IPHV8I1026X

International Journal Of Advanced Research and Innovation -Vol.8, Issue .I

ISSN Online: 2319 – 9253
Print: 2319 – 9245

EFFICIENT ATTRIBUTE BASED SIGNATURE SCHEME FOR USER ACCESS CONTROL IN DATABASE ENVIRONMENT

[#1]PEDDI KISHOR, Associate Professor& HOD, Dept of CSE,

[#2]N.DIVYA, Dept of CSE,

SREE CHAITANYA INSTITUTE OF TECHNOLOGICAL SCIENCES, KARIMNAGAR, T.S., INDIA.

Abstract: Security and privacy are very important issues in cloud computing. In existing system access control in clouds are centralized in nature. The scheme uses a symmetric key approach and does not support authentication. 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. Specifically, the proposed work integrates key management from pseudorandom number generator for unlink ability, a suitable indexing technique for maintaining confidential keyword based search which hides both surfing and data access patterns based on repetitive structures, and also binds up attribute based encryption with threshold signature exchange with audit ability for issuing role-based access control to prevent potential misbehavior, in both normal and emergency cases. The system devises mechanisms that can detect whether users health data have been illegally distributed and identifies possible sources of leakage. The main intruding network and the authorized party that did it will also be found, thus destroying the particular request of an intruder by crashing the attributes assigned to them. In this paper, an ABE encryption and outsourced decryption with verification and recovery technique is proposed. This technique effectively secures the data and also provides the correctness of the retrieved data along with the recovery mechanism for the transmitted data in case of malicious attack. The implementation of this scheme will show the correctness of the secure data storage and the recovery process.

Keywords: Access control, auditability, eHealthcare, Privacy, Threshold, ABE.

I.INTRODUCITON

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.

Internet technology is growing quickly, and people can process, store, or share with their data by using its ability. Recently, the cloud has emerged to provide various application services to satisfy user's requirement. Cloud computing provides the tools and technologies to build data/compute intensive parallel applications with much more affordable prices compared to traditional parallel computing techniques. It has three servicing models Infrastructure as a Service, Software-as-a-Service, and Platform as a Service. Saas type of cloud computing allow users to run existing online applications. It delivers a single application through the browser to thousands of customers using a multitenant architecture. PaaS allow users to run their own applications using supplier specific tools and languages. It comprises the environment for developing and provisioning cloud applications. The users of this layer are developers seeking to develop and run a cloud application for a particular platform. Iaas services on the infrastructure layer allow user to run any application they please on cloud hardware on their own choice. IT resources that are combined under the heading Infrastructure-as-a-Service (IaaS) include services linked to computing resources, data storage resources, and the communications channel. In the storage service application, the cloud can let the user, data owner to store his data, and share this data with other users via the cloud, because the cloud can provide the pay as you go environment where people just need to pay the money for the storage space they use. For protecting the confidentiality of the stored data, the data must be encrypted before uploading to the cloud. The encryption scheme used is attribute-based encryption. The ABE scheme used a user's identity as attributes, and a set of attributes were used to encrypt and decrypt data. One of the main efficiency drawbacks of the most existing ABE schemes is that decryption is expensive for resource-limited devices due to pairing operations, and the number of pairing operations required to decrypt a cipher text grows with the complexity of the access policy. The ABE scheme can result the problem that data owner needs to use every authorized user's public key to encrypt data. Key-policy attribute-based encryption (KP-ABE) scheme built the access policy into the user's private key and described the encrypted data with user's attributes. The KP-ABE scheme can achieve the grained access control and more exibility to control users than ABE scheme. But the disadvantage of KP-ABE is that the access policy is built into an user's private key, so data owner can't choose who can decrypt the data except choosing a set of attributes which can describe this data. And it is unsuitable in certain application because a data owner has to trust the key issuer. CP-ABE scheme built the access policy into the encrypted data; a set of attributes is in a user's key. The CP-ABE scheme addresses the problem of KP-ABE that data owner only trusts the key issuer [13]. To assess the performance of our ABE scheme with verifiable outsourced decryption, we implement the CP-ABE scheme with verifiable outsourced decryption and conduct experiments.

