Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

SURVEY ON KEY AGGREGATE CRYPTOSYSTEM FOR SCALABLE DATA SHARING

Nandhini.J [1],Ramasamy.S [2]
[1]PG Scholar, Vivekananda College of Engineering for Women
[2]Assistant professor Department of Computer Science and Engineering,
Vivekananda College of Engineering for Women

ABSTRACT: Public-key cryptosystems produce constant-size cipher texts with efficient delegation of decryption rights for any set of cipher texts. One can aggregate any set of secret keys and make them as compact as a single key. The secret key holder can release a constant-size aggregate key for flexible choices of cipher text set in cloud storage. In KAC, users encrypt a message not only under a public-key, but also under an identifier of cipher text called class. That means the cipher texts are further categorized into different classes. The key owner holds a master-secret called master-secret key, which can be used to extract secret keys for different classes. More importantly, the extracted key have can be an aggregate key which is as compact as a secret key for a single class, but aggregates the power of many such keys, i.e., the decryption power for any subset of cipher text classes. The key aggregate cryptosystem is enhanced with boundary less cipher text classes. The system is improved with device independent key distribution mechanism. The key distribution process is enhanced with security features to protect key leakage. The key parameter transmission process is integrated with the cipher text download process.

Keywords- key aggregate encryption, public key cryptosystem, cloud storage, attribute based encryption

I. INTRODUCTION

Cloud computing has become a significant technology trend either in the industrial or the academic field, and most of the experts expect that cloud computing will reshape information technology (IT) processes and the IT market place. In Cloud Computing, users connect to the 'Cloud', which appears as if it is a single entity as opposed to multiple servers. In this model, users can remotely store their data so as to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources Although this pay-per-use model of the cloud services brings significant savings for users and offers flexibility and scalability in terms of capacity and performance, it involves giving the cloud service provider (CSP) some form of control over the user's data.

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model promotes availability and is composed of five essential characteristics, three service models, and four deployment models**.** A 'cloud' is an elastic execution environment of resources involving multiple stakeholders and providing a metered service at multiple granularities for a specified level of quality (of service).

1.1. TYPES OF CLOUD

Cloud providers typically centre on one type of cloud functionality provisioning: Infrastructure, Platform or Software / Application, though there is potentially no restriction to offer multiple types at the same time, which can often be observed in PaaS (Platform as a Service) providers.

1.1.1. Infrastructure as a Service (IaaS) also referred to as Resource Cloud, provide scalable resources as services to the user. Accordingly, different resources may be provided via a service interface: Data & Storage Cloud deal with reliable access to data of potentially dynamic size, weighing resource usage with access requirements.
Examples: Amazon S3, SQL Azure.

1.1.2. Platform as a Service (PaaS), provide computational resources via a platform upon which applications and services can be developed and hosted. PaaS typically makes use of dedicated APIs to control the behaviour of a server hosting engine which executes and replicates the execution according to user requests.
Examples: Force.com, Google App Engine, Windows Azure (Platform).

1.1.3. Software as a Service (SaaS), also sometimes referred to as Service or Application Cloud are offering implementations of specific business functions and business processes that are provided with specific cloud capabilities, i.e. they provide applications / services using a cloud infrastructure or platform, rather than providing cloud features themselves.
Examples: Google Docs, Salesforce CRM.

1.2. DEPLOYMENT TYPES

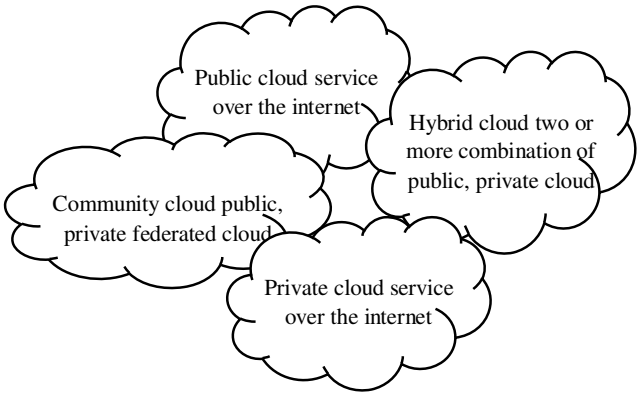

Figure 1: Deployment types of cloud

1.2.1. Private Cloud are typically owned by the respective enterprise and / or leased. Functionalities are not directly exposed to the customer, though in some cases services with cloud enhanced features may be offered – this is similar to (Cloud) Software as a Service from the customer point of view.
Example: eBay.

