IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661, p- ISSN: 2278-8727Volume 16, Issue 2, Ver. III (Mar-Apr. 2014), PP 25-32
www.iosrjournals.org

Enhancing Security and Fine-Grained Access Control for Personal Health Record in Cloud Computing

Rinkumol Kuriakose, R .Tamilarasu,
PG student & Assistant Professor (S.G.)
Department of CSE, SVS College of Engineering, Coimbatore, TamilNadu

Abstract: *Now a days , personal health record become patient-centric model in which PHR service allows a patient to manage and control personal health data in one place through the web, which has made the storage, retrieval and sharing of the medical information more efficient . For the wide range access it has out sourced in third party storage such as cloud providers. These PHR contains sensitive data that should be protected from unauthorized parties. But while using third party service providers there are many security and privacy risks for PHR. The one way to protect PHR data is to encrypt data before out sourcing. Since the PHR has been accessed by the multiple authorities; we introduce a fine grained attribute based access control in which each party is assigned with access permission for a set of attributes. Division of personal health records users into multiple security domains which reduce key management complexity for owners and users. A high degree of patient privacy is guaranteed simultaneously by exploiting multi-authority ABE.*
Keywords: *Personal Health Records, Cloud Computing Hybrid cryptography, Symmetric Key Encryption, Asymmetric Key Encryption, Attribute Based Encryption.*

I. Introduction

Personal Health Record (PHR) became an emerging patient centric model for the exchange of personal health information through third party such as cloud providers. Since it is being out sourced, the patients have no more physical control over the health information. There are wide ranges of security and privacy concerns while storing the data to the cloud. To ensure the security and privacy of the health record, the data should encrypt before outsource it. Yet, there are other issues toward achieving fine-grained, cryptographically enforced data access control are efficient key management, dynamic user revocation and flexible access.

This paper presents a different approach from the previous works for encrypting the information. The patient can decide with whom the information should be shared. For the fine-grained access of information, here introduces an architecture in which upon request from a person, a virtual proxy server will be created for the purpose of accessing the information. Only the person who has the decryption key can enter in to the virtual proxy server. This approach has made the user revocation process much easier since the key can be used only one time. Since the health records are being stored in a semi trusted third parties here it uses Attribute Based Encryption (ABE) for the privacy of the data.

Scalability: Any number of users can added to the system using the proposed approach.

Security: Here for the encryption hybrid cryptography is used. Symmetric key cryptography is used for data encryption and Asymmetric key cryptography is used for key encryption.

Efficiency: Since it uses both symmetric key and asymmetric key cryptography it is fast and very secure.

II. Related Work

Role-Based Access Control(R-BAC):

Role-based access control provides secure accessing of information while preventing unauthorized access. A person can access the data according to the role assigned and restrict the same person from accessing the data other than he has the right for.

Key Policy Attribute Based Encryption (KP-ABE):

In this, the primitive enables the senders to encrypt messages under a set of attributes and private keys are associated with access structures that specify the cipher text the key holder will be allowed to decrypt.

Cipher text Policy Attribute Based Encryption (CP-ABE):

Using this technique encrypted data can be kept confidential even if the server is untrusted. Prevents collusion attack.

Multi-Authority Attribute Based Encryption(MA-ABE):

MA-ABE have any number of independent authorities that monitor attributes and distribute secret keys. For each authority, an encryptor has a number dk and a set of attributes. The encryptor can then encrypt a message

such that a user can only decrypt if he has at least dk of the given attributes from each authority k.

III. Problem Definition

In the existing system either it uses symmetric key encryption or it uses Asymmetric key encryption. Symmetric key encryption is simple and easy to carry out. It is faster than any other cryptographic techniques. Since it uses same key for both encryption and decryption, the keys should be exchanged between sender and receiver .Exchanging the key is a disadvantage, since it can be retrieved by a third party. It will leads to key escrow problem. Assign a central authority to keep the key will lead to trust that party completely. Collusion attack is a disadvantage of the existing system. Origin and authenticity of message cannot be guaranteed. Asymmetric key encryption is more secure than symmetric key encryption since it uses different key for encryption and decryption. But compared to symmetric key encryption asymmetric key encryption is relatively slow and is not feasible for decrypting messages. It uses more computational resources.

