IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

A Privacy Preserving cloud Storage Framework by using Server Re-encryption Mechanism (SRM)

Nagaraju. P[1*], Nagamalleswara Rao. N[2], Vinod Kumar. Ch.R [3]

[1] Dept. of CSE, GMR Institute of Technology, Rajam, India
[2] Dept. of IT, RVR&JC College of Engineering, Guntur, India
[3] Dept. of IT, GMR Institute of Technology, Rajam, India

[*]*Corresponding Author: nagaraj528@gmail.com, Tel.: +91-99511-61129*

Available online at: www.ijcseonline.org

Accepted: 20/July/2018, Published: 31/July/2018

Abstract— Cloud computing is an emerging technology the way that organizations manage their data, owing to its attractive features such as robustness, low cost, and ubiquitous nature. However, privacy concerns arise whenever sensitive data is out-sourced to the cloud where the data is processed and stored. The fact that users no longer have physical possession of the outsourced data makes it a formidable task to achieve the data confidentiality and integrity. As the data, in most cases encrypted, have to be not only stored, but also processed in clouds, the cryptography-based data confidentiality and integrity protection approaches are not adequate to satisfy the security requirements. Proxy re-encryption serves as a promising solution to secure the data sharing in the cloud computing. It enables a data owner to encrypt shared data in cloud under its own public key, which is further transformed by a semi-trusted cloud server into an encryption intended for the legitimate recipient for access control. To achieve a flexible and fine-grained access control on the outsourced data in cloud environment, in this paper we design a functional encryption system.

Keywords— Cloud Computing Component, Privacy-Preserving methods, Privacy-Preserving Algorithms**.**

I. INTRODUCTION

Cloud storage can provide on-demand, scalable and QoS guaranteed storage resource, and users can operate their data anytime and anywhere by a simple and function limited device which can be connected with Internet to visit the cloud.

Facing the powerful and appealing advantages of cloud storage, however, a lot of people and companies are hesitant to put their data in cloud. The main reason is that people and companies are afraid of loss of control on their data. Without appropriate privacy solution for cloud become a large failure. There is a lot of research techniques made to provide security. Privacy means that the person to be free from all interference. Privacy control allows the person to maintain a degree of intimacy. Privacy is the protection for the truthful use of personal information of cloud user. Privacy problems have become very challenge in a cloud computing environment. So we use several techniques to protect the confidentiality of the data.

Because of the cloud's nature as a shared resource, identity management, privacy and access control are of particular concern. With so many organizations using cloud computing and associated cloud providers for data operations, proper security in these and other potentially vulnerable areas have become a priority for organizations contracting with a cloud computing provider. Cloud computing security processes should address the security controls the cloud provider will incorporate to maintain the customer's data security, privacy and compliance with necessary regulations. The processes will also likely include a business continuity and data backup plan in the case of a cloud security breach.

And there are some incidents of data leakage and losing which verify people's fears: a cloud storage-provider named LinkUp(MediaMax) went out of business last year after losing 45% of stored client data due to an error of a system administrator [1]; in 2007, criminals targeted the prominent cloud service provider Salesforce.com, and succeeded in stealing customer emails and addresses using a phishing attack [2]; Google's Docs was visited by unauthorized attacker, which caused data leakage [3]. Therefore, to be sustainable, in-depth development, cloud storage must

address the privacy concern, efficient data storage and access.

There have been many works on outsourced storage. [4] Developed a privacy-preserving electronic health record system. Based on symmetric and asymmetric encryption, it designed two key derivation schemes and compared the advantages and disadvantages of both. But it didn't consider the effects of change of user access right and the dynamic operations of data which would influence the effectiveness of key derivation.

Cloud computing security or, more simply, cloud security refers to a broad set of policies, technologies, and controls deployed to protect data, applications, and the associated infrastructure of cloud computing. It is a sub-domain of computer security, network security, and, more broadly, information security.

