ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

Survey Paper on Effective Access Control on Cloud

Archana Vilas Kshirsagar[1], KanchanDoke[2]

Student of M.E, Dept. of Computer Engineering, Bharati Vidyapeeth College of Engineering, Mumbai University, India

Assistant Professor, Dept. of I.T., B.N.Bandodkar College of science, Mumbai University, Thane, Maharashtra, India[1]

Assistant Professor, Dept. of I.T., Bharati Vidyapeeth College of Engineering, Mumbai University, Navi Mumbai, Maharashtra, India[2]

ABSTRACT: Attribute Based Encryption (ABE) provides direct control to data owner over data stored in cloud. Ciphertext Policy- Attribute Based Encryption (CP-ABE) provides all access control in user hands, where Attribute Control Policy (ACP) can be implemented by Owner. System can provide two layer encryption which divides ACP between owner and cloud to have less workload on owner. ABE scheme provide single authority to control whole attribute set which can leads toward single point bottleneck on performance. Some ABE scheme provide multi-authority by managing disjoint attribute subset, still single point bottleneck problem remain unsolved. The scheme provides threshold multi-authority CP-ABE access control in cloud names as TMACS. In which multiple authorities jointly manage the uniform attribute set, taking advantage of (t, n) threshold secret sharing. Here the master key can be shared among multiple authorities and legal user can generate his/her secret key by interacting any *t* out of *n* authorities.

KEYWORDS: Cloud,Attribute Based Encryption(ABE),Attribute Control Policy (ACP),single authority,multi-authority,threshold secret sharing

I. INTRODUCTION

Cloud computing is meant to satisfy requirement of data storage and data outsourcing for data owners. Thought cloud service provides such services but security and privacy of owner's data is major concern in cloud storage. Therefore secure data access is critical issue in cloud storage.

Attribute-Based Encryption (ABE) is one of the suitable technique to access data securely on cloud where control of data access is in the hand of data owner [3]. Ciphertext- Policy Attribute-Based Encryption (CP-ABE) is among the category of ABE scheme where, data owners can define access policy for each file based on users attribute. In most existing CP-ABE scheme the attribute managed and key distribute only by one authority. Thus one authority can lead system toward single point failure which directly effect on system performance and security. In this case single point failures can occur. In multi-authority CP-ABE scheme, the whole attribute set is divided into multiple subsets, each subset is now maintained by single authority. In this case adversary cannot compromise all authority at a same time. In addition single point failure has not solved.

Another technique which solves the problem of single point failure is Threshold multi-authority CP-ABE access control that is TMACS [1]. In this technique multiple authorities jointly manage the whole attribute set but no one has full control over any specific attribute. In this technique secret sharing key is used among different authorities with (t, n) threshold secret sharing. In TMACS secrete key is known as a Master Key which cannot be obtained by any single authority alone.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

II. LITERATURE SURVEY

Distributed, Concurrent and Independent Access to Encrypted Cloud Databases is the encrypted cloud database access provides multiple, independent and geographically distributed client to execute concurrent queries on encrypted data [6]. Here even SQL statements are in modified encrypted structure to provide confidentiality. The above goals are designed by proxy less cloud-client communication. To achieve goals like availability, scalability, SecureDBaaS prototype is used to support mentioned goals. Here SecureDBaaS process plaintext data, encrypted data, metadata and encrypted metadata. Data and metadata are stored in cloud database. SecureDBaaS clients can retrieve the required metadata from cloud through SQL statements. Secure table contain data where secure table is nothing but encrypted tables. The problem with this approach is all the SQL commands types need to predefine during design phase which seems impractical i.e. the set of SQL operations does not change after database design.

In Adaptive encryption architecture for cloud databases this approach access to cloud is adaptive that is change in workload doesn't cost to performance degradation, it also bring us privileges to change the set of SQL queries even after database design [5]. This is proxy less architecture. All metadata and data are stored in cloud database and can access by client through encrypted database engine. Encrypted engine fetch required metadata to execute SQL queries from cloud database and decrypt it through master key which is with client side application. Adaptive encryption scheme consider many SQL aware encryption algorithm such as *Random, Deterministic* which supports equality operators, *order preserving encryption, homomorphic sums, plain* and *search.* Adaptive encryption scheme consider many SQL aware encryption algorithm such as *Random, Deterministic* which supports equality operators, *order preserving encryption, homomorphic sums, plain* and *search.* If each column is encrypted through only one algorithm then administrator has to decide database operations at design time only for each column [6]. Here encryption algorithms are organized into structure called onions, where each onion is made up of ordered set of encryption algorithm called layer. Onions layers are used for equality, comparison, summation, string equality operators. Each plaintext column is encrypted into one or more encrypted column each one corresponding to an onion. Each plain text is encrypted through all the layers of its onion i.e. encrypted through more than on encryption algorithm. Thought this approach provides more adaptive mechanism for accessing cloud database, access policies are assigned by data owner or single authority only which can result in system bottleneck.

