Anonymous Authentication of Data by Attribute Based Signature scheme using KDC in Clouds

[I]Manjunath, [II]Prof. Virupakshappa

[I]M. Tech, Dept. of CSE, APPA Institute of Engineering and Technology, Gulbarga, Karnataka, India
[II]Asst. Professor, Dept. of CSE, APPA Institute of Engineering and Tech., Gulbarga, Karnataka India

Abstract

In this paper, a new attribute based signature scheme is introduced for secure data stored using KDC in clouds that supports anonymous authentication. In this scheme before storing data, the cloud has the ability to check the privilege of user for authentication. The scheme also provides the authenticated users to decrypt the information stored in the cloud and it also adds the access control method. The attribute based signature scheme prevents attacks and support updating of information stored in clouds. Updating includes the creation, modification, and reading of information. Attribute based signature scheme is decentralized and uses Key distribution center (KDC), and other schemes other than attribute based signature scheme are centralized and all these schemes are designed for clouds.

Keywords

Storage cloud, Decentralized, KDC, access control.

I. Introduction

Cloud Computing is an emerging technology that is receiving a lot of attention from the generation of the academic and Industrial worlds. In cloud computing, people or the user can provide their resources such as computation and storage to servers via the internet. Sharing the user resources is provided by Clouds. By applying this, makes free to users from the maintaining the resources. Cloud computing provides many types of services such as applications (e.g., Google products etc), Infrastructures (e.g., window's services, Amazon's EC2 etc), and provides platforms for software engineers to write applications (e.g., windows azure, Amazon's S3 etc) [1].

Information stored in the cloud server is highly sensitive for example medical records, and social networks [12]. Parameters such as security and privacy are important parts in the cloud computing. First, user itself must be authenticated before any transaction initiated. Lastly, It should be verified that the cloud do not make mess with information that is outsourced by the users. Providing user privacy is also a main concern so that the identity of the user is protected from the cloud and other users.

Recently in 2012, wang et al [2] presented secure and dependable storage in cloud. Cloud servers liable to byzantine failure, where failure can occur in many ways in a cloud server. The cloud is also liable to modification of information and colluding attacks in server. Colluding attack in server, storage servers can compromise conflicts, so that data files are modified as long as they are constant. Information is to be encrypted to provide data storage.

The main concern in cloud computing is the searching on the encrypted data [3]. The query is not known to the clouds but it is able to return the records to satisfy the query. This is done by the searchable encryption. The cloud receives the encrypted keywords and results will be returned by the cloud without the knowing the correct keyword to search. The information records should have the keywords that are associated with records that enables for the search. While searching in clouds the actual keywords are returned when matched with exact keywords in the cloud storage.

Challenging task in cloud computing is the accountability and includes technical issues. In cloud computing the operations performed or requested cannot be denied by either clouds or by the users. It is important to maintain the log file for every transactions performed on the clouds and it important to maintain that log file so to decide how much information kept in that log file. Accountability is addressed in TrustCloud [4].

Access control in cloud computing is paying an important role, because only authorized users can have access for the valid services In cloud server large amount of data is stored and this stored information is highly sensitive. Access controls in cloud are classified as three types: *User-based Access Control, Role-based Access Control, and Attribute-based Access Control*. In first type i.e. User-based Access Control (UAC), a list is maintained called as access control list (ACL) which contains users list who are authorized to access information. In second type i.e. Role-based Access Control (RAC), based on their individual roles, the users are classified and data is accessed by the users who match with their roles. For example only faculties and secretaries have the privilege to access the data from cloud not by others who do not have access privilege such as students. Last type i.e. Attribute-based Access Control (AAC), attributes are issued to the users, and information is attached to access policy. Accessing the information by the users, they should have the valid set of attributes and should satisfy the access policy. The uses and limitations of RAC and AAC are discussed in [5]. Some work has been done in AAC in clouds (e.g. [6], [7], [8]). All these use Attribute Based Encryption (ABE).

A. Digital signature

A digital signature [13] is a technique used to validate the digital document or message. Digital signatures are based on public key cryptography. Using public key algorithm user can generate two keys one is the private key and another is the public key. To create digital signature, signing software creates a one-way hash of electronic data to be signed. The private key is then used to encrypt the hash. The encrypted hash along with the other information such as the hashing algorithm is the digital signature. A digital signature consists of three algorithms [13].

1. A key generation that selects a private key uniformly at random from set of possible private keys.
2. A signing algorithm, given a message and private key that produces a signature.
3. A signature verifying algorithm, given a message, public key and a signature, either accepts or rejects the message.

