud storage the number of cipher texts grow rapidly. It will be better if maximum number of ciphertext classes is unlimited. IDS should be more efficient so that when mobile users carries the delegated key the cryptosystem is leakage resilient. Key delegation can be more flexible.

ACKNOWLEGEMENT

This is an opportunity to express my gratitude towards everyone who suggested and helped me in this Review paper. I wish devote my sincere thanks to my guide Prof. Lakshmimadhuri, Head of the Department Prof. Soumitra Das, Director of DYPSOE(Technical campus) Dr. S.S.Sonawane and ME Coordinator Prof. P.M Agarkar. I am also thankful to technical staff of Computer Department.for their support.

REFERENCES

[1]. Key-Aggregate Cryptosystem for Scalable Data Sharing in Cloud Storage Cheng-Kang Chu, Sherman S. M. Chow, Wen-Guey Tzeng, Jianying Zhou, and Robert H. Deng, Senior Member, IEEE.

[2]. S. S. M. Chow, Y. J. He, L. C. K. Hui, and S.-M. Yiu, "SPICE - Simple Privacy-Preserving Identity-Management for Cloud Environment," in Applied Cryptography and Network Security – ACNS 2012 Springer, 2012.

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

[4]. Chenfeng Vincent Zhou, Shanika Karunasekera and Christopher Leckie National ICT Australia Department of Computer Science and Software Engineering University of Melbourne, Australia Email: {cvzhou, shanika, caleckie}@cs.mu.oz.au "A Peer-to-Peer Collaborative Intrusion Detection System,"

[5]. RSA security Releases RSA Encryption Algorithm into Public Domain at the Wayback Machine(archived June 2007)

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

[7]. S. G. Akl and P. D. Taylor, Cryptographic Solution to a Problem of Access Control in a Hierarchy," ACM Transactions on Computer Systems (TOCS), vol. 1, no. 3, pp. 239–248, 1983.

[8]. G. C. Chick and S. E. Tavares, "Flexible Access Control with Master Keys," in Proceedings of Advances in Cryptology – CRYPTO '89, ser. LNCS, vol. 435. Springer, 1989, pp. 316–322.

[9]. G. Ateniese, A. D. Santis, A. L. Ferrara, and B. Masucci, "Provably-Secure Time-Bound Hierarchical Key Assignment Schemes," J. Cryptology, vol. 25, no. 2, pp. 243–270, 2012.

[10]. R. S. Sandhu, "Cryptographic Implementation of a Tree Hierarchy for Access Control," Information Processing Letters, vol. 27, no. 2,

[11]. Christof Paar, Jan Pelzl, "Introduction to Public-key Cryptography", Understanding Cryptography, Springer, 2009.

[12]. Q. Zhang and Y. Wang, "A Centralized Centralized Key Management Scheme for Hierarchical Access Control," in Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '04). IEEE, 2004, pp. 2067–2071.

[13]. D. Boneh and M. K. Franklin, "Identity-Based Encryption from the Weil Pairing," in Proceedings of Advances in Cryptology – CRYPTO '01, ser. LNCS, vol. 2139. Springer, 2001, pp. 213–229.

[14]. A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in Proceedings of Advances in Cryptology - EUROCRYPT '05, ser. LNCS, vol. 3494. Springer, 2005, pp. 457–473.

[15]. 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. 152–161.

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

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

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

[19]. R. Canetti and S. Hohenberger, "Chosen-Ciphertext Secure Proxy Re-Encryption," in Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS '07). ACM, 2007, pp. 185–194.

[20]. C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption Without Random Oracles," in Information Security Conference