1.2.2. Public Cloud Enterprises may use cloud functionality from others, respectively offer their own services to users outside of the company. Providing the user with the actual capability to exploit the cloud features for his / her own purposes also allows other enterprises to outsource their services to such cloud providers, thus reducing costs and effort to build up their own infrastructure
Example: Amazon, Google Apps, Windows Azure.

1.2.3. Hybrid Cloud Though public cloud allow enterprises to outsource parts of their infrastructure to cloud providers, they at the same time would lose control over the resources and the distribution / management of code and data. Hybrid cloud consist of a mixed employment of private and public cloud infrastructures so as to achieve a maximum of cost reduction through outsourcing

1.2.4. Community Cloud Typically cloud systems are restricted to the local infrastructure, i.e. providers of public cloud offer their own infrastructure to customers. Though the provider could actually resell the infrastructure of another provider, cloud do not aggregate infrastructures to build up larger, cross-boundary structures. In particular smaller SMEs could profit from community cloud to which different entities contribute with their respective (smaller) infrastructure. Community cloud can either aggregate public cloud or dedicated resource infrastructures

1.3. SECURITY ISSUES ASSOCIATED WITH THE CLOUD

There are number of security issues/concerns associated with cloud computing but these issues fall into two broad categories: Security issues faced by cloud providers (organizations providing infrastructure as a service (IaaS), platform as a service (PaaS), or software as a service (SaaS) via the cloud) and security issues faced by their customers .In most cases, the provider must ensure that their infrastructure is secure and that their clients' data and applications are protected while the customer must ensure that the provider has taken the proper security measures to protect their information.

II. LITERATURE SURVEY

2.1. Privacy Preserving Public Auditing For Secure Cloud Storage

In cloud storage, users can remotely store their data and using the on-demand high-quality applications and services from a shared pool of configurable computing resources. Users should be able to just use the cloud storage as if it is local, without worrying about data integrity. So the public auditability for cloud storage is of critical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data .This TPA with secure cloud storage system supporting privacy preserving public auditing. Further, extend of the result to enable the TPA to perform audits for multiple users concurrently and efficiently.

The technique of public key-based homomorphic linear authenticator (HLA) which enables TPA to perform the auditing without demanding the local copy of data and thus reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the HLA with random masking, guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server (CS) during the efficient auditing process. The aggregation and algebraic properties of the authenticator provides additional benefits for the auditing. To enable privacy-preserving public auditing for cloud data storage design should achieve the following security and performance guarantees:

- Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users.
- Storage correctness: to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact.
- Privacy preserving: to ensure that the TPA cannot derive users' data content from the information collected during the auditing process.
- Batch auditing: to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.

- Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.

2.2. Decentralized Access Control with Anonymous Authentication of Data Stored in Cloud

A new decentralized access control scheme for secure data storage in cloud that supports anonymous authentication. In the decentralised access control scheme, the cloud verifies the authenticity of the series without knowing the user's identity before storing data and also added the feature of access control in which only valid users are able to decrypt the stored information.

According to access control scheme a user can create a file and store it securely in the cloud. This scheme consists of use of the two protocols Attribute based encryption (ABE) and Attribute based signature (ABS), respectively.