A. Data security

A number of security threats are associated with cloud data services: not only traditional security threats, such as network eavesdropping, illegal invasion, and denial of service attacks, but also specific cloud computing threats, such as side channel attacks, virtualization vulnerabilities, and abuse of cloud services. The following security requirements limit the threats.[14]

1) Confidentiality

Data confidentiality is the property that data contents are not made available or disclosed to illegal users. Outsourced data is stored in a cloud and out of the owners' direct control. Only authorized users can access the sensitive data while others, including CSPs, should not gain any information of the data. Meanwhile, data owners expect to fully utilize cloud data services, e.g., data search, data computation, and data sharing, without the leakage of the data contents to CSPs or other adversaries.

2) Access controllability

Access controllability means that a data owner can perform the selective restriction of access to his data outsourced to cloud. Legal users can be authorized by the owner to access the data, while others can't access it without permissions. Further, it is desirable to enforce fine-grained access control to the outsourced data, i.e., different users should be granted different access privileges with regard to different data pieces. The access authorization must be controlled only by the owner in untrusted cloud environments.

3) Integrity

Data integrity demands maintaining and assuring the accuracy and completeness of data. A data owner always expects that his data in a cloud can be stored correctly and trustworthily. It means that the data should not be illegally tampered, improperly modified, deliberately deleted, or maliciously fabricated. If any undesirable operations corrupt or delete the data, the owner should be able to detect the corruption or loss. Further, when a portion of the outsourced data is corrupted or lost, it can still be retrieved by the data users.

II. BACKGROUND

A. Security issues associated with the cloud

Cloud computing and storage provides users with capabilities to store and process their data in third-party data centers. Organizations use the cloud in a variety of different service models (with acronyms such as SaaS, PaaS, and IaaS) and deployment models (private, public, hybrid, and community). Security concerns associated with cloud computing fall into two broad categories: security issues faced by cloud providers (organizations providing software-, platform- , or infrastructure-as-a-service via the cloud) and security issues faced by their customers (companies or organizations who host applications or store data on the cloud).The responsibility is shared, however. The provider must ensure that their infrastructure is secure and that their clients' data and applications are protected, while the user must take measures to fortify their application and use strong passwords and authentication measures.

When an organization elects to store data or host applications on the public cloud, it loses its ability to have physical access to the servers hosting its information. As a result, potentially sensitive data is at risk from insider attacks. According to a recent Cloud Security Alliance Report, insider attacks are the sixth biggest threat in cloud computing.[4] Therefore, Cloud Service providers must ensure that thorough background checks are conducted for employees who have physical access to the servers in the data center. Additionally, data centers must be frequently monitored for suspicious activity.

In order to conserve resources, cut costs, and maintain efficiency, Cloud Service Providers often store more than one customer's data on the same server. As a result, there is a chance that one user's private data can be viewed by other users (possibly even competitors). To handle such sensitive situations, cloud service providers should ensure proper data isolation and logical storage segregation.[2]

B. Cloud security controls

Cloud security architecture is effective only if the correct defensive implementations are in place. Efficient cloud security architecture should recognize the issues that will arise with security management.[8] The security management addresses these issues with security controls. These controls are put in place to safeguard any weaknesses in the system and reduce the effect of an attack. While there are many

types of controls behind a cloud security architecture, they can usually be found in one of the following categories:[8]

Deterrent controls

These controls are intended to reduce attacks on a cloud system. Much like a warning sign on a fence or a property, deterrent controls typically reduce the threat level by informing potential attackers that there will be adverse consequences for them if they proceed. (Some consider them a subset of preventive controls.)

Preventive controls

Preventive controls strengthen the system against incidents, generally by reducing if not actually eliminating vulnerabilities. Strong authentication of cloud users, for instance, makes it less likely that unauthorized users can access cloud systems, and more likely that cloud users are positively identified.

Detective controls

Detective controls are intended to detect and react appropriately to any incidents that occur. In the event of an attack, a detective control will signal the preventative or corrective controls to address the issue.[8] System and network security monitoring, including intrusion detection and prevention arrangements, are typically employed to detect attacks on cloud systems and the supporting communications infrastructure.