II. RELATED WORKS

As a lot of sensitive information is shared and hold on by third-party sites on the net, there'll be a desire to cipher information hold on at these sites. One disadvantage of encrypting information is that it will be by selection shared solely at a coarse-grained level (i.e., giving another party your A Verifiable Cloud Storage using Attribute Based Encryption and Outsourced Decryption with Recoverability personal key). Goyal et al [2] proposed a scheme for fine-grained sharing of encrypted information that it has the tendency to developed Key-Policy Attribute-Based coding. In that, attributes and personal keys are related to access structures that manage the cipher texts that the user is ready to rewrite. It didn't hide the set of attributes underneath that the information is encrypted. Cheung et al [1] proposed a ciphertext policy attribute-based coding (CP-ABE), every secret key is associated with a set of attributes, and each ciphertext is associated with access structure on attributes. Secret writing is enabled if and only if the user’s attribute set satisfies the ciphertext access structure. This provides fine-grained access management on shared information in several sensible settings, likewise as secure databases and secure multicast. It gifts a variant with well smaller ciphertexts and faster encryption/decryption operations. The most arrange is to form a hierarchy of attributes, so fewer cluster components square measure required to represent all attributes within the system. This economical variant is proven to be controller secure. Herranz et al [5] proposed the first attribute-based encryption (ABE) schemes allowing for truly expressive access structures and with constant ciphertext size. First it results a ciphertext-policy attribute-based encryption (CP-ABE) scheme with O(1)-size ciphertexts for threshold access policies and where private keys remain as short as in previous systems. As a second result, a certain class of identity-based broadcast encryption schemes generically yields monotonic key-policy attribute-based encryption (KP-ABE) systems in the selective set model. Waters et al [4] proposed ciphertext-policy attribute based encryption. to kept encrypted data confidential even if the storage server is untrusted. It is secure against collusion attacks. Previous ABE systems used attributes to describe the encrypted data and built policies into user’s keys and a party encrypting data determines a policy for who can decrypt. Thus, their methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). Raykova et al [3] proposed the VC scheme that verifies any function in the class of functions covered by the permissible ABE policies (currently Boolean formulas). It is very efficient verification algorithm that depends only on the output size.[7] presented a multi-function VC scheme allows the verifiable evaluation of multiple functions on the same pre-processed input. Green et al [12] proposed Outsourced decryption; it has two keys called secret key and transformation key. Proxy will be able to transform any ABE cipher text into a short cipher text for the user. While the security definitions show that an attacker will not be able to learn an encrypted message, there is no guarantee on the transformation's correctness. Junzuo et al [20] proposed a scheme to verify the retrieved content. It just gives a result whether the content is original or modified. It does not specify where the data get modified. If it results the content is modified then the data cannot be used without knowing the where the modification is present.

III. PRELIMINARIES

A. Access Tree

Let T be a tree representing an access structure. Each non-leaf node is described by its children and a threshold value. Non-leaf nodes of the tree represented as threshold gate. If numx.is the number of children of a node x and kx is its threshold value, then 0 < kx ≤ numx. When kx= 1, the threshold gate is an OR gate and when kx = numx, it is an AND gate. Each leaf node x of the tree is described by an attribute and a threshold value kx = 1. To facilitate working with the access trees, we define a few functions. Parent of node x is denoted as parent(x). The function att(x) is denotes the attribute associated with the leaf node x in the tree. In a access tree T the children of a node are numbered from 1 to n(num) . It also defines an ordering between the children of every node. For a node in an access structure, based on the given key the index values are uniquely assigned in an arbitrary manner. The function index(x) returns a number associated with the node x.

B. Cyclic Group

In algebra, a cyclic group is a group that is generated by a single element. Every element can be written as a power of some particular element "g" in multiplicative notation, or .as a multiple of "g" in additive notation. This element "g" is called a "generator" of the group. Any infinite cyclic group is isomorphic to Z, the integers with addition as the group operation. Any finite cyclic group of order n is isomorphic to Z/nZ, the integers modulo n with addition as the group operation [16].

C .Bilinear Paring Algorithm

Pairing-based cryptography is the use of a pairing between elements of two cryptographic groups to a third group to construct cryptographic systems. If the same group is used for the first two groups, the pairing is called symmetric and is a mapping from two elements of one group to an element from a second group [15]. In this way, pairings can be used to reduce a hard problem in one group to a different, usually easier problem in another group.

D. Secured Hashing Algorithm

There are several similarities in the evolution of hash function and that of symmetric block ciphers. We have seen that the increasing power of brute-force attacks[12].Cryptanalysis have led to the decline in the popularity of DES and in the design of newer algorithm with longer key lengths and with features designed to resist specific cryptanalytic attacks. Similarly, advances in computing power and hash function cryptanalysis have led to the decline in the popularity of first MD4 and then MD5, two very popular hash functions. In response, newer hash algorithm have been developed with longer hash code length and with features designed to resist specific cryptanalytic attacks. 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. [33] 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. Yang *et al.* [34] presented a modification of [33], 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.* [23]. This was also a centralized approach. A recent scheme by the same authors [24] 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.

