International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 12 Issue 2 –JANUARY 2015.

Entrusting Data with Pliable, High Quality Access Control in Cloud Computing

M.PRASANTH KUMAR[1], S.SRAVANI[2], and G.PEDDIRAJU[3]

[1]*M.Tech (CSE), CRIT, Affiliated to JNTUA University*

[2]*Assistant Professor, Dept of CSE, CRIT, Affiliated to JNTUA University*

[3]*Assistant Professor, Dept of CSE, Noble College of Engineering, Hyderabad*

Abstract— Trust negotiation is an approach to establishing trust between strangers through iterative disclosure of digital credentials. In this model, the access control policy for a resource is usually unknown to the party requesting access to the resource, when trust negotiation starts. Many organizations outsource their data management needs to an external service provider to relief the burden of storage management, universal data access with independent geographical locations, and avoidance of capital expenditure on hardware, software, personnel maintenance.

To ensure the data security and integrity in cloud storage, CP-ABE scheme a promising cryptographic solution, enables data owners to define their own access policies over user attributes and enforce the policies on the data to be distributed. Eventhough, eliminates the need to rely on the data storage server for preventing unauthorized data access and integrity, this scheme is not scalable, flexible; do not provide fine grained access control when it involves multiple value assignments for access expiration time and user revocation.

The proposed scheme not only achieves scalability due to its hierarchical structure, but also inherits flexibility and fine-grained access control in supporting compound attributes of ASBE.

Keywords: Access control, user revocation, HASBE.

I. Introduction

In a managed system security domain, the system either grants or denies an entity's requests to access certain resources according to its access control policies and the authenticated identities of the requester. Open systems such as Internet provide an environment where two or more parties who are virtually strangers to each other can make connections, conversations and do business together. As the communicating parties come from different domains, the identity information such as user names and passwords, or identity certificates, is usually inadequate to determine whether or not a party should be trusted. In such systems, the clients either unconditionally disclose their information to the server, or do not get the service at all. Domain policies are those which state conditional requirements and restrictions placed on the session. Through policy, a system may address needs of all communication participants in real time. A group security policy is a statement of the entirety of security relevant parameters and facilities used to implement the group. When it comes to the perspective of cloud computing, the prominent security concerns are about the data security and privacy in cloud computing because of its Internet- based data storage and management. In addition to the data confidentiality, flexible and fine-grained access control is also strongly desired in the service-oriented cloud computing model.

In this paper, we propose a hierarchical attribute-set-based encryption (HASBE) scheme for access control in cloud computing. This scheme extends the CP-ASBE (ciphertext-policy attribute- set-based encryption) for the system users to achieve scalable, flexible and fine-grained access control.

II. EXISTING SYSTEM

Access control is a classic security area of research where various security models has been proposed since 1960s. Bell-La Padula and BiBa are the two significant security models among them. Further the schemes which are been proposed to achieve flexibility and a fine-grained access control are been applicable only to the data owners and service providers but within the same trusted domain. The conventional method to protect sensitive data outsourced to third parties is to store encrypted data on servers, while the decryption keys are disclosed to authorize users only. But this scheme requires an efficient key management but lacks scalability and flexibility.

In the existing CP-ABE schemes key authority that generates public and secret parameters for CP-ABE. It is in charge of issuing, revoking, and updating attribute keys for users. It grants differential access rights to individual users based on their attributes. A data owner is responsible for defining (attribute -based) access policy, and enforcing it on its own data by encrypting the data under the policy before distributing it.

