IJCSN International Journal of Computer Science and Network, Volume 6, Issue 2, April 2017
ISSN (Online): 2277-5420
www.IJCSN.org
Impact Factor: 1.5

Attribute Based Access Control Policy in Cloud Systems

[1] K.Gayathri; [2] K.Selvasheela

[1, 2] Veerammal Engineering College, Singarakottai, Dindigul(DT)
Tamilnadu(ST), India

Abstract - Recently, there has been considerable interest in attribute-based access control (ABAC) to overcome the limitations of the classical access control models while unifying their advantages. The general idea of ABAC is to determine access control based on the attributes of involved entities. Authorization results are computed based on subject and object attributes and authorization policies. As attributes can be engineered to reflect appropriately detailed information about users, subjects and objects, ABAC ensures great flexibility in expressing fine- grained policies which are increasingly required by applications. Here introduce key update algorithm as a part of access control service that is specifically aimed at optimizing user key update processing cost in multi-authority cloud. To this end, a multi-agent system (MAS) to perform the access control functions including user authentication, key update handling, and authorization. To support key update process, the agents will execute key update algorithm by updating all user's keys containing changed attributes on behalf of the attribute authority (AA). In addition, we provide the security proof of our key updating scheme in the general security model. Finally, the performance evaluation is provide to substantiate the efficiency of our proposed scheme.

Keywords - attribute-based access control, key update, attribute authority.

1. Introduction

With the emergence of sharing confidential corporate data-on cloud servers, it is imperative to adopt an efficient encryption system with a fine-grained access control to encrypt outsourced data. Ciphertext-policy attribute-based encryption (CP-ABE), as one of the most promising encryption systems.

In this field, allows the encryption of data by specifying an access control policy over attributes, so that only users with a set of attributes satisfying this policy can decrypt the corresponding data. However, a CP-ABE system may not work well when enterprise users outsource their data for sharing on cloud servers, due to the following reasons:

First, one of the biggest merits of cloud computing is that users can access data stored in the cloud anytime and anywhere using any device, such as thin clients with limited bandwidth, CPU, and memory capabilities. Therefore, the encryption system should provide high performance.

Second, in the case of a large-scale industry, a delegation mechanism in the generation of keys inside an enterprise is needed. Although some CP-ABE schemes support delegation between users, which enables a user to generate attribute secret keys containing a subset of his own attribute secret keys for other users, we hope to achieve a full delegation, that is, a delegation mechanism between attribute authorities (AAs), which independently make decisions on the structure and semantics of their attributes.

Third, in case of a large-scale industry with a high turnover rate, a scalable revocation mechanism is a must. The existing CP-ABE schemes usually demand users to heavily depend on AAs and maintain a large amount of secret keys storage, which lacks flexibility and scalability.

2. Motivation

Our main design goal is to help the enterprise users to efficiently share confidential data on cloud servers. Specifically, we want to make our scheme more applicable in cloud computing by simultaneously achieving fine-grained access control, high performance, practicability, and scalability.

3. Contribution

In this paper, we first propose a hierarchical attribute-based encryption (HABE) model by combining a HIBE system and a CP-ABE system, to provide fine-grained access control and full delegation. Based on the HABE model, we construct a HABE scheme by making a performance-expressivity tradeoff, to achieve high performance. Finally, we propose a scalable revocation scheme by delegating to the CSP most of the computing

Copyright (c) 2017 International Journal of Computer Science and Network. All Rights Reserved.

tasks in revocation, to achieve a dynamic set of users efficiently.

The HABE Model

- The HABE model consists of a root master (RM) that corresponds to the third trusted party (TTP), multiple domain masters (DMs) in which the top-level DMs.
- The RM, whose role closely follows the root private key generator (PKG) in a HIBE system, is responsible for the generation and distribution of system parameters and domain keys.
- The DM, whose role integrates both the properties of the domain PKG in a HIBE system and AA in a CP-ABE system, is responsible for delegating keys to DMs at the next level and distributing keys to users.
- Notice that other DMs administer an arbitrary number of disjoint attributes, and have full control over the structure and semantics of their attributes.
- In the HABE model, we first mark each DM and attribute with a unique identifier (ID), but mark each user with both an ID and a set of descriptive attributes.

