International Transaction of Electrical and Computer Engineers System, 2015, Vol. 3, No. 1, 12-18
Available online at http://pubs.sciepub.com/iteces/3/1/2
© Science and Education Publishing
DOI:10.12691/iteces-3-1-2

Cryptography during Data Sharing and Accessing Over Cloud

Shyam Nandan Kumar[*]

M.Tech-Computer Science and Engineering, Lakshmi Narain College of Technology-Indore (RGPV, Bhopal), MP, India
[*]Corresponding author: shyamnandan.mec@gmail.com

Received March 02, 2015; Revised March 12, 2015; Accepted March 17, 2015

Abstract Data sharing is an important functionality in cloud environment. With the advent of the World Wide Web and the emergence of e-commerce applications and social networks, organizations across the world generate a large amount of data daily. This data would be more useful to cooperating organizations if they were able to share their data. In this article, an efficient methodology is provided to securely, efficiently, and flexibly share data with others in cloud computing. In this technique, the secret key holder can release a constant-size aggregate key for flexible choices of cipher-text set in cloud storage, but the other encrypted files outside the set remain confidential. Secure cryptographic architecture and working methodology are proposed in this paper for optimal services over the cloud.

Keywords: *cloud computing, data sharing, decryption, encryption*

Cite This Article: Shyam Nandan Kumar, "Cryptography during Data Sharing and Accessing Over Cloud." *International Transaction of Electrical and Computer Engineers System*, vol. 3, no. 1 (2015): 12-18. doi: 10.12691/iteces-3-1-2.

1. Introduction

Cloud computing (so-called, cloud) represents one of the magnificent shifts in information technology which can enhance collaboration, agility, scaling and availability, and provide the potential for cost reduction through optimized and efficient computing. Different from the existing technologies and computing approaches, cloud is defined with five essential characteristics (on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service), SPI service models (Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS)), and deployment models (Public, Private, Hybrid, Community).

Cloud Sharing is gaining popularity recently. In enterprise settings, we see the rise in demand for data outsourcing, which assists in the strategic management of corporate data. It is also used as a core technology behind many online services for personal applications. Access control is generally a policy or procedure that allows, denies or restricts access to a system. It may, as well, monitor and record all attempts made to access a system. Access Control may also identify users attempting to access a system unauthorized. It is a mechanism which is very much important for protection in computer security. Various access control models are in use, including the most common Mandatory Access Control (MAC), Discretionary Access Control (DAC) and Role Based Access Control (RBAC). All these models are known as identity based access control models. In all these access control models, user (subjects) and resources (objects) are identified by unique names. Identification may be done directly or through roles assigned to the subjects. These access control methods are effective in unchangeable distributed system, where there are only a set of Users with a known set of services.

There are two important challenges in secure outsourcing. First, the stored data must be protected against unauthorized access. Second, both the data and the access to data need to be protected from cloud storage service providers (e.g., cloud system administrators). In these scenarios, relying on password and other access control mechanisms is insufficient. Cryptographic encryption mechanisms are typically employed. However, simply having encryption and decryption implemented in the cloud database systems is insufficient. In order to support both challenges, data should be encrypted first by users before it is outsourced to a remote cloud storage service and both data security and data access privacy should be protected such that cloud storage service providers have no abilities to decrypt the data, and when the user wants to search some parts of the whole data, the cloud storage system will provide the accessibility without knowing what the portion of the encrypted data returned to the user is about.

The challenging problem is how to effectively share encrypted data. Of course users can download the encrypted data from the storage, decrypt them, then send them to others for sharing, but it loses the value of cloud storage. Users should be able to delegate the access rights of the sharing data to others so that they can access these data from the server directly. However, finding an efficient and secure way to share partial data in cloud storage is not trivial. In a shared-tenancy cloud computing environment, things become even worse. Data from different clients can be hosted on separate virtual machines (VMs) but reside on a single physical machine. Data in a target VM could be stolen by instantiating another VM co-resident with the target one [1].

In this paper sections are organized as follows: Section 2 gives the idea about types of attacks on cloud. Section 3 reviews some related works. Section 4 deals with data sharing and access approach. Section 5 describes the proposed scheme. Section 6 concludes the paper and presents avenues for future work. References for this paper are given in section 7.

