IJERMT International Journal of Engineering Research & Management Technology ISSN: 2348-4039
May-2015 Volume 2, Issue-3
Email: editor@ijermt.org www.ijermt.org

Enhancing the Integrity of Dynamic Data Stored in Cloud

Kamlesh Kumar Pathak
M.Tech Scholar
S.I.T.E
Meerut, India

Manik Chandra Pandey
Department of Computer Science
S.I.T.E
Meerut, India

ABSTRACT:

Cloud computing is a internet based computing which enables sharing of services. Many users place their data in the cloud, so correctness of data and security is a prime concern. This work studies the problem of ensuring the integrity and security of data storage in Cloud Computing. Security in cloud. To ensure the correctness of data, we consider the task of allowing a third party auditor (TPA), on behalf of the Cloud client[1], to monitor the integrity of the data stored in the cloud by utilizing public key and private key based. There is a probable risk while exchanging confidential data in the cloud computing environment. To preserve data integrity and confidentiality existing solutions adopt cryptographic methods and disclose data decryption keys only to authorized users. These solutions inevitably introduce a processing overhead on the data owner for key distribution and data management. These issues are addressed in this work by defining and enforcing access policies based on data attributes, and allowing the data owner to delegate most of the computation tasks involved in data access control to cloud servers [6] without affecting the integrity of data. The proposed work is accomplished by exploiting and uniquely combining some techniques of RSA [14] and proxy reencryption (PRE) [16]. This proposed scheme is implementing for simplifying the development efforts and provides direct support for security.

KEYWORDS:
Attribute-based encryption, Cloud Computing, Data dynamics, Proxy re-encryption, Monitoring, SaaS, TPA**.**

INTRODUCTION:

Cloud computing is a general term for anything that involves delivering hosted services over the internet. These services are broadly divided into three categories: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). A cloud service has three distinct characteristics that ifferentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic - a user can have as much or as little of a service as they want at any given time and the service is fully managed by the cloud service provider (the consumer needs nothing but a personal computer and Internet access).The advantage of cloud is cost savings. The prime disadvantage is security. Cloud computing is used by many software industries nowadays. Since the security is not provided in cloud, many companies adopt their unique security structure [18]. For eg) Amazon has its own security structure. Introducing a new and uniform security structure for all types of cloud is the problem we are going to tackle in this paper. Since the data placed in the cloud is accessible to everyone, security is not guaranteed. To ensure security, cryptographic techniques cannot be directly adopted. Sometimes the cloud service provider may hide the data corruptions to maintain the reputation. To avoid this problem, we introduce an effective third party auditor to monitor the user's outsourced data when needed. TPA is the third party auditor who will monitor the data of data owner or client. TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud are indeed intact. The released audit report helps the owner or client to evaluate the risk of their subscribed cloud data services and also it will be beneficial for the cloud As promising as it is, this paradigm also give rise to many new challenges for data security and access control. Data security, as it exists in many other applications, is among these challenges that would raise great concerns from users when they store sensitive information on cloud servers. These concerns originate from the fact that cloud servers are usually operated by commercial providers which are very likely to be outside of the trusted domain of the users. Data confidential against cloud servers is hence frequently desired when users outsource data for storage in the cloud. As a significant research area for system protection, data access control has been evolving in the past thirty years and various techniques have been developed to effectively implement access control, which allows flexibility in specifying differential access rights of individual users. Traditional access control architectures [6] usually assume the data owner and the servers storing the data are in the same trusted domain, where the servers are fully entrusted as reference monitor responsible for defining and enforcing access control policies. This assumption however no longer holds in cloud computing since the data owner and cloud servers are very likely to be in two different domains. On one hand, cloud servers are not entitled to access the outsourced data content for data confidentiality; on the other hand, the data resources are not physically under the full control of the owner. For the purpose of helping the data owner enjoy access control of data stored on untrusted cloud servers, a feasible would be encrypting data through certain cryptographic primitive(s), and disclosing decryption keys only to authorized users. Unauthorized users, including cloud servers, are not able to decrypt since they do not have the data decryption keys. This general method actually has been widely adopted by existing works which aim at securing data storage on un trusted servers [10]. One critical issue with this branch of approaches is how to achieve the desired security goals without introducing a high complexity on key management and data encryption. Another biggest concern with cloud data storage is that of data integrity verification at untrusted servers. For example, the storage service provider, which experiences intricate failures occasionally, may decide to hide the data errors from the clients for the benefit of their own. What is more serious is that for saving money and storage space the service provider might neglect to keep or deliberately delete rarely accessed data files which belong to an ordinary client. Previously many cryptographic methods were adopted to preserve data integrity and confidentiality. They are classified as:

SECRET KEY CRYPTOGRAPHY (SKC):

Uses a single key for both encryption and decryption

PUBLIC KEY CRYPTOGRAPHY (PKC):

Uses one key for encryption and another for decryption

HASH FUNCTIONS:

Uses a mathematical Transformation to irreversibly "encrypt “information. At present, these methods alone were proven inefficient for providing security. This inefficiency is a result of advancement in technology. The challenges in cloud

COMPUTING PARADIGM ARE:

Storing large data files correctly and maintaining them.

Verify correctness of the remote data even without the existence of local copies.

DATA SECURITY:

These concerns originate from the fact that cloud servers are usually operated by commercial providers which are very likely to be outside of the trusted domain of the users. Data confidential against cloud servers when users outsource data for storage in the cloud. These challenges, if not well resolved, may impede its fast growth. In this work the challenges are addressed by using Key-policy RSA (KP)) cryptographic primitive and Proxy re-encryption. KP is proposed to resolve the problem of data access control in one-to-many communications, while proxy re-encryption is a technique which is mainly used to protect the data from its disclosure in the cloud storage server. This technique helps the cloud servers to provide the requested data to the client without knowing the underlying content.

RELATED WORK:

A. UNIQUENESS OF CLOUD COMPUTING:

Cloud computing is cost-effective. Here, cost is greatly reduced as initial expense and recurring expenses are much lower than traditional computing. Maintenance cost is reduced as a third party maintains everything from running the cloud to storing data. Cloud is characterized by features such as platform, location and device independency, which make it easily adoptable for all sizes of businesses, in particular small and mid-sized. However, owing to redundancy of computer system networks and storage system cloud may not be reliable for data, but it scores well as far as security is concerned. In cloud computing, security is tremendously improved because of a superior technology security system, which is now easily available and affordable. Yet another important characteristic of cloud is scalability, which is achieved through server virtualization. Some of [19] the most important five key characteristics are,

1. On-demand nature Service
2. Open Network Access
3. Resource Pooling
4. Calculated Service
5. Selection of Provider

B. DEFINITIONS:

Various terminologies related to this paper have been explained in this section [1][2].

CLOUD SERVICE PROVIDER (CSP):

A CSP is the most important part of cloud architecture. CSP acts as an interface between the cloud resources and cloud prescribes. It provides all the services (SaaS, PaaS, IaaS) available for a client in “pay as you use” manner.

CLOUD STORAGE SERVER (CSS):

A CSS is an entity, which is managed by Cloud Service Provider (CSP) .It has significant storage space and computation resource to maintain client’s data.

THIRD PARTY AUDITOR (TPA):

A TPA is an entity which maintains records for clients and data owner. Records include login, file access and logout information.

SOFTWARE AS A SERVICE (SAAS):

This is one of the services provided by cloud, in which a user no longer owns the software that is utilized but instead uses it when required.

KEY-POLICY (KP):

KP a public key cryptography primitive for one-to many communications. In this, data are associated with attributes for each of which a public key component is defined.

PROXY RE-ENCRYPTION (PRE):

PRE is also a cryptography primitive that allows a proxy to transform a cipher text computed under A’s public key into one that can be opened by B’s secret key.