Corrective controls

Corrective controls reduce the consequences of an incident, normally by limiting the damage. They come into effect during or after an incident. Restoring system backups in order to rebuild a compromised system is an example of a corrective control.

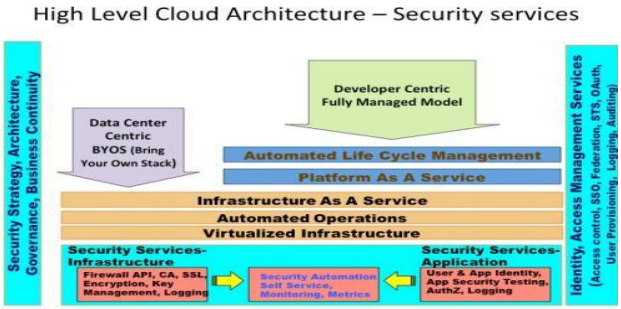

Fig.1: Cloud Security Architecture

C. Cloud Security Principles

Every enterprise has different levels of risk tolerance and this is demonstrated by the product development culture, new technology adoption, IT service delivery models, technology strategy, and investments made in the area of security tools and capabilities. When a business unit within an enterprise decides to leverage SaaS for business benefits, the technology architecture should lend itself to support that model. Additionally the security architecture should be aligned with the technology architecture and principles. Following is a sample of cloud security principles that an enterprise security architect needs to consider and customize:

- Services running in a cloud should follow the principles of least privileges.
- Isolation between various security zones should be guaranteed using layers of firewalls – Cloud firewall, hypervisor firewall, guest firewall and application container. Firewall policies in the cloud should comply with trust zone isolation standards based on data sensitivity.
- Applications should use end-to-end transport level encryption (SSL, TLS, IPSEC) to secure data in transit between applications deployed in the cloud as well as to the enterprise.
- Applications should externalize authentication and authorization to trusted security services. Single Sign-on should be supported using SAML 2.0.
- Data masking and encryption should be employed based on data sensitivity aligned with enterprise data classification standard.
- Applications in a trusted zone should be deployed on authorized enterprise standard VM images.
- Industry standard VPN protocols such as SSH, SSL and IPSEC should be employed when deploying virtual private cloud (VPC).
- Security monitoring in the cloud should be integrated with existing enterprise security monitoring tools using an API.

D. **Security and privacy**

Identity management

Every enterprise will have its own identity management system to control access to information and computing resources. Cloud providers either integrate the customer's identity management system into their own infrastructure, using federation or SSO technology, or a biometric-based identification system,[1] or provide an identity management system of their own.[13] CloudID,[1] for instance, provides privacy-preserving cloud-based and cross-enterprise biometric identification. It links the confidential information

of the users to their biometrics and stores it in an encrypted fashion. Making use of a searchable encryption technique, biometric identification is performed in encrypted domain to make sure that the cloud provider or potential attackers do not gain access to any sensitive data or even the contents of the individual queries.[1]

Physical security

Cloud service providers physically secure the IT hardware (servers, routers, cables etc.) against unauthorized access, interference, theft, fires, floods etc. and ensure that essential supplies (such as electricity) are sufficiently robust to minimize the possibility of disruption. This is normally achieved by serving cloud applications from 'world-class' (i.e. professionally specified, designed, constructed, managed, monitored and maintained) data centres.

Personnel security

Various information security concerns relating to the IT and other professionals associated with cloud services are typically handled through pre-, para- and post-employment activities such as security screening potential recruits, security awareness and training programs, proactive.

Privacy

Providers ensure that all critical data (credit card numbers, for example) are masked or encrypted and that only authorized users have access to data in its entirety. Moreover, digital identities and credentials must be protected as should any data that the provider collects or produces about customer activity in the cloud.

III. EFFECTIVE ENCRYPTION

In cryptography, encryption is the process of encoding a message or information in such a way that only authorized parties can access it. Encryption does not itself prevent interference, but denies the intelligible content to a would-be interceptor. In an encryption scheme, the intended information or message, referred to as plaintext, is encrypted using an encryption algorithm, generating cipher text that can only be read if decrypted. For technical reasons, an encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. It is in principle possible to decrypt the message without possessing the key, but, for a well-designed encryption scheme, considerable computational resources and skills are required. An authorized recipient can easily decrypt the message with the key provided by the originator to recipients but not to unauthorized users.

