Secure Forensic Report Retrieval Application using Cipher Text-Policy Attribute-Based Encryption

Vinod I. Jondhale[1], Manisha Y. Joshi[2]

[1]Student, Department of Computer Science and Engineering, MGM's College of Engineering, Nanded, Maharashtra, India

[2]Associate Professor, Department of Computer Science and Engineering, MGM's College of Engineering, Nanded, Maharashtra, India

Abstract*: In cryptography, confidentiality, authenticity and anonymity are studied for long, and to provide security is the reason for all modern cryptographic research. Attribute based view has developed gradually by the requirements of security in a distributed setting. Attribute based encryption have been developed to give a fine grained access control on the data and at present attribute-based systems have wide range of applications in new decentralized settings. When the list of users may not be known prior, attribute based encryption mechanism is useful in these settings. In these settings, all users may possess some credentials or attributes, and these are used to determine access control and also to provide some degree of anonymity with respect to the user's identity. Cipher text policy attribute based encryption is a scheme that gives a way to separate the credentials or attributes from the access policy and combine them at a later stage to provide secure access to protected data. We have proposed similar application for secure forensic data report retrieval where the setup algorithm first generates keys using the attributes of the users, the forensic analyst defines a set of attributes to encrypt the report and store the report in temporary storage. The receivers such as policeman, relatives, doctor will have to possess the defined set of attributes to decrypt the report.*

Keywords: Cipher text-policy attribute-based encryption (CP-ABE), attribute-based encryption (ABE), access policy, fine-grained access control**.**

1. Introduction

In a public key cryptosystem, there are two different keys: a public key and a private key. For example, John uses Bella's public key to encrypt a message to Bella. Bella uses her private key to decrypt the message. In many situations, when the user encrypts data, it is important that he has a specific access control policy for deciding on who can decrypt the data. For example, suppose that a commander stores confidential information at the storage node and ants the information to be accessed by members of "BSF Battalion 1" who are participating in "Kashmir". The soldier satisfying the attributes can access it. The commander may specify the following access structure for accessing this information - ("ARMY GENERAL") **OR** **((**"BSF BATTALION 1") **AND** ("Kashmir")).By the above predicate, the commander would mean that army general and the soldiers belonging to BSF battalion 1 and Kashmir can only access the data. The existing public key encryption methods allow a party to encrypt data to a particular user. It is unable to efficiently handle more users by applying expressive types of encryption access control. The attribute-based encryption allows the access control over the encrypted data for multiple users.

Attribute-based encryption is a type of public-key encryption in which the secret key of a user and the cipher text are dependent upon attributes and in such a system, the decryption of a cipher text is possible only if the set of attributes of the user key matches the attributes of the cipher text [1]. The concept of attribute-based encryption was first proposed by Amit Sahai *et al.* and Brent Waters *et al.* There are two types of Attribute-Based Encryption schemes they are Key-Policy Attribute-Based Encryption (KP-ABE) and Cipher text-Policy Attribute-Based Encryption (CPABE).

The cipher text-policy ABE (CP-ABE) provides a way of encrypting data such that the encryptor defines the set of attributes that the decryptor needs to possess in order to decrypt the cipher text. Thus, different users are allowed to decrypt different pieces of data per the security access policy. In CP-ABE, the key authority generates private keys of users by applying the authority's master secret keys to users associated set of attributes [1]. In KP-ABE, the encryptor only labels a cipher text with a set of attributes. The key authority chooses a policy for each user that determines which cipher texts he can decrypt and issues the key to each user by embedding the policy into the user's key [1].The secure forensic report retrieval application uses the concept of cipher text-policy attribute-based encryption for storage and retrieval of the stored data. The forensic analyst encrypts the autopsy reports using set of attributes which needs to be possessed by the receiver to decrypt the data. The receivers are the doctor, police and the relatives of the patient.