IV. PROPOSED WORK

The main contributions of this paper are the following:

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 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.

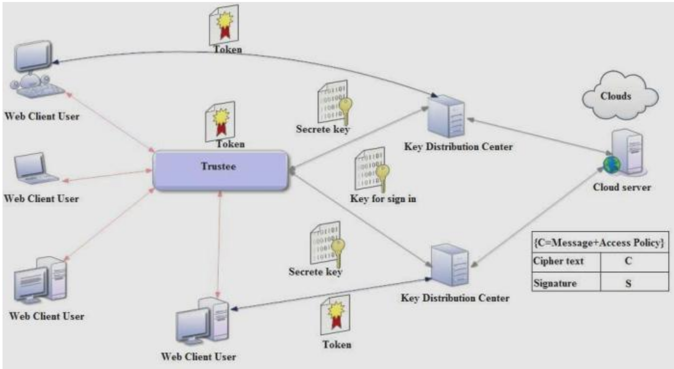

Fig 1: Secure Cloud storage model

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 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 messageunder this claim. The ciphertext C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the ciphertext C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message. Write proceeds in the same way as file creation. By designating the verification process to the cloud, it relieves the individual users from time consuming verifications. When a reader wants to read some data stored in the cloud, it tries to decrypt it using the secret keys it receives from the KDCs. If it has enough attributes matching with the access policy, then it decrypts the information stored in the cloud.

A. Creation of KDC

Different number of KDC's are created and to register a user details. KDC name, KDC id and KDC password are given as input to create KDC. Inputs will save in a database and to register a user details given a input as username and user id.

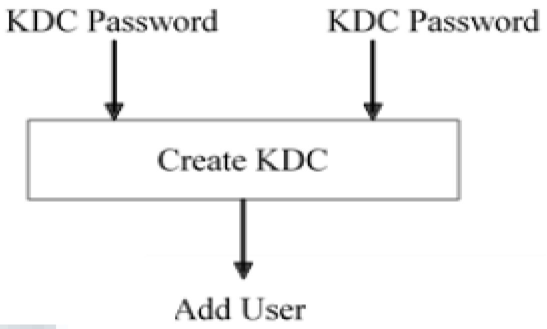

Fig 2: Creation of KDC

B. KDC Authentication

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

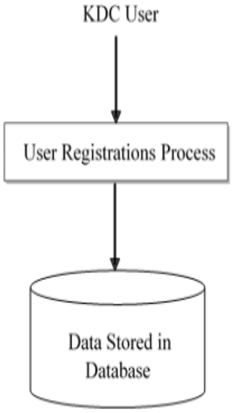

Fig 3: KDC Authentication

C. Trustee and User Accessibility

Users can get the token from trustee for the file upload. After trustee was issuing a token, trustee can view the logs. 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).

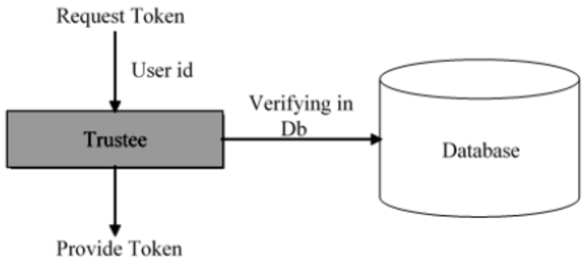

Fig 4: User Accessibility

D. Creation of access policy

After the key was received by the User, the message MSG is encrypted under the access policies. The access policies decide who can access the data stored in the cloud. The cipher text C with signature is c, and is sent to the cloud. The cloud verifies the signature and stores the cipher text C. When a reader wants to read, the cloud sends C. If the user has attributes matching with access policy, it can decrypt and get back original message and user can upload the file after user get key from the KDC.

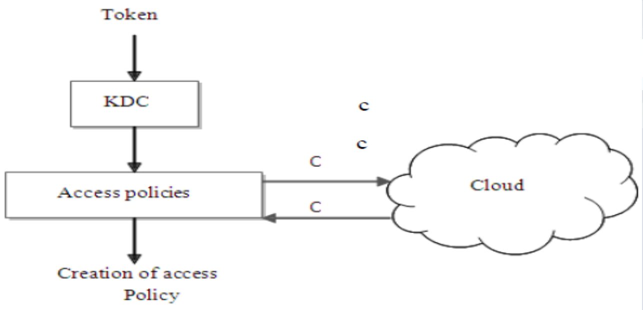

Fig 5: Creation of access policy

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. 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).