B. About Key Distribution Center(KDC)

In cryptography, a key distribution center [14] (KDC) is a part of cryptosystem. KDCs often operate in systems within which some users may have permission to use certain services at some time and not at others. Typical operation with KDC involves a request from user to use some service. The KDC will use cryptographic [9] techniques to authenticate requesting users as themselves. It will check whether an Individual user has right to access the service requested if the authenticated user meets all the prescribed conditions, the KDC can issue a ticket permitting access.

KDCs mostly operated with symmetric encryption [14]. In most (but not all) cases the KDC shares a key with all other parties.KDC produces a ticket based on server key and that ticket is received by client and submits it to the appropriate server. The server can verify submitted ticket and grant access to the user submitting it. Security systems in KDCs include Kerberos which partitions KDC functionality between two different agents the authentication server (AS) and Ticket Granting Service (TGS).

II. Literature Survey

Towards secure and dependable storage services in cloud computing [2]: In this paper a flexible distributed storage integrity auditing mechanism, utilizing the homomorphic token and distributed erasure-coded data was proposed. Their proposed design allowed users to audit the cloud storage with very lightweight communication and computation cost. The auditing results not only ensured strong cloud storage correctness guarantee, but also simultaneously achieved fast data error localization. i.e., The identification of misbehaving server. Considering, the cloud data are dynamic in nature, their proposed design further supported secure and efficient dynamic operations on outsourced data, including block modification, deletion, and append. Analysis shown that proposed scheme is highly efficient and resilient against Byzantine failure, and even server colluding attacks.

TrustCloud: A Framework for Accountability and Trust in Cloud Computing [4]: The key barrier to widespread uptake of cloud computing is the lack of trust in clouds by potential customers. While preventive controls for security and privacy measures detective controls related to cloud accountability and auditability. The complexity resulting from the sheer amount of virtualization and data distribution carried out in current clouds has also revealed an urgent need for research in cloud accountability, as has the shift in focus of customer concerns from server health and utilization to the integrity and safety of end-users data. This topic discussed key challenges in achieving a trusted cloud through the use of detective controls, and presented the TrustCloud framework, which addressed accountability in cloud computing via technical and policy based approaches.

Attribute based data sharing with attribute revocation [7]: Ciphertext-Policy Attribute Based Encryption (CP-ABE) is a promising cryptographic primitive for fine-grained access control of shared data. In Cp-ABE, each user is associated with a set of attributes and data are encrypted with access structures on attributes. A user is able to decrypt a ciphertext if and only if his attributes satisfy the ciphertext access structure. In their paper they focused on an important issue of attribute revocation which is cumbersome for CP-ABE schemes. They resolved this challenging issue by considering more practical scenarios in which semi-trustable on-line proxy servers are available. As compared to existing schemes, proposed solution enabled the authority to revoke user attributes with minimal effort. They achieved uniquely integrating the technique of proxy re-encryption with CP-ABE, and enabled the authority to delegate most of laborious tasks to proxy servers. Analysis shown that proposed scheme is provably secure against chosen ciphertext attacks.

DACC: Distributed access control in clouds [9]: They proposed a new model for data storage and access in clouds. Their scheme avoided storing multiple encrypted copies on same data. In their framework for secure data storage, cloud stores encrypted data (without being able to decrypt them). The main novelty of our model is addition of key distribution centers (KDCs). They proposed DACC (Distributed Access Control in Clouds) algorithm, where one or more KDCs distribute keys to data owners and users. KDC may provide access to particular fields in all records. Thus, a single key replaces separate keys from owners. Owners and users are assigned certain set of attributes. Owner encrypts the data with the attributes it has and stores them in the cloud. The users with matching set of attributes can retrieve the data from the cloud. They applied attribute-based encryption based on bilinear pairings on elliptic curves. Their scheme is collusion secure, two users cannot together decode any data that none of them has individual right to access. DACC also supported revocation of users, without redistributing keys to all the users of cloud services.

EASiER: Encryption-based Access Control in Social Networks with Efficient Revocation [10]: A promising approach to mitigate the privacy risks in Online Social Networks (OSNs) is to shift access control enforcement from the OSN provider to the user by means of encryption. This creates the challenge of key management to support complex policies involved in OSNs and dynamic groups. To address this, they proposed EASiER, architecture that supported fine-grained access control policies and dynamic group membership by using attribute based encryption. A key and novel feature of this architecture is that it is possible to remove access from a user without issuing new keys to other users or re-encrypting existing ciphertexts. They achieved this by creating a proxy that participates in the decryption process and enforces revocation constraints. The proxy in minimally trusted and cannot decrypt ciphertexts or provide access to previously revoked users.

III. Background

Attribute Based Encryption

Attribute based encryption with multiple authorities as proposed by lewko and waters [11] proceeds as follows [9].