2. Types of Attacks on the Cloud

There are a number of types of privacy and security attacks in the Cloud. The following contains a summary of the common types of attacks that may occur in the Cloud.

2.1. Flooding Attacks

A malicious user can send requests to the Cloud; he/she can then easily overload the server by creating bogus data requests to the Cloud. The attempt is to increase the workload of the Cloud servers by consuming lots of resources needlessly.

2.2. Law Enforcement Requests

When the FBI or government demand a Cloud Service Provider access to its data, the Cloud Service Provider is least likely to deny them. Hence, there may an inherent threat to user privacy and confidentiality of data.

2.3. Data Stealing Attacks

A term used to describe the stealing of a user account and password by any means such as through brute-force attacks or over-the-shoulder techniques. The privacy and confidentiality of user’s data will be severely breached. A common mechanism to prevent such attacks is to include an extra value when authenticating. This value can be distributed to the right user by SMS and hence mitigate the likelihood of data confidentiality issues.

2.4. Denial-of-Service Attacks

Malicious code is injected into the browser to open many windows and as a result deny legitimate users access to services.

2.5. XML Signature Wrapping Attacks

Using different kinds of XML signature wrapping attacks, one can completely take over the administrative rights of the Cloud user and create, delete, modify images as well as create instances.

2.6. Cross site scripting attacks

Attackers can inject a piece of code into web applications to bypass access control mechanisms. Researchers found this possible with Amazon Web Services [9] in November 2011. They were able to gain free access to all customer data, authentication data, and tokens as well as plaintext passwords.

3. Related Work

In recent years, the research areas on secure data processing have gained more and more attention. There exist several expressive Attribute Based Encryption (ABE) schemes where the decryption algorithm only requires a constant number of pairing computations. Recently, Green et al. proposed a remedy to this problem by introducing the notion of ABE with outsourced decryption, which largely eliminates the decryption overhead for users. Based on the existing ABE schemes, Green et al. also presented concrete ABE schemes with outsourced decryption.

Searchable encryption schemes are designed to solve security problems for remote cryptographic storage while enabling search for the expected contents corresponding to an encrypted keyword securely. Symmetric searchable encryption (SSE) scheme introduced in [2] is suitable for the setting where a party searching over the data is also the one who generates it. Such scenario is referred to as single writer and single reader (SW/SR).

Asymmetric searchable encryption (ASE) is designed for the scenario where a party searching over the data can be different from the party who generates it [3]. Such scenario is referred to as many writers and single reader (MW/SR). Since writers and readers can be different, ASE schemes are more suitable for the setting with a larger number of users. Both SSE and ASE protocols did not completely solve the problem that one can privately retrieve segments of encrypted data from remote databases. Since the database server can learn by passive logging with statistical inference which encrypted keyword matches the submitted search keyword and which encrypted document is retrieved.

Attribute-based encryption (ABE) [4,5] allows each cipher-text to be associated with an attribute, and the master-secret key holder can extract a secret key for a policy of these attributes so that a cipher-text can be decrypted by this key if its associated attribute conforms to the policy. For example, with the secret key for the policy (2 v 3 v 6 v 8), one can decrypt cipher-text tagged with class 2; 3; 6 or 8. However, the major concern in ABE is collusion-resistance but not the compactness of secret keys. Indeed, the size of the key often increases linearly with the number of attributes it encompasses, or the cipher-text-size is not constant.

To delegate the decryption power of some cipher-texts without sending the secret key to the delegates, a useful primitive is proxy re-encryption (PRE) [6]. A PRE scheme allows Alice to delegate to the server (proxy) the ability to convert the cipher-texts encrypted under her public-key into ones for Bob. PRE is well known to have numerous applications including cryptographic file system [7]. Nevertheless, Alice has to trust the proxy that it only converts cipher-texts according to her instruction, which is what we want to avoid at the first place. Even worse, if the proxy colludes with Bob, some form of Alice’s secret key can be recovered which can decrypt Alice’s (convertible) cipher-texts without Bob’s further help. That also means that the transformation key of proxy should be well-protected. Using PRE just moves the secure key storage requirement from the delegates to the proxy. It is thus undesirable to let the proxy reside in the storage server.