Multi-User Encrypted SQL Operation on Cloud approach provides scalable and confidential access to cloud database. This architecture called Multi-User relaTional Encrypted DataBase (MuteDB) that guarantees data confidentiality by executing SQL operation on data by applying access control policies [4]. The MuteDB does not rely on any intermediate proxy to avoid single point bottleneck. Here every data and metadata is stored on cloud in encrypted format. Here data managed and create by DBA, who is also responsible storing encrypted data and metadata on the cloud. DBA is the trusted entity who owns root credentials, manages user accounts and enforces access control policies. This ACP defines which user can have access on which data. Each user will be provided set of credentials including all the information that allows him/her to access legitimate data. In this case access policies are also encrypted and stored in cloud. The DBA is the only authority who can have control on all system entity; this can leads toward DBA overloading and can result on performance degradation.

III.METHODOLOGIES

Multi-authority cloud access control has following access control approach:

1. Revocable data access control for multi – authority cloud Storage
2. Delegated Access Control in clouds
3. Threshold Multi-Authority CP-ABE Access Control Scheme

1. Revocable data access control for multi – authority cloud Storage:

CP-ABE access control scheme gives more access control to data owner, where there are multiple authorities co-exist and each authority is able to manage attributes independently. This scheme provides forward and backward security for access control [3].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

1.1. System Model

There are five types of entities in the system: a Certificate Authority (CA), Attribute Authority (AA), Data owners (Owners), Cloud server (Server) and Data consumer (users).

- Certificate Authority (CA): It is global trusted authority in the system. It sets up the system and accepts registration of all users and *AA*s in the system assigns global unique identity for each user and *AA* and also assign global public key for them. CA is not involved in attribute management and secret key creation for user.
- Attribute Authority (AA): it is responsible managing attributes independently. It also entitling and revoking user's attribute according to their role and domain. Here every attribute is associated with single *AA*, but each *AA* can manage multiple attributes. *AA* is responsible for generating secret key for each user reflecting his/her attribute.
- Data consumer (User): it has global identity in the system. A user may have assigned a set of attribute from multiple *AA*. The user will receive a secret key associated with its attributes from corresponding *AA*.
- Data owners (Owners): owner first divides data into different component and encrypt each component separately using different content key using symmetric encryption techniques, defines access policy on attributes defines from multiple *AA* and encrypt content key under access policies. Owner sends this encrypted data with encrypted key to cloud.

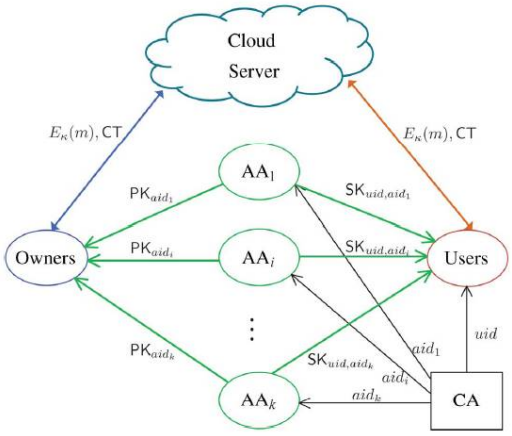

Fig. 1. System model of data access control in multi-authority cloud storage

1.2. Framework

The framework of data access control scheme for multi-authority cloud storage system consists following phases:

Phase 1: System Initialization

- CASetup: The CA setup algorithm is run by CA, generate global master key for system. It generates global unique ID known as *uid*for each user and *aid* for each AA. Generate certificate for each legal user of the system.
- AASetup: it is run by each *AA*. It produces secret and public key pair (SK*aid*, PK*aid*) for each *AA*.

Phase 2: Secret Key Generation by *AA*s.

