INTERNATIONAL JOURNAL OF COMPUTATIONAL MATHEMATICAL IDEAS

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6314-6319 (2014)

Better Safety and Efficiency in Attribute Based Data Sharing

G. Kishore Kumar [#1], G. Sreenivasa Reddy, Associate Professor, M .Tech [#2]

Department Of Computer Science and Engineering.

Buchepalli Venkayamma Subbareddy Engineering College, *Affiliated to JNTUK*, Chimakurthy, Andhra Pradesh, India.

[1*] kishore.pandu67@gmail.com

[2*] bvsrhodcse@gmail.com

Abstract -- **The Key Generation Center (KGC) could decrypt any messages addressed to specific users by generating their private keys. This is not suitable for data sharing scenarios where the data owner would like to make their private data only accessible to designated users key. To overcome this problem we propose escrow problem which means a written agreement delivered to a third party and Attribute-Based Encryption (ABE). Attribute-based encryption is a promising cryptographic approach, is a fine-grained data access control which provides a way of defining access policies based on different attributes of the requester, environment and the data object. 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. 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 an 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.**

Keywords- **Attribute-based encryption, revocation, access control Data sharing.**

I. INTRODUCTION

People can share their lives with friends by uploading their private photos or messages into the online social networks; or upload highly sensitive personal health records (PHRs) into online data servers such as Microsoft Health Vault, Google Health for ease of sharing with their primary doctors or for cost saving. As people enjoy the advantages of these new technologies and services, their concerns about data security and access control also arise. Improper use of the data by the storage server or unauthorized access by outside users could be potential threats to their data. People would like to make their sensitive or private data only accessible to the authorized people with credentials they specified. 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 to encrypt or to define the attribute set over a universe of attributes that a decrypt or needs to possess in order to decrypt the cipher text, and enforce it on the contents. 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 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.

II. RELATED WORK

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, 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. 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 encryptor determines a policy on who can decrypt the data. We create public key revocation encryption systems with small cryptographic private and public keys. Our systems have two important features relating respectively to public and private key size. First, public keys in our two systems are short and enable a

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6314-6319 (2014)

user to create a cipher text that revokes an unbounded number of users. This is in contrast to other systems where the public parameters bound the number of users in the system and must be updated to allow more users. Second, the cryptographic key material that must be stored securely on the receiving devices is small. Keeping the size of private key storage as low as possible is important as cryptographic keys will often be stored in tamper-resistant memory, which is more costly. This can be especially critical in small devices such as sensor nodes, where maintaining low device cost is particularly crucial. 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. This results in OðN2Þ communication overhead on the system setup phase and on any rekeying phase, and requires each user to store OðN2Þ additional auxiliary key components besides the attributes keys, where N is the number of authorities in the system. 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. The random secret is unique and should be consistent with the same user for any possible attribute change (such as adding some attributes) of the user. However, it is impossible for the KGC to issue a personalized key component with the same random secret as that of attribute key components to a user, since the KGC can by no means know which random secrets (used to issue a set of attributes key components) are assigned to which users in the Chow's key issuing protocol. 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]. To do so, one just adds conjunctively the AND of negation of revoked user identities (where each is considered as an attribute here). 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 non-monotonic ABE, where public parameters is only Oð1Þ group elements, and private keys for access structures involving t leaf attributes is of size. 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 schemes 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. Yu et al. [13] also recently addressed the user revocation in the ABE-based data sharing system. In this scheme, the user revocation is realized using proxy reencryption by the data server. However, in order to revoke users, the KGC should generate all secret keys including the proxy key on behalf of the data server. Then, the server would reencrypt the ciphertext under the proxy key received from the KGC to prevent revoked users from decrypting the ciphertext. Thus, the key escrow problem is also inherent in this scheme, since the KGC manages all secret keys of users as well as the proxy keys of the data server. 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 datastoring 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 ciphertext 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, more finegrained 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.

