Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 76 (2014) 28562-28565

ISSN: 2229-712X

Exploring the data access control for multi-authority cloud storage with efficient revocation

R.Balaji and N.Arthi

Sir Issac Newton College of Engineering and Technology, Nagapattinam, TamilNadu, India.

ARTICLE INFO

Article history:

Received: 14 September 2014;
Received in revised form:
10 November 2014;
Accepted: 17 November 2014;

Keywords

Cloud server,
Business record,
Multi-authority,
Cloud storage,
Attribute revocation,
Attribute Based Encryption (ABE).

ABSTRACT

Business record (BR) is an emerging patient-centric model of information exchange, which is outsourced to be stored at a third party, such as cloud providers. To assure business data control over access to their own BRs, it is promising method to encrypt the BRs before outsourcing. To achieve scalable data access control for BRs, we leverage attribute based encryption techniques to encrypt business file. In data owner, the users divide to the BR system into multiple security domains that reduces key management complexity for owner and users. A degree of privacy is guaranteed by exploiting multi-authority ABE.

Introduction

Cloud storage is an important service of cloud computing [1], which offers services for data owners to host their data in the cloud. This new paradigm of data hosting and data access services introduces a great challenge to data access control. Because the cloud server cannot be fully trusted by data owners, they can no longer rely on servers to do access control. Cipher text-Policy Attribute-based Encryption (CP-ABE) [2], [3] is regarded as one of the most suitable technologies for data access control in cloud storage systems, because it gives the data owner more direct control on access policies. In CP-ABE scheme, there is an authority that is responsible for attribute management and key distribution. The authority can be the registration office in a university, the human resource department in a company, etc. The data owner defines the access policies and encrypts data according to the policies. Each user will be issued a secret key reflecting its attributes. A user can decrypt the data only when its attributes satisfy the access policies.

In multi-authority cloud storage systems, users' attributes can be changed dynamically. A user may be entitled some new attributes or revoked some current attributes. And his permission of data access should be changed accordingly. However, existing attribute revocation methods [9] either rely on a trusted server or lack of efficiency, they are not suitable for dealing with the attribute revocation problem in data access control in multi-authority cloud storage systems. Attribute based encryption is implemented based on AES technique, which is an encryption algorithm for securing sensitive but unclassified material by U.S. Government agencies and, as a likely consequence, may eventually become the de facto encryption standard for commercial transactions in the private sector. It is used to provide encrypt and decrypt the attribute values of patient records.

AES Features

AES was designed to be efficient in both hardware and software, and supports a block length of 128 bits and key lengths of 128, 192, and 256 bits. AES allows for three different key lengths: 128, 192, or 256 bits. Most of our discussion will assume that the key length is 128 bits. Encryption consists of 10 rounds of processing for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. Except for the last round in each case, all other rounds are identical.

System Model And Security Model

System Model

We consider a data access control system in multi-authority cloud storage, as described in Fig. 1. There are five types of entities in the system: a certificate authority (CA), attribute authorities (AAs), data owners (owners), the cloud server (server) and data consumers (users). The CA is a global trusted certificate authority in the system. It sets up the system and accepts the registration of all the users and AAs in the system. For each legal user in the system, the CA assigns a global unique user identity to it and also generates a global public key for this user. However, the CA is not involved in any attribute management and the creation of secret keys that are associated with attributes. For example, the CA can be the Social Security Administration, an independent agency of the United States government. Each user will be issued a Social Security Number (SSN) as its global identity. Every AA is an independent attribute authority that is responsible for entitling and revoking user's attributes according to their role or identity in its domain. In our scheme, every attribute is associated with a single AA, but each AA can manage an arbitrary number of attributes. Every AA has full control over the structure and semantics of its attributes. Each AA is responsible for generating a public attribute key for each attribute it manages and a secret key for each user reflecting his/her attributes.

Each user has a global identity in the system. A user may be entitled a set of attributes which may come from multiple attribute authorities. The user will receive a secret key associated with its attributes entitled by the corresponding attribute authorities.