4. Proposed Model

In the Proposed, based on hierarchical identity-based encryption (HIBE) and combined with Key isolation mechanism (IKE), a new hierarchical key update algorithm is proposed. Above all, the security mechanism based on private key update on IKE is demonstrated; then a new encryption model (HIBE-IKE) is constructed, which combines hierarchical identity-based encryption and private Key isolation mechanism and provides a new Private Key update method to solve the problems of private key update based on identity-based encryption system.

In this, we first propose a hierarchical attribute-based encryption (HABE) model by combining a HIBE system and a CP-ABE system, to provide fine-grained access control and full delegation. Based on the HABE model, we construct a HABE scheme by making a performance-expressivity tradeoff, to achieve high performance.

Finally, we propose a scalable revocation scheme by delegating to the CSP most of the computing tasks in revocation, to achieve a dynamic set of users efficiently.

5. Design for Proposed

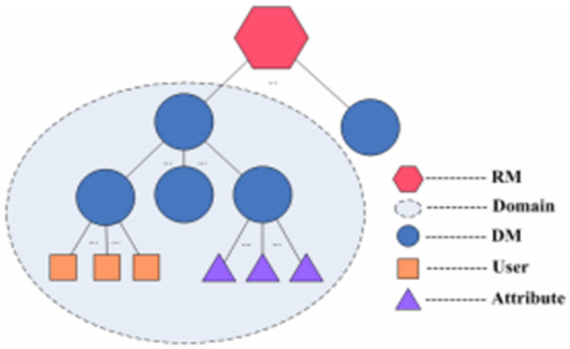

Fig 1: Overall design

6. Modules

- User Login
- Creation of KDC
- KDC Authentication
- Trustee and User Accessibility
- File accessing
- File Restoration

7. Modules description

7.1 User Login

- In general computer usage, logon is the procedure used to get access to an operating system or application, usually in a remote computer. Almost always a logon requires that the user have (1) a user ID and (2) a password.
- Often, the user ID must conform to a limited length and the password must contain at least one digit and not match a natural language word. The user ID can be freely known and is visible when entered at a keyboard or other input device.
- The password must be kept secret (and is not displayed as it is entered). Some Web sites require users to register in order to use the site; registered users can then enter the site by logging on.
- In our Login module, user has to enter his id and password to login. This is mainly for security purpose; those who have not registered cannot login. Then user can apply the proposed algorithm to improve the system performance.

Copyright (c) 2017 International Journal of Computer Science and Network. All Rights Reserved.

7.2 Creation of KDC

- All the approaches take a centralized approach and allow only one KDC, which is a single point of failure.

- Different number of KDC's is 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.

7.3 KDC Authentication

- After the generation KDC, It will give 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 verifying the user details and it will insert it in a Database.

- 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.

7.4 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.

- On presenting her id to the trustee, trustee gives a token. There are multiple KDCs, which can be scattered.

- Users on presenting the token to KDC receive keys for encryption/decryption and signing. SK are secret keys given for decryption, Kx are keys for signing.

- User can login with their credentials and request the token from trustee for the file upload using the user id.

- 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).

7.5 File accessing

- 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.

- To write to an already existing file, the user must send its message with the claim policy as done during file creation.

- The cloud verifies the claim policy and only if the user is authentic, is allowed to write on the file.

- The current time stamp T is attached to the cipher text to prevent replay attacks.

- 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).

7.6 File Restoration

- A restore is performed in order to return data to its original condition if files have become damaged or to copy or move data to a new location.

- 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.

8. Input and Output Design

8.1 Input Design:

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product

Copyright (c) 2017 International Journal of Computer Science and Network. All Rights Reserved.

development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

Input Design is the process of converting a user oriented description of the inputs to a computer-based business system into a programmer-oriented specification.

- Input data were found to be available for establishing and maintaining master and transaction files and for creating output records
- The most suitable types of input media , for either off-line or on-line devices, where selected after a study of alternative data capture techniques.