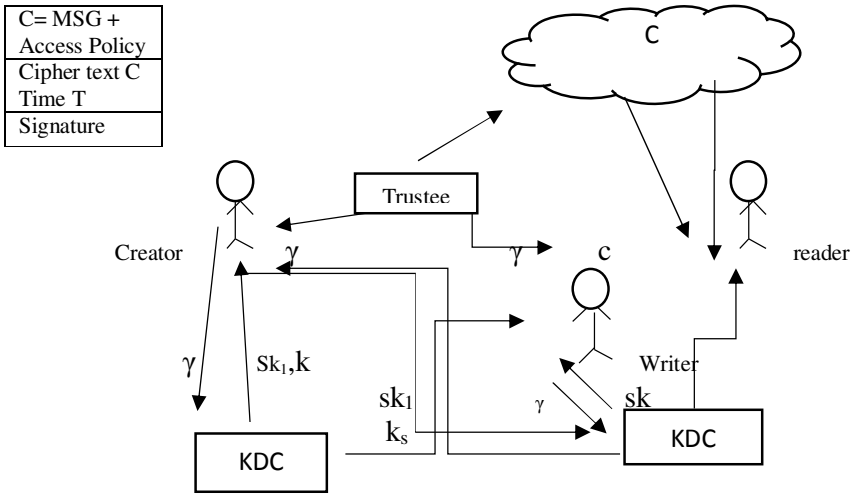

Figure 2. Secure cloud storage model.

We refer to the Fig. 2. There are three users, a creator, a reader, and writer. Creator receives a token (γ) from the trustee, who is assumed to be honest. There are multiple Key Distribution Centre (KDCs) (here 2), which can be scattered. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. In the Figure. 2, SKs are secret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message under this claim. The cipher text C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the cipher text C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud. In this scheme, the cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Key distribution is done in a decentralized way. One limitation is that the cloud knows the access policy for each record stored in the cloud.

2.3. Provably-Secure Time-Bound Key Assignment Scheme

A key assignment scheme is a cryptographic technique for implementing an information flow policy, sometimes it may be known as hierarchical access control. A time-bound hierarchical key assignment scheme is a method to assign time-dependent encryption keys to a set of classes in a partially ordered hierarchy, in such a way that the key of a higher class can be used to derive the keys of all classes lower down in the hierarchy, according to temporal constraints. A hierarchical key assignment scheme is a method to assign an encryption key and some private information to each class in the hierarchy. This assignment is carried out by a central authority, the Trusted Authority (TA), which is active only at the distribution phase. In a time bound hierarchical key assignment scheme users need a different key for each time period which implies that the key derivation procedure should also depend on the time period other than the hierarchy of the classes. Once a time period expires, users in a class should not be able to access any subsequent keys if they are not authorized to do so. The algorithm used in this scheme is RSA based time bound hierarchical key assignment.

There are two different construction of hierarchical key assignment scheme. The first one is based on symmetric encryption schemes, whereas, the second one makes use of bilinear maps. These appear to be the first constructions of time-bound hierarchical key assignment schemes which are simultaneously practical and provably-secure.

2.4. Scalable and Secure Sharing of Personal Health Records in Cloud Computing Using Attribute-Based Encryption

Personal health record (PHR) is an emerging patient-centric model of health information exchange, which is often outsourced to be stored at a third party, such as cloud providers. A PHR service allows a patient to create, manage, and control her personal health data in one place through the web, which has made the storage, retrieval, and sharing of the medical information more efficient. To achieve fine-grained and scalable data access control for PHRs, we leverage attribute-based encryption (ABE) techniques to encrypt each patient's PHR file. A high degree of patient privacy is guaranteed simultaneously by exploiting multiauthority ABE. This scheme also enables dynamic modification of access policies or file attributes, supports efficient on-demand user/attribute revocation and break-glass access under emergency scenarios.

The main goal of this scheme is to provide secure patient-centric PHR access and efficient key management. The key idea is to divide the system into multiple security domains (namely, public domains and personal domains) according to the different users' data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors and medical researchers. For each PSD, its users are personally associated with a data owner (such as family members or close friends), and they make accesses to PHRs based on access rights assigned by the owner.

2.5. Scalable and Efficient Data Possession

Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client's (potentially very large) outsourced data.

The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or

requiring the client to outsource its data in encrypted form. In this paper, we construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, our PDP technique allows outsourcing of dynamic data, i.e, it efficiently supports operations, such as block modification, deletion and append.

Two recent results PDP and POR have highlighted the importance of the problem and suggested two very different approaches. The first is a public key- based technique allowing any verifier (not just the client) to query the server and obtain an interactive proof of data possession. This property is called public verifiability. The interaction can be repeated any number of times, each time resulting in a fresh proof. The POR scheme uses special blocks (called sentinels) hidden among other blocks in the data. During the verification phase, the client asks for randomly picked sentinels and checks whether they are intact. If the server modifies or deletes parts of the data, then sentinels would also be affected with a certain probability.