C. PURPOSE & AIM:

The purpose of this paper is to provide security to the confidential data that the data owner store on the cloud storage server, to preserve the data integrity while it is transmitted from data owner to CSS and downloaded by the authorized client and this application assigns unique access structure to each client to access to the data stored on the cloud monitored by **TPA.** The application is desktop based, and also user can access the CSS from any location using the data owner and client interface. The application manages and provides security to all the confidential data that the data owner stores and this application also stores information regarding each interaction that data owner and the authorized clients perform on the CSS. This project also aims to reduce the burden on the data owner begin online to distribute the cryptographic keys to the authorized clients. The application facilitates data storage and data integrity. The application GUI is intuitive and user friendly. For the purpose of helping the data owner to secure data stored on untrusted cloud servers, the existing solution is to encrypt data through certain cryptographic primitive(s), and disclosing decryption keys only to authorized users. Unauthorized users, including cloud servers, are not able to decrypt since they do not have the data decryption keys. This general method actually has been widely adopted by existing works which aim at securing data storage on untrusted servers. In the existing technique there was a possible threat while exchanging the confidential data as data is stored on the untrusted cloud servers. One critical issue with this approach is how to achieve the desired security goals without introducing a high complexity on data owner of key management and data encryption using numerous secret keys. In the existing approach there exists heavy computation overhead on the data owner and there is no facility access to the data stored on the cloud server. And also in the existing technique the TPA can only keep track of the different interactions on the data stored in the cloud server [3]. By this information a data owner can only know, the culprits (unauthorized users) who made the access and what data have been modified or deleted. But this technique cannot prevent the cloud provider or other culprits from modifying or deleting the contents on cloud. With this motivation, an application is developed to reduce the burden on the data owner by assigning most of the computation intensive tasks on the cloud server and also it won't allow the provider or any others to know the underlying contents stored in the cloud server. Any platform is used and the deployment is remote desktop based. The remoting concept is used provide an abstract approach to inter process communication that separates the removable object from a specific client or server application domain and from a specific mechanism of communication.

CRITICS ON EXISTING WORKS:

Ateniese et al. are the first to consider public audit ability in their defined "provable data possession" (PDP) model for ensuring possession of files on untrusted storages. In their scheme, they utilize RSA based homomorphism tags for auditing outsourced data, thus public audit ability is achieved. Disadvantage Ateniese et al. do not consider the case of dynamic data storage, and the direct extension of their scheme from static data storage to dynamic case may suffer design and security problems. Ateniese et al**.** propose a dynamic version of the prior PDP scheme. However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations. Disadvantage It only allows very basic block operations with limited functionality, and block insertions cannot be supported. Wang et al**.** consider dynamic data storage in a distributed scenario, and the proposed challenge-response protocol can both determine the data correctness and locate possible errors. Disadvantage They only consider partial support for dynamic data operation. Juels et al. describe a "proof of retrievability" (PoR) model, where spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose, and F is further encrypted to protect the positions of these special blocks. Disadvantage The number of queries a client can perform is also a fixed priori, and the introduction of pre-computed "sentinels" prevents the development of realizing dynamic data updates. In addition, public audit ability is not supported in their scheme. Shacham et al**.** design an improved PoR scheme with full proofs of security. They use publicly verifiable homomorphism authenticators built from BLS signatures, based on which the proofs can be aggregated into a small authenticator value, and public retrievability is achieved.

PROBLEM STATEMENT:

The figure 1. Shows the three different network entities for cloud data storage. A number of problems exist [3] to providing secured data storage and data integrity which are addressed by the following:

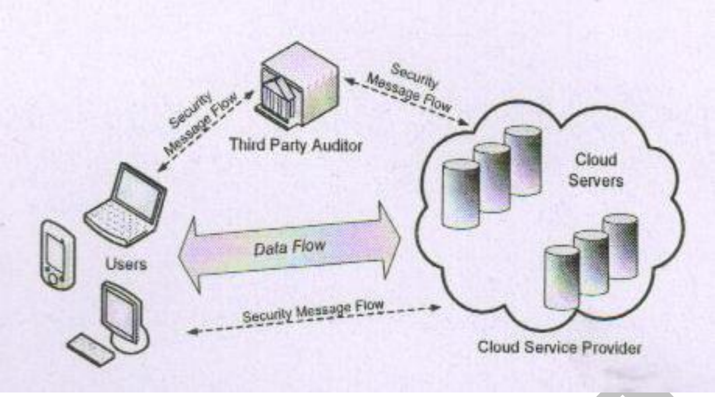


Figure 1. Cloud data storage architecture

The data owner can only know who are all the culprits (unauthorized users) who made the access and what data have been modified or deleted. The existing technique cannot prevent the cloud provider or other culprits from modifying or deleting the contents on cloud. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when data access control is desired, and thus do not scale well. The data owner should be always online to provide decryption keys to authorized clients.

SYSTEM ANALYSIS:

The verification schemes with public audit ability any TPA in possession of the public key can act as a verifier. We assume that TPA is unbiased while the server is untrusted. For application purposes, the clients may interact with the cloud servers via CSP to access or retrieve their pre-stored data. More importantly, in practical scenarios, the client may frequently perform block-level operations on the data files. The most general forms of these operations we consider in this paper are modification, insertion, and deletion.

