International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 03 Issue: 08 | Aug-2016 www.irjet.net p-ISSN: 2395-0072

Privacy Preserving using KP-ABE Technique in Cloud Computing

B. Nanda Kishore, Dr. G. Naga Rama Devi, Professor

B.Nanda Kishore, PG Scholar, Dept, of CSE, CREC, Tirupati, AP, India
Dr. G. Naga Rama Devi, Professor &HOD, Dept. of CSE, CREC, Tirupati, AP, India

Abstract: *Symmetric key algorithm uses same key for both encryption and decryption. The authors take a centralized approach where a single key distribution center (KDC) distributes secret keys and attributes to all users. A new decentralized access control scheme for secure data storage in clouds that supports anonymous authentication. The validity of the user who stores the data is also verified. The proposed scheme is to hide the user's attributes using SHA algorithm. The KP-ABE cryptosystem is a asymmetric algorithm for public key cryptography. ABE algorithm use for Creation of access policy, file accessing and file restoring process and also hiding the access policy to the user using query based algorithm.*

Index Terms: user privacy Anonymity,Multi-authority, attribute-based encryption.

I. INTRODUCTION

The mainstay of this is to propose a new decentralized Access control scheme for secure data storage in clouds that supports anonymous authentication. The proposed is hiding the access policy to the user(access policy hidden) using query based algorithm and using SHA algorithm we are hiding the users attributes. A writer whose attributes and keys have been revoked cannot write back stale information. 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. 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 Revoked users cannot access data after they have been revoked.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 A writer whose attributes and keys have been revoked cannot write back stale information.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 writes 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. Proposing privacy preserving authenticated access control scheme. According to our scheme a user can create a file and store it securely in the cloud. This scheme consists of use of the two protocols ABE and ABS. The cloud verifies the authenticity of the user without knowing the user's identity before storing data. The 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.

II. RELATED WORK

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 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 (ABE). 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 ABE 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]. There has been some work on ABAC in clouds (for example, [3], [4], [5], [6], [7]). All these work use a cryptographic primitive known as Attribute Based Encryption (ABE). ABE was proposed by Sahai and Waters [10]. In ABE, a user has a set of attributes in addition to its unique ID. There are two classes of ABEs. In Key-policy ABE or KP- ABE (Goyal et al. [11]), the sender has an access policy to encrypt data. 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. 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. Multi- authority ABE protocol was studied in [5], [6], which required no trusted authority which requires every user to have attributes from at all the KDCs. A multi-authority Cipher text-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.

Encrypt(M,(A,ρ),GP,{PK}) → CT The encryption algorithm takes in a message M, an access matrix (A,ρ), the set of public keys for relevant authorities, and the global parameters. It outputs a ciphertext CT.

KeyGen(GID,GP,i,SK) → Ki,GID The key generation algorithm takes in an identity GID, the global parameters, an attribute i belonging to some authority, and the secret key SK for this authority. It produces a key Ki, GID for this attribute, identity pair.

Decrypt(CT,GP,{Ki,GID}) → M The decryption algorithm takes in the global parameters, the ciphertext, and a collection of keys corresponding to attribute, identity pairs all with the same fixed identity GID. It outputs either the message M when the collection of attributes i satisfies the access matrix corresponding to the ciphertext. Otherwise, decryption fails.

Recently, Lewko and Waters [9] proposed a fully decentralized ABE where users could have zero or more attributes from each authority and did not require a trusted server. In all these cases, decryption at user's end is computation intensive. So, this technique might be inefficient when users access using their mobile devices. To get over this problem, Green et al. [1] proposed to outsource the decryption task to a proxy server, so that the user can compute with minimum resources (for example, hand held devices). However, the presence of one proxy and one key distribution center makes it less robust than decentralized approaches. Both these approaches had no way to authenticate users, anonymously.. presented a modification of , authenticate users, who want to remain anonymous while accessing the cloud.

To ensure anonymous user authentication Attribute Based Signatures were introduced by Maji et al. [8]. This was also a centralized approach. A recent scheme by the same authors [9] takes a decentralized approach and provides authentication without disclosing the identity of the users. However, as mentioned earlier in the previous section it is prone to replay attack.

III. PROPOSED WORK

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 KP-ABE 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 writes 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. 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 are secret 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.UserEnrollment

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.TrusteeandUser 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.Hashalgorithm.

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.

IV. CONCLUSIONS

A decentralized access control technique with anonymous authentication, which provides user revocation and prevents replay attacks, is achieved. The cloud does not know the identity of the user who stores information, but only verifies the user's credentials. Key distribution is done in a decentralized way and also hides the attributes and access policy of a user. One limitation is that the cloud knows the access policy for each record stored in the cloud. In future, using SQL queries for hide the attributes and access policy of a user. Files stored in cloud can be corrupted. So for this issue using the file recovery technique to recover the corrupted file successfully and to hide the access policy and the user attributes.

V.REFERENCES

[1 D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls,"in15th National Computer Security Conference, 1992.

[2]D. R. Kuhn, E. J. Coyne, and T. R. Weil, "Adding attributes to role- based 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.

[6] F. Zhao, T. Nishide, and K. Sakurai, "Realizing fine-grained and flexible access control to outsourced data with attribute-based cryptosystems," in ISPEC, ser. Lecture Notes in Computer Science, vol. 6672. Springer, 2011, pp. 83–97.

[7] S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed access control in clouds," in IEEE TrustCom, 2011.

[8] H. K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-based signatures: Achieving attribute-privacy and collusion-resistance," IACR Cryptology ePrint Archive, 2008.