IV. Proposed System

Personal Health Record is patients centric web application in which a patient can access manage and control their health information in a fine-grained manner. For the wide range of access these health records are outsourced to cloud servers. In recent years many research have been done on the security and privacy of these cloud servers. Most of the researches have been concluded with the fact that the data should be kept securely by encrypting the data before out sourcing it.
In this proposed work, it uses a cryptographic technique called Hybrid Cryptosystem that is different from the previous work. An attribute based encryption is used for the fine-grained access control.

A. ATTRIBUTE BASED ENCRYPTION

Attribute-based encryption (ABE) is a type of public key encryption. Using ABE users can encrypt and decrypt messages based on desired attributes. 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.
In this proposed method, attribute based encryption technique provides security to the database. Personal health information is stored in the third party cloud server. Using this attribute based encryption; the patient can decide whom should have the access rights to the data. In Attribute based encryption cipher text labeled with set of attribute. Key associated with access structure that control which cipher text a user is able to decrypt. Using ABE, access policies are expressed based on the attributes of users or data, which enables a patient to selectively share her PHR among a set of users by encrypting the file under a set of attributes, without the need to know a complete list of the users. The complexity based on the key management can be reduced by dividing the domain.

Advantages of ABE

1. To implement fine-grained access control over encrypted data.
2. Preventing unauthorized access.
3. Preventing collusion attack.

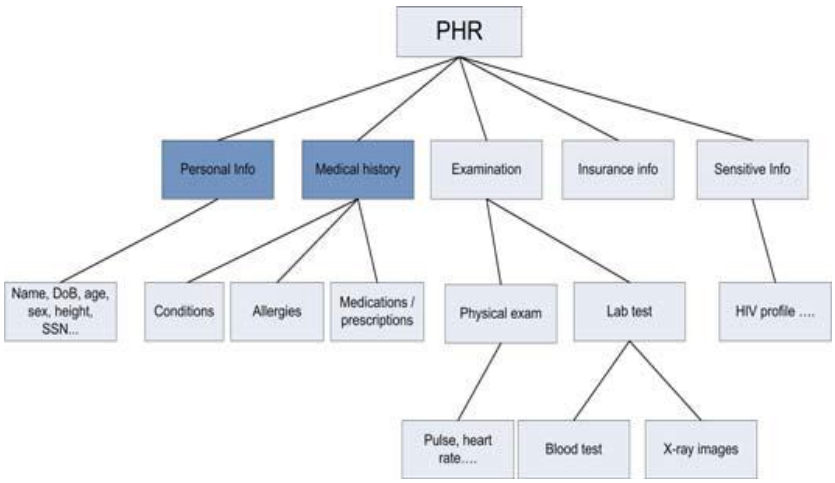

Figure 1. Attributes

B. HYBRID CRYPTOGRAPHY

In cryptography, public-key cryptosystems are convenient in that they do not require the sender and receiver to share a common secret in order to communicate securely (among other useful properties). However, they often rely on complicated mathematical computations and are thus generally much more inefficient than comparable symmetric-key cryptosystems.

In many applications, the high cost of encrypting long messages in a public-key cryptosystem can be prohibitive. A hybrid cryptosystem is one which combines the convenience of a public-key cryptosystem with the efficiency of a symmetric-key cryptosystem.

A hybrid cryptosystem can be constructed using any two separate cryptosystems:

1. A key encapsulation scheme, which is a public-key cryptosystem, and
2. A data encapsulation scheme, which is a symmetric-key cryptosystem.

The hybrid cryptosystem is itself a public-key system, whose public and private keys are the same as in the key encapsulation scheme.