Modern cryptographic systems include symmetric-key algorithms (such as DES and AES) and public-key algorithms (such as RSA). Symmetric-key algorithms use a single shared key; keeping data secret requires keeping this key secret. Public-key algorithms use a public key and a private key. The public key is made available to anyone (often by means of a digital certificate). A sender encrypts data with the public key; only the holder of the private key can decrypt this data.Since public-key algorithms tend to be much slower than symmetric-key algorithms, modern systems such as TLS and SSH use a combination of the two: one party receives the other's public key, and encrypts a small piece of data (either a symmetric key or some data used to generate it). The remainder of the conversation uses a (typically faster) symmetric-key algorithm for encryption.

Many modern protocols are designed to have forward secrecy, which requires generating a fresh new shared key for each session. Classic cryptosystems invariably generate two identical keys at one end of the communication link and somehow transport one of the keys to the other end of the link. However, it simplifies key management to use Diffie–Hellman key exchange instead.

Symmetric Keys

The symmetric encryption classes require a key and a new initialization vector (IV) to encrypt and decrypt data. Whenever you create a new instance of one of the managed symmetric cryptographic classes using the default constructor, a new key and IV are automatically created. Anyone that you allow to decrypt your data must possess the same key and IV and use the same algorithm. Generally, a new key and IV should be created for every session and neither the key nor IV should be stored for use in a later session.

To communicate a symmetric key and IV to a remote party, you would usually encrypt the symmetric key by using asymmetric encryption. Sending the key across an insecure network without encrypting it is unsafe, because anyone who intercepts the key and IV can then decrypts your data. For more information about exchanging data by using encryption, see Creating a Cryptographic Scheme.

A.

B. Creating a Cryptographic Scheme

A simple cryptographic scheme for encrypting and decrypting data might specify the following steps:

1. Each party generates a public/private key pair.
2. The parties exchange their public keys.
3. Each party generates a secret key for Triple DES encryption, for example, and encrypts the newly created key using the other's public key.
4. Each party sends the data to the other and combines the other's secret key with its own, in a particular order, to create a new secret key.

5. The parties then initiate a conversation using symmetric encryption.

IV. PRIVACY PRESERVING METHODS

Several methods have been put forward to tackle this issue of privacy preserving. It is important, that the privacy has to be preserved anytime and anywhere.

A) Anonymity-based Method

Jiang Wang et al. put forward an Anonymity-based method to achieve and preserve privacy in cloud computing [1]. The anonymity algorithm processes the data and anonymizes all or some information before releasing it in the cloud milieu. Tis approach differs from the traditional cryptography technology for preserving user's privacy as it gets rid of key management and thus it stands simple and flexible.

B) privacy-preserving Architecture

Architecture for database storage [2] in cloud is proposed in this paper, which preserves the privacy of users' data. Tis approach prevents the risk of both external and internal attacks to the outsourced data. The main architectural elements are the user interface, user engine, rule engine and the cloud database.

C) privacy-preserved Access Control

Miao Zhou et al. [3] considered the privacy of users in the cloud environment and proposed a flexible method of access control. Each cloud user is linked with certain attributes, which determines their access rights. The paper propounded a two-tier encryption model in which the base phase and surface phase builds up the two tiers of the model respectively. At the first phase, the data owner performs local attribute-based encryption on the data that has to be outsourced. The surface phase on the other hand is performed by the cloud servers, after the initialization done by the cloud data owner. This phase implements the Server re-encryption mechanism (SRM). The SRM dynamically re-encrypts the encrypted data in the cloud, when the owner of that data requests. The request for SRM arises either when a new user has to be created or an existing user has to be repealed. Tough the re-encryption takes place in cloud server, the privacy of users data is not compromised as the access policies remains hidden to the cloud servers. Tus, in this paper privacy of data is preserved by providing full access control to the owner of the data and by disallowing the cloud provider to gain knowledge about the data.

