AN ADVANCED SECURITY SYSTEM ON MULTI-CLOUD DATA SHARING

[1]N.Pavithra, [2]M.Rajagopal
[1]Research Scholar, Department of Information Technology, Jayam College of Engineering and Technology
[2]Assistant Professor Department of Information Technology, Jayam College of Engineering and Technology,
[1]pavithra1812@gmail.com, [2]Raja82gopal@gmail.com

Abstract: With the recent adoption and diffusion of the data sharing paradigm in distributed systems such as online social networks or cloud computing, there have been increasing demands and concerns for distributed data security. One of the most challenging issues in data sharing systems is the enforcement of access policies and the support of policies updates. Cipher text policy attribute-based encryption (CP-ABE) is becoming a promising cryptographic solution to this issue. It enables data owners to define their own access policies over user attributes and enforce the policies on the data to be distributed. However, the advantage comes with a major drawback which is known as a key escrow problem. The key generation center could decrypt any messages addressed to specific users by generating their private keys. The proposed scheme features the following achievements: 1) the key escrow problem could be solved by escrow-free key issuing protocol, which is constructed using the secure two-party computation between the key generation center and the data-storing center, and 2) fine-grained user revocation per each attribute could be done by proxy encryption which takes advantage of the selective attribute group key distribution on top of the ABE. The performance and security analyses indicate that the proposed scheme is efficient to securely manage the data distributed in the data sharing system.

Keywords: Data sharing, attribute-based encryption, revocation, access control, removing escrow.

1. INTRODUCTION

Cloud computing has received considerable attention from both academia and industry due to a number of important advantages including: cost effectiveness, low management overhead, immediate access to a wide range of applications, flexibility to scale up and down information technology (IT) capacity, and mobility where customers can access information wherever they are, rather than having to remain at their desks. Cloud computing is a distributed computational model over a large pool of shared-virtualized computing resources (e.g., storage, processing power, memory, applications, services, and network bandwidth).

Attribute-based encryption (ABE) is a promising cryptographic approach that achieves a fine-grained data access control. It provides a way of defining access policies based on different attributes of the requester, environment, or the data object. Especially, cipher text policy attribute-based encryption (CP- ABE) enables an encrypt tor to define the attribute set over a universe of attributes that a decryptor needs to possess in order to decrypt the cipher text, and enforce it on the contents [5]. Thus, each user with a different set of attributes is allowed to decrypt different pieces of data per the security policy. This effectively eliminates the need to rely on the data storage server for preventing unauthorized data access, which is the traditional access control approach of such as the reference monitor [1]

Nevertheless, applying CP-ABE in the data sharing system has several challenges. In CP-ABE, the key generation center (KGC) generates private keys of users by applying the KGC's master secret keys to users' associated set of attributes. Thus, the major benefit of this approach is to largely reduce the need for processing and storing public key certificates under traditional public key infrastructure (PKI). However, the advantage of the CP-ABE comes with a major drawback which is known as a key escrow problem.

The KGC can decrypt every cipher text addressed to specific users by generating their attribute keys. This could be a potential threat to the data confidentiality or privacy in the data sharing systems. Another challenge is the key revocation. Since some users may change their associate attributes at some time, or some private keys might be compromised, key revocation or update for each attribute is necessary in order to make systems secure. This issue is even

more difficult especially in ABE, since each attribute is conceivably shared by multiple users (henceforth, we refer to such a set of users as an attribute group). This implies that revocation of any attribute or any single user in an attribute group would affect all users in the group. It may result in bottleneck during rekeying procedure or security degradation due to the windows of vulnerability.

1.1 Related Work

ABE comes in two flavors called key-policy ABE (KP-ABE) and cipher text-policy ABE. In KP-ABE, attributes are used to describe the encrypted data and policies are built into users' keys; while in CP-ABE, the attributes are used to describe users' credentials, and an encrypt or determines a policy on who can decrypt the data. Between the two approaches, CP-ABE is more appropriate to the data sharing system because it puts the access policy decisions in the hands of the data owners.

1.1.1 Removing Escrow

Most of the existing ABE schemes are constructed on the architecture where a single trusted authority or KGC has the power to generate the whole private keys of users with its master secret information. Thus, the key escrow problem is inherent such that the KGC can decrypt every cipher text addressed to users in the system by generating their secret keys at any time. Chase and Chow [22] presented a distributed KP-ABE scheme that solves the key escrow problem in a multi authority system. In this approach, all (disjoint) attribute authorities are participating in the key generation protocol in a distributed way such that they cannot pool their data and link multiple attribute sets belonging to the same user.