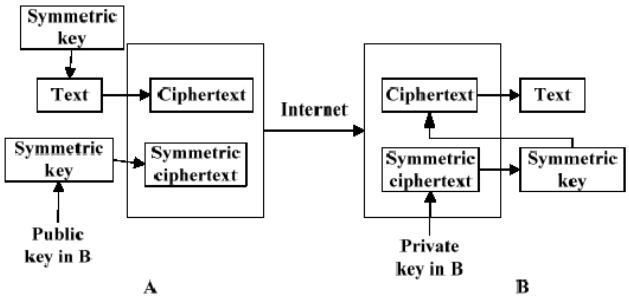

Figure 2. Hybrid Cryptosystem

A simple and efficient way to build an encryption scheme that has an unrestricted message is to build a hybrid encryption scheme. Loosely speaking, such a scheme uses public-key encryption techniques to encrypt a key K that is then used to encrypt the actual message using symmetric key encryption techniques. One key ingredient in any hybrid scheme is a key encapsulation mechanism. This is like a public-key encryption scheme, except that the job of the encryption algorithm is to generate the encryption of a random key K. Of course, one can always use a general-purpose public-key encryption scheme to do this, by simply generating K at random, and then encrypting it.

RELATED TERMS

Plain Text: A message in its natural format readable by an attacker.
Cipher Text: Message altered to be unreadable by anyone except the intended recipients.
Key: Sequence that controls the operation and behavior of the cryptographic algorithm.
Key space: Total number of possible values of keys in a crypto algorithm.
Cryptosystem: The combination of algorithm, key and key management functions used to perform cryptographic operations.

C. SYMMETRIC KEY ENCRYPTION

A single-key encryption system (also known as "symmetric key encryption", since the same key is used for both encryption and decryption) works by means of an algorithm that transforms the input data based upon the value of an encryption key, using some combination of mathematical and logical operations. To decrypt, the algorithm runs the same set of operations in reverse, using the same key value. The Vignere cipher is a very simple example of this, where the algorithm is simply to add the input value to the corresponding key value. Decrypting a Vignere cipher is simply a matter of subtracting.

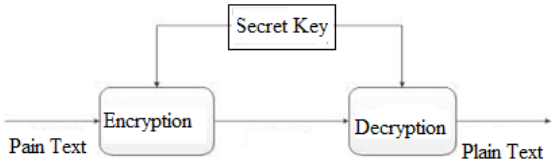

Figure 3. Symmetric Key Cryptography

ADVANCED ENCRYPTION STANDARD (AES)

Advanced Encryption Standard (AES) is based on a design principle known as a substitution-permutation network.AES is a block cipher with a block length of 128 bits.AES allows key size of 128, 192, and 156.Key size of 128 itself proven to unbreakable. Unlike DES and triple DES, it is fast in both software and hardware. AES does not use a Feistel network .The128 bit block is arranged like a 4×4 matrix. The 4×4 matrix bytes are referred to as state.Each round of processing works on the input state array and produces an output state array. The output state array produced by the last round is rearranged into a 128-bit output block. According to the key size AES decide on how many round it needed for the encryption.

Key Size (bytes)	Block Size (bytes)	Rounds
16	16	10
24	16	12
32	16	14

Table. 1 AES Rounds

State: Defines the current condition (state) of the *block*. That is the block of bytes that are currently being worked on. The state starts off being equal to the block, however it changes as each round of the algorithms executes. Plainly said this is the block in progress.
XOR: Refers to the bitwise operator Exclusive Or. XOR operates on the individual bits in a byte in the following way:
0 XOR 0 = 0
1 XOR 0 = 1
1 XOR 1 = 0
0 XOR 1 = 1

Steps included in encryption are:
1) Substitute bytes
2) Shift rows
3) Mix columns
4) Add round key
The last step consists of XORing the output of the previous three steps.

Steps included in decryption are:
1) Inverse shift rows
2) Inverse substitute bytes
3) Add round key
4) Inverse mix columns.
The third step consists of XORing the output of the previous two steps.

Step1: Substitute bytes

- This is a byte-by-byte substitution. The substitution byte for each input byte is found by using lookup table.
- The entries in the lookup table are constructed by a combination of GF(2^8) arithmetic and bit mangling.
- The substitution step will reduce the correlation between input bits and output bits (at the byte level). The bit mangling part of the substitution step ensures that the substitution cannot be described in the form of evaluating a simple mathematical function.
- The corresponding substitution step used during decryption is called Inverse Substitute Bytes.