V. DESIGN GOAL AND MAIN IDEA

i. Design Goal

Our main design goal is to help the data owner achieve a flexible and fine-grained access control on the outsourced data in clouds. We want to prevent the cloud provider from learning the data contents and user information, and allow the data owner to define users who can get access to data files. Specifically, we do not want user's creation or revocation affects other users, namely other users do not need to update their secret keys. In addition, our scheme also features policy-hiding and is secure against to the collusion attacks from malicious users.

ii. Main Idea

Considering to achieve a flexible and fine-grained access control on the outsourced data in cloud environment, we design a functional encryption system named as encryption, which proposes attribute-set-based encryption as a base tier and server re-encryption mechanism as a surface tier.

As any access structure can be represented as an access tree T , we associate each data file with an access structure from where different attribute sets can be generated. Take the example, which gives an instance of access structure and attribute sets that can be generated. We describe a data file that can be accessed by CS staff, CS students from class one, or CS students from class two.

Specifically, the privileged users hold an attribute set as one of the following: (University, CS, Student, Class.1), (University, CS, Student, Class.2), or (University, CS, Staff). These attribute sets are used to generate the private keys of the privileged users. Each attribute in the attribute sets is given a chosen value Si. A student in class one who holds an attribute set (University, CS, Student, Class.1) can decrypt this data file using his private key computed as h(Suni+Scs+Sstudent+Sclass.1). Analogically, a professor in CS department decrypts the data file by using his secret key computed as h(Suni + Scs + Sstaf f). Our construction of attribute-set based encryption allows different users to decrypt the data file with the corresponding secret keys, which does not encrypt access structure into a cipher-text.

However, we cannot consider this access control as flexible when the attribute-set based scheme runs alone. One challenging issue here, caused by user revocation, is to require a system-wide private keys update against the expedience of users. Thus, we propose a server re-encryption mechanism combining with the attribute-set based scheme, to eliminate users from updating their private keys when a user joins or leaves. In addition, in scenarios involving potentially huge sets of data files of considerable size, re-encryption and re-transmission by data owner may not be acceptable. As the cloud servers are assumed to be more powerful, the task of data file re-encryption is done by the cloud server without disclosing file content and attribute list.

iii. The Encryption System

We assume that the two-tier encryption system consists of the following three parties: the Data Owner who is also the cloud user, the Cloud Provider who provides cloud servers, and many data consumers that we refer them as users for brevity. The date owner encrypted the data files first before sent them into the cloud and built a server re-encryption mechanism (SRM) that works as a second level dynamic password generator. We will introduce each tier model of this system in the following chapters in detail, as base model and surface model respectively.

• Base Phase: The data owner at local, before outsourcing data into the clouds, performs a attribute-set based encryption on the data files according to the access policies.

• Surface Phase: The cloud server performs the dynamic encryption operations over the encrypted data files, when receiving request messages from the data owner.

To access the data files stored in cloud, users download the ones of their interest from the cloud provider and decrypt them with their own decryption key. The data owner is not required to be always online unless there are necessary changes in the access structure that caused by the user grant or revocation. As the servers in cloud are assumed to have abundant storage capacity and computation power, we transfer the task of data file re-encryption to the cloud servers without the leakage of data file contents and any information about the users, including the number of users and users' ID list. The server re-encryption mechanism which ran on the servers in cloud will handle the data file re-encrypt task in a imperceptible way without requiring the users to re-key their decryption keys for re-encrypted data files.

5.1. BASE PHASE

A. Access Structure and Attribute Sets Consistently with the data outsourcing scenario, we assume the existence of several attribute sets in the system and the data owner therefore defines access structures for users to access the outsourced data. These access structures are abstracted in a down-top manner to generate authorizations that can be modeled via an access matrix. Each row of the access matrix is set to one privilege attribute set for a specific user or users, with a generated secret key for corresponding data file.