One disadvantage of this kind of fully distributed approach is the performance degradation. Since there is no centralized authority with master secret information, all attribute authorities should communicate with the other authorities in the system to generate a user's secret key. Recently, Chow [23] proposed an anonymous private key generation protocol in identity-based literature such that the KGC can issue a private key to an authenticated user without knowing the list of users' identities. It seems that this anonymous private key generation protocol works properly in ABE systems when we treat an attribute as an identity in this construction. However, we found that this cannot be adapted to ABE systems due to mainly two reasons. First, in Chow's protocol, identities of users are not public anymore, at least to the KGC, because the KGC can generate users' secret keys otherwise. Since public keys (attributes in the ABE setting) are no longer "public," it needs additional secure protocols for users to obtain the attribute information from attribute authorities. Second, since the collusion attack between users is the main security threat in ABE, the KGC issues different personalized key components to users by blinding them with a random secret even if they are associated with the same set of attributes.

1.1.2 Revocation

Be then court et al. [5] and Boldyreva et al. [8] proposed first key revocation mechanisms in CP- ABE and KP-ABE settings, respectively. These schemes enable an attribute key revocation by encrypting the message to the attribute set with its validation time. These attribute-revocable ABE schemes have the security degradation problem in terms of the backward and forward secrecy. They revoke attribute itself using timed rekeying mechanism, which is realized by setting expiration time on each attribute. In ABE systems, it is a considerable scenario that membership may change frequently in the attribute group.

Then, a new user might be able to access the previous data encrypted before his joining until the data are reen crypted with the newly updated attribute keys by periodic rekeying (backward secrecy). On the other hand, a revoked user would still be able to access the encrypted data even if he does not hold the attribute any more until the next expiration time (forward secrecy). Such an uncontrolled period is called the window of vulnerability.

Recently, the importance of immediate user revocation (rather than attribute revocation) has been taken notice of in many practical ABE-based systems [6], [7], [12]. The user revocation can be done by using ABE that supports negative clauses, proposed by Ostrovsky et al. [6]. One drawback in this scheme is that the private key size increases by a multiplicative factor of logn, where n is the maximum number of attributes. Lewko et al. [7] proposed more efficient instantiations of Ostrovsky et al.'s framework [6] for nonmonotonic ABE, where public parameters is only Oð1Þ group elements, and private keys for access

structures involving t leaf attributes is of size OðtÞ. However, these user- revocable schemes also have a limitation with regard to the availability. When a user is revoked even from a single attribute group, he loses all the access rights to the system, which is not desirable in many pragmatic scenarios since the other attributes may be still valid. Attrapadung and Imai [9] suggested another user-revocable ABE scheme addressing this problem by combining broadcast encryption schemes with ABE schemes.

However, in this scheme, the data owner should take full charge of maintaining all the membership lists for each attribute group to enable the direct user revocation. This scheme is not applicable to the data sharing system, because the data owners will no longer be directly in control of data after storing their data to the external storage server.

1.2 Contribution

In this paper, we propose a novel CP-ABE scheme for a secure data sharing system, which features the following achievements. First, the key escrow problem is resolved by a key issuing protocol that exploits the characteristic of the data sharing system architecture. The key issuing protocol generates and issues user secret keys by performing a secure two-party computation (2PC) protocol between the KGC and the data-storing center with their own master secrets. The 2PC protocol deters them from obtaining any master secret information of each other such that none of them could generate the whole set of user keys alone. Thus, users are not required to fully trust the KGC and the data storing center in order to protect their data to be shared. The data confidentiality and privacy can be cryptographically enforced against any curious KGC or data-storing center in the proposed scheme.

Second, the immediate user revocation can be done via the proxy encryption mechanism together with the CP-ABE algorithm. Attribute group keys are selectively distributed to the valid users in each attribute group, which then are used to reencrypt the cipher text encrypted under the CPABE algorithm. The immediate user revocation enhances the backward/forward secrecy of the data on any membership changes. In addition, as the user revocation can be done on each attribute level rather than on system level, finer grained user access control can be possible. Even if a user is revoked from some attribute groups, he would still be able to decrypt the shared data as long as the other attributes that he holds satisfy the access policy of the cipher text.