2. Literature Survey

Sahai *et al.* and Waters *et al.* [1] introduced attribute-based encryption (ABE) as a method for encrypted access control. In an attribute-based encryption system cipher texts are not necessarily encrypted to one particular user as in traditional public key cryptography. Instead both user's cipher texts and private keys will be associated with a set of attributes or it will be associated as a policy over attributes. A user is able to decrypt a ciphertext if there is a "match" between his private key and the ciphertext. Sahai *et al.* and Waters *et al.* presented a Threshold ABE system in which cipher texts were labeled with a set of attributes S and a user's private key was associated with both a threshold parameter k and another set of attributes S'. In order for a user to decrypt a ciphertext at least k attributes must overlap between the ciphertext and his private keys. Fuzzy identity-based encryption [2,3,4] scheme was designed that could use biometric identities as attributes. The primary drawback of the Sahai-Waters [1] threshold ABE system is that the threshold semantics are not very expressive and therefore are limiting for designing more general systems. Goyal *et al.* introduced the idea of a more general key-policy attribute-based encryption system. In their construction a ciphertext is associated with a set of attributes and a user's key can be associated with any monotonic tree access structure. The construction of Goyal *et al.* can be viewed as an extension of the Sahai-Waters techniques which embedded a Shamir [5] secret sharing scheme in the private key, the authority embeds a more general secret sharing scheme for monotonic access trees. Goyal *et al.* also suggested the possibility of a ciphertext-policy ABE scheme, but did not offer any constructions. Pirretti *et al.* [6] gave an implementation of the threshold ABE encryption system, demonstrated different applications of attribute-based encryption schemes and addressed several practical notions such as key-revocation. In recent work, Chase [7] gave a construction for a multi-authority attribute-based encryption system, where each authority would administer a different domain of attributes. The primary challenge in creating multi-authority ABE is to prevent collusion attacks between users that obtain key components from different authorities. While the Chase system used the threshold ABE system as its underlying ABE system at each authority, the problem of multi-authority ABE is in general orthogonal to finding more expressive ABE systems. In addition, there is a long history of access control for data that is mediated by a server.

Fine-grained access control systems facilitate granting differential access rights to a set of users and allow flexibility in specifying the access rights of individual users. Several techniques are known for implementing fine grained access control. Access control relies on software checks to ensure that a user can access a piece of data only if he is authorized to do so. This situation is not particularly appealing from a security standpoint. In the event of server compromise, for example, as a result of a software vulnerability exploit, the potential for information theft is immense. Furthermore, there is always a danger of "insider attacks" wherein a person having access to the server steals and leaks the information, for example, for economic gains. Some techniques (see, e.g., [8]) create user hierarchies and require the users to share a common secret key if they are in a common set in the hierarchy. The data is then classified according to the hierarchy and encrypted under the public key of the set it is meant for. Clearly, such methods have several limitations. If a third party must access the data for a set, a user of that set either needs to act as an intermediary and decrypt all relevant entries for the party or must give the party its private decryption key, and thus let it have access to all entries. In many cases, by using the user hierarchies it is not even possible to realize an access control equivalent to monotone access trees.

3. Background

The formal definitions for the security of ciphertext policy attribute based encryption (CPABE) are given below. The work of Goyal *et al.* [13] is used to define an access structure and security definitions.

Access Structure [12] - Let $\{P_1, P_2, \ldots, P_n\}$ be a set of parties. A collection $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}}$ is monotone if $\forall B, C$: if $B \in A$ and $B \subseteq C$ then C ∈ A. An access structure is a collection A of non-empty subsets of $\{P_1, P_2, \ldots, P_n\}$ that is $A \subseteq 2^{\{P_1, P_2, \ldots, P_n\}} \setminus \{\emptyset\}$. The sets in A are called the authorized sets, and the sets not in A are called the unauthorized sets.

The role of the parties is taken by the attributes. Thus, the authorized sets of attributes will be present in access structure A . It is possible to realize general access structures by having the 'not' of an attribute as a separate attribute altogether. Therefore, the number of attributes will be doubled in the system. A cipher text-policy attribute based encryption scheme consists of four fundamental algorithms: Setup, Encrypt, KeyGen, and Decrypt.

Setup- The setup algorithm takes no input other than the implicit security parameter. It outputs the public parameters PK and a master key MK.

Encrypt (PK, M, A) - The public parameters PK, a message M, and an access structure A are taken as input in encryption algorithm over the universe of attributes. The algorithm will encrypt M and produce a cipher text CT such that only a user that possesses a set of attributes that satisfies the access structure will be able to decrypt the message.

Key Generation (MK, S)- The key generation algorithm takes as input the master key MK and a set of attributes S that describe the key. It outputs a private key SK.

Decrypt (PK, CT, SK)- The public parameters PK, a cipher text CT, which contains an access policy A, and a private key SK, which is a private key for a set S of attributes are taken as the inputs for the decryption algorithm. If the set S of attributes satisfies the access structure A then the algorithm will decrypt the cipher text and return a message M.

Delegate (SK, S˜) - A secret key SK for some set of attributes S and a set S' ⊆ S are taken as inputs for the delegate algorithm. It output a secret key SK for the set of "attributes S". Like identity-based encryption schemes [2, 3, 4] the security model allows the user to query for any private keys that cannot be used to decrypt the challenge cipher text. In CP-ABE the cipher texts are identified with access structures and the private keys with attributes. It follows the security definition that the user will choose to be challenged on an encryption to an access structure A^* and can ask for any private key S such that S does not satisfy S^*.