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6314-6319 (2014)

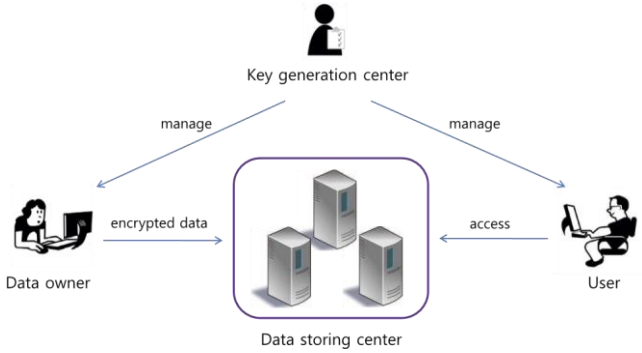


Fig. 1. Architecture of a data sharing system.

We propose a novel CP-ABE scheme for a secure data sharing system. 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. The data confidentiality and privacy can be cryptographically enforced against any curious KGC or data storing center in the proposed scheme.

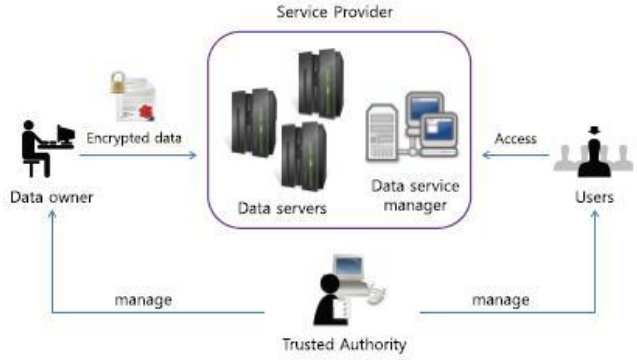
 the key escrow problem could be solved by escrow-free key issuing protocol, which is constructed using the secure two-party computation between the key generation center and the data storing center.

 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.

III. ATTRIBUTE BASED DATA SHARING

3.1 System Description and Key Management

The architecture of the data sharing system, which consists of the following system entities: 1. 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. That is, it will honestly execute the assigned tasks in the system; however, it would like to learn information of encrypted contents as much as possible. 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 the KGC, and issues and revokes attribute group keys to valid users per each attribute, which are used to enforce a fine-grained user access control. Similar to the previous schemes [2], [13], [14], we assume the data-storing center is also semi trusted (that is, honest-but-curious) like the KGC. 3. 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 datastoring center, are semitrusted, 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. In order to realize this somewhat contradictory requirement, the two parties engage in the arithmetic 2PC protocol with master secret keys of their own, and issue independent key components to users during the key issuing phase. The 2PC protocol deters them from knowing each other's master secrets so that none of them can generate the whole set of secret keys of users individually. Thus, we take an assumption that the KGC does not collude with the data-storing center since they are honest as in [2], [13], [14] otherwise, they can guess the secret keys of every user.



Fine-grained user access control. 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 is an entity who wants to access the data. 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. The data confidentiality and privacy can be cryptographically enforced against any curious KGC or data storing center in the proposed scheme. Efficiency Attribute Based Data Sharing System encrypting the content, hence solving the performance degradation problem of distributed approach.

3.2. Methodology

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 centre with their own master secrets. Most of the existing ABE schemes are constructed

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6314-6319 (2014)

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.

A. Cipher text – Policy Attribute Based Encryption with User Revocation We define the CP-ABE with user revocation capability scheme. The scheme consists of the following six algorithms: Setup: The setup algorithm is a randomized algorithm that takes no input other than the implicit security parameter. It outputs the public key PK and a master key MK. AttrKeyGen: The attribute key generation algorithm takes as input the master key MK, a set of attributes and a set of user indices. It outputs a set of private attribute keys for each user in U that identifies with the attributes set. KEKGen: The key encrypting key (KEK) generation algorithm takes as input a set of user indices and outputs KEKs for each user in U, which will be used to encrypt attribute group keys. Encrypt: The encryption algorithm is a randomized algorithm that takes as input the public parameter PK, a message M, and an access structure AA over the universe of attributes. It outputs a cipher text such that only a user who possesses a set of attributes that satisfies the access structure will be able to decrypt the message.