A. EXISTING SYSTEM:

In the present world the integrity and confidentiality of data storage in Cloud Computing is maintained by delegating the task of allowing a third party auditor (TPA) on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or back up data only. To ensure cloud data storage security, it is critical to enable a third party auditor (TPA) to evaluate the service quality from [5] an objective and independent perspective. Public verifiability also allows clients to delegate the integrity verification tasks to TPA while they themselves can be unreliable or not be able to commit necessary computation resources performing continuous verifications. But in this technique the TPA can only keep track of the different interactions on the data stored in the cloud. By this information a data owner can only know, the culprits (unauthorized users) who made the access and what data have been modified or deleted. But this technique cannot prevent the cloud provider or other culprits from modifying or deleting the contents on cloud. In other techniques to keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when data access control is desired, and thus do not scale well. Here the data owner should be online always to provide decryption keys to authorized users. At present, a number of challenges exist in providing secured data storage and data integrity which are addressed in this work. The challenges faced at present are:

1. The data owner can only know who are all the culprits (unauthorized users) who made the access and what data have been modified or deleted.
2. The existing technique cannot prevent the cloud provider or other culprits from modifying or deleting the ontents on cloud.
3. To keep sensitive user data confidential against untrusted servers, existing solutions usually apply cryptographic methods by disclosing data decryption keys only to authorized users. However, in doing so, these solutions inevitably introduce a heavy computation overhead on the data owner for key distribution and data management when data access control is desired, and thus do not scale well.
4. The data owner should be always online to provide decryption keys to authorized clients.

B. PROPOSED SYSTEM:

There are many challenges in Cloud Computing, if not well resolved, may impede its fast growth. In this work the challenges of the existing techniques are addressed by using some techniques of Key-policy RSA encryption (KP) and Proxy re-encryption. KP is proposed to resolve the problem of data access control in one-to-many communications, while proxy re-encryption is a technique which is mainly used to protect the data from its disclosure in the cloud storage server. This technique helps the cloud servers to provide the requested data to the client without knowing the underlying content. The **.**Net Framework used facilitates ease of use, reduces development efforts and supports improved security and communication features platform remoting allows objects to interact with each other across application domains**.** The figure 2 shows the major network entities in the proposed cloud architecture are [1]:

DATA OWNER:

An entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations.

CLOUD STORAGE SERVER (CSS):

An entity, which is managed by Cloud Service Provider (CSP) .It, has significant storage space and computation resource to maintain client's data.

EFFECTIVE THIRD PARTY AUDITOR (ETPA):

A TPA is an entity which maintains records for clients and data owner. Records include login, file access and logout information.

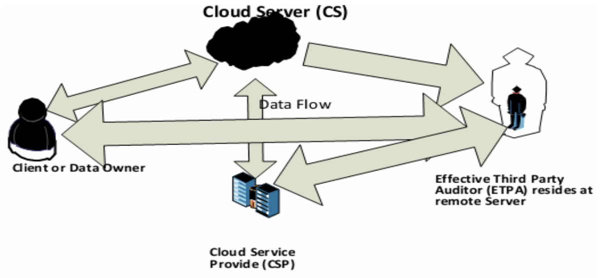

Cloud Server (CS)
Data Flow
Client or Data Owner
Effective Third Party Auditor (ETPA) resides at remote Server
Cloud Service Provide (CSP)

Figure 2. Network Entities of Proposed Cloud architecture

SYSTEM DESIGN:

A.SYSTEM ARCHITECTURE:

Large systems are always decomposed into subsystems that provide some related set of services. The initial design process of identifying these sub-systems and establishing a framework for sub-system control and communication is called Architecture design and the output of this design process is a description of the software architecture. The architectural design process is concerned with establishing a basic structural framework for a system. It involves identifying the major components of the system and communications between these components. The system architecture shows the blocks required for the paper. Figure 3 shows the existing system architecture.