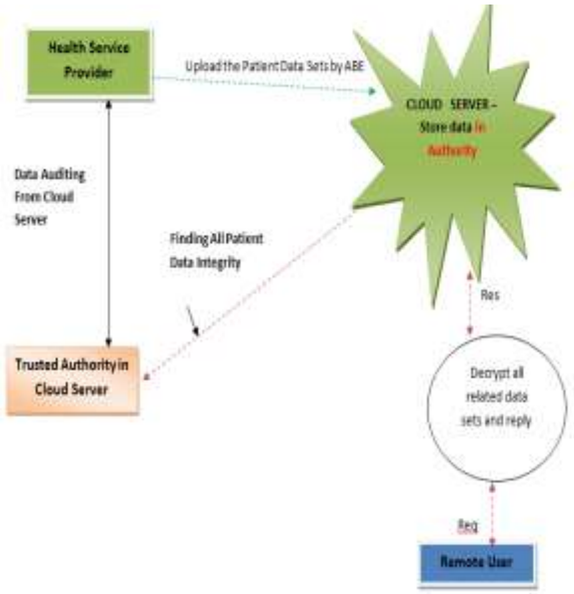

Health Service Provider
Upload the Patient Data Sets by ABE
CLOUD SERVER – Store data in Authority
Data Auditing From Cloud Server
Finding All Patient Data Integrity
Res
Trusted Authority in Cloud Server
Decrypt all related data sets and reply
Req
Remote User

Figure 1 System model of data access control in Multi authority Cloud Environment

Each owner first divides the data into several components according to the logic granularities and encrypts each data component with different content keys by using symmetric encryption techniques. Then, the owner defines the access policies over attributes from multiple attribute authorities and encrypts the content keys under the policies. Then, the owner sends the encrypted data to the cloud server together with the ciphertexts.2 they do not rely on the server to do data access control. But, the access control happens inside the cryptography. That is only when the user's attributes satisfy the access policy defined in the cipher text; the user is able to decrypt the cipher text. Thus, users with different attributes can decrypt different number of content keys and thus obtain different granularities of information from the same data.

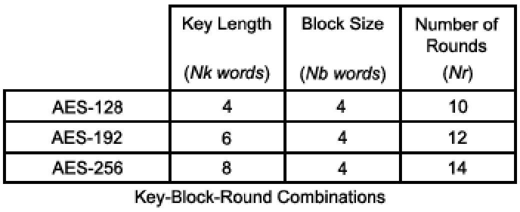
Frame work

The framework of our data access control scheme is defined as follows.

Definition 1 (Framework of Multi-Authority Access Control Scheme). The framework of data access control scheme for multi-authority cloud storage systems contains the following phases:

System Initialization- we consider the server to be semi-trusted, i.e., honest but curious as those in [28] and [15]. That means the server will try to find out as much secret information in the stored BR files as possible, but they will honestly follow the protocol in general. On the other hand, some users will also try to access the files beyond their privileges. For example, a pharmacy may want to obtain the prescriptions of patients for marketing and boosting its profits. To do so, they may collude with other users, or even with the server. In addition, we assume each party in our system is preloaded with a public/private key pair, and entity authentication can be done by traditional challenge-response protocols.

Key Generator - The Key Generator used to generate the key for encryption based on available preferred techniques. AES will produce compact keys with the additional benefit that the cryptosystem is not burdened with patent compliance. However, should a binary fall to reverse Engineering, the key will become compromised (note that AES is a Symmetric Cipher - not an Asymmetric Cipher which has Public and Private Keys). Currently, there are three FIPS (Federal Information Processing Standards) approved symmetric encryption algorithms: AES, Triple DES, and Skipjack. This article will use AES or the Advanced Encryption Standard in CBC Mode. Note that DES (FIPS 46-3) was withdrawn in May 2005, and is no longer approved for Federal use. AES (or Rijndeal - pronounced "Rhine dahl") is the work of Joan Daemen and Vincent Rijmen - hence the portmanteau Rijndael. AES is a 128 bit block cipher that accepts key lengths of 128, 192, and 256 bits. The required number of rounds (i.e., linear and non-linear transformations), depend on the key size. Below are the FIPS 197 conformant Key – Block- Round combinations.



