Trust Enforcement in Cloud Storage Systems with Mutual Trust Model

Mr. Anup R. Nimje[1], Prof. V. T. Gaikwad[2], Prof. H. N. Datir[3]
[1] M.E. Student, Computer Engineering
Sipna College Of Engineering and Technology, Amravati, India
[2]Associate Professor and Head , Information Technology
Sipna College Of Engineering and Technology, Amravati, India
[3]Assistant Professor, Computer Science and Engineering
Sipna College Of Engineering and Technology, Amravati, India

***Abstract:-* Cloud computing has become very popular not only in industry to an individual levels but also in researchers to the end users. Use of cloud computing for storing large amount of data, that may be confidential and expensive as well. The concept of cloud computing storage or the implementation of cloud computing as storage as a service, brings itself the concerns about various issues such as confidentiality, integrity, access control and newness of data.**

In short, the mutual trust must be established between owner and the cloud service provider. There is a mutual trust model proposed by Ayad Barsoum and Anwar Hasan, in which these issues are said to be resolved. This model consists of the three techniques such as lazy revocation, key rotation and broadcast encryption. The given model consists of four modules i.e. Owner Module, User Module, Cloud Service Provider (CSP) Module and Trusted Third Party (TTP) Module. In this paper, we review this model and implement the same and study the functionality of each module and how the mutual trust between the owner and CSP is established.

I INTRODUCTION:

Cloud computing has become a very popular and high demanded from industries as well as researchers. Huge research is going on this systems. This is due to its cost effectiveness, low management overhead, immediate access to a wide range of applications, flexibility to scale up and down capacity, and mobility where customers can access the data from anywhere, instead of staying on their desks. Cloud computing is the computational model over a large pool of shared virtualized computing resources (e.g., storage, processing power, memory, applications, service and network bandwidth). In previous access control techniques it is assumed that the data exists with the data owner and the storage servers in the same *trust domain*. [1] Data is outsourced to a remote CSP, thus this assumption does not applicable for this situation as it resides outside the trust domain of the data owner by taking the full access of the outsourced data management. The data to be stored is very huge, at corporate level which may be very confidential and expensive. That is transmitted over the cloud systems to perform various computational operations. The managing and monitoring of the data is comparatively difficult task. [2] So, we have modified cloud systems in which the storage is offered by cloud service providers (CSPs) in which the data storage is outsourced. [3]-[4].This outsourcing of data storage is known as *Storage-as-a-Service*. In the process of outsourcing the storage and store the sensitive data to remote CSP, there must be *confidentiality*, *integrity*, and *access control* for the data. The confidentiality can be brought by traditional cryptography before outsourcing to CSP. For the data integrity over cloud servers, there is provable data possession (PDP) technique has been proposed.

Various *PDP protocols* have been proposed to verify the integrity of static data efficiently. [5] Whereas an another class of PDP schemes is concerned with the dynamic behavior of data over remote servers. It allows the owner to outsource a data file and perform updating or scaling operations, dynamic updates on the outsourced data.

Proof of irretrievability (POR) consists of the entire data file can be reconstructed from portions of the data that are reliably stored on the servers. whereas HAIL(High-Availability and Integrity Layer) [6] is a distributed cryptographic system. It allows the client to a store the files are intact and retrievable. It is robust and improves on the security and efficiency of existing models such as Proofs of Retrievability (PORs) [7] that is deployed on individual servers. The data storage center stores all of a client's data. In this system the important task to provide efficient and provably secure to extract the client's data from any prover that passes a verification check.

Attribute-based encryption scheme is an another solution for getting fine-grained access control [8].For user and server in non trusted domain, the 'Attribute Based Encryption (ABE)' scheme has been introduced in which the user's secret key and the ciphertext are associated with a set of attributes. A user is able to decrypt the cipher-text if and only if at least a specified number of attributes get overlapped between the cipher-text and user's secret key. The encryption algorithm consists of encryption with attributes as well as the policies specified. But however it fails with respect to flexibility and scalability when authorities at multiple levels are considered.