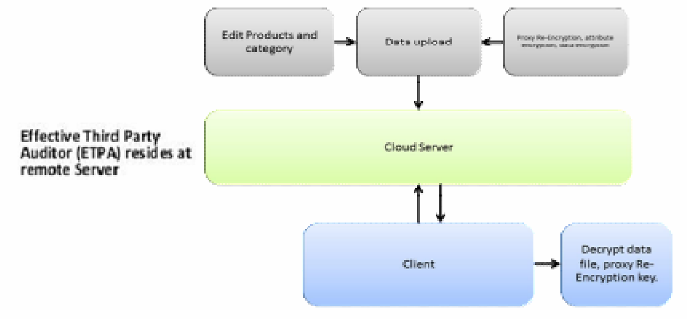

Edit Products and category
Data upload
Proxy Re-Encryption, attribute
Effective Third Party Auditor (ETPA) resides at remote Server
Cloud Server
Client
Decrypt data file, proxy Re-Encryption key.

Figure 3 Cloud Computing System Architecture

The data owner is the one who encrypts the document (text, image and word document) that he wants to upload on the cloud server. After which he performs the AES encryption to encrypt the plain document, proxy reencryption and attribute based encryption to provide unique access structure to each client. Then he categorizes the data into different categories by linking different proxy key generated using the proxy re-encryption technique. The data owner is the one who has the priority of editing and also deleting the contents stored on the cloud server. Then he performs the attribute based encryption to provide unique access structure to each client. After all these steps the data owner uploads the encrypted file, re-encryption key and access structure on to the cloud server. Only authorized client can log in using his private key. After successful he gets a unique access structure that has been assigned by the data owner. Clients request the cloud server for the re-encryption key and files that he wants to download. Finally client decrypts the re-encryption key and then decrypts the encrypted file using this key.

SYSTEM IMPLEMENTATION:

A. ESTABLISHING CLIENT AND TPA:

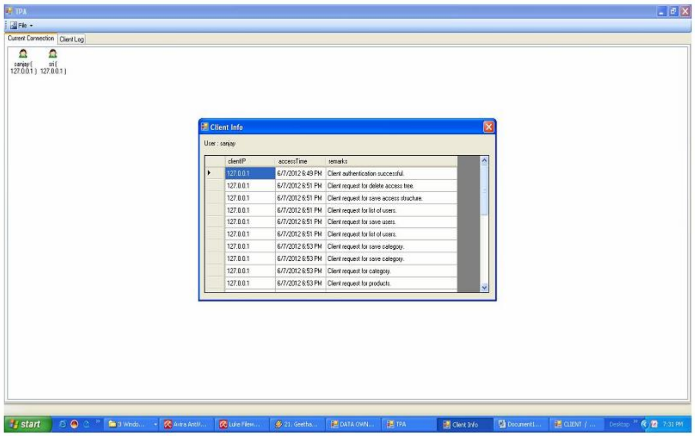
An entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations. TPA which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request

B. ALGORITHM FOR OVERALL IMPLEMENTATION:

Step 1: Initially the cloud server is started.
Step 2: Then the data owner login using the data upload interface.
Step 3: He encrypts the data.
Step 4: Perform PRE (Proxy Re-Encryption), using client's public key.
Step 5: Further he performs KP (Key Policy).
Step 6: He then generates different categories using the proxy key generated during the PRE, Upload the data.
Step 7: Authorized clients who are authenticated by the data owner login using their private key.
Step 8: Each client gets access structure of separate files based on access policy defined by the data owner to that particular client.
Step 9: Download required files.

C. RESULTS:

The figure 4 shows the monitored information of the data owner or the cloud clients/users by the TPA.

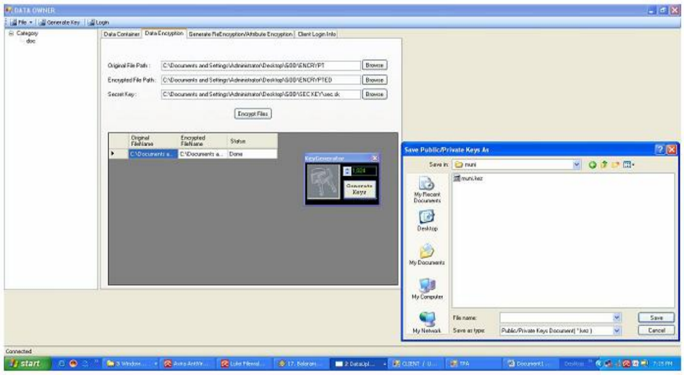


User : sanjay

	clientIP	accessTime	remarks
▸	127.0.0.1	6/7/2012 6:49 PM	Client authentication successful.
	127.0.0.1	6/7/2012 6:51 PM	Client request for delete access tree.
	127.0.0.1	6/7/2012 6:51 PM	Client request for save access structure.
	127.0.0.1	6/7/2012 6:51 PM	Client request for list of users.
	127.0.0.1	6/7/2012 6:51 PM	Client request for save users.
	127.0.0.1	6/7/2012 6:51 PM	Client request for list of users.
	127.0.0.1	6/7/2012 6:53 PM	Client request for save category.
	127.0.0.1	6/7/2012 6:53 PM	Client request for save category.
	127.0.0.1	6/7/2012 6:53 PM	Client request for category.
	127.0.0.1	6/7/2012 6:53 PM	Client request for products.