8.2 Input Design Consideration

- The field length must be documented.
- The sequence of fields should match the sequence of the fields on the source document.
- The data format must be identified to the data entry operator.

Design input requirements must be comprehensive. Product complexity and the risk associated with its use dictate the amount of detail

- These specify what the product does, focusing on its operational capabilities and the processing of inputs and resultant outputs.
- These specify how much or how well the product must perform, addressing such issues as speed, strength, response times, accuracy, limits of operation, etc.

8.3 Output Design:

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs.

In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

1.Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

9. The HABE Model

The HABE model consists of a root master (RM) that corresponds to the third trusted party (TTP), multiple domain masters (DMs) in which the top-level DMscorrespond to multiple enterprise users, and numerous users that correspond to all personnel in an enterprise.

The RM, whose role closely follows the root private key generator (PKG) in a HIBE system, is responsible for the generation and distribution of system parameters and domain keys. The DM, whose role integrates both the properties of the domain PKG in a HIBE system and AA in a CP-ABE system, is responsible for delegating keys to DMs at the next level and distributing keys to users. Specifically, we enable the leftmost DM at the second level to administer all the users in a domain, just as the personnel office administers all personnel in an enterprise, and not to administer any attribute. Notice that other DMs administer an arbitrary number of disjoint attributes, and have full control over the structure and semantics of their attributes.

In the HABE model, we first mark each DM and attribute with a unique identifier (ID), but mark each user with both an ID and a set of descriptive attributes. we enable an entity's secret key to be extracted from the DM administering itself, and an entity's public key, which denotes its position in the HABE model, to be an IDtuple consisting of the public key of the DM administering itself and its ID, e.g., the public key of DM_i with ID_i is in the form of $(PK_i{-}1, ID_i)$, the public key of user U with ID_u is in the form of $(PK\diamondsuit, ID_u)$, and the public key of attribute a with ID_a is in the form of (PK_i, ID_a), where $PK_i{-}1$, $PK\diamondsuit$, and PK_i are assumed to be the public keys of the DMs that administer DM_i, U, and a, respectively.

Steps

Step 1:

Setup(K) → (params, MK_0): The RM first picks $mk_0 \in Zq$, and then chooses groups G_1 and G_2 of order q, a bilinear map $\hat{e} : G_1 \times G_1 \rightarrow G_2$, two random oracles H1: {0, 1} ∗ →

Copyright (c) 2017 International Journal of Computer Science and Network. All Rights Reserved.

G_1 and H_2: $G_2 \rightarrow \{0, 1\}$ n for some n, and a random generator $P_0 \in G_1$. Let $Q_0 = mk_0P_0 \in G_1$. The system parameters params = (q, G_1, G_2, e, n, P ˆ 0, Q_0, H_1, H_2) will be publicly available, while $MK_0 = (mk_0)$ will be kept secret.

Step 2:
CreateDM(params, MK_i,PK_i+1) → (MK_i+1) : To generate the master key for DM_i+1 with PK_i+1, the RM or DM_i first picks a random element mk_i+1 ∈Zq, and then computes SK_i+1 = SK_i + mkiPi+1 where P_i+1 = H1(PK_i+1) ∈ G_1, and Q_i+1 = mk_i+1$P_0 \in G_1$, finally sets MK_i+1 = (mk_i+1, SKi+1, Q-tuple$_i$+1) where Q-tuple$_i$+1 = (Q-tuple$_i$, Q_i+1), and gives the random oracle HA : {0, 1} → Zq that is chosen by the RM and shared in a domain. Here, we assume that SK0 is an identity element of G_1, and Q-tuple$_0$ = (Q_0).

Step 3:
CreateUser(params, MK_i,PK_u,PK_a) → (SK_i,u, SK_i,u,a) : To generate a secret key for user U with PK_u on attribute a with PKa, DMi first checks whether U is eligible for a, and a is administered by itself. If so, it first sets mku = HA(PKu) ∈Zq, SK_i,u = mkimkuP0 ∈ G1, and SK_i,u,a = SK_i+ $mk_imk_uP_a \in$ G1, where Pa = H1(PK_a) ∈ G1, and then gives Q-tuple$_i$. Otherwise, it outputs "NULL".