Third party auditor (TPA) is again one of the solution for cloud storage systems. It verifies the integrity of the dynamic data stored in the cloud behalf of the client. The TPA auditing of whether his data stored in the cloud is indeed intact, which can be commercially important Cloud Computing. The various data operations like modification, insertion and deletion are also significantly performed. The public verifiability and dynamic data operations are provided in TPA model. [9]

Plutus is again solution for cloud storage system. It consists of file sharing without placing much trust on the file servers is carried out. It uses cryptography to protect and share files. Plutus achieves strong security with overhead comparable to systems that encrypt all network traffic. [10]

SiRiUS: It consists of network un-trusted storage and provides its own read-write cryptographic access control for file level sharing and File systems are supported by using hash tree constructions. It performs well relative to the file system despite using cryptographic operations. [11]

In the existing schemes discussed, access control techniques assume the existence of the data owner and the storage servers in the same trust domain. Such system, no longer holds when the data is outsourced to a remote CSP.

Enforcing access control and newness of data, mutual trust between various components are beyond the setting of both PDP and POR. There are verifier and auditing process introduced in systems. These models have focused on access control and secure sharing of data on un-trusted servers. There are various issues with the previous models such as dynamic data over remote servers, data integrity, newness property and mutual trust etc.

II THE MUTUAL TRUST MODEL:

It is given by Ayad Barsoum and Anwar Hasan has addressed the important issues related to outsourcing the storage of data such as dynamic data, newness, mutual trust, and access control. That is, the remotely stored data can be accessed by authorized users, as well as updated and scaled by the owner. The newness of data i.e. after updating, the authorized users should receive the latest version of the data. The mutual trust between the data owner and the CSP has been addressed in the model. Determination of the dishonest party, from User or CSP and the responsible party is provided. The access control allows the data owner to grant or revoke access rights to the outsourced data.

Our contributions: The designing and implementation of a cloud-based storage scheme having the following features:

1. It allows a data owner to outsource the data to a remote CSP,and perform dynamic operations.
2. It ensures the newness property, i.e., the authorized users receive the most recent version of the data when an update is performed.
3. It establishes indirect mutual trust between the data owner and the CSP since each party resides in a different trust domain.
4. It enforces the access control for the outsourced data.

Ayad Barsoum and Anwar Hasan, proposed that In the mutual trust model given provides solutions to the important issues related to outsourcing the storage of data, namely *dynamic data*, *newness*, *mutual trust*, and *access control*. [12]. The owner outsources the data storage from the cloud service provider and it is made available for the authorized users. Also the data is updated, scaled and monitored by the Trusted Third Party. After updating, authorized users should receive the latest version of the data.

In Trusted Third Party the mechanism has been introduced to determine the dishonest party. The dishonesty or attack on the data from any entity is detected and the responsible party is identified. For addressing issue of access control the mechanism has been provided for granting and revoke the access rights to the outsourced data. This scheme addresses important issues related to outsourcing the storage of data over the cloud storage systems.

The cloud computing storage model considered in this scheme as shown in Fig. 1:

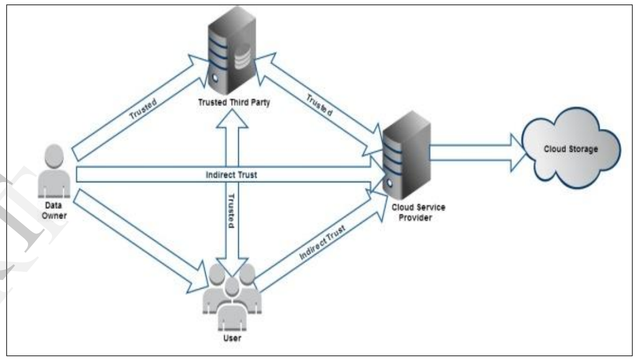

Fig. 1: Cloud Computing Data Storage System Model

The mutual trust for cloud computing storage model given by Ayad Barsoum and Anwar Hasan consists of the components as given in fig.1.

(i) Data owner: it is able to generate sensitive data to be stored in the cloud and that would be made available for controlled usage

(ii) Cloud Service Provider (CSP): It provides infrastructure for data storage and manages the cloud servers. It stores the owner's files and make them available for authorized users;

(iii) Authorized Users: it is a set of owner's clients who have provided rights to access the remote data; and

(iv) Trusted Third Party (TTP) : It is trusted by all other system components, and has capabilities to detect and specify dishonest parties or give some notification to the data owner.