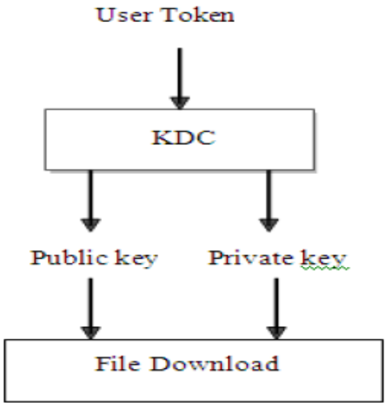

Fig 6: File accessing

F. File Restoration

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.

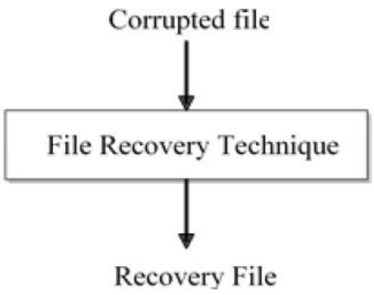

Fig 7: File Restoration

G. Secure 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 2^{32}.

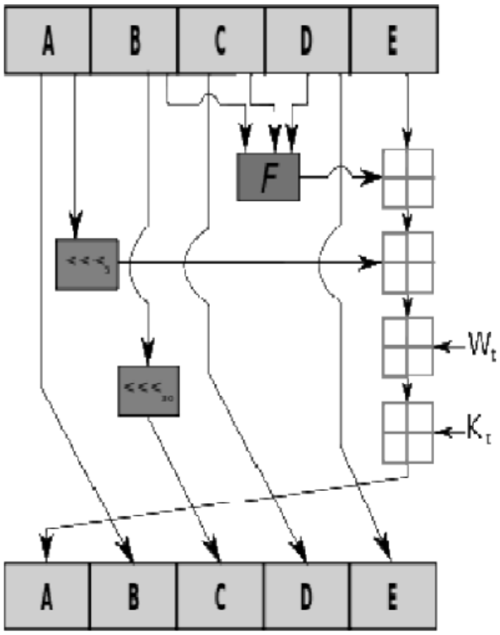

Fig 8: Secure Hash Algorithm

H. Paillier Algorithm

The Paillier cryptosystem, named after and invented by Pascal Paillier is a probabilistic asymmetric algorithm for public key cryptography. *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= $(L(g_{\lambda}$ mod $n_2))_{-1}$mod n, where function L is defind as The public (encryption) key is (n,g).. The private (decryption) key is (λ,μ).

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

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

V.CONCULSION

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 hide 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.

REFERENCES

[1] S. Ruj, M. Stojmenovic and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds", *IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 556–563, 2012.

[2] C. Wang, Q. Wang, K. Ren, N. Cao and W. Lou, "Toward Secure and Dependable Storage Services in Cloud Computing", *IEEE T. Services Computing*, vol. 5, no. 2, pp. 220–232, 2012.

[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in *IEEE INFOCOM.* , pp. 441–445, 2010.

[4] S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Financial Cryptography Workshops*, ser. Lecture Notes in Computer Science, vol. 6054. Springer, pp. 136–149, 2010.

[5] H. Li, Y. Dai, L. Tian, and H. Yang, "Identity-based authentication for cloud computing," in *CloudCom*, ser. Lecture Notes in Computer Science, vol. 5931. Springer, pp. 157–166, 2009.

[6] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009, http://www.crypto.stanford.edu/craig.

[7] A.-R. Sadeghi, T. Schneider, and M. Winandy, "Token-based cloud computing," in *TRUST*, ser. Lecture Notes in Computer Science, vol. 6101. Springer, pp. 417–429, 2010.

[8] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "Trust cloud: A framework for accountability and trust in cloud computing," HP Technical Report HPL-2011-38. Available at http://www.hpl.hp.com/techreports/2011/HPL-2011-38.html.

[9] R. Lu, X. Lin, X. Liang, and X. Shen, "Secure Provenance: The Essential of Bread and Butter of Data Forensics in Cloud Computing," in *ACM ASIACCS*, pp. 282–292, 2010.

[10] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," in *15th National Computer Security Conference*, 1992.

[11] 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.

[12] 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 multiowner settings," in *SecureComm*, pp. 89–106, 2010.

[13] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in *ACM ASIACCS*, pp. 261–270, 2010.

[14] G. Wang, Q. Liu, and J. Wu, "Hierarchical attribute-based encryption for fine-grained access control in cloud storage services," in *ACM CCS*, , pp. 735–737, 2010.