Data owners need not be concerned about defining any access policy for users, but just need to define only the access policy for attributes as in the previous ABE schemes. The proposed scheme delegates most laborious tasks of membership management and user revocation to the data storing center while the KGC is responsible for the attribute key management as in the previous CP-ABE schemes without leaking any confidential information to the other parties. Therefore, the proposed scheme is the most suitable for the data sharing scenarios where users encrypt the data only once and upload it to the data-storing centers, and leave the rest of the tasks to the data- storing centers such as reencryption and revocation.

1.3 Organization

The rest of the paper is organized as follows. We analyze the efficiency and security of the proposed scheme.

2. DATA SHARING ARCHITECTURE

2.1 System Description and Key Management

Fig. 1 shows the architecture of the data sharing system, which consists of the following system entities:

- **Key generation center**. It is a 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. It is assumed to be honest-but-curious.; thus, it should be prevented from accessing the plaintext of the encrypted data even if it is honest.
- **Data-storing center**. It is an entity that provides a data sharing service. It is in charge of controlling the accesses from outside users to the storing data and providing corresponding contents services. The data- storing center is another key authority that generates personalized user key with KGC and issues and revokes attribute group keys to valid users per each attribute.

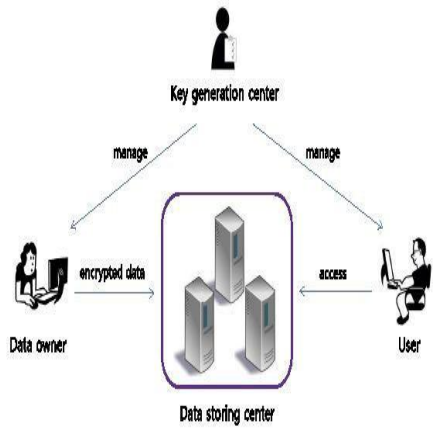

Figure 1: Architecture of a data sharing system.

- **Data owner.** It is a client who owns data, and wishes to upload it into the external data-storing center for ease of sharing or for cost saving. 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.
- **User.** It is an entity who wants to access the data. If a user possesses a set of attributes satisfying the access policy of the encrypted data, and is not revoked in any of the valid attribute groups, then he will be able to decrypt the cipher text and obtain the data. Since both of the key managers, the KGC and the data storing center, are semi trusted, they should be deterred from accessing plaintext of the data to be shared; meanwhile, they should be still able to issue secret keys to users.

2.1 Threat Model and Security Requirements

- **Data confidentiality**. Unauthorized users who do not have enough attribute satisfying the access policy should be prevented from accessing the plaintext of the data. Additionally, the KGC is no longer fully trusted in the data sharing system. Thus, unauthorized access from the KGC as well as the data-storing center to the plaintext of the encrypted data should be prevented.

- **Collusion resistance.** Collusion resistance is one of the most important security property required in ABE systems. If multiple users collude, they may be able to decrypt a cipher text by combining their attributes even if each of the users cannot decrypt the cipher text alone. We do not want these colluders to be able to decrypt the private data in the server by combining their attributes. Since we assume the KGC and data- storing center are honest, we do not consider any active attacks from them by colluding with revoked users.
- **Backward and forward secrecy.** In the context of attribute-based encryption, backward secrecy means that any user who comes to hold an attribute (that satisfies the access policy) should be prevented from accessing the plaintext of the previous data distributed before he holds the attribute. On the other hand, forward secrecy means that any user who drops an attribute should be prevented from accessing the plaintext of the subsequent data distributed after he drops the attribute, unless the other valid attributes that he is holding satisfy the access policy.

3. PRELIMINARIES AND DEFINITION

3.1 Cryptographic Background

We first provide a formal definition for access structure by recapitulating the definition. Then, we will briefly review the cryptographic background about the bilinear map and its security assumption.

3.2 Notations

In this paper, x 2R S denotes the operation of picking an element x at random and uniformly from a finite set S. For a probabilistic algorithm A; x $ A assigns the output of A to the variable x. 1_ denotes a string of _ ones, if _ 2 IN. A function _ : IN ! IR is negligible (negl(k)) if for every constant c _ 0 there exists kc such that _ðkÞ < k_c for all k > kc.

