Advanced Attribute based Encryption & Signature to Control Access on Cloud Storage System

Bachalakuri Sumathi[1] **V.Vijay Kumar**[2]
[1]M.Tech. Student [2]Sr. Assistant Professor
[1,2]Department of Computer Science & Engineering
[1,2]Sir Venkateshwara Engineering College, Suryapet

Abstract— This survey proposes an incipient decentralized access control scheme for secure data storage in clouds which fortifies in nominate authentication. The cloud verifies the authenticity of the series without paramount cognizance in the user’s identity afore storing information. This scheme additionally has the integrated feature of access control. In access control scheme only valid users are able to decrypt the stored data/information. This scheme averts replay attacks additionally fortifies engenderment, modification, and reading information stored in the cloud. These schemes additionally address utilize revocation. Moreover, the authentication and access control scheme is decentralized and robust in nature unlike other access control schemes designed for clouds which are centralized. The computation, communication, and storage overheads are commensurable to centralized approaches.

Key words: Decentralized Access, Access Control, Attribute Based Encryption, Attribute Based Signature, Cloud Storage

I. Introduction

Cloud computing is a promising computing model which currently has drawn far reaching consideration from both the educational community and industry. By joining a set of existing and new procedures from research areas, for example, Service-Oriented Architectures (SOA) and virtualization, cloud computing is viewed all things considered a computing model in which assets in the computing infrastructure are given as services over the Internet. It is a new business solution for remote reinforcement outsourcing, as it offers a reflection of interminable storage space for customers to have data reinforcements in a pay-asyou- go way [1]. It helps associations and government offices fundamentally decrease their financial overhead of data administration, since they can now store their data reinforcements remotely to thirdparty cloud storage suppliers as opposed to keep up data centers on their own. Numerous services like email, Net banking and so forth… are given on the Internet such that customers can utilize them from anyplace at any time. Indeed cloud storage is more adaptable, how the security and protection are accessible for the outsourced data turns into a genuine concern. The three points of this issue are availability, confidentiality and integrity. To accomplish secure data transaction in cloud, suitable cryptography method is utilized. The data possessor must encrypt the record and then store the record to the cloud. Assuming that a third person downloads the record, they may see the record if they had the key which is utilized to decrypt the encrypted record. Once in a while this may be failure because of the technology improvement and the programmers. To overcome the issue there is lot of procedures and techniques to make secure transaction and storage.

Apart from the technical solutions to ascertain security and privacy, there is withal a desideratum for law enforcement. Cloud servers are prone to Byzantine failure, where a storage server can fail in arbitrary ways. The cloud is additionally prone to data modification and server colluding attacks. In server colluding attack, the adversary can compromise storage servers, so that it can modify data files as long as they are internally consistent. To provide secure data storage, the data needs to be encrypted. However, the data is often modified and this dynamic property needs to be taken into account while designing efficient secure storage techniques. Efficient search on encrypted data is withal a paramount concern in clouds. The clouds should not ken the query but should be able to return the records that gratify the query. This is achieved by betokens of searchable encryption. Access control in clouds is gaining attention because it is paramount that only sanctioned users have access to valid accommodation.

II. Related work

A. *Existing System:*

Access control in clouds is gaining attention because it is important that only authorized users have access to valid service. A huge amount of information is being stored in the cloud, and much of this is sensitive information. Care should be taken to ensure access control of this sensitive information which can often be related to health, important documents (as in Google Docs or Dropbox) or even personal information (as in social networking). There are broadly three types of access control: User Based Access Control (UBAC), Role Based Access Control (RBAC), and Attribute Based Access Control (ABAC).

In UBAC, the access control list (ACL) contains the list of users who are authorized to access data. This is not feasible in clouds where there are many users. In RBAC (introduced by [1]), users are classified based on their individual roles. Data can be accessed by users who have matching roles. The roles are defined by the system. For example, only faculty members and senior secretaries might have access to data but not the junior secretaries. The ABAC is more extended in scope, in which users are given attributes, and the data has attached access policy. Only users with valid set of attributes, satisfying the access policy, can access the data. For instance, in the above example certain records might be accessible by faculty members with more than 10 years of research experience or by senior secretaries with more than8 years experience. The pros and cons of RBAC and

ABAC are discussed in [2]. A writer whose attributes and keys have been revoked cannot write back stale information. The receiver receives attributes and secret keys from the attribute authority and is able to decrypt information if it has matching attributes.

In Cipher textpolicy, CP-ABE the receiver has the access policy in the form of a tree, with attributes as leaves andmonotonic access structure with AND, OR and other threshold gates. All the approaches take a centralized approach and allow only one KDC, which is a single point of failure. Chase [14] proposed a multi-authority ABE, in which there are several KDC authorities (coordinated by a trusted authority) which distribute attributes and secret keys to users. Multiauthority ABE protocol was studied in which required no trusted authority which requires every user to have attributes from at all the KDCs.