Step2: Shift rows

- In this step, a circular shift is performed. The circular shift that moves each byte a space. A byte that was in the second position may end up in the third position after the shift. The circular part of it specifies that the byte in the last position shifted one space will end up in the first position in the same row.

- The state is arranged in a 4x4 matrix.
- Each row is then moved over (shifted) 1, 2 or 3 spaces over to the right, depending on the row of the state.
- First row is never shifted.
- The second row is shifted one byte to the left in a circularly.
- The third row is shifted two bytes to the left in a circularly.
- The fourth row is shifted three bytes to the left in a circularly.

Step3: Mix columns

- This step replaces each byte of a column by a function of all the bytes in the same column.
- Each byte in a column is replaced by two times that byte, plus three times the next byte, plus the byte that comes next, plus the byte that follows.
- The bytes refer here are the bytes in the same column, and it is taken circularly, i.e. the byte that is next to the one in the last row is the one in the first row.
- By 'two times' and 'three times', it means multiplications in GF(2^8) by the bit patterns 000000010 and 00000011, respectively.

Step4: Add round key

- Each of the 16 bytes of the state is XORed against each of the 16 bytes of the expanded key for the current round.
- The Expanded Key bytes are never reused. So once the first 16 bytes are XORed against the first 16 bytes of the expanded key then the expanded key bytes 1-16 are never used again. The next time the Add Round Key function is called bytes 17-32 are XORed against the state.
- The operation is viewed as column wise operation between is 4 bytes of state column and one word of the round key.
- During decryption this procedure is reversed. Therefore the state is first XORed against the last 16 bytes of the expanded key, then the second last 16 bytes and so on.

Key Expansion

Each time the Add Round Key function is called and a different part of the expanded key is XORed against the state. For this to work the Expanded Key must be large enough so that it can provide key material for every time the Add Round Key function is executed.

Advantages of AES

1) AES is very secure.
2) It provides the key length of 128,192, and 256 bits.
3) Key length of 128 itself unbreakable.
4) AES is suitable for both software and hardware.

Functions involved in each round in encryption:

Round	Function
-	AddRoundKey (State)
0	AddRoundKey(MixColumn(ShiftRow(ByteSyb(State))))
1	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
2	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
3	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
4	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
5	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
6	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
7	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
8	Add Round Key (Mix Column(Shift Row(ByteSyb(State))))
9	AddRoundKey(ShiftRow(ByteSyb(State))))

Table 2. Encryption

Functions involved in each round in decryption:

Round	Function
-	Add Round Key (State)
0	MixColumn (AddRoundKey(ByteSub(ShiftRow(State))))
1	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
2	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
3	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
4	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
5	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
6	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
7	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
8	Mix Column (Add Round Key(ByteSub(Shift Row(State))))
9	Add Round Key (Shift Row(ByteSyb(State))))

Table 3. Decryption

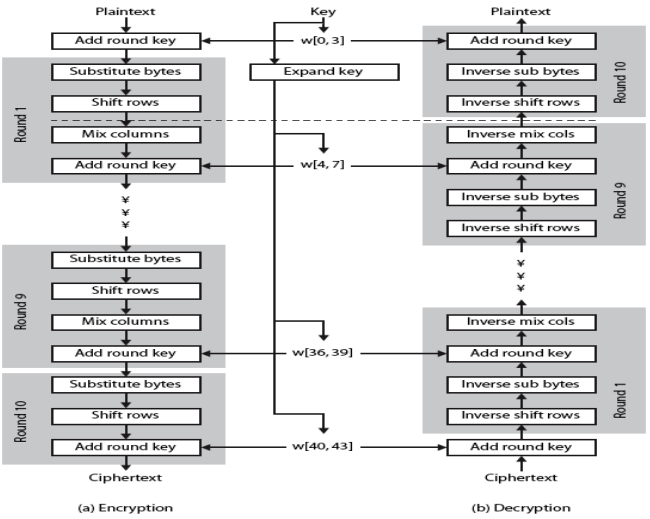

Figure 4. AES Structure

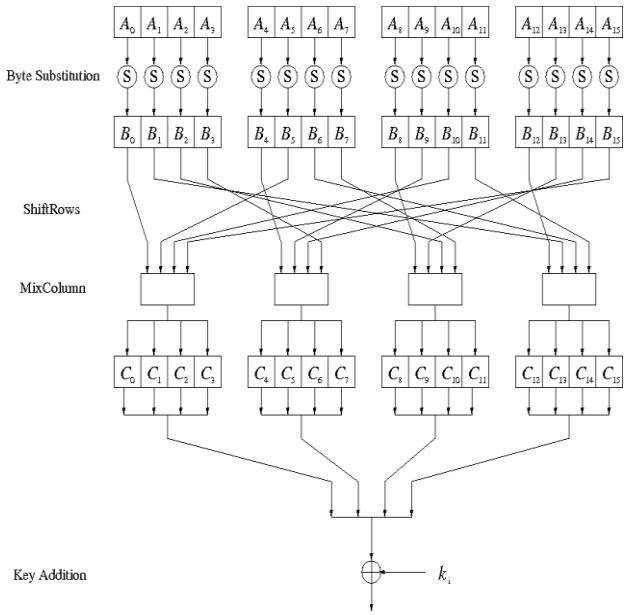

Figure 5.Internal structure of AES

D. ASYMMETRIC KEY ENCRYPTION

In designing security systems, it is wise to assume that the details of the cryptographic algorithm are already available to the attacker. The history of cryptography provides evidence that it can be difficult to keep the details of a widely used algorithm secret. A key is often easier to protect (it's typically a small piece of information) than an encryption algorithm, and easier to change if compromised. Thus, the security of an encryption system in most cases relies on some key being kept secret. Trying to keep keys secret is one of the most difficult problems in practical cryptography; see key management. An attacker who obtains the key can recover the original message from the encrypted data.

Encryption algorithms which use the same key for both encryption and decryption are known as symmetric key algorithms.

The asymmetric key algorithms allow one key to be made public while retaining the private key in only one location. They are designed so that finding out the private key is extremely difficult, even if the corresponding public key is known. A user of public key technology can publish their public key, while keeping their private key secret, allowing anyone to send them an encrypted message.

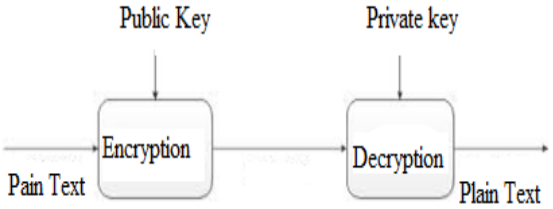

Figure 6. Asymmetric key Cryptography

RSA

RSA is a public key cryptography used for the key encapsulation process in this paper. In RSA, encryption keys are public, while the decryption keys are not, so only the person with the correct decryption key can decipher an encrypted message. Everyone has their own encryption and decryption keys. The keys must be made in such a way that the decryption key may not be easily deduced from the public encryption key.

The receiver may need to verify that a transmitted message actually originated from the sender (signature), and didn't just come from there (authentication). This is done using the sender's decryption key, and the signature can later be verified by anyone, using the corresponding public encryption key. Signatures therefore cannot be forged. Also, no signer can later deny having signed the message.

Key Generation

1. Choose two distinct prime numbers p and q randomly.
2. Compute $n=pq$ .n will be used as the modulus for both the public and private keys.
3. Compute $\varphi(n) = \varphi(p)\varphi(q) = (p-1)(q-1)$, where φ is Euler's totient function.
4. Choose an e such that $1 < e < \phi(n)$, and such that e and $\phi(n)$ share no divisors other than 1 (e and $\phi(n)$ are relatively prime).e is kept as the public key exponent.
5. Determine d as $d \equiv e^{-1} \pmod{\varphi(n)}$;
   d isthe multiplicative inverse of e *(modulo* $\varphi(n)$*).*

Encryption

$$c \equiv m^e \pmod{n}.$$

c –Cipher text
m – Plaintext
n - The modulus for both the public and private keys.
e - Public key

Decryption

$$m \equiv c^d \pmod{n}.$$

c – Cipher text
m – Plaintext
n - The modulus for both the public and private keys.
e - Private key

RSA Algorithm Example

- Choose $p = 3$ *and* $q = 11$
- Compute $n = p \times q = 3 \times 11 = 33$
- Compute $\varphi(n) = (p - 1) \times (q - 1) = 2 \times 10 = 20$
- Choose e such that $1 < e < \varphi(n)$ and e and n are co prime. Let $e = 7$
- Compute a value for d such that $(d \times e) \% \varphi(n) = 1$. One solution is $d = 3$ $[(3 \times 7) \% 20 = 1]$
- Public key is (e, n) => (7, 33)
- Private key is (d, n) => (3, 33)
- The encryption of $m = 2$ is $c = 2^7 \% 33 = 29$
- The decryption of $c = 29$ is $m = 29^3 \% 33 = 2$

Advantages of RSA

- RSA with AES is an efficient cryptographic technique recently.
- To provide privacy and authenticity.
- The main advantage of RSA is that it is a public key cryptography in which it does not have to share the key.ie it uses two different keys (public and private key) for the encryption and decryption.

V. Advantages Of Proposed System

1. Health information can be accessed easily and quickly.
2. Treatment can be quickly started using the information recorded
3. Patient can shown the details to other doctor for medical advice in a controlled manner.
4. Reimbursement procedure can be done easily.
5. A user friendly environment is provided.
6. Data confidentiality.

VI. Conclusion And Future Enhancement

This paper provides privacy and security to the medical data which is stored in the third party cloud storage. It prevents attackers and hackers by using new cryptographic techniques like hybrid encryption and attribute based encryption.

However, using hybrid encryption is more secure than any other techniques that are used till now; data needs more security. If the key that has to be encrypted is as same length as the message then, using public key cryptography is of no use. So a new technique should be introduced for the same. Now days there are many cryptographic present for the efficient encryption and decryption. These can be adopted for the better security for the data's that are stored in the cloud storage.

REFERENCES

[1] U.Jyothi K., Nagi Reddy, B. Ravi Prasad," Review of "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing"" International Journal Of Engineering And Computer Science ISSN:2319-7242Volume 2 Issue 8 August, 2013 Page No. 2440-2447

[2] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized Private Keyword Search over Encrypted Personal Health Records in Cloud Computing," Proc. 31st Int'l Conf. Distributed Computing Systems (ICDCS '11), June 2011.

[3] M. Li, S. Yu, K. Ren, and W. Lou, "Securing Personal Health Records in Cloud Computing: Patient-Centric and Fine-Grained Data Access Control in Multi-Owner Settings," Proc. Sixth Int'l ICST Conf. Security and Privacy in Comm. Networks (SecureComm '10), pp. 89-106, Sept. 2010.

[4] H. Lo¨ hr, A.-R. Sadeghi, and M. Winandy, "Securing the E-Health Cloud," Proc. First ACM Int'l Health Informatics Symp. (IHI '10), pp. 220-229, 2010.\

[5] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-Grained Data Access Control in Cloud Computing," Proc. IEEE INFOCOM '10, 2010.

[6] C. Dong, G. Russello, and N. Dulay, "Shared and Searchable Encrypted Data for Untrusted Servers," J. Computer Security, vol. 19, pp. 367-397, 2010.

[7] J. Benaloh, M. Chase, E. Horvitz, and K. Lauter, "Patient Controlled Encryption: Ensuring Privacy of Electronic Medical Records," Proc. ACM Workshop Cloud Computing Security (CCSW '09), pp. 103-114, 2009.

[8] G. Ateniese, K. Benson, and S. Hohenberger, "Key-Private Proxy Re-Encryption, "Proc. Topics in Cryptology (CT-RSA),pp. 279-294, 2009.

[9] J. Shao and Z. Cao, "CCA-Secure Proxy Re-Encryption without Pairings,"Proc. 12th Int'l Conf. Practice and Theory in Public Key Cryptography (PKC),pp. 357-376, 2009.

[10] Q. Tang, "Type-Based Proxy Re-Encryption and Its Construction," Proc. Ninth Int'l Conf. Cryptology in India: Progress in Cryptology (INDOCRYPT),pp. 130-144, 2008.