	Key Length (*Nk words*)	Block Size (*Nb words*)	Number of Rounds (*Nr*)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Key-Block-Round Combinations

Taking from FIPS 197:

For both its Cipher and Inverse Cipher, the AES algorithm uses a round function that is composed of four different byte-oriented transformations: 1) byte substitution using a substitution table (S-box), 2) shifting rows of the State array by different offsets, 3) mixing the data within each column of the State array, and 4) adding a Round Key to the State.

Data encryption By owners

The main goal of our framework is to provide secure patient-centric BR access and efficient key management at the same time. The key idea is to divide the system into multiple security domains (namely, public domains (PUDs) and personal domains (PSDs)) according to the different users' data access requirements. The PUDs consist of users who make access based on their professional roles, such as doctors, nurses and medical researchers. In practice, a PUD can be mapped to an independent sector in the society, such as the health care, government or insurance sector. For each PSD, its users are personally associated with a data owner (such as family members or close friends), and they make accesses to BRs based on access rights assigned by the owner.

Each data owner (e.g., patient) is a trusted authority of her own PSD, who uses a KP-ABE system to manage the secret keys and access rights of users in her PSD. Since the users are personally known by the BR owner, to realize patient-centric access, the owner is at the best position to grant user access privileges on a case-by-case basis. For PSD, data attributes are defined which refer to the intrinsic properties of the BR data, such as the category of a BR file. For the purpose of PSD access, each BR file is labeled with its data attributes, while the key size is only linear with the number of file categories a user can access. Since the number of users in a PSD is often small, it reduces the burden for the owner. When encrypting the data for PSD, all that the owner needs to know is the intrinsic data properties.

Data encryption By Users

In our framework, there are multiple SDs, multiple owners, multiple AAs, and multiple users. In addition, two ABE systems are involved. We term the users having read and write access as data readers and contributors, respectively. The owners upload ABE-encrypted BR files to the server. Each owner's BR file is encrypted both under a certain fine grained and role-based access policy for users from the PUD to access, and under a selected set of data attributes that allows access from users in the PSD. Only authorized users can decrypt the BR files, excluding the server.

Attribute Revocation

Security Model

In multi-authority cloud storage systems, we make the following assumptions:

The CA is fully trusted in the system. It will not collude with any user, but it should be prevented from decrypting any cipher texts by itself. . Each AA is trusted but can be corrupted by the adversary. The server is curious but honest. It is curious about the content of the encrypted data or the received message, but will execute correctly the task assigned by each attribute authority. Each user is dishonest and may collude to obtain unauthorized access to data.

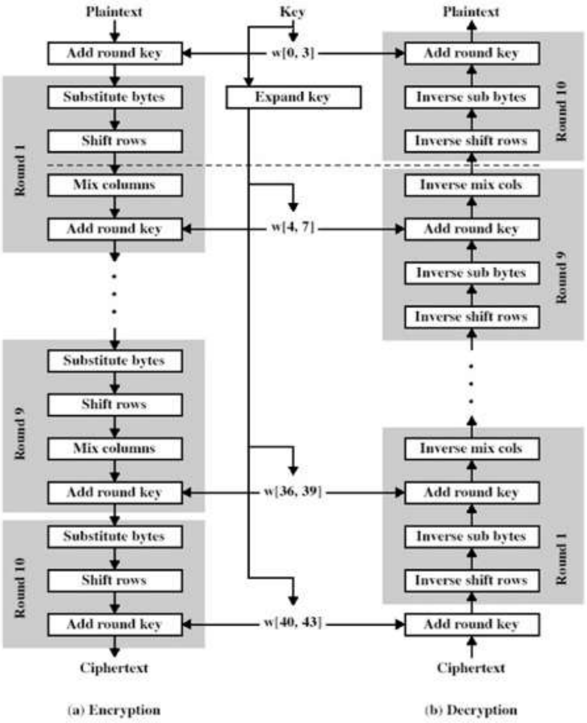
All of the cryptographic algorithms we have looked at so far have some problem. The earlier ciphers can be broken with ease on modern computation systems. The DES algorithm was broken in 1998 using a system that cost about $250,000. It was also far too slow in software as it was developed for mid-1970’s hardware and does not produce efficient software code. Triple DES on the other hand, has three times as many rounds as DES and is correspondingly slower. As well as this, the 64 bit block size of triple DES and DES is not very efficient and is questionable when it comes to security. What was required was a brand new encryption algorithm that would be resistant to all known attacks. The National Institute of Standards and Technology (NIST) wanted to help in the creation of a new standard. However, because of the controversy that went with the DES algorithm, and the years of some branches of the U.S. government trying everything they could to hinder deployment of secure cryptography this was likely to raise strong skepticism. The problem was that NIST did actually want to help create a new excellent encryption standard but they couldn’t get involved directly. Unfortunately they were really the only ones with the technical reputation and resources to the lead the effort.