2.6. Proxy Re-encryption Systems for Identity-based Encryption

A proxy re-encryption system allows the proxy to transform cipher texts encrypted under public key into the different cipher texts that can be decrypted by secret key. The proxy re-encryption system should at least satisfy the following requirements; 1) a proxy alone cannot obtain the underlying plaintext, 2) and delegatee cannot obtain the underlying plaintext without the proxy cooperating. The proxy re-encryption system can be a primitive for various attractive applications.

In the IBE system, a sender encrypts a message to an IBE receiver by using receiver's identity as a public key. The IBE system can dramatically improve the workload for public key certificate management, while it is heavy burden in the traditional certificate based public key encryption (CBE) system premised on Public Key Infrastructure (PKI) service. The IBE system necessarily requires a third party called Public Key Generator (PKG) which generates all secret keys for IBE users by using its master-secret key, and thus the IBE system works where the PKG operation can be allowed.

2.7 Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data

Attribute-based encryption for fine-grained access control of encrypted data is a new cryptosystem for fine-grained sharing of encrypted data that we call Key-Policy Attribute-Based Encryption (KP-ABE).As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites. One drawback of encrypting data, is that it can be selectively shared only at a coarse-grained level (i.e., giving another party your private key). In this cryptosystem, cipher text is labelled by the encryptor with a set of descriptive attributes. Each private key is associated with an access structure that specifies which type of cipher texts the key can decrypt. We call such a scheme a Key-Policy Attribute-Based Encryption (KP-ABE), since the access structure is specified in the private key, while the cipher texts are simply labelled with a set of descriptive attributes. A user is able to decrypt a cipher text if the attributes associated with a cipher text satisfy the key's access structure.

This setting is reminiscent of secret sharing schemes. Secret-sharing schemes (SSS) are used to divide a secret among a number of parties. The information given to a party is called the share (of the secret) for that party. For example, one can specify a tree access structure where the interior nodes consist of AND and OR gates and the leaves consist of different parties. Any set of parties that satisfy the tree can reconstruct the secret and this construction each user's key is associated with a tree-access structure where the leaves are associated with attributes. Key Policy Attribute Based Encryption (KP-ABE) scheme is designed for one-to-many communications.

2.8. Improved Proxy Re-Encryption Schemes with Applications to Secure Distribute Storage

Proxy re-encryption allows a proxy to transform a cipher text computed under Alice‘s public key into one that can be opened by Bob‘s secret key. There are many useful applications of this primitive. For instance, Alice might wish to temporarily forward encrypted email to her colleague Bob, without giving him her secret key. In this case, Alice the delegator could designate a proxy to re-encrypt her incoming mail into a format that Bob the delegate can decrypt using his own secret key. Alice could simply provide her secret key to the proxy, but this requires an unrealistic level of trust in the proxy. It present several efficient proxy re-encryption schemes that offer security improvements over earlier approaches. The primary advantage of our schemes is that they are unidirectional (i.e., Alice can delegate to Bob without Bob having to delegate to her) and do not require delegators to reveal all of their secret key to anyone – or even interact with the delegatee – in order to allow a proxy to re-encrypt their cipher texts. In our schemes, only a limited amount of trust is placed in the proxy. Most useful properties of proxy re-encryption protocols:

- Unidirectional: Delegation from A→B does not allow re-encryption from B→A
- Non-interactive: Re-encryption keys can be generated by Alice using Bob‘s public key; no trusted third party or interaction is required.
- Proxy invisibility: both sender and recipient are aware of the proxy re-encryption protocol but do not know whether the proxy is active, has performed any action or made any changes, or even if it exists.
- Original-access: Alice can decrypt re-encrypted cipher texts that were originally sent to her.

2.9 Aggregate and Verifiably Encrypted Signatures from Bilinear Maps

An aggregate signature scheme is a digital signature that supports aggregation: Given n signatures on n distinct messages from n distinct users, it is possible to aggregate all these signatures into a single short signature. This single signature (and the n original messages) will convince the verifier that the n users did indeed sign the n original messages (i.e., user I signed message Mi for i = 1 . . . n). The concept of an aggregate signature, present security models for such signatures, and give several applications for aggregate signatures. Aggregate signatures are useful for reducing the size of certificate chains (by aggregating all signatures in the chain) and for reducing message size in secure routing protocols such as SBGP. An aggregate signature scheme enables us to achieve precisely this type of compression. Suppose each of n users has a public-private key pair (PKi, SKi). User ui signs message Mi to obtain a signature σi. Then there is a public aggregation algorithm that takes as input all of σ1. . . σn and outputs a short compressed signature σ.