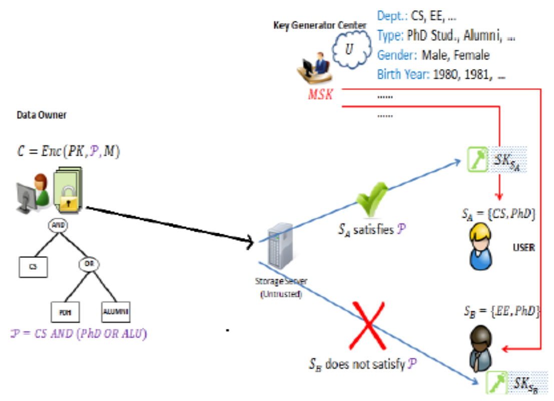
Key Generator Center
Dept.: CS, EE, ...
Type: PhD Stud., Alumni, ...
Gender: Male, Female
Birth Year: 1980, 1981, ...
MSK
Data Owner
$C = Enc(PK, \mathcal{P}, M)$
AND
CS
OR
PDH
ALUMNI
$\mathcal{P}$ = CS AND (PhD OR ALU)
Storage Server (Untrusted)
S_A satisfies $\mathcal{P}$
SK_{S_A}
$S_A = \{CS, PhD\}$
USER
$S_B = \{EE, PhD\}$
S_B does not satisfy $\mathcal{P}$
SK_{S_B}

Fig 1: CP-ABE based storage architecture

In a CP-ABE scheme, decryption keys only support user attributes that are organized logically as a single set. CP-ABE scheme facilitates a recursive set based tree structure, and user can decrypt a ciphertext with a given key if and only if there is an assignment of attributes from the private key to nodes of the tree such that the tree is satisfied. In KP-ABE, each attribute private key is associated with an access structure that specifies for which type of cipher texts the key is able to decrypt, and cipher text is labeled with sets of attributes. In this scheme we have a monotonic access structures for key policies whereas in CP-ABE scheme a user's key is associated with a set of attributes and an encrypted ciphertext will specify an access policy over attributes.

III. Proposed System

The proposed implementation for HASBE scheme deals with the ASBE algorithm with a hierarchical user structure can be successfully applied for hierarchical user grant , data file creation, file access, user revocation, and file deletion. In this scheme, a data encryptor is the one who specifies an access structure for a cipher text, refers to a cipher text policy. Only the users with decryption keys whose associated attributes, specified in their key structures, who satisfy such access structure can decrypt the ciphertext. The key structure is a recursive set based in which each element of the set is either a set or an element corresponding to an attribute.

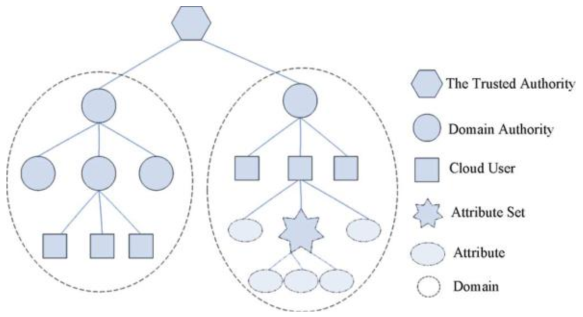

Fig. 2. Hierarchical structure of system users

In this system, data owners, data consumers, domain authorities, and the trusted authority are organized in a hierarchical manner as shown below:

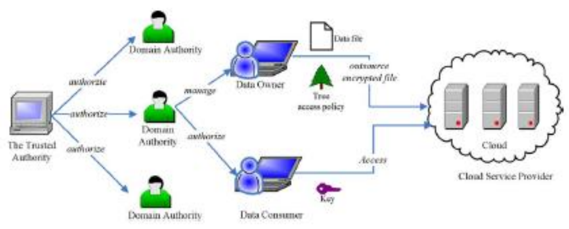

Fig. 3. System Model for HASBE

The trusted authority is the root authority and responsible for managing top-level domain authorities. In this implementation, neither data owners nor data consumers will be always online. They come online only when necessary, while the cloud service provider, the trusted authority, and domain authorities are always online.

In the hierarchical structure of the system users, each user is associated with a public key and a private key. Also it is assumed that all parties or users are secured using standard security protocols such as SSL. The recursive set based key structure for either a set or an element can be given as shown below in Fig. 3. The key structure defines unique labels for sets in it. Individual attributes inherit the label of the set they are contained in and are uniquely defined by the combination of their name and their inherited label.