The system components are having relations indicated by double-sided arrows, where solid and dashed arrows represent trust and distrust relations, respectively. It means that the TTP is trusted by data owner, the authorized users, and the CSP. Whereas the data owner and the authorized users have mutual distrust relations with the CSP. Hence the TTP is used to enable indirect mutual trust between these three components. data owner and the authorized users are directly trusted.

III THE ALGORITHMIC TECHNIQUES INVOLVED FOR GIVEN MODEL:

In this scheme, the data owner enforces access control for the outsourced data by combining three cryptographic techniques:, *lazy revocation* [13], *and key rotation*[10][14],*broadcast encryption*[15].

A. Lazy Revocation

Lazy revocation technique allows the data owner to allowing for access and revoking the access for the outsourced data file. It allows to read (decrypt)data blocks. However, updated or new blocks must not be accessed by such revoked users.

It is for allowing the revoked users to read unchanged data blocks. Updated or new blocks following a revocation are encrypted under new keys. Lazy revocation trades re-encryption and data access cost for a degree of security. However, it causes fragmentation of encryption keys, i.e., data blocks could have more than one key.

B. Key Rotation

Key rotation is a technique in which a sequence of keys can be generated from an initial key and a master secret key. It consists of the sequence of keys. The secret keys are specialized for the two points: Only the owner of the master secret key can generate the next key in the sequence from the current key, and any authorized user knowing a key in the sequence can generate all previous versions of that key. It means that, given the i-th key Ki in the sequence, it is computationally infeasible to compute keys K*l* for *l*>*i* without having the master secret key, but it is easy to compute keys Kj for *j* <*i*.

The first property enables the data owner to revoke access to the data by producing new keys in the sequence, which are used to encrypt updated/new blocks following a revocation. It is used to prevent a user revoked during the *i*-th time from getting access to data blocks encrypted during the *l*-th time for*l*>*i*. The second property allows authorized users to maintain access to data blocks that are encrypted under older versions of the current key. Data owner can transfer only a single key K*i* to authorized users for accessing all data blocks that are encrypted under keys K*ig* (rather than transferring a potentially large set of keys f{K1,K2, …., K*ig*}. Thus, the second property reduces the communication overhead on the owner side. The mutual trust model considered in this work makes use of the key rotation technique.

Let *N* =*pq* be the RSA modulus (p and q are prime numbers), a public key = *(N, e)* and a master secret key d. The key d is known only to the data owner, and *ed = 1mod (p- 1) (q - 1)*.Whenever a user's access is revoked, the data owner generates anew key in the sequence (rotating forward).

Let ctr indicate the index/version number of the current key in the keys sequence. The owner generates the next key by exponentiation Kctr with
the master secret key d: Kctr+1 = Kdctr mod N.

Then the authorized users can recursively generate older versions of the current key by exponentiations with the public key component e:Kctr-1 = Kectr mod N (rotating backward).The RSA encryption algorithm gives repeatedly encryption that results in cycling. It can be a pseudorandom number generator. However it is used to factor the RSA modulus N.

C. Broadcast Encryption

In the Broadcast encryption (bENC) technique, it allows a broadcaster to encrypt a message for an arbitrary subset of a group of users. That subset of group of users are allowed to decrypt the message. If all users are outside the subset, they cannot access the encrypted data. To enforce access control in outsourced data in this work bENC encryption. The bENC is composed of three algorithms:

Setup, Encrypt, and Decrypt:

Broadcast encryption (bENC) [15] allows a broadcaster/data owner to encrypt a message for an arbitrary subset of a group of users. The proposed scheme uses bENC to enforce access control in outsourced data. The bENC consists of three algorithms: SETUP, ENCRYPT, and DECRYPT.

1) Setup:

This algorithm takes system users *n* as input. It defines a bilinear group G of prime order *p* with a generator *g*, a cyclic multiplicative group GT , and a bilinear mapê: G×G →GT. The algorithm picks a random α∈Zp, computes gi= g(αi)∈G for i= 1, 2,..,n,n + 2,.., 2n, and sets v = gγ∈G for γ ∈R Zp.
The outputs of above operations are a public keyPK =(g, g1,.., gn, gn+2,.., g2n, v)∈G^{2n+1}, and *n* private keys{di}1≤i≤n, where di= gγi∈G.