- SKeyGen: it is run by each *AA*. It takes system parameters, global secret key of each user *uid*, the SK*aid*, and set of attributes which are manages by *AA*. It outputs the secret key SK*uid.aid*for user *uid* which is used for decryption.

Phase 3: Data Encryption by Owner

- The owner first encrypts the data m with content keys by using symmetric encryption method, then encrypt content key.
- Encrypt: The encryption algorithm is run by data owner to encrypt content keys. It takes as input, global system parameters, the set of public keys PK*aid* of each *AA* in the encryption set; content key *k* and access policy *A*. The algorithm encrypts *k* according to access policy *A*.

Phase 4: Data Decryption by Users

- Decrypt: It takes as input, encrypted content key with access policy *A* and secret key of user SK*uid*. If the user *uid* access policy *A*, the algorithm will decrypt the content key *k*.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

Phase 5: Attribute Revocation: Here only the components associated with revokes attributes needs to update instead of whole data set. This revocation is done by cloud.

- Update Key generation by AAs – UKeyGen: This algorithm is run by corresponding AA*aid* that manages the revoked attribute. It takes inputs, the secret key of AA, SK*aid*revoked attribute and update key for secret key update and update key for ciphertext update.
- Secret Key Update by non-revoked users – SKUpdate: this algorithm is run by each non-revoked user *uid.aid.* It outputs new secret key SK*uid.aid*for each non-revokes user *uid.*
- Ciphertext Update by Sever- CTUpdate: this algorithm runs by cloud server. Accepts ciphetext which contain revoked attribute and update key. Outputs new ciphetext CT. This updation is done by re-encryption method.

Though system provides better access control and more efficiency, if any authority fails or crashes the attribute belong to that authority remains unavailable for all users.

2. Two Layer Encryption Approach

The TLE system consist four entities, Data owner (Owner), Data consumer (User), Identity provider (IdP), Cloud.

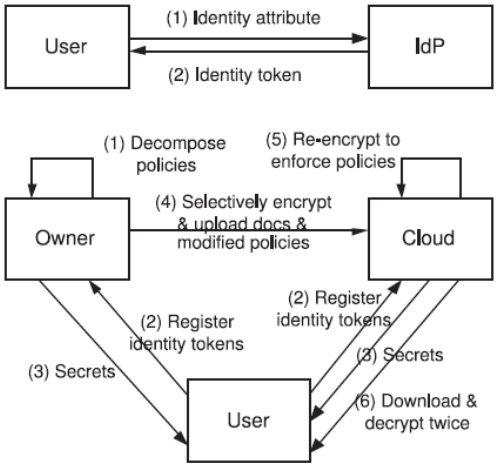

Fig.2. TLE Approach

- Identity Token Issuance: IdPs are trusted third parties which provide identity tokens to User based on their identity attribute which is unique.
- Policy Decomposition: The owner decomposes each ACP into two sub ACPs. It provides two sub sets of ACPs, $ACPB_{owner}$ and $ACPB_{cloud}$. The owner enforces the confidentiality related to $ACPB_{owner}$and cloud enforces the remaining ACPs in $ACPB_{cloud..}$
- Identity Token Registration: Usrs register their identity tokens to cloud as well as owner so that User can get secret to decrypt required data. When user register with owner, the owner issues them two sets of secrets for attribute conditions. The owner keeps one set and gives another set to cloud.
- Data Encryption and Upload: The owner encrypts the data based on the sub ACPs in $ACPB_{owner}$withunique symmetric key, called an ILE keyand uploads it with corresponding public information. The cloud now encrypts the data again gained from owner based on ACPs in $ACPB_{cloud}$with unique symmetric key, called an OLE key.In this case instead of sharing secret key, users are given secret which combine with public information to obtain actual private keys.
- Data Downloading and Decryption: Users download the data from cloud and decrypt it twice to get access of data. ILE key and OLE key both keys can be generated using public information provided by owner and cloud using secret given to User.
- Encryption Evolution management: After the initial encryption is performed, affected data items need to be re-encrypted in case of any changes with assigned credentials with new symmetric key. This process does not involve

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

owner in it, instead cloud generate re-encryption with new symmetric key. If ACPs are modified then owner need to perform new policy decomposition again.

3.Threshold Multi-Authority CP-ABE Access Control Scheme

3.1. (t, n) Threshold Secret Sharing

To address the problem of single point failure, TMACS multi-authority jointly manages whole attribute set but no one has full access over any attribute set.