Figure 4. Monitoring of Cloud Clients by TPA

The figure 5 shows the Data owner uploading the Information of the encrypted format with the help of Public Key, Private Key and Attribute Key.



Original File Path : C:\Documents and Settings\Administrator\Desktop\GOD\ENCRYPT

Encrypted File Path : C:\Documents and Settings\Administrator\Desktop\GOD\ENCRYPTED

Secret Key : C:\Documents and Settings\Administrator\Desktop\GOD\SECKEY\sec.sk

	Original FileName	Encrypted FileName	Status
▸	C:\Documents a...	C:\Documents a...	Done

Save in: muni

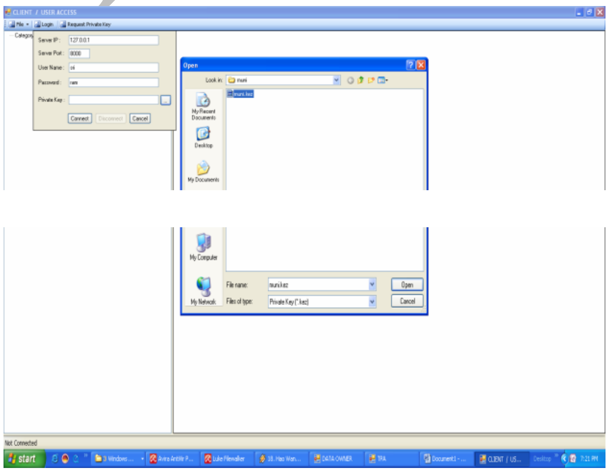
muni.kez

File name:

Save as type: Public/Private Keys Document(*.kez)

Figure 5. Data Owner Uploading data to the Cloud

The figure 6 shows the Authenticated Cloud client's can access the uploaded data by the data owner with the help of its private key only when the data owner permits.



Server IP : 127.0.0.1

Server Port : 8000

User Name : sri

Password : ram

Private Key :

Look in: muni

muni.kez

File name: muni.kez

Files of type: Private Key (*.kez)

Figure 6. Cloud Users/ Clients downloading the data

CONCLUSION & FUTURE ENHANCEMENT:

This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In particular, we consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to monitor the integrity of the dynamic data stored in the cloud. We have achieved data confidentiality and preserved data integrity with the help of KP and proxy re-encryption techniques. We utilize and uniquely combine the public key based homomorphism authenticator with random masking to achieve the privacy-preserving public cloud data auditing system. This scheme is the first to support scalable and efficient public auditing in the Cloud Computing. The technique of AES algorithm where TPA can perform the auditing tasks simultaneously. The data in the cloud does not remain static. Unlike most prior works, the new scheme further supports secure and efficient dynamic operations on data stored in the cloud, including: data update, delete and append. Unique access structure enhances security. Implementation can be simplifies the development efforts and provides direct support for security. Ex-tensive security and performance analysis show that the proposed scheme is highly efficient and provably secure.

In Future, the Author intends to enhance the system by developing a module to write files to the developing a module storage server. In addition the author is interested in handling different types of multimedia, as part of future enrichment. The author is interested in making this application web based

APPENDIX:

B. KP FOR CONSTRUCTING PUBLIC AUDIT:

The public auditing system can be constructed in four phases by using the Key Policy of RSA Encryption (KP) algorithm.

STEP1: SETUP:

This algorithm takes as input a security parameter and the attribute universe U = {1, 2. . . N}of cardinality N. It returns the public key PK (e,n) as well as a system master key MK(d,n). While PK is publicly known to all the parties in the system, MK is kept as a secret by the authority party.

STEP2: ENCRYPTION:

This algorithm takes a message M, the public key PK. It outputs the cipher text E.

STEP3: KEY GENERATION:

This algorithm takes as input an access tree T, the master key MK, and the public key PK. It outputs a user secret key SK.