B. Definition of Attribute-set Based Encryption

We first give formal definition of our attribute-set based encryption. An attribute-set based encryption scheme consists of four probabilistic polynomial-time algorithms: Setup, KeyGen, Encryption and Decryption.

C. Main Construction

The main construction of the base phase is performed by the data owner, before outsourcing the data item into the cloud. It enforces policy hidden attribute-set based encryption on the data files according to the access policies. We borrow the polynomial function introduced in [8].

Setup(T, A) → (SK1, SK2, ..., SKm, Matrix(A)). In the basic construction, it chooses two large prime numbers p and q such that q|p - 1, and chooses a generator element g ∈ Zp ∗ of order q. For each attribute ai ∈ A, it chooses a random value si ∈ Zp. With the access structure T, x root may have numxroot values the different attribute sets, that is, m secret keys associated with attribute sets A ˜1 to A ˜m.

By generating the Matrix(A), we can compute these secret keys as SKj = h(ai⊆$\boldsymbol{A}$ ˜i siai), 1 ≤ i ≤ n, 1 ≤ j ≤ m. KeyGen(SK1, SK2, ..., SKm) → PK. On input the secret keys, the key generation algorithm constructs a polynomial function as

$$f(x) = \prod_{j=1}^{m}(x - SK_j) \equiv \sum_{j=0}^{m} a_j x^j \mod q,$$

where a_j are coefficients. It generates the encryption key PK = (g^{a0}, g^{a1}, ..., g^{am}).

Encryption(PK, M) → CT. The encryption algorithm then chooses random r ∈ Z_q and generators h ∈ Z_p ∗, and outputs a cipher-text (c_1, c_2):

$$CT = \begin{cases} c_1 \leftarrow (h^r \cdot g_0^r, g_1^r, ..., g_m^r) \\ c_2 = M \cdot h^r \end{cases}$$

Decrypt(CT, SKj) → M. For each decryption key SKj, the decryption algorithm computes

$$\begin{cases} h^r \leftarrow h^r \cdot g_0^r \cdot g_1^{SK_j r} \cdot ... \cdot g_m^{SK_j{}^m r} \\ M = c_2/h^r \end{cases}$$

5.2. SURFACE PHASE

The surface phase is initialized by the data owner and performed by the cloud servers over the outsourced data files. It enforces the dynamic encryption operations over the encrypted data files, when receiving request messages from the data owner. The request messages contain new encryption keys for cloud servers as input. Combining with the base phase, the surface phase allows the server to conduct re-encryption for the users.

A. Server re-encryption Mechanism

Our Server re-encryption mechanism (SRM) is a mechanism that runs by the cloud server, especially for new user creation or user revocation. This mechanism proceeds in rounds as a state transition diagram, shown as Figure 3. During each round, the server listens to the request from the data owner with an encrypted data file index CTi corresponding to a new public key PK∗, and then performs re-encryption on CTi with PK∗ and associate the re-encrypted ciphertext CT ∗ with index i. Finally SRM updates the re-encrypted data CT ∗ to replace the previous CT and record this replacement in the system.

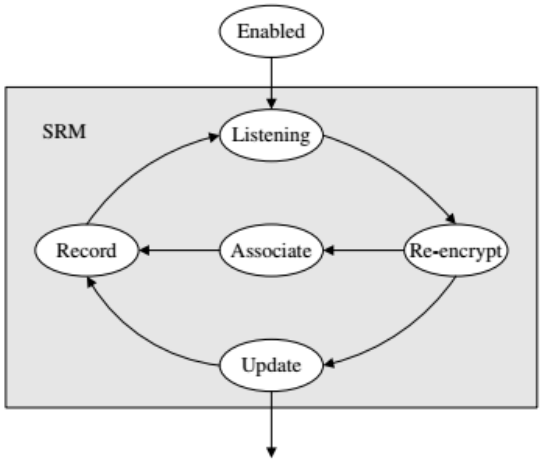

Fig.2: The state transition diagram of SRM.

The main difference between our server re-encryption mechanism and proxy re-encryption is that the proxy re-encryption allows a proxy to transform a ciphertext under a delegator's public-key into a delegatee's ciphertext on the same message by using some additional information, and the re-encryption key is generated by the decryptor. However, in a server re-encryption mechanism, the re-encryption key is generated by the encryptor, i.e., the data owner.