2) Encrypt:

This algorithm takes a subsetS ⊆{1, 2, . . . , n}, and a public key PKas input. And outputs a pair (Hdr, K), where Hdr is called the header (broadcast ciphertext), and K is a encryption key for message.
Hdr= (C0, C1) ∈G^2, where for t∈R Zp, C0 = g^tandC1 = $(v \cdot \prod_{j \in S} g_{n+1-j})$t. The key K = ê($g_{n+1}$, g)^t is used to encrypt a message M (symmetric encryption) to bebroadcast to the subset S.

3) Decrypt:

This algorithm takes as input a subset S⊆{1, 2,.., n}, a user-IDi∈{1, 2, . . . , n},the private key di for user i, the header Hdr = (C0, C1), and the public key PK. If i∈S, the algorithm outputs the key K =ê (gi, C1)/ê $(di \cdot \prod_{j \in S, j \neq i} g_{n+1-j+i}, C_0)$, which can be used to decrypt the encrypted version of M.[16]

The data that is out sourced requires metadata and block indices. The metadata and block indices provides the modifications made by owner. Every time the user and owner can get the recent copy of data. The block status table is introduced in this model in which *combined* hash values and a small data structure are provided indicating the status of blocks. Also the Trusted Third Party is to establish the trust among different components. To enforce access control and mutual trust in this model these three cryptographic techniques are used [12].

IV IMPLEMENTATION

The architecture of the given model is shown in fig. 2 in which the four models are shown. The cryptographic algorithms wherever used are shown as ‘key’ between the components of the system. There is a database maintained for the implementation of these components and finally the data is stored in the cloud storage. The trust between the owner and user are maintained using lazy revocation and key rotation. Also the trust between the CSP and TTP is established using broadcast encryption. The hash values are maintained for each file.

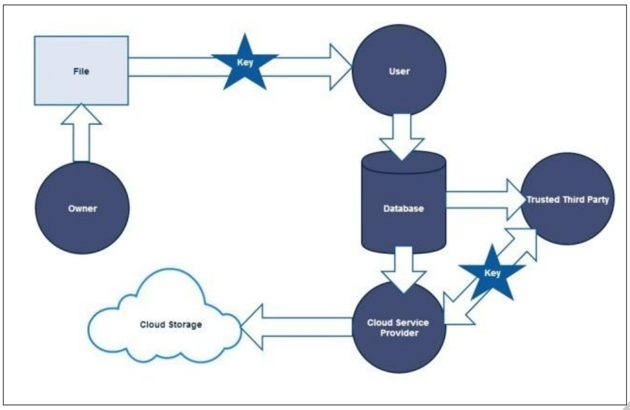

Fig. 2. Architecture of the model.

Cloud storage service consists of the storage of the data on the cloud and the accessibility of data from anywhere using the internet connectivity. A database server is similar to this concept at individual level. It consists of the data stored on it and can be accessible from any client using connectivity. In this work we are using MySQL database server for storing the data before sending to cloud. In our implementation we use a MySql for database. And a cloud storage space for storing the files, approved by CSP. Algorithms in which the encryption-decryption mechanisms RSA module, hash key generation etc are implemented using Java. Web pages are developed using java server pages and CSS3. Also we use glassfish server for executing the web project.

We have implemented the given mutual trust model for uploading a file. The implementation consists of four modules:

1. OModule (owner module),
2. CModule (CSP module),
3. UModule (user module), and
4. TModule (TTP module).

OModule runs on the owner side, is used by the owner to perform the owner role in the setup and file preparation phase. CModule runs is used by the CSP to store, update, and retrieve data from database server which is assumed to be cloud server. UModule runs at the authorized users’ side, and include functionalities that allow users to interact with the TTP and the CSP to retrieve and access the outsourced data. TModule is used by the TTP to perform the TTP role in the setup and file preparation phase. Moreover, the TTP uses this module to determine the un-trusted party in the system.

Procedure:

i) Owner and User Registration at Registration Portal with passwords.

ii) Owner Login

iii) File Upload by Owner
  a. File gets encrypted and the Secret Key is generated.
  b. User Grant Access
  c. Key is sent to the selected User

iv) User Login
  a. Inputs the filename and key.
  b. User is allowed to view and download the file.
  c. User cannot modify the file. An alert message is generated and detected.

v) CSP Login
  a. It can view the file (since it is having the set of keys) but cannot modify. An alert message is generated and detected.
  b. It has to send the file to cloud storage once verified.

vi) TTP Login
  a. It consists of the file details and the owner, user data. Matches the hash key with the CSP for proceeding with file.
  b. An alert table is maintained with file details, owner and user details, action noted, timestamp for detecting the un-trusted parties.
  c. This alert table is also provided to the owner portal for its reference.