System Initialization: Select a prime p, generator g of G_0, groups G_0 and G_T of order p, a map $e : G_0 \times G_0 \rightarrow G_T$, and a hash function $H : \{0.1\}^* \rightarrow G_0$ which maps the identities of users to G_0. The SHA-1 is the hash operation used. Each KDC $A_j \in A$ has set of attributes L_j. The attributes disjoint ($L_i \cap L_j = \emptyset$ for $i \neq j$). each KDC also chooses two random exponents $\alpha_i, y_i \in z_q$ the secret key of KDC A_j is

$$SK[j] = \{\alpha_i, y_i, i \in L_j\}. \qquad (1)$$

KDC'spublic key A_j is published

$$PK[j] = \{e\,(g, g)^{\alpha_i}, g^{\alpha_i}, i \in L_j\}. \qquad (2)$$

Key Generation and Distribution by KDCs: User U_u receives a set of attributes I[j, u] from KDC A_j, and appropriate secret key $sk_{i,u}$ for each $i \in I[j, u]$

$$SK_{i,\,u} = g^{\alpha_i} H(u)^{y_i}\ , \qquad (3)$$

Where $\alpha_i, y_i \in Sk[j]$.

Sender operation Encryption: ABE.Encrypt(MSG, x) will be the encryption operation. Sender decides about access x. Message

MSG is encrypted by sender as follows:

- Sender chooses a random value $s \in Z_q$ and a random vector $v \in Z_q$, with s as its first entry; h is the number of leaves in the access tree.
- Calculate $\lambda_x = R_x \cdot v$, where R_x is a row of R.
- Choose a random vector $w \in Z_q$ with 0 as the first entry.
- Calculate $w_x = R_x \cdot w$.
- For each row of R, choose a random $\rho_x \in Z_q$.
- The following parameters are calculated:

$$C_0 = MSGe(g, g)^s,$$

$$C_{1,x} = e(g,g)^{\lambda_x} e(g,g)^{\alpha_{x(x)}\rho_x}, x_{1,} \qquad (4)$$

$$C_{2,x} = g^{\rho_x} x_1,$$

$$C_{3,x} = g^{y_{x(x)}\rho_x} g^{\omega_x} x_1,$$

The ciphertext C is sent by the sender.

$$C = \langle R, \pi, C_0, \{C_{1,x}, C_{2,x}, C_{3,x}, x_1\} \rangle. \qquad (5)$$

Decryption operation by receiver: The ABE.Decrypt(C, $\{sk_{i,u}\}$) is decryption operation, where C is given by (5). Receiver U_u takes ciphertext C as input, secret keys $\{sk_{i,u}\}$, group G_0, and output is returned as message msg. After that it executes following steps:

User U_u calculates the set of attributes $\{\pi(x) : x \in X\} \cap I_u$, where X is the set rows of R.

For each of these attributes, it checks if there is a subset X' of rows of R, such that vector is their combination. If not, decryption is impossible. If yes it calculates constants.

Decryption is done as follows:

For each $x \in X^{'}$,

$dec(x) = C_{1,x} e(H(u), C_{3,x}) / e(sk_{x(x),u}, C_{2,x})$.

U_u Computes $MSG = C_0 / \Pi_{x \in X'} dec(x)$.

Attribute-Based Signature scheme

ABS scheme [12] has the following steps.

System Initialization: select prime q, and groups G_1 and G_2, which are of order q. mapping is done as $e : G_1 \times G_1 \rightarrow G_2$. Let g_1, g_2 be the generators of G_1 and h_j be generators of G_2, let H be a hash function. The secret key for the trustee is TSK = (a_0, TSig) and public key is

$TPK = (G_1, G_2, H, g_1, A_0, h_0, h_1, \ldots, h_{tmax}, g_2, TVer)$.

Where, TSig is the private key with message signed and TVer is the public key used for verification.

User Registration: For a user with U_u the KDC draws at random $K_{base} \in G$. The following token γ is output

$\gamma = (u, K_{base}, K_0, \rho)$, (6)

Where $K_0 = K^{1/a}$.

KDC setup: choose $a, b \in Z_q$ randomly and compute: $A_{ij} = h_j$, $B_{ij} = h_{j,}$ for $A_i \in A$, $j \in [t_{max}]$. The private key of ith KDC is ASK[i] = (a, b) and public key $APK[i] = (A_{ij}, B_{ij} | j \in [t_{max}])$.