Instead of designing or helping to design a cipher, what they did instead was to set up a contest in which anyone in the world could take part. The contest was announced on the 2nd of January 1997 and the idea was to develop a new encryption algorithm that would be used for protecting sensitive, non-classified, U.S. government information. The ciphers had to meet a lot of requirements and the whole design had to be fully documented (unlike the DES cipher). Once the candidate algorithms had been submitted, several years of scrutinisation in the form of cryptographic conferences took place. In the first round of the competition 15 algorithms were accepted and this was narrowed to 5 in the second round. The fifteen algorithms are shown in table 7 of which the 5 that were selected are shown in bold. The algorithms were tested for efficiency and security both by some of the world’s best publicly renowned cryptographers and NIST itself.

Inner Workings of a Round

The algorithm begins with an Add round key stage followed by 9 rounds of four stages and a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of its counterpart in the encryption algorithm. The four stages are as follows:

1. Substitute bytes
2. Shift rows
3. Mix Columns
4. Add Round Key



The tenth round simply leaves out the Mix Columns stage. The first nine rounds of the decryption algorithm consist of the following:

1. Inverse Shift rows
2. Inverse Substitute bytes
3. Inverse Add Round Key
4. Inverse Mix Columns

Again, the tenth round simply leaves out the Inverse Mix Columns stage. Each of these stages will now be considered in more detail.

Mix Column Transformation

This stage (known as MixColumn) is basically a substitution but it makes use of arithmetic of GF(2^8). Each column is operated on individually. Each byte of a column is mapped into a new value that is a function of all four bytes in the column. The transformation can be determined by the following matrix multiplication on state (see figure 7.6):

$$
\begin{aligned}
s^{0}\ {}_{0,j} &= (2 \bullet s_{0,j}) \oplus (3 \bullet s_{1,j}) \oplus s_{2,j} \oplus s_{3,j} \\
s^{0}\ {}_{1,j} &= s_{0,j} \oplus (2 \bullet s_{1,j}) \oplus (3 \bullet s_{2,j}) \oplus s_{3,j} \\
s^{0}\ {}_{2,j} &= s_{0,j} \oplus s_{1,j} \oplus (2 \bullet s_{2,j}) \oplus (3 \bullet s_{3,j}) \\
s^{0}\ {}_{3,j} &= (3 \bullet s_{0,j}) \oplus s_{1,j} \oplus s_{2,j} \oplus (2 \bullet s_{3,j})
\end{aligned}
$$

Equation 1

where • denotes multiplication over the finite field GF(2^8).

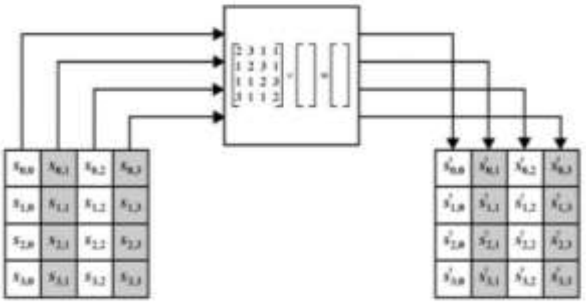


Figure 7.6: MixColumns stage 1

As and example, lets take the first column of a matrix to be s0,0 = {87}, s1,0 ={6E}, s2,0 = {46}, s3,0 = {A6}. This would mean that s0,0 = {87} gets mapped to the value s0 = {47} which can be seen by working out the first line of equation in 1 with j = 0. Therefore we have:

$(02 \bullet 87) \oplus (03 \bullet 6E) \oplus 46 \oplus A6 = 47$

So to show this is the case we can represent each Hex number by a polynomial:

{02} = x

{87} = x7 + x2 + x + 1

Multiply these two together and we get:

x • (x7 + x2 + x + 1) = x8 + x3 + x2 + x

The degree of this result is greater than 7 so we have to reduce it modulo an irreducible polynomial m(x). The designers of AES chose m(x) = x8 + x4 + x3 + x + 1. So it can be seen that

$(x^8+x^3+ x^2 + x) \bmod (x^8 + x^4 + x^3 + x + 1)= x^4+ x^2+ 1$

This is equal to 0001 0101 in binary. This method can be used to work out the other terms. The result is therefore:

$$\begin{array}{ll} & 0001\ 0101 \\ & 1011\ 0010 \\ & \\ & 0100\ 0110 \\ \oplus & \underline{1010\ 0110} \\ & 0100\ 0111 \quad = \{47\} \end{array}$$

This first matrix of equation 1 can be shown to be the inverse of the first matrix in equation 7.3. If we label these A and A−1 respectively and we label state before the mix columns operation as S and after as S0, we can see that:
AS = S0 therefore
A−1S0
= A−1AS = S

Conclusion

In this system, we proposed a revocable multi-authority CPABE scheme that can support efficient attribute revocation. Then, we constructed an effective data access control scheme for multi-authority cloud storage systems. We also proved that our scheme was provable secure in the random oracle model. The revocable multi-authority CPABE is a promising technique, which can be applied in any remote storage systems and online social networks etc.

References:

[1] B. Waters, ‘‘Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization,’’ in Proc. 4th Int’l Conf. Practice and Theory in Public Key Cryptography (PKC’11), 2011, pp. 53-70.

[2] V. Goyal, A. Jain,O. Pandey, andA. Sahai, ‘‘Bounded Ciphertext Policy Attribute Based Encryption,’’ in Proc. 35th Int’l Colloquium on Automata, Languages, and Programming (ICALP’08), 2008, pp. 579-591.

[3] A.B. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B.Waters, ‘‘Fully Secure Functional Encryption: Attribute-Based Encryption and (Hierarchical) Inner Product Encryption,’’ in Proc.Advances in Cryptology-EUROCRYPT’10, 2010, pp. 62-91.

[4] M. Chase, ‘‘Multi-Authority Attribute Based Encryption,’’ in Proc. 4th Theory of Cryptography Conf. Theory of Cryptography (TCC’07), 2007, pp. 515-534.

[5] M. Chase and S.S.M. Chow, ‘‘Improving Privacy and Security in Multi-Authority Attribute-Based Encryption,’’ in Proc. 16th ACM Conf. Computer and Comm. Security (CCS’09), 2009, pp. 121-130.

[6] A.B. Lewko and B. Waters, ‘‘Decentralizing Attribute-Based Encryption,’’ in Proc. Advances in Cryptology-EUROCRYPT’11, 2011, pp. 568-588.

[7] S. Yu, C. Wang, K. Ren, and W. Lou, ‘‘Attribute Based Data Sharing with Attribute Revocation,’’ in Proc. 5th ACM Symp. Information, Computer and Comm. Security (ASIACCS’10), 2010, pp. 261-270.

[8] S. Jahid, P. Mittal, and N. Borisov, ‘‘Easier: Encryption-Based Access Control in Social Networks with Efficient Revocation,’’ in Proc. 6th ACM Symp. Information, Computer and Comm. Security (ASIACCS’11), 2011, pp. 411-415.

[9] D. Boneh and M.K. Franklin, ‘‘Identity-Based Encryption from the Weil Pairing,’’ in Proc. 21st Ann. Int’l Cryptology Conf.: Advances in Cryptology - CRYPTO’01, 2001, pp. 213-229.