Assume there are *n* participants in the system, denoted as *P = {P1, P2,…,Pn}.* And threshold value is *t (t<=n).* Firstly each participant in the system *Pi* calculates his/her sub-secret, and then the master key can be set as $S = \sum_{i=1}^{n} Si$. Each participant sends his/her sub-share to the *n-1* participants in the system. After receiving all *n* sub-shares *Pi* calculates his/her master share as *Si.* After that the sharing master secret *S* can be reconstructed by any *t* of the *n* participants.

3.2. System Model

In multi-authority cloud storage system, there exist five entities: a global Certificate Authority (CA), Attribute Authority (AA), data owner (Owner), data consumer (User) and cloud server.

- The Certificate Authority (CA)is a global trusted entity which is responsible for setting system parameters and public key for attributes, assigns unique *uid*for each legal user and unique *aid* for each AA. CA also decides parameter *t* for threshold value of AA that is involved in the user secret key generation.
- The Attribute Authority (AA) is responsible forattribute management and key generation. Here all AA's are jointly managing the whole attribute set. All AA cooperate with each other to share master key. Whenever user request for secret key to AAs. Each AA separately generates its corresponding secret key independently.
- The data owner (Owner) encrypts his/her own files and define who can get access of that data. First of all each owner encrypts his/her data with symmetric key algorithm like AES or DES, formulates access policy over an attribute set and encrypts the symmetric key under that policies. After that owner sends whole encrypted data and encrypted symmetric key to cloud storage. Any user can gain this encrypted data from cloud if and only if the attribute set satisfy the access policy for that user.
- The data consumer (User) is assigned with global unique ID that is *uid*by CA and applies for his/her secret keys from AA with own identification. The user can get interested ciphertext from cloud server but can decrypt it only if he/she satisfies the access policy of attribute set.
- The cloud server only provides platform for owner; the encrypted data can be available by any legal user freely. TMACS helps to reduce this communication overhead. Here rather than entire secret key can be reconstructed by t AAs, The task of key reconstruction is done by user to avoid synchronization overheads.

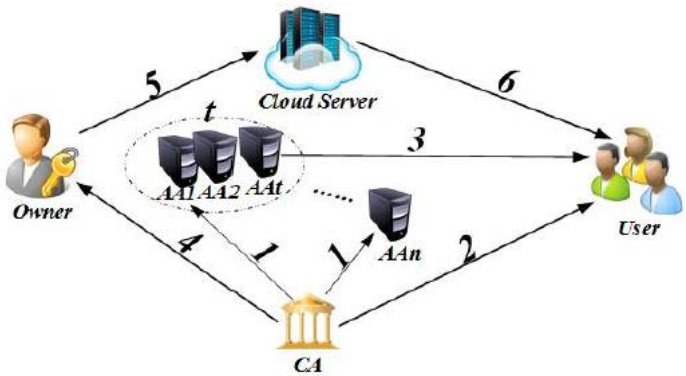

Fig. 3. Framework of TMACS

(1) AA registers to CA to gain *(aid, aid.cert)*;
(2) User registers to CA to gain *(uid, uid.cert)*;
(3) User gains his/her SK from any *t* out of *n* AAs.
(4) Owner gain PK from CA;
(5) Owner upload (CT) to cloud server;

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

(6) Users download (CT) from cloud server;

3.3. Steps in data access control scheme

3.3.1. System Initialization

The operation of system initialization is divided into three sub process: CAsetup1, AASetup and CASetup2.

- CAsetup1: This operation is performed by CA where CA works on user registration and AA registration, and unique ID generation and certification.
- AASetup: All n AA cooperate with each other to call (t, n) threshold secret sharing. Each AA (*AAi, i=1, 2,…, n)*selects the random number *αi.* In this way the master key is implicitly decided: $\alpha = \sum_{i=1}^{n} \alpha i$, from which each AA calculates the sub-share *Si* separately and sends to all *n-1* AAs. After receiving *n-1* sub-shares AA calculate its master key share $ski = \sum_{j=1}^{n} sji$, *(AAj, j=1, 2,…, n)* and further calculate its relevant public key share. Here public key *pki* can be shared with any other entities.
- CASetup2: This operation is run by CA where it calculates the global public key. CA randomly chooses *t* out of *n* AA public key shares.