3.2.1Access Structure

Definition 1 (Access structure). Let fP1; P2; Png be a set of parties. A collection AA _ 2fP1;P2;...;Png is monotone if 8B;C: if B 2 AA and B _ C, then C 2 AA. An access structure (respectively, monotone access structure) is a collection (respectively, monotone collection) AA of nonempty subsets of fP1; P2; . . . ; Png, i.e., AA _ 2fP1;P2;...;Png n f;g.

The sets in AA are called the authorized sets, and the sets not in AA are called the

unauthorized sets. In CP-ABE schemes, the role of the parties is taken by the attributes. Thus, the access structure AA will contain the authorized sets of attributes. From now on, by an access structure we mean a monotone access structure.

4. SECURITY

In this section, we prove the security of the proposed scheme with regard to the security requirements discussed in Section 2.

4.1 Collusion Resistance

In the cipher text-policy attribute-based encryption, the secret sharing must be embedded into the cipher text instead to the private keys of users. Like the previous ABE schemes, the private keys (SK) of users are randomized with personalized random values selected by the KGC such that they cannot be combined in the proposed scheme. In order to decrypt a cipher text, the colluding attacker should recover eðg; gÞ_s. To recover this, the attacker must pair Cy from the cipher text and Dy from the other colluding users' private keys for an attribute _y (we suppose that the attacker does not hold the attribute _y).

However, this results in the value eðg; gÞ_s blinded by some random r, which is uniquely assigned to each user, even if the attribute group keys for the attributes that the user holds are still valid.2 This value can be blinded out if and only if the user has the enough key components to satisfy the secret sharing scheme embedded in the cipher text. Therefore, the desired value eðg; gÞ_s cannot be recovered by collusion attack since the blinding value is randomized from a particular user's private key.

4.2 Data Confidentiality

In our trust model, the KGC is no longer fully trusted as well as the data-storing center even if they are honest. Therefore, the plain data to be shared should be kept secret from them as well as from unauthorized users. Data confidentiality on the shared data against outside users who have not enough attributes can be trivially guaranteed. If the set of attributes of a user cannot satisfy the access tree in the cipher text, he cannot recover the desired value eðg; gÞrs during the decryption process, where r is a random value uniquely assigned to him. On the other hand, when a user is revoked from some attribute groups that satisfy the access policy, he cannot decrypt the Cipher text either unless the rest of the attributes of him satisfy the access policy. In order to decrypt a node x for an attribute _x, the user needs to pair C0x from the cipher text and D0 x from its private key. However, this cannot result in the value eðg; gÞrqxð0Þ, which is desired to generate eðg; gÞrs, since C0x is blinded by the updated attribute group key that the revoked user from the attribute group can by no means obtain (by key secrecy property of the one-way key agreement protocol).

Another attack on the shared data can be launched by the data-storing center and the KGC. Since they cannot be totally trusted by users (suppose that the data-storing center could be compromised and the KGC tries to exploit private user data maliciously for its profit), the confidentiality for the shared data against them is another essential security criteria for secure data sharing. The KGC issues a set of attribute keys, SKK;u, to an authenticated user u for the attributes that the user is entitled. The data-storing center issues a user a personalized secret key, SKD;u, by performing a secure 2PC protocol with the KGC.

As we discussed in Theorem 1, this key generation protocol discourages the two parties to obtain each other's master secret key and determine the secret key issued from each other. Therefore, they could not have enough information to decrypt the data. Even if the data-storing center manages each attribute group key, it cannot decrypt any of the nodes in the access tree in the cipher text. This is because it is only authorized to reencrypt the cipher text with each attribute group key, but is not allowed to decrypt it (that is, any of the attribute keys for the corresponding attributes in the Cipher text issued by the KGC are not given to the data-storing center). Therefore, data confidentiality against the honest-but curious KGC and data-storing center is also guaranteed.

4.3 Backward and Forward Secrecy

When a user comes to hold a set of attributes that satisfy the access policy in the cipher text at some time instance, the corresponding attribute group keys are updated and delivered to the valid attribute group members securely (including the user). In addition, all of the components encrypted with a secret key s in the cipher text are reencrypted by the data-storing

center with a new secret s0, and the cipher text components corresponding to the attributes are also reencrypted with the updated attribute group keys.