Attribute Generation: The token verification algorithm verifies the signature contained in γ using the signature verification key TVer in TPK. This algorithm extracts K_{base} from γ using (a, b) from ASK[i] and computes K_x, $x \in J[i, u]$. The key K_x can be checked for consistency using algorithm ABS.KeyCheck(TPK,APK[i], γ, K_x), which checks

$\hat{e}(K_x, A_{ij} B_{ij}) = \hat{e}(K_{base}, h_j)$,

For all x □ J[i, u] and j □ [t_{max}].

Sign: The algorithm

ABS.Sign(TPK,{APK[i] : i □ AT[u]},γ, {K_x : x □ J_u}, MSG, y),

Has input the public key of the trustee, the secret key of the signer, the message to be signed and the policy claim γ. The policy claim is first converted into the span program with rows labeled with attributes. A vector v is computed that satisfies the assignment $\{x : x \in J[i, u]\}$.Compute $\mu = H(MSG||\gamma)$. And compute:

$$Y = K^{r_0}, S_i = (K^{u_i})^{r_0} . (g_2\, g_1)^{r_i} (i \in J_u) \ (7)$$

$$W = K_0, P_j = \Pi_{i \in AT[u]} (A_{ij} B^{\pi'(i)})^{M_{ij} r_i} (j \in [y]). \quad (8)$$

The signature is calculated as

$$\sigma = (Y, W, S_1, S_2, \ldots.., S_t, P_1, P_2, \ldots., P_t). \qquad (9)$$

Verify: Algorithm ABS.Verify(TPK, $\sigma = (Y, W, S_1, S_2, \ldots, S_t, P_1, P_2, \ldots P_t)$, MSG, y), Converts γ to the corresponding monotone program with rows labeled with attributes. Compute $\mu = H(MSG||\gamma)$. If Y=1, ABS.Verify=0 meaning false. Otherwise following constraints are checked:

$$\hat{e}(W, A_0) = \hat{e}(Y, h_0), \qquad (10)$$

$\Pi_{i \in 1} \hat{e}(S_i, A_{ij} B^{\pi'(i)})^{Mij} = \hat{e}(Y, h_1) \hat{e}(g_2 g^{\mu}, P_1), j = 1,$

$$\hat{e}(g_2 g^{\mu}, P_j), j > 1, \qquad (11)$$

Where i' =AT[i].

IV. Conclusion

In this paper, an attribute based Signature scheme using KDC for anonymous authentication is presented. The proposed scheme provided the full secure for the Information that is stored in the clouds by using the encryption technique. The scheme prevents attacks by hackers. User stores information in the cloud, but the cloud simply stores data without the identity of the user but it verifies the user's privileges. Key distribution Center (KDC) provides unique key to user and this distribution is decentralized. The cloud should know the access policy for the record which it stores.

References

[1] S. Ruj, M. Stojmenovic, and A. Nayak, "Privacy Preserving Access Control with Authentication for Securing Data in Clouds," Proc. IEEE/ACM Int'l Symp. 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 Trans. Services Computing, vol. 5, no. 2, pp. 220-232, Apr.- June 2012.

[3] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," Proc. IEEE INFOCOM, pp. 441-445, 2010.

[4] R.K.L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B.S. Lee, "Trustcloud: A Framework for Accountability and Trust in Cloud Computing," HP Technical Report HPL-2011-38, http://www.hpl.hp.com/techreports/2011/HPL-2011-38.html, 2013.

[5] 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, June 2010.

[6] *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," Proc. Sixth Int'l ICST Conf. Security and Privacy in Comm. Networks (SecureComm), pp. 89-106, 2010.*

[7] *S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute Based Data Sharing with Attribute Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), pp. 261-270, 2010.*

[8] *G. Wang, Q. Liu, and J. Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services," Proc. 17th ACM Conf. Computer and Comm. Security (CCS), pp. 735-737, 2010.*

[9] *S. Ruj, A. Nayak, and I. Stojmenovic, "DACC: Distributed Access Control in Clouds," Proc. IEEE 10th Int'l Conf. Trust, Security and Privacy in Computing and Communications (TrustCom), 2011.*

[10] *S. Jahid, P. Mittal, and N. Borisov, "EASiER: Encryption-Based Access Control in Social Networks with Efficient Revocation," Proc. ACM Symp. Information, Computer and Comm. Security (ASIACCS), 2011.*

[11] *A.B. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption," Proc. Ann. Int'l Conf. Advances in Cryptology (EUROCRYPT), pp. 568-588, 2011.*

[12] *H.K. Maji, M. Prabhakaran, and M. Rosulek, "Attribute-Based Signatures," Topics in Cryptology - CT-RSA, vol. 6558, pp. 376-392, 2011.*

[13] *http://en.m.wikipedia.org/wiki/Digital_signature.*

[14] *http://en.m.wikipedia.org/wiki/Key_Distribution_Center.*