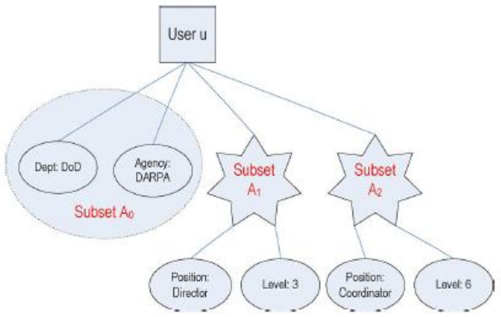

Fig. 4. Hierarchical Key structure

The trusted authority calls the System Setup algorithm to create system public parameters PK (which will be made public) and master key MK0 (secret key)

$$\mathbf{PK} = \Big(\mathbb{G}, g, h_1 = g^{\beta_1}, \quad f_1 = g^{\frac{1}{\beta_1}}, \quad h_2 = g^{\beta_2}, \quad f_2 = g^{\frac{1}{\beta_2}}, e(g,g)^{\alpha}\Big)$$

$$\mathbf{MK}_0 = (\beta_1, \beta_2, g^{\alpha}).$$

When a new top-level domain authority, wants to join, the trusted authority first verifies for the validity of the domain authority. If valid, then CreateDA algorithm generates the master key for the DAi.

$$\mathbf{MK}_i = \Big(\mathbb{A}, D = g^{\frac{(\alpha + r^{\{u\}})}{\beta_1}}, D_{i,j} = g^{r_i^{\{u\}}} \cdot H(a_{i,j})^{r_{i,j}^{\{u\}}}, \quad D'_{i,j} = g^{r_{i,j}^{\{u\}}} \text{ for } 0 \le i \le m,\ 1 \le j \le n_i, \quad E_i = g^{\frac{(r^{\{u\}} + r_i^{\{u\}})}{\beta_2}} \text{ for } 1 \le i \le m\Big).$$

The secret key for the new user or a new domain authority can be given as

$$\mathbf{SK}_u \text{ or } \mathbf{MK}_{i+1} = \Big(\widetilde{\mathbb{A}}, \widetilde{D} = D \cdot f_1^{\widetilde{r}^{\{u\}}}, \widetilde{D}_{i,j} = D_{i,j} \cdot g^{\widetilde{r}_i^{\{u\}}} \cdot H(a_{i,j})^{\widetilde{r}_{i,j}^{\{u\}}}, \quad \widetilde{D}'_{i,j} = D'_{i,j} \cdot g^{\widetilde{r}_{i,j}^{\{u\}}} \text{ for } a_{i,j} \in \widetilde{\mathbb{A}}, \quad \widetilde{E}_i = E_i \cdot f_2^{\widetilde{r}^{\{u\}} + \widetilde{r}_i^{\{u\}}} \text{ for } A_i \in \widetilde{\mathbb{A}}\Big).$$

The new secret key $\mathbf{SK}_u$ or $\mathbf{MK}_{i+1}$ is a secret key for the key structure $\widetilde{\mathbb{A}}$.

For the data protection on the cloud, it has to be first encrypted by the data owner and then upload the encrypted data files on the cloud. This encrypted file is encrypted with a symmetric key and then with HASBE. Any data file should be processed as below before uploading by the data owner.

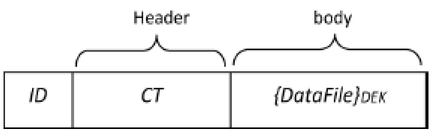

Fig. 5. Format of a data file on the cloud

- Pick a unique ID for this data file.
- Randomly choose a symmetric data encryption key $\mathrm{DEK} \overset{R}{\leftarrow} \kappa$, where κ is the key space, and encrypt the data file using DEK.
- Define a tree access structure $\mathcal{T}$ for the file and encrypt DEK with $\mathcal{T}$ using algorithm $\mathrm{Encrypt}(\mathbf{PK}, \mathrm{DEK}, \mathcal{T})$ of HASBE which returns ciphetext CT.