Even if the user has stored the previous cipher text before, he obtains the attribute keys and the holding attributes satisfy the access policy, he cannot decrypt the pervious cipher text. This is because, even if he can succeed in computing eðg; gÞrðsþs0Þ from the current cipher text, it would not help to recover the desired value eðg; gÞ_s for the previous cipher text since it is blinded by a random s0. Therefore, the backward secrecy of the shared data is guaranteed in the proposed scheme.

On the other hand, when a user comes to drop a set of attributes satisfying the access policy in the cipher text at some time instance, the corresponding attribute group keys are also updated and delivered to the valid attribute group members securely (excluding the user). Then, all of the components encrypted with a secret key s in the cipher text are reencrypted

4.3.1 HUR: IMPROVING SECURITY AND EFFICIENCY IN ATTRIBUTE-BASED DATA SHARING 2281

Another collusion attack scenario is the collusion between revoked users in order to obtain the valid attribute group keys for some attributes that they are not authorized to have (e.g., due to revocation). The attribute group key distribution protocol in the proposed scheme is secure in terms of the (established) key secrecy as we discussed in Section 3.1.5.

Thus, the colluding revoked users can by no means obtain any valid attribute group keys for attributes that they are not authorized to hold. By the data- storing center with a new secret s0, and the cipher text components corresponding to the attributes are also reencrypted with the updated attribute group keys. Then, the user cannot decrypt any nodes corresponding to the attributes after his revocation due to the blindness resulted from newly updated attribute group keys. In addition, even if the user has recovered eðg; gÞ_s before he was revoked from the attribute groups and stored it, it would not help to determine the desired value eðg; gÞ_ðsþs0Þ in the subsequent cipher text since it is also reencrypted with a new random s0. Therefore, the forward secrecy of the shared data is also guaranteed in the proposed scheme.

5. CONCLUSION

The enforcement of access policies and the support of policy updates are important challenging issues in the data sharing systems. In this study, we proposed a attribute based data sharing scheme to enforce a fine-grained data access control by exploiting the characteristic of the data sharing system. The proposed scheme features a key issuing mechanism that removes key escrow during the key generation.

The user secret keys are generated through a secure two-party computation such that any curious key generation center or data-storing center cannot derive the private keys individually. Thus, the proposed scheme enhances data privacy and confidentiality in the data sharing system against any system managers as well as adversarial outsiders without corresponding (enough) credentials. The proposed scheme can do an immediate user revocation on each attribute set while taking full advantage of the scalable access control provided by the cipher text policy attribute-based encryption.

Therefore, the proposed scheme achieves more secure and fine-grained data access control in the data sharing system. We demonstrated that the proposed scheme is efficient and scalable to securely manage user data in the data sharing system.

REFERENCES

[1] J. Anderson, “Computer Security Planning Study,” Technical Report 73-51, Air Force Electronic System Division, 1972.

[2] L. Ibraimi, M. Petkovic, S. Nikova, P. Hartel, and W. Jonker, “Mediated Ciphertext-Policy Attribute- Based Encryption and Its Application,” Proc. Int’l Workshop Information Security Applications (WISA ’09), pp. 309-323, 2009.

[3] A. Sahai and B. Waters, “Fuzzy Identity-Based Encryption,” Proc. Int’l Conf. Theory and Applications of Cryptographic Techniques (Eurocrypt ’05), pp. 457-473, 2005.

[4] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data,” Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006.

[5] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-Policy Attribute-Based Encryption,” Proc. IEEE Symp. Security and Privacy, pp. 321-334, 2007.

[6] R. Ostrovsky, A. Sahai, and B. Waters, “Attribute-

Based Encryption with Non-Monotonic Access Structures," Proc. ACM Conf. Computer and Comm. Security, pp. 195-203, 2007.

[7] A. Lewko, A. Sahai, and B. Waters, "Revocation Systems with Very Small Private Keys," Proc. IEEE Symp. Security and Privacy, pp. 273-285, 2010.

[8] A. Boldyreva, V. Goyal, and V. Kumar, "Identity- Based Encryption with Efficient Revocation," Proc. ACM Conf. Computer and Comm. Security, pp. 417-426,2008.

[9] N. Attrapadung and H. Imai, "Conjunctive Broadcast and Attribute-Based Encryption," Proc. Int'l Conf. Palo Alto on Pairing-Based Cryptography (Pairing), pp. 248-265, 2009.