ReEncrypt: The re-encryption algorithm is a randomized algorithm that takes as input the cipher text including an access structure and a set of attribute groups. If the attribute groups appear in AA, it re-encrypts for the attributes; else, returns specifically, it outputs a re-encrypted cipher text such that only a user who possesses aset of attributes that satisfies the access structure and has a valid membership for each of them at the same time will be able to decrypt the message. Decrypt: The decryption algorithm takes as input the cipher text which contains an access structure AA, a private key SK, and a set of attribute group keys for a set of attributes.

IV. CONCLUSION AND FUTURE WORK

To 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. Data privacy and confidentiality in the data sharing system against any system managers as well as adversarial outsiders without corresponding (enough) credentials. In the future, it would be interesting to consider attribute-based encryption systems by applying advanced cryptosystem for data sharing. In future, we encrypt multimedia content, Solve the performance degradation of fully distributed approach, Neglected key expired time, we can use multi Data Storing Centre, Proxy servers to update user secret key without disclosing user attribute information.

V. REFERENCES

[1]. Luan Ibraimi, Milan Petkovic, Svetla Nikova, Pieter Hartel and Willem Jonker, ―Mediated Cipher text-Policy Attribute-Based Encryption and Its Application‖, Information Security Applications, Lecture Notes in Computer Science, DOI: 10.1007/978-3-642-10838-9_23, pp 309-323,2009.

[2]. Junbeom Hur and Dong Kun Noh, ―Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems‖, IEEE Transactions on Parallel and Distributed Systems, pp 1214-1221, 2011.

[3].Alexandra Boldyreva, Vipul Goyal, Virendra Kumar, ―Identity-based encryption with efficient revocation‖, Proceedings of the 15th ACM conference on Computer and communications security, ISBN: 978-1-59593-810-7, pp 417-426, 2008.

[4].Shucheng Yu, Cong Wang, Kui Ren, Wenjing Lou, ―Attribute based data sharing with attribute revocation‖, Proceedings of the 5th ACM Symposium on Information, ISBN: 978-1-60558-936-7, pp 261-270, 2010.

[5]. L. Cheung and C. Newport, "Provably Secure Ciphertext Policy ABE," Proc. ACM Conf. Computer and Comm. Security, pp. 456-465, 2007.

[6]. X. Liang, Z. Cao, H. Lin, and D. Xing, "Provably Secure and Efficient Bounded Ciphertext Policy Attribute Based Encryption," Proc. Int'l Symp. Information, Computer, and Comm. Security (ASIACCS), pp. 343-352, 2009.

[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.

[10] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters, "Secure Attribute-Based Systems," Proc. ACM Conf. Computer and Comm. Security, 2006.

[11] S. Rafaeli and D. Hutchison, "A Survey of Key Management for Secure Group Communication," ACM Computing Surveys, vol. 35, no. 3, pp. 309-329, 2003.

[12] P. Golle, J. Staddon, M. Gagne, and P. Rasmussen, "A Content- Driven Access Control System," Proc. Symp. Identity and Trust on the Internet, pp. 26-35, 2008.

[13] 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 '10), 2010.

[14] S.D.C. Vimercati, S. Foresti, S. Jajodia, S. Paraboschi, and P. Samarati, "Over-Encryption: Management of Access Control Evolution on Outsourced Data," Proc. Int'l Conf. Very Large Data Bases (VLDB '07), 2007.

[15] 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.

[16] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-Based Encryption for Fine-Grained Access Control of Encrypted

ISSN: 0974-8652 / www.sijcmi.weebly.com / VOL 6-NO3-PP 6314-6319 (2014)

Data," Proc. ACM Conf. Computer and Comm. Security, pp. 89-98, 2006.
[17] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-Policy Attribute-Based Encryption," Proc. IEEE Symp. Security and Privacy, pp. 321-334, 2007.
[18] 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.