A multi-authority Ciphertext-Policy Attribute-Based Encryption system is comprised of the following five algorithms: Global Setup(λ) → GP The global setup algorithm takes in the security parameter λ and outputs global parameters GP for the system. Authority Setup(GP) → SK, PK Each authority runs the authority setup algorithm with GP as input to produce its own secret key and public key pair, SK, PK.

B. *Proposed System:*

We propose a new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. In the proposed scheme, the cloud verifies the authenticity of the series without knowing the user's identity before storing data. Our scheme also has the added feature of access control in which only valid users are able to decrypt the stored information. The scheme prevents replay attacks and supports creation, modification, and reading data stored in the cloud.

1) *Advantages of Proposed System:*

- Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them.
- Authentication of users who store and modify their data on the cloud.
- The identity of the user is protected from the cloud during authentication.

III. IMPLEMENTATION

Distributed access control of data stored in cloud so that only authorized users with valid attributes can access them. The identity of the user is protected from the cloud during authentication. The architecture is decentralized, meaning that there can be several KDCs for key management.

The access control and authentication are both collusion resistant, meaning that no two users can collude and access data or authenticate themselves, if they are individually not authorized. Revoked users cannot access data after they have been revoked. The proposed scheme is resilient to replay attacks. A writer whose attributes and keys have been revoked cannot write back stale information. The protocol supports multiple read and write on the data stored in the cloud. The costs are comparable to the existing centralized approaches, and the expensive operations are mostly done by the cloud. Authentication of users who store and modify their data on the cloud. The identity of the user is protected from the cloud during authentication.

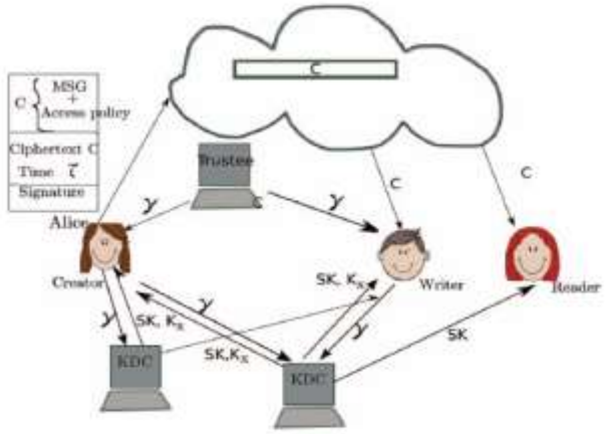


Fig 1: Architecture Diagram.

The architecture is decentralized, meaning that there can be several KDC's for key management. There are three users, a creator, a reader and writer. Creator Alice receives a token γ from the trustee, who is assumed to be honest. A trustee can be someone like the federal government who manages social insurance numbers etc. On presenting her id the trustee gives her a token γ.. For example, these can be servers in different parts of the world. A creator on presenting the token to one or more KDCs receives keys for encryption/decryption and signing. In the Fig. 1, SKs aresecret keys given for decryption, Kx are keys for signing. The message MSG is encrypted under the access policy X. The access policy decides who can access the data stored in the cloud. The creator decides on a claim policy Y, to prove her authenticity and signs the message.

A. *Creation of KDC*

To create a different number of KDC's given a input as KDC name, KDC id and KDC password it will save in a database and to register a user details given an input as user name and user-id.

B. *User Enrollment*

After KDC given a user id to a user, the user will enrolled the personal details to KDC's given an input as user-name user-id, password etc. The KDC will be verify the user details and it will insert it in a Database.

C. *Trustee and User Accessibility*

User can login with their credentials and request the token from trustee for the file upload using the user id. After the user id received by the trustee, trustee will be create token using user id,key and user signature(SHA). Then the trustee will issue a token to the particular user and then trustee can view the logs.

D. *Creation of Access Policy*

After trustee token issuance for the users, the users produce the token to the KDC then the

token verify by the KDC if it is valid then KDC will provide the public and Private key to the user. After users received the keys the files are encrypt with the public keys and set their Access policies (privileges).

E. File Accessing

Using their access policies the users can download their files by the help of kdc's to issue the private keys for the particular users.

F. Hash algorithm

Definition: SHA-1 is one of several cryptographic hash functions, most often used to verify that a file has been unaltered. SHA is short for Secure Hash Algorithm. File verification using SHA-1 is accomplished by comparing the checksums created after running the algorithm on the two files you want to compare. SHA-1 is the second iteration of this cryptographic hash function, replacing the previous SHA-0. An SHA-2 cryptographic hash function is also available and SHA-3 is being developed. One iteration within the SHA-1 compression function. A, B, C, D and E are 32bit words of the state. F is a nonlinear function that varies. N denotes a left bit rotation by n places. N varies for each operation. Wt is the expanded message word of round t. Kt is the round constant of round t. denotes addition modulo 232.