A weaker re-encryption scheme is one in which the proxy possesses both parties' keys simultaneously. One key decrypts a plaintext, while the other encrypts it. Since the goal of many proxy re-encryption schemes is to avoid revealing either of the keys or the underlying plaintext to the proxy, this method is not ideal.

Defining functions

Proxy re-encryption schemes are similar to traditional symmetric or asymmetric encryption schemes, with the addition of two functions:

- **Delegation** – allows a message recipient (keyholder) to generate a re-encryption key based on his secret key and the key of the delegated user. This re-encryption key is used by the proxy as input to the re-encryption function, which is executed by the proxy to translate ciphertexts to the delegated user's key. Asymmetric proxy re-encryption schemes come in bi-directional and uni-directional varieties.

  - In a bi-directional scheme, the re-encryption scheme is reversible—that is, the re-encryption key can be used to translate messages from Bob to Charlie, as well as from Charlie to Bob. This can have various security consequences, depending on the application. One notable characteristic of bi-directional schemes is that both the delegator and delegated party (e.g., Charlie and Bob) must combine their secret keys to produce the re-encryption key.

  - A uni-directional scheme is effectively one-way; messages can be re-encrypted from Bob to Charlie, but not the reverse. Uni-directional schemes can be constructed such that the delegated party need not reveal its secret key. For example, Bob could delegate to Charlie by combining his secret key with Charlie's public key.

Transitivity – Transitive proxy re-encryption schemes allow for a cipher-text to be re-encrypted an unlimited number of times. For example, a cipher-text might be re-encrypted from Bob to Charlie, and then again from Charlie to David and so on. Non-transitive schemes allow for only one (or a limited number) of re-encryptions on a given cipher-text. Currently, there is no known uni-directional, transitive proxy re-encryption scheme. It is an open problem as to whether such constructions are possible.

VI. MAIN CONSTRUCTION OF SRM

The main construction of SRM is composed of three algorithms: Setup, Re-encrypt, and Decrypt. Noted that the decrypted algorithm is not run on SRM, but can be operated on other side. i.e. on user's side. We present this algorithm as part of the main construction only for completeness. Setup(λ, CTi) → I. The setup algorithm takes in the security parameters λ and every index CTi of the outsourced data, generates an index list I as output.

Re-encrypt(CT, PK) → CT *. On receiving PK* as* $((h^v \cdot (g_0 *)^v, (g_1 *)^v, ..., (g_m *)^v), c_3)$, *where* $c_3 = h^{-r} \cdot h^v$, *output the re-encrypted cipher text as* (c^*_1, c^*_2):

$$CT = \begin{cases} c_1^* \leftarrow (h^v \cdot (g_0^*)^v, (g_1^*)^v, ..., (g_m^*)^v) \\ c_2^* = M \cdot h^v \end{cases}$$

Decrypt(CT^*, SK_j) → M. For each SK_j, the decryption algorithm computes M as in the base phase.

VII. CONCLUSION

There is an emerging trend towards data resourcing where data management is outsourced to clouds that provide storage capabilities and high-bandwidth distribution channels. In this paper we presented an encryption scheme for a two-tier system to achieve flexible and fine-grained access control in clouds, while delegating most of computation-intensive tasks to cloud servers without leaking private data. we design a functional encryption system, to achieve a flexible and fine-grained access control on the outsourced data in cloud environment.

References

[1] Wang J, Zhao Y et al, "Providing Privacy Preserving in cloud computing", International Conference on Test and Measurement, vol 2(2009), 213–216.

[2] Greveler U, Justus b et al., "A Privacy Preserving System for Cloud Computing", 11th IEEE International Conference on Computer and Information Technology, 648–653(2011).

[3] G. Ateniese, K. Benson, and S. Hohenberger, "KeyPrivate Proxy Re-Encryption", Proc. Topics in Cryptology, 2009, pp. 279-294.