4. System Description

There are four modules in the implementation of secure forensic report. The modules are as follows:

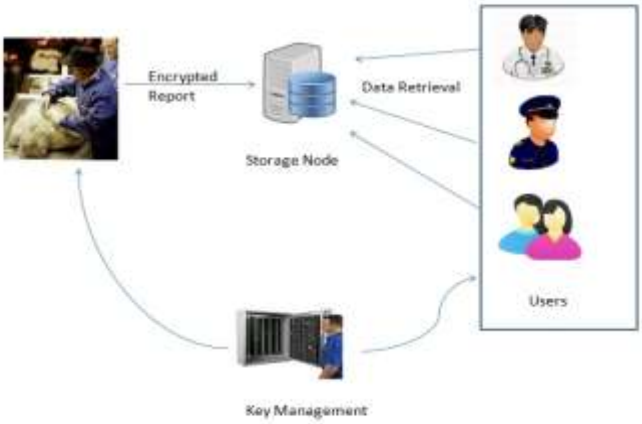

Figure 1: System Architecture of Forensic Report Application

Forensic analyst (Sender) - It is an entity that owns autopsy reports and stores them into the external data storage node for sharing and reliable delivery to users. A forensic analyst is responsible for defining (attribute based) access policy and enforcing it on its own data by encrypting the data under the policy before storing it to the storage node. The attributes used to encrypt the data are patient id, patient name, age, date of birth, relative of patient, doctor id, doctor name, date of death, police id, police name, and death reason.

Storage Node - This is an entity that stores data from forensic analyst and provide corresponding access to users. It may be mobile or static. The storage node to be semi trusted, that is honest-but-curious.

Doctor (User) - This is a user who wants to access the data stored at the storage node. If a doctor possesses a set of attributes satisfying the access policy of the encrypted data defined by the forensic analyst, and is not revoked in any of the attributes, then he will be able to decrypt the cipher text and obtain the data. The attributes the doctor has to possess are doctor id, doctor name, patient id, patient name, date of birth, and date of death.

Police (User) - This is a user who wants to access the data stored at the storage node. If a police man possesses a set of attributes satisfying the access policy of the encrypted data defined by the forensic analyst, and is not revoked in any of the attributes, then he will be able to decrypt the ciphertext and obtain the data. The attributes the police man has to possess are police id, police name, doctor name, patient name, death reason, and date of death.

Relatives (User) - This is a user who wants to access the data stored at the storage node. If a relative possesses a set of attributes satisfying the access policy of the encrypted data defined by the forensic analyst, and is not revoked in any of the attributes, then he will be able to decrypt the ciphertext and obtain the data. The attributes the relative has to possess are patient id, patient name, patient age, patient name, date of birth, relation of the relative, and doctor name.

5. Implementation

The implementation uses a160-bit elliptic curve group based on the super singular curve $y^2 = x^3 + x$ over a 512-bit finite field. On the test machine, the PBC library can compute pairings in approximately 5.5ms, and exponentiations in G_0 and G_1 take about 6.4ms and 0.6ms respectively. Fig. 2 shows the key generation time, as the number of attributes increases the time for key generation also increases. The encryption time is depended on the leaf nodes in the policy, as the leaf nodes increases the encryption time also increases. The performance of decryption is slightly more difficult to measure in the absence of a precise application, since the decryption time can depend significantly on the particular access trees and set of attributes involved.

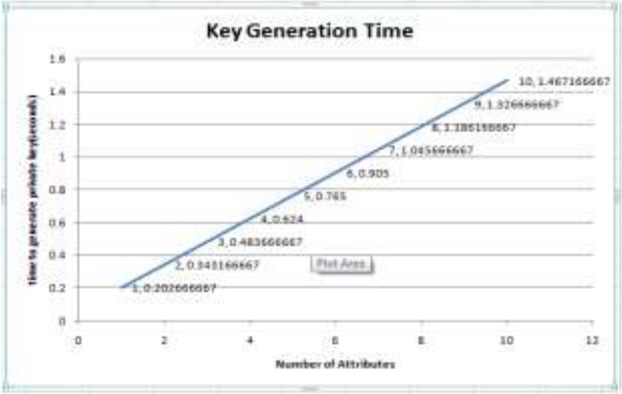

Figure 2: Key Generation Time

For the below measurement the time is calculated taking into consideration that all the attributes are required to decrypt the data.