3.3.2. Encryption

The operation Encryption is performed by data owner independently. Owner first chooses random number *k* as a symmetric key and performs encryption on plaintext *M* using symmetric key *k.* Finally owner sends the encrypted data E_k *(M,)* encrypted by symmetric key *k* to cloud server.

3.3.3. Secret Key Generation

This operation is run by one user and *t* out of n AAs. If less than t AAs is alive in the system then secret key cannot be generated. Therefore user can contact with any *t* AAs to have share of secret key which is generated by each AA separately. After getting *t* shares from any *t* AAs, the user can generate his/her secret key. To gain the secret key share from *AAi*, the user uid_j, first send his/her signed request with his/her identity and sends his certificate $uid_j.certi$to*AAi*.

3.3.4. Decryption

The decryption operation is run by each user. The user can download data according to his/her interest from cloud server. However he/she can't decrypt the data unless his/her attribute set satisfies the access structure hidden inside the ciphertext. After getting secret key from *t* AAs the user can construct symmetric key *k* to decrypt downloaded ciphertext.

Table 1. Comparison between different Multi- authority access controls

Parameters	Reference [3]	Reference [2]	Reference [1]
Access Control	CP-ABE	CP-ABE	(t, n) threshold CP-ABE
Single point failure	Possible	Possible	Not Possible
Encryption Layer	One	Two	One
Key used	Secret key	Secret key	Shared Secret key

IV. CONCLUSION

The cloud accesses mentioned above provide CP-ABE access control scheme where data owner has direct access on who can access data which also support forward as well as backward security. Delegate access control provides TLE which reduce communication overhead and workload on data owner, where cloud and owner together provides access control. The access such as TMACS provides new multi-authority CP-ABE access control. In this scheme all AAs jointly manage whole attribute set and share master key, by providing *(t, n)* threshold secret sharing. Here by interacting with any *t* out of *n AAs* the legal user can generate his/her secret key. Thus TMACS avoids single-point bottleneck of both security and performance. TMACS not only secure when less than *t* authorities are compromised, but also robust when no less then *t* authority alive in the system.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 1, January 2016

REFERENCES

[1] Wei Li, KaipingXue, YingjieXue, and Jianan Hong, "TMACS: A Robust and Verifiable Threshold Multi-Authority Access Control System in Public Cloud Storage", IEEE Transactions on Parallel and Distributed Systems,Vol. PP,Issue 99, pp.1-12, 2015.
[2] Mohamed Nabeel and Elisa Bertino, Fellow, IEEE, "Privacy Preserving Delegated Access Control in Public Clouds", IEEE Transactions on Knowledge and Data Engineering, Vol. 26, Issue 9, pp.2268-2280, 2014
[3] Kan Yang, Student Member, IEEE, and XiaohuaJia, Fellow, IEEE, "Expressive, Efficient, and Revocable Data Access Control for Multi-Authority Cloud Storage", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, Issue 7, pp. 1735-1744,2014
[4] Luca Ferretti, Fabio Pierazzi, Michele Colajanni, and MircoMarchetti, "Scalable Architecture for Multi-User Encrypted SQL Operations on Cloud Database services", IEEE Transactions on Cloud Computing, Vol. 2, Issue4, pp. 448-458, 2014
[5] Luca Ferretti, Fabio Pierazzi, Michele Colajanni, and MircoMarchetti, "Performance and cost evaluation of an adaptive encryption architecture for cloud databases", IEEE Transactions on Cloud Computing, Vol. 2, Issue 2,pp.143-155, 2014
[6] Luca Ferretti, Michele Colajanni, and MircoMarchetti, "Distributed, Concurrent, and Independent Access to Encrypted Cloud Databases", IEEE Transactions on Parallel and Distributed Systems, Vol. 25, Issue 2, pp.437-446,2014

BIOGRAPHY

Archana Vilas Kshirsagaris Student of M.E. Computer Engineering, BharatiVidyapeeth College of Engineering, Mumbai University, Navi Mumbai, Maharashtra, India. She is an Assistant Professor in the Department of I.T., B.N.Bandodkar College of science, Mumbai University,Thane, Maharashtra, India. Her area of interest is Cloud computing and image processing.

KanchanDokeis aProfessorin the Department of I.T. and computer science, BharatiVidyapeeth College of Engineering, Mumbai University, Navi Mumbai, Maharashtra, India. Her area of interest is cloud computing, algorithms and programming.