[4] Assad Abbas and Samee U. Khan, Senior Member, IEEE, "A Review on the State-of-the-Art Privacy-Preserving Approaches in the e-Health Clouds", IEEE journal of biomedical and health informatics, vol. 18, no. 4, july 2014.

[5] Zhiguang Qin , Hu Xiong, Shikun Wu, and Jennifer Batamuliza, "A Survey of Proxy Re-Encryption for Secure Data Sharing in Cloud Computing", JOURNAL OF L ATEX CLASS FILES, VOL. 13, NO. 9, SEPTEMBER 2014.

[6] P. Angin et al., "An entity-centric approach for privacy and identity management in cloud computing," in Reliable Distributed Systems, 201029th IEEE Symposium on, New Delhi, IEEE. 2010, pp. 177–183.

[7] Zhou M, Mu Y et al, "Privacy-Preserved Access Control 3. for Cloud Computing", International Joint Conference of IEEE TrustCom-11/IEEE ICESS- 11/FCST-11, 83–90(2011).

[8] Mu, Y., Varadharajan, V., and Nguyen, K. Q. (1999) "Delegated Decryption", Proceedings of the 7th IMA International Conference on Cryptography and Coding, Cirencester, UK, pp.258-269.

[9] Amazon Elastic Computer Cloud (EC2). http://aws.amazon.com/ec2/

[10] Amazon Simple Storage Service (S3). http://aws.amazon.com/s3/

[11] RuWei Huang, Si Yu, Wei Zhuang, XiaoLin Gui, Design of Privacy-Preserving Cloud Storage Framework, 2010 Ninth International Conference on Grid and Cloud Computing

[12] Yu, S., Wang, C., Ren, K., Lou W, "Achieving secure, scalable, and fine-granied data access control in cloud computing", INFOCOM'10: 29th IEEE international conference on computer communications, San Diego, CA, USA, pp, 534-542(2010).

[13] W. Sharon Inbarani , G. Shenbagamoorthy, C. Kumar Charlie Paul, "Proxy Re-encryption Schemes for Data Storage Security in Cloud- A Survey", International Journal of Engineering Research & Technology (IJERT) Vol. 2 Issue 1, January- 2013 ISSN: 2278-0181.

[14] Nagaraju.P,Nagamalleswara rao .N.," A Detailed Study of Security Aspects in Cloud Computing", International Journal of Emerging Technologies in Engineering Research (IJETER),Volume 4, Issue 6, June (2016).

Authors Profile

Mr.P.Nagaraju pursed Bachelor of Technology from JNT University in 2007 and Master of Technology from Andhra University in year 2009. He is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Computational Science, GMR Institute of Technology, Rajam since 2012.He has published more than 5 research papers in reputed international journals. His main research work focuses on Cryptography Algorithms, Network Security, Cloud Security and Privacy, Big Data Analytics, IoT and Computational Intelligence based education. He has 9 years of teaching experience and 4 years of Research Experience.

Dr. N. Nagamalleswara Rao received his B.Tech degree in Computer Science and Engineering in 1991 and M.E. degree in CSE in 1993. He is in the teaching field since January 1993. He is currently working as Professor in Department of Information Tchnology, RVR&JC Colleg of enginring, Guntur since 2012 He completed his Doctorate degree in 2000. His area of specialization is Computer Algorithms, Compilers, Image Processing. He has more than 30 International publications in Journals and fifteen papers in International Conferences to his credit. He is a Senior Member of IEEE. He has 25 years of teaching experience and 20 years of Research Experience.

.

Mr.R.Ch.Vinodkumar pursed Bachelor of Computer Application from GITAM University in 2003 and Master of Technology from Andhra University in year 2007. He is currently pursuing Ph.D. and currently working as Assistant Professor in Department of Information Technology, GMR Institute of Technology, Rajam since 2012.He has published more than 10 research papers in reputed international journals and International Conferences. His main research work focuses on Software Engineering, Cloud Security and Privacy, Big Data Analytics, IoT and Image procssing. He has 12 years of teaching experience and 7 years of Research Experience.