That will also be inconvenient since every decryption requires separate interaction with the proxy.

An Efficient Certificate-less Encryption for Secure Data Sharing in Public Clouds [8] is proposed by Seung-Hyun Seo, Nabeel, M, Bertino, E. and Xiaoyu Ding. It deals with a mediated certificateless encryption scheme without pairing operations for securely sharing sensitive information in public clouds. Mediated certificate-less public key encryption (mCL-PKE) solves the key escrow problem in identity based encryption and certificate revocation problem in public key cryptography. However, existing mCL-PKE schemes are either inefficient because of the use of expensive pairing operations or vulnerable against partial decryption attacks. It was not a decentralized approach.

One of the main efficiency drawbacks of the most existing ABE schemes is that decryption is expensive for resource-limited devices due to pairing operations, and the number of pairing operations required to decrypt a cipher-text grows with the complexity of the access policy. The above observation motivates us to study ABE with verifiable outsourced decryption in this thesis work. Here emphasized that an ABE scheme with secure outsourced decryption does not necessarily guarantee verifiability (i.e., correctness of the transformation done by the cloud server).

In order to set up a suitable cryptography over cloud, in the paper, inspired by above related work, a novel mechanism is proposed for secure communication.

4. Data Sharing and Accessing in the Cloud

With the advancements in Cloud computing, there is now a growing focus on implementing data sharing capabilities in the Cloud. With the ability to share data via the Cloud, the number of benefits increases multifold. As businesses and organizations are now outsourcing data and operations to the Cloud, they benefit further with the ability to share data between other businesses and organizations. Employees also benefit as they can share work and collaborate with other employees and can also continue working at home or any other place such as the library. They don’t need to worry about losing work as it is always in the Cloud. With social users, the ability to share files, including documents, photos and videos with other users provides great benefit to them.

When considering data sharing and collaboration, simple encryption techniques do not suffice, especially when considering key management. To enable secure and confidential data sharing and collaboration in the Cloud, there needs to first be proper key management in the Cloud.

However, the main problem with data sharing in the Cloud is the privacy and security issues. As discussed in Section. 2, the Cloud is open to many privacy and security attacks, which make many users wary of adopting Cloud technology for data sharing purposes.

4.1. Requirements of Data Sharing in the Cloud

To enable data sharing in the Cloud, it is imperative that only authorised users are able to get access to data stored in the Cloud. We summarise the ideal requirements of data sharing in the Cloud below.

- The data owner should be able to specify a group of users that are allowed to view his/her data
- Any member of the group should gain access to the data anytime without the data owner’s intervention.
- No other user, other than the data owner and the members of the group, should gain access to the data, including the Cloud Service Provider.
- The data owner should be able to revoke access to data for any member of the group.
- The data owner should be able to add members to the group.
- No member of the group should be allowed to revoke rights of other members of the group or join new users to the group.
- The data owner should be able to specify who has read/write permissions on the data owner’s files.

Achieving privacy and security requirements in the Cloud architecture can go a long way to attracting large numbers of users to adopting and embracing Cloud technology.

- ***Data Confidentiality***: Unauthorized users (including the Cloud), should not be able to access data at any given time. Data should remain confidential in transit, at rest and on backup media. Only authorized users should be able to gain access to data.
- ***User revocation***: When a user is revoked access rights to data, that user should not be able to gain access to the data at any given time. Ideally, user revocation should not affect other authorized users in the group for efficiency purposes.
- ***Scalable and Efficient***: Since the number of Cloud users tends to be extremely large and at times unpredictable as users join and leave, it is imperative that the system maintain efficiency as well as be scalability.
- ***Collusion between entities***: When considering data sharing methodologies in the Cloud, it is vital that even when certain entities collude, they should still not be able to access any of the data without the data owner’s permission. Earlier works of literature on data sharing did not consider this problem, however collusion between entities can never be written off as an unlikely event.

4.2. Need for Key Management in Cloud

Encryption provides data protection while key management enables access to protected data. It is strongly recommended to encrypt data in transit over networks, at rest, and on backup media. In particular, data encryption at rest (e.g., for long-term archival storage) can avoid the risk of malicious cloud service providers or malicious multi-tenants abuse. At the same time, secure key stores (including key backup and recoverability) and access to key stores must be securely implemented since improper (or access to) key storage could lead to the compromise of all encrypted data. Key management is anything you do with a key except encryption and decryption and covers the creation/deletion of keys, activation/deactivation of keys, transportation of keys, storage of keys and so on. Most Cloud service provider’s provide basic key encryption schemes for protecting data or may leave it to the user to encrypt their own data. Both encryption and key management are very important to help secure applications and data stored in the Cloud. Requirements of effective key management are discuss below.

- ***Secure key stores***: The key stores themselves must be protected from malicious users. If a malicious user gains access to the keys, they will then be able to access any encrypted data the key is corresponded to. Hence the key stores themselves must be protected in storage, in transit and on backup media.
- ***Access to key stores***: Access to the key stores should be limited to the users that have the rights to access data. Separation of roles should be used to help control access. The entity that uses a given key should not be the entity that stores the key.
- ***Key backup and recoverability***: Keys need secure backup and recovery solutions. Loss of keys, although effective for destroying access to data, can be highly devastating to a business and Cloud providers need to ensure that keys aren't lost through backup and recovery mechanisms.

4.3. Identity and Access Management

Secure management of identity and access control is a critical factor to prevent account and service hijacking. It is strongly recommended to prohibit sharing of account credentials, to leverage strong (multi-factor) authentication if possible, and to consider delegated authentication and managing trust across all types of cloud services. Access control is a security features that control how users and systems communicate and interact with one another. Access means flow of information between subject and object. Subject is an active entity that requests access to an object or the data in an object whereas object is a passive entity that contains information. There are broadly three types of access control:

- Role Based Access Control (RBAC),
- User Based Access Control (UBAC), and
- Attribute Based Access Control (ABAC).

In UBAC, the access control list (ACL) contains the list of users who are authorized to access data. This is not feasible in clouds where there are many users. In RBAC, users are classified based on their individual roles. Data can be accessed by users who have matching roles. The roles are defined by the system. For example, only faculty members and senior secretaries might have access to data but not the junior secretaries. ABAC is more extended in scope, in which users are given attributes, and the data has attached access policy. Only users with valid set of attributes, satisfying the access policy, can access the data.

An area where access control is widely being used is health care. Clouds are being used to store sensitive information about patients to enable access to medical professionals, hospital staff, researchers, and policy makers. It is important to control the access of data so that only authorized users can access the data. Using ABE, the records are encrypted under some access policy and stored in the cloud. Users are given sets of attributes and corresponding keys.

Access control is also gaining importance in online social networking where users (members) store their personal information, pictures, and videos and share them with selected groups of users or communities they belong to.

5. The Proposed Scheme

In this section cryptographic architecture model is proposed in Cloud environment as shown in Fig. 1. In this technique a user or sender interacts with three elements: Data, Attribute and Key. These elements are used to encrypt the message. Encrypted messages are also known as cipher –text. Now cipher-text is sent to the receiver via cloud or network. There may be the process of verification of message if needed. To get the message in plain-text, cipher-text is decrypted by using key and attributes values in public-key encryption fashion.

5.1. Assumptions

In this paper, assumptions are made as follows:

1) The cloud is honest-but-curious, which means that the cloud administrators can be interested in viewing user's content, but cannot modify it. Honest-but-curious model of adversaries do not tamper with data so that they can keep the system functioning normally and remain undetected.

2) Users can have either read or write or both accesses to a file stored in the cloud.

3) All communications between users/clouds are secured by Secure Shell Protocol, SSH.

5.2. Mathematical Background

Bilinear pairings on elliptic curves is used. Let G be a cyclic group of prime order q generated by g. Let G_T be a group of order q. We can define the map $e : G \times G \rightarrow \mathrm{G}_T$. The map satisfies the following properties:

1) $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G$ and $a,\ b \in Zq$, $Zq = \{0,\ 1,\ 2,\ \ldots, q-\ 1\}$.

2) Non-degenerate: $e(g,\ g) = 1$.

Bilinear pairing on elliptic curves groups is used. The choice of curve is an important consideration because it determines the complexity of pairing operations.

5.3. Formats of Access Policies

Access policies can be in any of the following formats: 1) Boolean functions of attributes, 2) Linear Secret Sharing Scheme (LSSS) matrix, or 3) Monotone span programs. Any access structure can be converted into a Boolean function. An example of a Boolean function is $((a_1 \wedge a_2 \wedge a_3) \vee (a_4 \wedge a_5)) \wedge (a_6 \vee a_7))$, where a_1, a_2, . . ., a_7 are attributes.

Let $Y : \{0,\ 1\}^{\mathrm{n}} \rightarrow \{0,\ 1\}$ be a monotone Boolean function [25]. A monotone span program for Y over a field F is an $l \times t$ matrix M with entries in F, along with a labeling function $a : [\mathrm{l}] \rightarrow [\mathrm{n}]$ that associates each row of M with an input variable of Y, such that, for every $(\mathrm{x}_1,\ \mathrm{x}_2 \ldots, \mathrm{x}_\mathrm{n}) \in \{0,\ 1\}^{\mathrm{n}}$, the following condition is satisfied:

$Y(\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_\mathrm{n}) = 1 \Leftrightarrow \exists v \in \mathrm{F}^{1 \times l} : v\mathrm{M} = [1, 0, 0, \ldots, 0]$

and $(\forall i : x_{a(i)} = 0 \Rightarrow v_i = 0)$

In other words, $Y(x_1, x_2, \ldots, x_n) = 1$ if and only if the rows of M indexed by $\{i \mid x_{a(i)} = 1\}$ span the vector [1, 0, 0, . . ., 0].

For handling the fault tolerance, there should be several Key Distribution Centre (KDC) located at multiple servers over the cloud. It help in parallel encryption and distributed processing of message. Attributed should also be distributed at multiple servers. The use of public-key encryption gives more flexibility for the cloud applications. For example, in enterprise settings, every employee can upload encrypted data on the cloud storage server without the knowledge of the company's master-secret key. In Cloud Computing, outsourced data might not only be accessed but also updated frequently by users for various application purposes. Hence, supporting data dynamics for privacy-preserving public risk auditing is also of paramount importance. Now we show how our main scheme can be adapted to build upon the existing work to support data dynamics, including block level operations of modification, deletion and insertion. Data dynamics support is achieved by replacing the index information i with the mi in the computation of block signatures and using the classic data structure-Merkle hash tree (MHT) for the underlying block sequence enforcement.

There are three users, a creator, a reader and writer. Creator receives a token γ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting her id (like health/social insurance number), the trustee gives her a token γ. There are multiple KDCs, 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. 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.

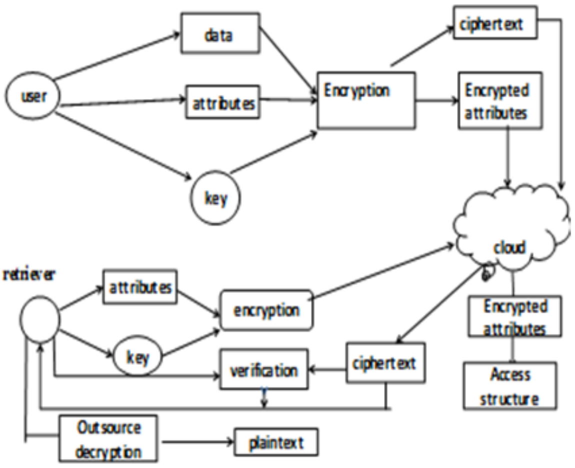

Figure 1. Cryptosystem Architecture Model for Secure Data Sharing over Cloud

5.4. System Module for Encrypted Data Sharing

The design of proposed scheme is inspired from the collusion-resistant broadcast encryption scheme proposed by Boneh et al. [10]. Although their scheme supports constant-size secret keys, every key only has the power for decrypting cipher-texts associated to a particular index. We thus need to devise a new Extract algorithm and the corresponding Decrypt algorithm. Proposed encryption scheme consists of five polynomial-time algorithms as follows:

- Setup Phase
- Key Generation Phase
- Encryption Phase
- Extract Phase
- Decryption Phase

The data owner establishes the public system parameter via Setup and generates a public/master-secret key pair via Key Generation. 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 delegates 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 Decrypt. Processes of modules are shown in Figure 1 and Figure 2.

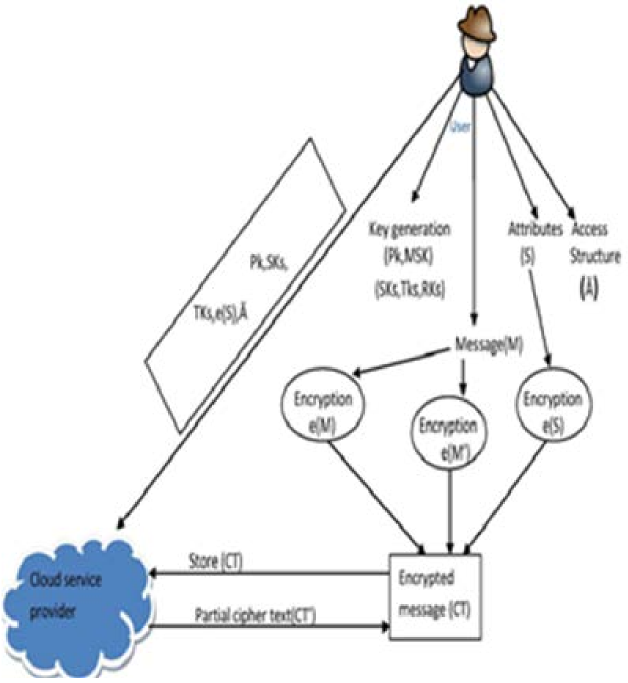

Figure 2. Cryptographic Module for secure Data Sharing over cloud

Suppose sender wants to share his data m_1, $m_{2,}$ …… m_i on the server. First Setup $(1^{\lambda}, n)$ is performed to get ***param*** and execute ***KeyGeneration*** phase to get the public/master-secret key pair (PK; MSK). The system parameter ***param*** and public-key PK can be made public and master-secret key MSK should be kept secret by sender. Anyone (including sender) can then encrypt each m_i by CT_i = ***Encrypt(PK, M, A)***. The encrypted data are uploaded to the server. With ***param*** and PK, people who cooperate with sender can update sender’s data on the server. Once sender is willing to share a set S of his data with receiver, he can compute the aggregate key KS for receicer by performing ***Extract(MSK, S)***. Since KS is just a constantsize key, it is easy to be sent to receiver via a secure e-mail.

5.4.1. Setup Phase

The setup algorithm $\left(1^{\lambda}, \mathrm{n}\right)$ takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK. executed by the data owner to setup an account on an untrusted server. On input a security level parameter 1^{λ} 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. Randomly pick a bilinear group G of prime order p where $2^{\lambda} \leq p \leq 2^{\lambda+1}$, a generator $g \in G$ and $\alpha \in R\ Zp$. Compute $g_i = g^{\alpha \mathrm{i}} \in G$ for $i = 1, \ldots..n, n + 2, \ldots.. 2n$. Output the system parameter as ***param*** = < g, g_1, ….. g_n, g_{n+2}, g_{2n}>

5.4.2. Key Generation Phase

KeyGeneration (MSK,S). The key generation algorithm takes as input the master key MK and a set of attributes S that describe the key. Pick $\gamma \in_R Zp$. It outputs a private key SK. Executed by the data owner, to randomly generate a public/master-secret key pair $\left(\mathrm{PK} = g^{\gamma},\ \mathrm{MSK} = \ \gamma\right)$. The sizes of ciphertext, public-key, master-secret key and aggregate key in proposed scheme are all of constant size. The public system parameter has size linear in the number of ciphertext classes, but only a small part of it is needed each time and it can be fetched on demand from large (but non-confidential) cloud storage.

5.4.3. Encryption Phase

Encrypt (PK, M, A). The encryption algorithm takes as input the public parameters PK, a message $\mathrm{M} \in \mathrm{GT}$, and an access structure A over the universe of attributes. The algorithm will encrypt *M* and produce a ciphertext *CT* such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message. We will assume that the ciphertext implicitly contains *A*. It is executed by anyone who wants to encrypt data. Users encrypt a message not only under a public-key, but also under an identifier of ciphertext called class. That means the ciphertexts are further categorized into different classes.

5.4.4. Extract Phase

Extract $\left(\mathrm{MSK} = \gamma, S\right)$. It is executed by the data owner for delegating the decrypting power for a certain set of cipher-text classes to a delegate. 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 K_S where $K_S = \Pi_{j \in S}\, g^{\gamma}{}_{n+1-j}$. 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.

5.4.5. Decryption Phase

Decrypt (PK, CT=(CT_{i-2}, CT_{i-1}, CT_i), K_S). The decryption algorithm takes as input the public parameters PK, a cipher-text CT, which contains an access policy A, and an aggregate-key K_S, which is generated by ***Extract***. If the set S of attributes satisfies the access structure A then the algorithm will decrypt the cipher-text and return a message *M* where

$$\mathrm{M} = \mathrm{CT}_{\mathrm{i}}\ .\mathrm{e}(\mathrm{K}_{\mathrm{S}}.\ \Pi_{\mathrm{j} \in \mathrm{S},\ \mathrm{j} \neq \mathrm{i}}\ \mathrm{g}_{\mathrm{n+1-j+i}},\ \mathrm{CT}_{\mathrm{i-2}})/(\ \Pi_{\mathrm{j} \in \mathrm{S}}\ \mathrm{g}_{\mathrm{n+1-j},}\ \mathrm{CT}_{\mathrm{i-1}})$$

6. Conclusion and Future Work

Efficient search on encrypted data is also an important concern in clouds. However, security concern has become the biggest obstacle to adoption of cloud because all information and data (including reallocation of data, and security management level) are completely under the control of cloud service providers. This paper presented encrypted data sharing methodology for cloud environment which is decentralized and prevent from many attacks. Key Distribution is done in decentralized way. Public-key based on encryption is used. In cloud storage, the number of cipher-texts usually grows rapidly. So we have to reserve enough cipher-text classes for the future extension. Proposed technique does not authenticate users, who want to remain anonymous while accessing the cloud.

In the future, work can be done on distributed and scalable Big Data sharing methodology with anonymous authentication on clouds.

References

[1] L. Hardesty, *“Secure computers aren’t so secure,” MIT press*, 2009, http://www.physorg.com/news176107396.html.

[2] D. X. Song, D. Wagner, A. Perrig. *“Practical techniques for searches on encrypted data”. Proceedings of the IEEE Symposium on Security and Privacy*, 2000, pp. 44-55.

[3] D. Boneh, G. D. Crescenzo, R. Ostrovsky, G. Persiano. “*Public key encryption with keyword search*”*. Advances in Cryptology-EUROCRYPT*’04, 2004, LNCS 3027, Springer, pp. 506-522.

[4] 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. 89-98.

[5] 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.

[6] S. S. M. Chow, J. Weng, Y. Yang, and R. H. Deng, *“Efficient Unidirectional Proxy Re-Encryption,” in Progress in Cryptology-AFRICACRYPT* 2010, ser. LNCS, vol. 6055. Springer, 2010, pp. 316-332.

[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, *“Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage,” ACM Transactions on Information and System Security (TISSEC)*, vol. 9, no. 1, pp. 1-30, 2006.

[8] Seung-Hyun Seo, Nabeel, M, Bertino, E. and Xiaoyu Ding, *“An Efficient Certificateless Encryption for Secure Data Sharing in*

Public Clouds” *IEEE Transactions on Knowledge and Data Engineering*, Volume: 26, Issue: 9, pp. 2107-2119, 05 August 2013

[9] Ruhr (2011) “*Cloud computing: Gaps in the cloud*”. NewsRx Health Sci.

[10] D. Boneh, C. Gentry, and B. Waters, “*Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys,*” in Proceedings of Advances in Cryptology-CRYPTO ’05, ser. LNCS, vol. 3621. Springer, 2005, pp. 258-275.