The implementation of the given model provides the mutual trust in the different component of the cloud storage systems and satisfactory results.

V CONCLUSION

Outsourcing the data to the CSP or Storage is itself a concern regarding the trust. A data owner wants the service that it must be secured, authorized, integrated and trusted. There are various trust enforcement models like PDP, POR etc have been proposed. The mutual trust model given by Ayad Barsoum et al. proposed that it allows owner to outsource the file over the Cloud Storage. CSP offered by the CSP and enables indirect mutual trust between them. There is an entity called Trusted Third Party that enables to track the status of the files and detects any un-trusted activity on the data. It determines the un-trusted party in the system.

This establishes the mutual trust between the different component of the cloud system. It provides the solutions for confidentiality, integrity and access control of the outsourced data. The data that is being outsourced, is encrypted at CSP side. TTP manages the files and co-ordinates the CSP using the hash keys. The three techniques are implemented in this model i.e. broadcast encryption, lazy revocation, and key rotation.

REFERENCES

[1] S. D. C. di Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, “Over-encryption: Management of access control evolution on outsourced data,” in Proceedings of the 33rd International Conference on Very Large Data Bases. ACM, 2007, pp. 123–134.

[2] Meiko Jensen, JörgSchwenk, Nils Gruschka, Luigi Lo Iacono, ―On Technical Security Issues in Cloud Computing, 2009 IEEE International Conference on Cloud Computing

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,” in *Proceedings of the 14th ACM Conference on Computer and Communications Security*, ser. CCS ’07, 2007, pp. 598–609.

[4] F. Seb´e, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, “Efficient remote data possession checking in critical information infrastructures,” *IEEE Trans. on Knowl. AndData Eng.*, vol. 20, no. 8, 2008.

[5] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable data possession at untrusted stores,”

[6] K. D. Bowers, A. Juels, and A. Oprea, “HAIL: a high-availability and integrity layer for cloud storage,” in *CCS ’09: Proceedings of the 16th ACM conference on Computer and communications security*. New York, NY, USA: ACM, 2009, pp. 187–198.

[7] A. Juels and B. S. Kaliski, “PORs: Proofs of Retrievability for large files,” in *CCS’07: Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 584–597.

[8] Vipul Goyal, Omkant Pandey, Amit Sahai, Brent Waters , “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data” Proceedings of the 13th ACM conference on Computer and communications security,Pages 89 - 98

[9] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in Proceedings of the 14th European Conference on Research in Computer Security, 2009, pp. 355–370.

[10] M. Kallahalla, E. Riedel, R. Swaminathan, Q. Wang, and K. Fu, “Plutus: Scalable secure file sharing on untrusted storage,” in Proceedings of the FAST 03: File and Storage Technologies, 2003.

[11] E.-J. Goh, H. Shacham, N. Modadugu, and D. Boneh, “Sirius: Securing remote untrusted storage,” in *Proceedings of the Network and Distributed System Security Symposium, NDSS*, 2003.

[12] Ayad Barsoum and Anwar Hasan “Enabling Dynamic Data and Indirect Mutual Trust for Cloud Computing Storage Systems”, IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS-2013.

[13] M. Backes, C. Cachin, and A. Oprea, “Secure key-updating forlazy revocation,” in 11th European Symposium on Research in Computer Security, 2006, pp. 327–346.

[14] K. E. Fu, “Group sharing and random access in cryptographic storage file systems,” Master’s thesis, MIT, Tech. Rep., 1999.

[15] D. Boneh, C. Gentry, and B. Waters, “Collusion resistant broadcast encryption with short ciphertexts and private keys,” in *Advances in Cryptology - CRYPTO*, 2005, pp. 258–275.

[16] D. Boneh, C. Gentry, and B. Waters, “Collusion resistant broadcast encryption with short ciphertexts and private keys,” in *Advances in Cryptology - CRYPTO*, 2005, pp. 258–275.