The encryption algorithm associates a polynomial qx for each node x in the tree. The degree of qx should be always one less than the threshold value given as dx. The ciphertext can be computed as follows:

$$\mathbf{CT} = +\Big(\mathcal{T}, \widetilde{C} = M \cdot e(g,g)^{\alpha \cdot s}, C = h_1^s, \bar{C} = h_2^s, \forall y \in Y : \quad C_y = g^{q_y(0)}, C'_y = H(\mathbf{att}(y))^{q_y(0)}, \quad \forall x \in X : \hat{C}_x = h_2^{q_x(0)}\Big)$$

While accessing the data the system should make sure that the revoked user cannot access the associated data files and the users who have the access privileges to access these files can access them correctly.

For accessing the encrypted files on the cloud, the cloud sends the corresponding cipher texts to the receiving user. The user decrypts them by first calling $\mathrm{Decrypt}(\mathrm{CT}, \mathrm{SK}_u)$ to obtain DEK and then decrypt the files.

If t is a leaf node, and if $\mathbf{att}(t) \notin A_i$, where $A_i \in \mathbb{A}$, then $\mathrm{DecryptNode}(\mathrm{CT}, \mathrm{SK}_u, t, i) = \mathrm{null}$. If $\mathbf{att}(t) = a_{i,j} \in A_i$, where $A_i \in \mathbb{A}$, then $\mathrm{DecryptNode}(\mathrm{CT}, \mathrm{SK}_u, t, i) = e(D_{i,j}, C_t)/e(D'_{i,j}, C'_t) = e(g,g)^{r_i^{\{u\}}} \cdot q_t(0)$.

If t is a nonleaf node, then $\mathrm{DecryptNode}(\mathrm{CT}, \mathrm{SK}_u, t, i)$ is defined as follows:

- Let B_t be an arbitrary k_t sized set of child nodes z such that $z \in B_t$ only if (1) label $i \in S_z$ or (2) label $i' \in S_z$ for some $i' \neq i$ and z is a *translating node*. If no such set exists then return null.
- For each node $z \in B_t$, if $i \in S_z$, then call $\mathrm{DecryptNode}(\mathrm{CT}, \mathrm{SK}_u, z, i)$ and store output in F_z.
- For each node $z \in B_t$, if $i' \in S_z$ and $i' \neq i$, then call $\mathrm{DecryptNode}(\mathrm{CT}, \mathrm{SK}_u, z, i')$ and store output in F'_z. Then if $i \neq 0$, translate F'_z to F_z as follows:

$$F_z = e(\hat{C}_z, E_i/E_{i'}) \cdot F'_z = e(g,g)^{r_i^{\{u\}} \cdot q_z(0)}.$$

Otherwise, if $i = 0$, then translate F'_z to F_z as follows:

$$F_z = \frac{e(\hat{C}_z, E_{i'})}{F'_z} = e(g,g)^{r^{\{u\}}} \cdot q_z(0).$$

Then the message M can be computed as $M = \tilde{C} \cdot F/e(C, D)$.

For the deletion of an encrypted data file, the data owner should send the file's unique ID and its signature on this ID to the cloud. Upon successful verification of the data owner and the deletion request, the cloud deletes the data file.

IV. CONCLUSION

The implemented work HASBE provides a scalable, flexible, fine-grained access control in a cloud environment. This scheme incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. This scheme efficiently supports compound attributes due to flexible attribute set combinations. This scheme achieves a efficient user revocation because of multiple value assignments of attributes.

REFERENCES

[1] D. E. Bell and L. J. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation The MITRE Corporation, Tech. Rep., 1976.
[2] K. J. Biba, Integrity Considerations for Secure Computer Sytems The MITRE Corporation, Tech. Rep., 1977.
[3] H. Harney, A. Colgrove, and P. D. McDaniel, "Principles of policy in secure groups," in Proc. NDSS, San Diego, CA, 2001.
[4] P. D. McDaniel and A. Prakash, "Methods and limitations of security policy reconciliation," in Proc. IEEE Symp. Security and Privacy,Berkeley, CA, 2002.
[5] T. Yu and M. Winslett, "A unified scheme for resource protection in automated trust negotiation," in Proc. IEEE Symp. Security and Privacy, Berkeley, CA, 2003.
[6] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in Proc. ACM Conf. Computer and Communications Security (CCS), Alexandria, VA, 2005.