2.10. Decentralizing Attribute Based Encryption

Decentralizing attribute-based encryption propose a Multi-Authority Attribute-Based Encryption (ABE) system. In this system, any party can become an authority and there is no requirement for any global coordination other than the creation of an initial set of common reference parameters. A party can simply act as an ABE authority by creating a public key and issuing private keys to different users that react their attributes. A user can encrypt data in terms of any Boolean formula over attributes issued from any chosen set of authorities. Finally, the system does not require any central authority. Spreading a central authority's keys over several machines to alleviate performance pressures might simultaneously increase the risk of key exposure. In constructing this system, the largest technical hurdle is to make it collusion resistant. Prior Attribute-Based Encryption systems achieved collusion resistance when the ABE system authority ―tied" together different components (representing different attributes) of a user's private key by randomizing the key. However, in our system each component will come from a potentially different authority, where we assume no coordination between such authorities.

III. Conclusion

How to protect users' data privacy is a central question of cloud storage. With more mathematical tools, cryptographic schemes are getting more versatile and often involve multiple keys for a single application. In this article, we consider how to "compress" secret keys in public-key cryptosystems which support delegation of secret keys for different cipher text classes in cloud storage. No matter which one among the power set of classes, the delegate can always get an aggregate key of constant size. Our approach is more flexible than hierarchical key assignment which can only save spaces if all key-holders share a similar set of privileges.

References

[1] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Secure Cloud Storage," IEEE Trans. Computers, vol. 62, no. 2, pp. 362-375, Feb. 2013.

[2] 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.

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

[4] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '03), pp. 416-432, 2003.

[5] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), pp. 89-98, 2006.

[6] C.-K. Chu and W.-G. Tzeng, "Identity-Based Proxy Re-encryption without Random Oracles," Proc. Information Security Conf. (ISC '07), vol. 4779, pp. 189-202, 2007.

[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Trans. Information and System Security, vol. 9, no. 1, pp. 1-30, 2006.

[8] A. B. Lewko and B. Waters (2011) "Decentralizing attribute-based encryption" in Proc. EUROCRYPT, pp. 568–588.

[9] T.H. Yuen, S.S.M. Chow, Y. Zhang, and S.M. Yiu, "Identity-Based Encryption Resilient to Continual Auxiliary Leakage," Proc. Advances in Cryptology Conf. (EUROCRYPT '12), vol. 7237, pp. 117-134, 2012.

[10] D. Boneh, X. Boyen, and E.-J. Goh, "Hierarchical Identity Based Encryption with Constant Size Ciphertext," Proc. Advances in Cryptology Conf. (EUROCRYPT '05), vol. 3494, pp. 440-456, 2005.

[11] D. Boneh, R. Canetti, S. Halevi, and J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption," SIAM J. Computing, vol. 36, no. 5, pp. 1301-1328, 2007.

[12] D. Boneh, C. Gentry, and B. Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys," Proc. Advances in Cryptology Conf. (CRYPTO '05), vol. 3621, pp. 258-275, 2005.

[13] D. Boneh and M.K. Franklin, "Identity-Based Encryption from the Weil Pairing," Proc. Advances in Cryptology (CRYPTO '01), vol. 2139, pp. 213-229, 2001.

[14] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques (EUROCRYPT '05), vol. 3494, pp. 457-473, 2005.

[15] S.S.M. Chow, Y. Dodis, Y. Rouselakis, and B. Waters, "Practical Leakage-Resilient Identity-Based Encryption from Simple Assumptions," Proc. ACM Conf. Computer and Comm. Security, pp. 152-161, 2010.

[16] Q. Zhang and Y. Wang, "A Centralized Key Management Scheme for Hierarchical Access Control," Proc. IEEE Global Telecomm. Conf. (GLOBECOM '04), pp. 2067-2071, 2004.

[17] J. Benaloh, "Key Compression and Its Application to Digital Fingerprinting," technical report, Microsoft Research, 2009.