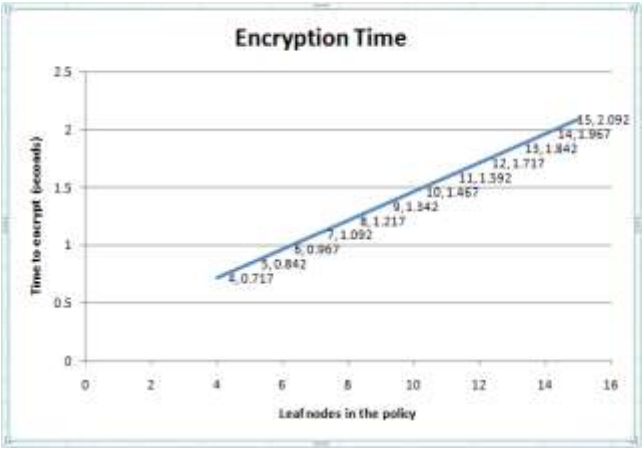

Figure 3: Encryption Time

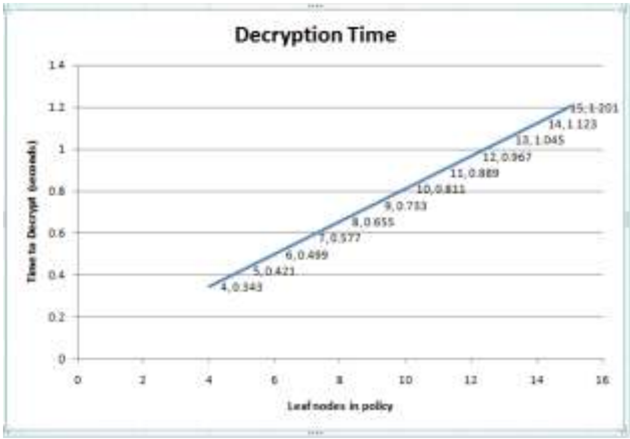

Figure 4: Decryption Time

6. Conclusions

Cipher text policy attribute based encryptions are scalable cryptographic solution to the access control and secure data retrieval issues. In recent years, attribute-based encryption is a relatively attractive research topic and has many attracting properties. It provides a fine-grained and no interactive access control mechanism of encrypted data and has great potential applications in many fields. Our project is not the unique one, but is an Endeavour attempt to have a precise scenario of what the terms “secure data retrieval using cipher text policy attribute based encryption” is meant to be and its implementation. Our system can enhance the security of the sensitive data by using cipher text policy attribute based encryption mechanism. In this project, we have implemented secure data retrieval for disruption tolerant military network using cipher text policy attribute based encryption and also designed secure forensic data retrieval using cipher text policy attribute based encryption.

References

[1] Sahai and B.Waters, “Fuzzy Identity Based Encryption,” Advances in Cryptology – Eurocrypt, volume 3494 of LNCS, pages 457–473. Springer, 2005.

[2] C. Cocks, “An identity based encryption scheme based on quadratic residues,” IMA Int. Conference., pages 360–363, 2001.

[3] A. Shamir, “Identity Based Cryptosystems and Signature Schemes,” Advances in Cryptology – CRYPTO, volume 196 of LNCS, pages 37–53. Springer, 1984.

[4] D. Boneh and M. Franklin, “Identity Based Encryption from the Weil Pairing,” Advances in Cryptology – CRYPTO, volume 2139 of LNCS, pages 213–229.Springer, 2001.

[5] A. Shamir, “How to share a secret,” Communication. ACM, 22(11):612–613, 1979.

[6] M. Pirretti, P. Traynor, P. McDaniel, and B. Waters,“ Secure Attribute-Based Systems,” ACM conference on Computer and Communications Security (ACM CCS), 2006.

[7] M. Chase, “Multi-authority attribute-based encryption,” the Fourth Theory of Cryptography Conference (TCC 2007), 2007.

[8] S. G. Akl and P. D. Taylor, “Cryptographic Solution to a Multi Level Security Problem,” Advances in Cryptology - CRYPTO, 1982.

[9] G. R. Blakley, “Safeguarding cryptographic keys,” National Computer Conference, pages 313-317. American Federation of Information Processing Societies Proceedings, 1979.

[10] J. Benaloh and Leichter J, “Generalized Secret Sharing and Monotone Functions,” Advances in Cryptology - CRYPTO, volume 403 of *LNCS*, pages 27-36. Springer, 1988.

[11] Y. Dodis, N. Fazio, A. Lysyanskaya, and D.F. Yao, “ ID-Based Encryption for Complex Hierarchies with Applications to Forward Security and Broadcast Encryption,” ACM conference on Computer and Communications Security (ACM CCS), pages 354-363, 2004.

[12] A. Beimel, “Secure Schemes for Secret Sharing and Key Distribution” PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[13] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute Based Encryption for Fine-Grained Access Control of Encrypted Data,” ACM conference on Computer and Communications Security (ACM CCS), 2006.