G. Paillier Algorithm

The Paillier cryptosystem, named after and invented by Pascal Paillier is a probabilistic asymmetric algorithm for public key cryptography.

1) Key generation

Choose two large prime number p and q randomly and independently of each other such that gcd (pq,(p-1)(q-1))=1. This property is assured if both primes are of equivalent length, i.e p,q {0,1 }s-1 for security parameter S . Compute n=pq and λ=lcm(p-1,q-1). Select random integer g where gZ*n2. Ensure n divides the order of g by checking the existence of the following modular multiplicative inverse. μ= (L(gλ mod n2))-1mod n, where function L is defined as The public (encryption) key is (n,g).. The private (decryption) key is (λ,μ).

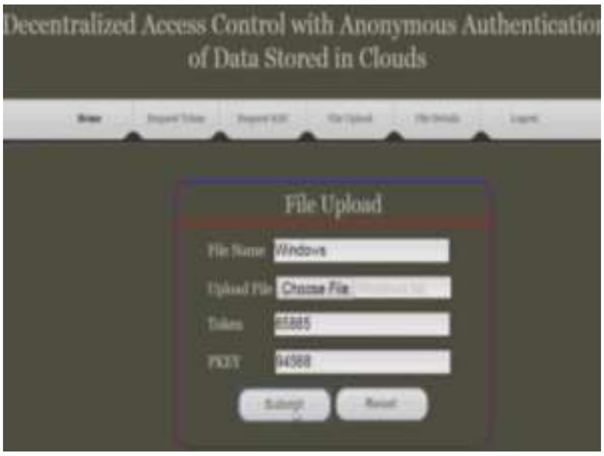
2) Encryption

Let m be a message to be encrypted where m Zn. Select random r where r Z*n. Compute cipher text as: c= gm .rn mod n2

3) Decryption

Cipher text: cZ*n2.Compute message: m =L(cλ mod n2).M mod n

IV. EXPERIMENTAL RESULTS



Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds

File Upload

File Name: Windows

Upload File: Choose File

Token: 85885

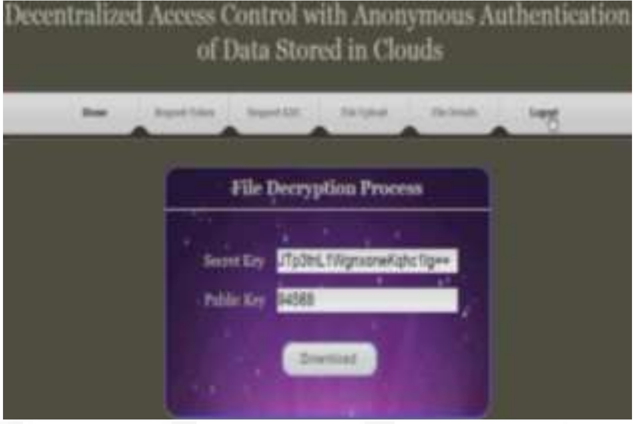
PKEY: 94568

Fig. 2: User File Upload Page.

Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds

Token ID

Fig 3: User Token receiving Page.



Decentralized Access Control with Anonymous Authentication of Data Stored in Clouds

File Decryption Process

Secret Key: lTp3thL fWgnsoneKqhcTlg==

Public Key: 94568

Fig 4: User File Decrypt Process using keys.

V. CONCLUSION

We propose secure cloud storage utilizing decentralized access control with incognito authentication. The files are associated with file access policies, that used to access the files placed on the cloud. Uploading and downloading of a file to a cloud with standard Encryption/Decryption is more secure. Revocation is the paramount scheme that should abstract the files of revoked policies. So no one can access the revoked file in future.

The policy instauration is made as facile as possible. The instaurate key is integrated to the file. Whenever the utilizer wants to instaurate the files he/she may directly download all instaurate keys and made changes to that keys, then upload the incipient instaurate keys to the files stored in the cloud.

REFERENCES

[1] D. F. Ferraiolo and D. R. Kuhn, “Role-based access controls,” in 15th National Computer Security Conference, 1992.

[2] D. R. Kuhn, E. J. Coyne, and T. R. Weil, “Adding attributes to rolebased access control,” IEEE Computer, vol. 43, no. 6, pp. 79–81, 2010.

[3] M. Li, S.Yu, K. Ren, and W.Lou, “Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings,” in SecureComm, 2010, pp. 89– 106.

[4] S. Yu, C. Wang, K. Ren, and W. Lou, “Attribute based data sharing with attribute revocation,” in ACM ASIACCS, 2010, pp. 261–270. [5] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryption for

fine-grained access control in cloud storage services," in ACM CCS, 2010, pp. 735–737.

[5] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-Based Authentication for Cloud Computing," Proc. First Int'l Conf. Cloud Computing (CloudCom), pp. 157-166, 2009.

[6] C.Gentry, "A Fully Homomorphic Encryption Scheme," PhD dissertation, Stanford Univ., http://www.crypto.stanford.edu/craig