Step 4:
Encrypt(params, A, {PK_{aij} |1 ≤ i ≤ N, 1 ≤ j ≤ n_i}, f) → (CT) : Given a DNF access control policy A = N ∨ i=1 (CC_i) = N ∨ i=1 (ni∧ j=1 a_{ij}), where N ∈ Z + is the number of conjunctive clause in A, $n_i \in$ Z + is the number of attributes in the i-th conjunctive clause CC_i, and a_{ij} is the j-th attribute in CC_i. Let DM_{iti} with (ID_i1, . . . ,ID_{iti}) be the DM at level t_i, administering all attributes in CC_i, where ID_{ik} for 1 ≤ k <t_i are IDs of DM_{iti} 's ancestors.

10. Conclusion

In this paper, we introduced the HASBE scheme for the purpose of experiencing scalable, flexible, and fine-grained access control in cloud computing. The HASBE scheme incorporates a hierarchical structure of system users by applying a delegation algorithm to ASBE. HASBE supports compound attributes due to flexible attribute set combinations, and also achieves efficient user revocation because of multiple value assignments of attributes. Finally, the proposed scheme, is implemented and conducted comprehensive performance analysis and evaluation, which showed its advantages and efficiency over existing schemes.

11. Future Scope

In future work, we will work towards designing a more expressive scheme, which can be proved to have full security under the standard model, with better performance.

References

[1] C. Gentry and A. Silverberg. Hierarchical ID-Based Cryptography. In Proceedings of ASIACRYPT 2002, pages 548-566.

[2] S. Muller, S. Katzenbeisser, and C. Eckert.Distributed Attribute-Based Encryption. In Proceedings of ICISC 2008, pages 20-36.

[3] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing. In Proceedings of IEEE INFOCOM 2010, pages 534-542.

[4] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in Proc. ACMConf. Computer and Communications Security (CCS), Alexandria, VA, 2005

[5] A. Ross, "Technical perspective: A chilly sense of security," Commun. ACM, vol. 52, pp. 90–90, 2009.

[6] D. E. Bell and L. J. LaPadula, Secure Computer Systems: Unified Exposition and Multics Interpretation The MITRE Corporation, Tech. Rep., 1976.

[7] K. J. Biba, Integrity Considerations for Secure Computer Systems The MITRE Corporation, Tech. Rep., 1977.

[8] H. Harney, A. Colgrove, and P. D. McDaniel, "Principles of policy in secure groups," in Proc. NDSS, San Diego, CA, 2001.

[9] R. Bobba, H. Khurana, and M. Prabhakaran, "Attribute-sets: A practically motivated enhancement to attribute-based encryption," in Proc.ESORICS, Saint Malo, France, 2009.

[10] A. Sahai and B. Waters, "Fuzzy identity based encryption," in Proc.Acvancesin Cryptology—Eurocrypt, 2005, vol. 3494, LNCS, pp. 457–473.

[11] G.Wang, Q. Liu, and J.Wu, "Hierachicalattibutebased encryption for fine-grained access control in cloud storage services," in Proc. ACMConf. Computer and Communications Security(ACM CCS), Chicago,

[12] 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. International Journal of Computer & Organization Trends –Volume 3 Issue 9 – Oct 2013 ISSN: 2249-2593 http://www.ijcotjournal.org Page 435

[13] J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials," in Proc. ACM Conf. Computer andCommunications Security (CCS),Alexandria, VA, 2005.

[14] V. Goyal, O. Pandey, A. Sahai, and B.Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in Proc. ACM Conf.Computer and Communications Security (ACMCCS), Alexandria,

Copyright (c) 2017 International Journal of Computer Science and Network. All Rights Reserved.

[15] Zhiguo Wan, Jun'e Liu, and Robert H. Deng, "HASBE: A Hierearchical Attribute-Based Solution for Flexible and Scalable Access Control in Cloud Computing

Copyright (c) 2017 International Journal of Computer Science and Network. All Rights Reserved.