STEP4: DECRYPTION:

This algorithm takes as input the cipher text E encrypted under the attribute set I, the user's secret key SK for access tree T, and the public key PK. Finally, it output's the message M if and only if I satisfies T.

RSA algorithm:

RSA consist of three steps:-

1. Key Generation Process
2. Encryption Process
3. Decryption Process

KEY GENERATION PROCESS:

1. Select p , q where p and q both prime , p is not equal to q.
2. Calculate $n = p xq$
3. Calculate $\phi(n) = (p-1) \ x \ (q-1)$
4. Select integer e whose $\gcd(\phi(n), e) = 1; \ 1 < e < \phi(n)$
5. Calculate d, $d = e^{-1} (\bmod \phi(n))$
6. Public key: $PU = \{e,n\}$

7. Private key: $PR = \{d,n\}$

ENCRYPTION PROCESS:

plain text : $M < n$

Cipher text: $C = M^e \bmod n$

DECRYPTION PROCESS:

Cipher text: C

Plain text: $M = C^d \bmod n$.

ACKNOWLEDGMENT:

The authors would like to thank the editor, mystifying reviewers for their priceless suggestions that appreciably improved the eminence of this paper. Finally thankful to respected supervisor Mr. Manik Chandra Pandey and Head-of-Departments SITE, Meerut (India).

REFERENCES:

1. Abhishek Mohta, Ravi Kant Sahu,Lalit Kumar Awasthi Dept. of CSE, NIT Hamirpur (H.P.) India, Robust Data Security for Cloud while using Third Party Auditor,International Journal of Advanced Research in Computer Science and Software Engineering Research Paper Volume 2, Issue 2, February 2012.
2. Kartthikeyan.S, Balakrishnan.S, Saranya.G, Shobana.S, Introducing Effective Third Party Auditing (TPA) for DataStorage Security in Cloud, International Journal of Computer Science and technology IJCST Vol. 2, Issue 2, June 2011.
3. Qian Wang1, Cong Wang1, Jin Li1, Kui Ren1, and Wenjing Lou2,"Enabling Public Audit ability and Data Dynamics for Storage Security in Cloud Computing",IEEE computer society , Vol 22, No 5, May 2011.
4. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H.Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I.Stoica, and M. Zaharia, "Above the clouds: A berkeley viewof cloud computing", University of California, Berkeley, Tech. Rep. USB-EECS-2009-28, Feb 2009.
5. Qian Wang1, Cong Wang1, Jin Li1, Kui Ren1, and Wenjing Lou2,"Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing", Springer-Verlag Berlin Heidelberg 2009.
6. Vipul Goyal□,etc,."Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data"CCS'06, October 30–November 3, 2006,
7. Giuseppe Ateniese† Kevin Fu‡ Matthew Green† Susan Hohenberger‡," Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage" TISSEC, 2005.
8. Ting Yu,etc,"A Unified Scheme for ResourceProtection in Automated Trust Negotiation" IEEE 2003.
9. T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation", in Proc. Of SP'03, 2003.
10. Mahesh Kallahalla_ Erik Riedel† Ram Swaminathan_Qian Wang‡ Kevin Fu§,"Plutus: Scalable secure file sharing on untrusted storage", File and Storage Technologies (FAST'03).
11. P. D. McDaniel and A. Prakash, "Methods and limitations of security policy reconciliation," in Proc. Of SP'02, 2002.
12. H. Harney, A. Colgrove, and P. D. McDaniel, "Principles of policy in secure groups," in Proc. of NDSS'01, 2001.
13. 104th United States Congress, "Health Insurance Portability and AccountabilityAct of 1996 (HIPPA)," Online at http://aspe.hhs.gov/admnsimp/pl104191.htm,1996.
14. Shucheng Yu1, Kui Ren2, Wenjing Lou1, and Jin Li2,"Defending Against Key Abuse Attacks in KP-ABE Enabled Broadcast Systems"
15. Eu-Jin Goh□, Hovav Shacham_, Nagendra Modadugu, Dan Boneh," SiRiUS: Securing Remote Untrusted Storage"
16. Giuseppe Ateniese_ Karyn Benson_ Susan Hohenberger," Key-Private Proxy Re-Encryption".
17. Shucheng Yu_, Cong Wang†, Kui Ren†, and Wenjing Lou_," Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing".