ACHIEVING ROBUST BIOMETRIC BASED ACCESS CONTROL MECHANISM FOR CLOUD COMPUTING

Kun Huang, Jiangyong Shi*, Ming Xian*, and Jian Liu**

**State Key Laboratory of Complex Electromagnetic Environment Effects on Electronics and Information System, National University of Defense Technology, Changsha, 410073, China*

khuang_123@163.com, sjy.paper@gmail.com, qwertmingx@tom.com, ljabc730@163.com

Keywords: Cloud Computing, Access Control, Fuzzy Identity Based Encryption, Key Insulated Encryption, Biometric

Abstract

Cloud computing is an arresting emerging computing paradigm that offers users on demand network access to a large shared pool of computing resources. This paper focuses on leveraging biometric identity to achieve access control in cloud. Biometric possesses a lot of advantages like portability, uniqueness and verification clearness; nevertheless biometric measurements are always noisy. Additionally, there always exists key exposure problem in the context of access control. To protect sensitive data along with private key confidential against malicious servers or other external attackers and meet the requirement of removing the biometric noisy property, we exploit and combine techniques of fuzzy identity based encryption (FIBE), biometric measurement, and key insulated encryption. Specifically, we based on the idea that every time legal user or malicious one makes the request of accessing data of his interest will the cloud severs update the corresponding header file which only the legal user has the ability to decrypt. To our best knowledge, it is the first to consider the key insulated encryption in the access control of cloud computing. Extensive analysis shows that our proposed scheme is provably secure under existing security models.

1 Introduction

As promising as it is, cloud computing is facing many challenges which may impede its fast growth if not well resolved. According to recent research conducted by CSA (Cloud Security Alliance), the top nine cloud computing threats for 2013 have been identified. The report reflects the current consensus among industry by CSA, focusing on threats specially related to the shared and on-demand nature of cloud computing. Data breaches and data loss are the very first two threat factors among "The Notorious Nine".

Data security that is protecting against data breaches and data loss, as it exists in many applications, is among these challenges that would raise the greatest concerns from users when they store sensitive information on cloud servers. These concerns stem from the fact that cloud servers are usually out of the trusted domain of the users. Data confidential against cloud servers is hence frequently desired when users outsource data for storage in the cloud.

As a significant research field in security protection, data access control has been evolving in the past thirty years and various techniques [1–9] have been developed to effectively implement fine-grained access control, which allows flexibility in specifying differential access rights of individual users. However, there always exists latent danger on the distribution and management of private key. The existing model of access control necessarily includes the very step of distributing the private key corresponded to one's identity. Exposure of private (secret) keys can be the most devastating attack on such an access control system since all security guarantees are lost in the situation that some malicious entity may pretend to be a legal user. In real world, this problem is probably the greatest threat to cryptography: it is easier to obtain a secret key from a stolen device than to break the computational assumption on which the system is based. The threat is increasing with users

carrying mobile devices which hold the corresponding secret key for producing signature. In identity based schemes, key exposure may means that the corresponding owner of the exposed key will leak the encrypted file to others with no idea of the existence of such an entity.

To resolve the key exposure problem, in 2002, Dodis et al. [10] proposed the method of key insulated cryptography. In key insulated scheme, physical security (and hence secrecy of stored data) is guaranteed with a single base device that holds a master key SK_0 corresponded to a fixed key PK. However, all decryption, is performed by an insecure device with key exposure risks. The lifetime of the protocol is divided into distinct periods of $1,\ldots,N$. At the beginning of each period, the insure device refreshes its temporary secret key by interacting with the secure base device. In fact, the insecure device can get any period at any time. The exposure of up to t out of the N periods, chosen adaptively by the adversary, still keeps secure any period that was not exposed. The optimal t achieved by some of the schemes can be $N-1$ while the remaining period is still secure. When applying the method of key insulation into identity based signature, the key exposure problem of identity based scheme can be solved efficiently.

In this paper, we address the key exposure problem in the cloud's access control. Similarly, we let the discrete period represents the frequency of access application instead of the discrete time domain. Through our elaborate design, our proposed scheme resembles the so called one-time pad, which is deemed definitely secure in key management.

The rest of paper is organized as follows: Section 2 introduces the system model, security model, and our design goal. Then we present some technique preliminaries and provide the detailed description of our scheme in Sections 3 and 4 respectively. Section 5 gives the security analysis and performance evaluations. Finally, Section 6 concludes the whole paper.

2 Models and Assumption

2.1 System Model

We assume that the system is composed of the following three parties: Data Owner, Data Visitor (user), and Cloud Servers (CS). To access data files stored in cloud and shared by the data owner, users download the ever updated data files of their interest from Cloud Servers and then decrypt it via the also updated private key. Once data owner distributes the private key to user, he may be offline for a long time. For simplicity, we assume that the only access privilege for users is data file reading. Cloud Servers are always online which not only play the role of storing data file but also update its header file which is small compared to the whole data file. They are assumed to have abundant storage capacity and computation power.

2.2 Security Model

In our model, cloud severs are assumed to be honest but curious. In other words, CSP follow our proposed scheme generally, but try their best to find some secret information. Additionally, certain external attackers like hacker is considered here to steal either CSP's file information or user's private keys. And we assume the attacker has high-tech ability to intercept both users' biometric identity sample and updated temporary key. Moreover, we make an assumption that CS don't collude with the sophisticated hacker. Otherwise, user can only resort to laws.

Definition 1: Our proposed scheme is strongly protected against external attacker's intercepting if any malicious attacker who is able to intercept both user's biometric identity sample and user's updated temporary key can't learn any information of data owner's data file except for a negligible probability.

2.3 Design Goals

Our main design goal is to help the data owner achieve biometric identity based access control of files stored on Cloud Servers and enable user to securely access data owner's files of his interest. Specifically, the data owner is enabled to enforce a unique access structure on each user based on their biometric identity which precisely indicates the set of files that the user is allowed to access. We also want to prevent data owner's files and user's updated private key from being learned by external attacker like hacker.

3 Technique Preliminaries

3.1 Fuzzy Identity-Based Encryption

FIBE [2] is a public key cryptography primitive for one-to-many communications. In Fuzzy IBE, an identity is viewed as a set of descriptive attributes. A Fuzzy IBE scheme allows for a private key for an identity ω to decrypt a

ciphertext encrypted with an identity ω' if and only if the identities ω and ω' are close enough to each other as measured by the "set overlap" distance metric. A Fuzzy IBE scheme can be applied to enable encryption using biometric inputs as identities; the error-tolerance property of a Fuzzy IBE scheme is precisely what needed for the use of biometric identities, which inherently have some noise when they are sampled. Additionally, Fuzzy-IBE can be used for a type of application that we term "attribute-based encryption".

For the construction of our scheme, some cryptographic techniques are recalled firstly. Let G_1 be a bilinear group of prime order p, and let g be a generator of G_1. Additionally, let $G_1 \times G_1 \to G_2$ denote the bilinear map. A security parameter κ is used to determine the size of the groups. And in order to create an IBE scheme in which a ciphertext created using identity ω' can be decrypted only by a secret key ω, set a threshold value d satisfying $|\omega \cap \omega'| \geq d$. Additionally, define the Lagrange coefficient $\Delta_{i,S}$ for $i \in Z_p$, and a set S of elements in Z_p:

$$\Delta_{i,S}(x) = \prod_{j \in S, j \neq i} \frac{x-j}{i-j} \tag{1}$$

Identities will be element subsets of some universe U of size $|U|$. We will associate each element with a unique integer in Z_p^*. (In practice an attribute will be associated with every element so that identities will have some semantics.) A FIBE scheme is composed of four algorithms which can be defined as follows:

Setup(d): First, define the universe, U of elements. For simplicity, we can take the first $|U|$ elements of Z_p^* to be the universe. Namely, the integers $1, \ldots, |U| \pmod p$. Next, choose $t_1, \ldots, t_{|U|}$ uniformly at random from Z_p. Finally, choose y uniformly at random form Z_p. The public parameter is: $PK = (T_1 = g^{t_1}, \ldots, T_{|U|} = g^{t_{|U|}}, Y = e(g,g)^y)$. The master key is $MK = (t_1, \ldots, t_{|U|}, y)$. While PK is publicly known to all the parties in the system, MK is kept as a secret by the authority party.

Encryption: Encryption with the public key ω' and message $M \in G_2$ proceeds as follows. First, a random value $s \in Z_p$ is chosen. The ciphertext is then published as: $E = (\omega', E' = MY^s, \{E_i = T_i^s\}_{i \in \omega'})$. Note that the identity ω' is included in the ciphertext.

Key Generation: To generate a private key for identity $\omega \in U$, the following steps are taken. A $d-1$ degree polynomial q is randomly chosen such that $q(0) = y$. The private key consists of components $(D_i)_{i \in \omega}$, where $D_i = g^{\frac{q(i)}{t_i}}$ for every $i \in \omega$.

Decryption: Suppose that a ciphertext, E, is encrypted with a key for identity ω' and we have a private key for identity ω' where $|\omega \cap \omega'| \geq d$. Choose an arbitrary d-element subset S, of $\omega \cap \omega'$. Then, the ciphertext can be decrypted as:

$$\begin{aligned} &\frac{E'}{\prod_{i \in S}(e(D_i, E_i))^{\Delta_{i,S(0)}}} \\ &= \frac{Me(g,g)^{sy}}{\prod_{i \in S}(e(g^{\frac{q(i)}{t_i}}, g^{st_i}))^{\Delta_{i,S(0)}}} \\ &= \frac{Me(g,g)^{sy}}{\prod_{i \in S}(e(g,g)^{sq(i)})^{\Delta_{i,S(0)}}} \\ &= M. \end{aligned} \tag{2}$$

The last equality is derived from using polynomial interpolation in the exponents. Since the polynomial $sq(x)$ is of degree $d-1$, it can be interpolated using d points.

3.2 Key Insulated Cryptosystem

Cryptographic computations (decryption, signature generation, etc.) are often performed on a relatively insecure device (e.g., a mobile device or an Internet-connected host) which cannot be trusted to maintain secrecy of the private key. Dodis et al. [10] proposed and investigated the notion of key-insulated security whose goal is to minimize the damage caused by secret-key exposures. In their model, the secret keys stored on the insecure device are refreshed at discrete time periods via interaction with a physically secure but computationally-limited device which stores a "master key". All cryptographic computations are still done on the insecure device, and the public key

remains unchanged.

Similarly, in our paper we let the index j represents the frequency of access application instead of the discrete time domain. In special, we set D_i^{j+1} to represents the jth updated secret key of D_i^1 , where $D_i^1 = D_i = g^{\frac{q(i)}{t_i}}$, $i \in \omega'$ is the initial secret key of user and assigned by data owner. The fraction of file $E_i^1 = g^{st_i}, i \in \omega'$ stored in cloud should also be updated in some special way. Only when user accesses the file of his interest will the cloud severs (CS) trigger the updating operation of E_i^1 . We equip user with a frequency recorder, which records the frequency of user's legal accesses. The key insulated cryptosystem is defined as following:

• At the initial time, data owner distributes CS and user some master key along with the key evolution method respectively. We set the master key by x_0^* and the temporary key by x_j, $j \geq 1$ where $x_0^*, x_j \in Z_p$. In addition, we set a semantically secure hash function H which maps Z_P to Z_P itself. In addition, the key evolution mechanism is $x_j = H((x_0^*)^j \bmod p)$.

• At the access time, CS updates the fraction of header file $E_i^1 = g^{st_i}$ as:

$$E_i^{j+1} = (E_i^j)^{\frac{1}{x_j}} = g^{\frac{st_i}{\prod_{k=1}^{k=j} x_k}}, i \in \omega' \qquad (3)$$

And then user updates his secret key as:

$$D_i^{j+1} = (D_i^j)^{x_j} = g^{\frac{q(i)\prod_{k=1}^{k=j} x_k}{t_i}}, i \in \omega' \qquad (4)$$

Afterwards, CS and user erase the previous E_i^j and D_i^j respectively. Additionally at any channel external attackers can only intercept the temporary key x_j instead of x_0^* . Our proposed scheme is similar to the $(1, N)$-key-insulated encryption scheme where N can be set to be the total frequency of legal user's qualified access.

4 Our Proposed Scheme

4.1 Main Ideal

In order to achieve key insulated, portable and fine-grained access control on outsourced data in the cloud, we utilize and uniquely combine the following three advanced techniques: FIBE, key insulated cryptosystem and biometric measurement. More specifically, data owner associates data file with an identity, that is biometric identity, and assigns each user a decrypted secret key which is defined by this identity. To enforce and secure this kind of access control, we utilize FIBE to escort data encryption keys of data files and adopt key insulated mechanism to intensify the update management of user's secret key. Such a construction enables us to immediately enjoy high level security of access control.

What's more, the cloud environment decides data owner shouldn't stay on-line all the time, or it would introduce heavy computation overhead and cumbersome online burden towards the data owner. In view of this restrictive consideration, we allow CS to execute this updating operation, that is to say, update a small fraction of data file every time. From the perspective of security, this updating operation isn't unnecessary or cumbersome. In addition, to better blend with the cloud environment and satisfy so called "one time one encryption" requirement, we introduce a novel key insulated cryptosystem based on the number of legal user's access instead of the traditional discrete time domain. This idea stems from the following scene: "Every time the legal user wants to access his interest file, CS first updates some header file and then user also updates the corresponding private key. Assume one external attacker has intercepted the user's temporary private key, the attacker may use the key to access the legal user's file and yet CS again updates the header of data file. Thus, the temporary key is meaningless to the attackers which ensure the security of access control.

In our model, we endow the attacker two significant abilities, namely intercepting the user's temporary key and pretending it to be a legal visitor. However, CS is regulated not to collude with the external attackers or pretend to be external attacker, else there is no way to defend against this kind of union attack. Thus CS is just curious about the data file. Due to the private key inside the user's domain, CS can't learn anything about the data file unless it obtain user's private master key. Furthermore,

in order to enable the legal user to access his interest file as usual, we employ two frequency recorders which record the frequency of the header file updated by CS and the frequency of legal access by users respectively. We require CS to be equipped with this record and send the recorded frequency number to user every access time. Afterwards, the user updates his private key based on the difference between the two recordings. To our best knowledge, this is the first to consider the aforementioned scheme in the access control of cloud. Under our regulated adversary model and security model, our proposed access control mechanism is robust and secure.

4.2 The Detailed Scheme

We will present our proposed scheme in System Level. The implementation of high level operations includes System Setup, New File Creation, and File Access.

System Level Operations: System level operations in our proposed scheme are designed as follows.

System Setup In this operation, the data owner chooses a security parameter κ and calls the algorithm level interface $FIBESetup(\kappa)$, which outputs the system public parameter PK and the system master key MK.

New File Creation Before uploading a file to Cloud Servers, the data owner processes the data file as follows.

• Select a unique identity ID of user for this data file such as user's biometric character attribute;

• Randomly select a symmetric data encryption key $DEK \leftarrow K$, where K is the key space, and encrypt the data file using DEK;

• Select one biometric attribute set ω' sampled from user for the data file and encrypt DEK with ω' using FIBE, i.e. $(\omega', E', \{E_i^1\}_{i\in\omega'}) \leftarrow FIBEEncrypt(\omega', DEK, PK)$.

Finally, the updating data file is stored on the cloud in the format as shown in Figure 1.

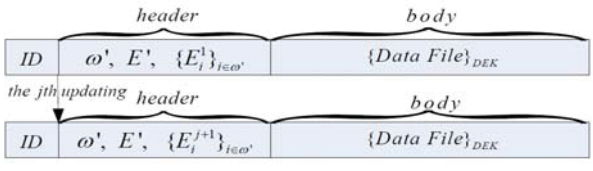

Figure 1. Format of an updating data file stored in the cloud

File Access In this operation, CS respond user's request of data file access, and then update the header file. Simultaneously, the frequency recording of CR_{cloud} is plus one where the initial frequency recording is set to be zero. Then, Cloud Servers send updated header file, ciphertexts of the requested body file and the frequency number of CR_{cloud} to the user, if |w n w'|>=d. On receiving the response from Cloud Servers, the user first compares the frequency recording between CR_{cloud} and CR_{user}, next, the user updates his secret key based on the difference between the two recordings. Finally, he decrypts data files by calling $FIBEEncrypt(\omega, \omega', D_i^{j+1}, E_i^{j+1})_{i\in(\omega\cap\omega')}$ to decrypt DEK and then decrypting data files using DEK.

Correctness According to the above Equation (2) (3), one easily deduces:

$$\begin{aligned}
&\frac{E'}{\prod_{i\in S}(e(D_i^{j+1}, E_i^{j+1}))^{\Delta_{i,S(0)}}} \\
&= \frac{DEK \times e(g,g)^{sy}}{\prod_{i\in S}(e(g^{\frac{st_i}{\prod_{k=1}^{k=j} x_k}}, g^{\frac{q(i)\prod_{k=1}^{k=j} x_k}{t_i}}))^{\Delta_{i,S(0)}}} \\
&= \frac{DEK \times e(g,g)^{sy}}{\prod_{i\in S}(e(g,g)^{sq(i)})^{\Delta_{i,S(0)}}} \\
&= DEK.
\end{aligned} \tag{5}$$

That is corresponded to Equation (2).

4.3 Summary

In our proposed scheme, we exploit the technique of hybrid encryption to protect data files, i.e, we encrypt data files using symmetric $DEKs$ and encrypt $DEKs$ with FIBE. Using FIBE, we are able to immediately enjoy convenient biometric data access control and good property such as error tolerance. To resolve the challenging issue of key exposure of user's private key, we combine the technique of key insulated cryptosystem with FIBE and free data owner from the interaction with user for the cumbersome access operations. We achieve this by letting Cloud Servers keep a CR_{cloud} corresponded to each user's CR_{user} and equipping both CS and user with the same type of updating mechanism due to key insulated cryptosystem. When user needs to access file of his interest, both CS and user will update the header file

and private key respectively which hinders the external attackers learning the symmetric key DEK. Thanks to the key insulated cryptosystem and this kind of access approach, even if the attacker intercepts certain temporary key x_j, he still can't erode our encrypted data. In addition, the data owner also does not need to always stay online since Cloud Servers will take over the burdensome task after having obtained the updating mechanism and the corresponding master key x_0^*. To some extent, our proposed scheme resembles the so called "one-time pad".

5 Evaluation

5.1 Security Analysis

Definition 2. [2](Decisional Modified Bilinear Diffie -Hellman (MBDH) Assumption) Suppose a challenger chooses $a,b,c,z \in Z_p$ at random. The Decisional MBDH assumption is that no polynomial time adversary can distinguish the tuple $(A=g^a, B=g^b, C=g^c, Z=e(g,g)^{\frac{ab}{c}})$ from $(A=g^a, B=g^b, C=g^c, Z=e(g,g)^z)$ with more than a negligible advantage.

Theorem 1. Under our regulated adversary model, the proposed scheme is strongly protected against external attacker's intercepting. Furthermore, it supports secure key updates.

Proof. Before giving the proof, we first intensify the ability of external attacker. Assume some attacker is able to obtain the sample of user's biometric identity and intercept the temporary key x_j of user's key. In our model, user's biometric identity isn't deemed as private key which may be stolen by some high-tech attacker such as hacker, otherwise according to the Selective-ID adversary model of Waters et al.[2] the security of our proposed scheme can be reduced to the hardness of Decisional MBDH assumption.

Suppose some attacker has the ability to get user's biometric attribute sample set along with the temporary secret key D^j and the temporary key x_{j-1}, so that he can be qualified to access data owner's data file due to user's biometric identity. Nevertheless, once the attacker makes the request of access to CS, CS always updates the header file like Equation (3) using the key insulated updating mechanism and respond the updated header file along with the body file to the attacker. Owing to the property of this updating mechanism, on no account can the attacker decrypt the header file except a negligible probability. To the contrary, only the legal user is able to update his secret key x_{j-1} to x_j in the method $x_j = H((x_0^*)^j) \bmod p$. In addition, the legal user has the privilege to compute the updated secret key $D_{i\in(\omega\cap\omega')}^{j+1}$ in the way $D_i^{j+1} = (D_i^j)^{x_j}$.

Therefore, our proposed scheme is strongly protected. Furthermore, thanks to the key insulated updating mechanism it also supports secure key updates.

5.2 Performance Analysis

This section numerically evaluates the performance of our proposed scheme in terms of the computation overhead in CS's updating operation and user's updating operation.

Lemma 1 In a prime p order group, any element g's y powers where y is chosen at random from Z_p can be computed in at most $O(\ln p)$ multiplicative operations.

Proof. First let y be expressed in binary like: $y = y_0 \cdot 2^0 + y_1 \cdot 2^1 + ... + y_n \cdot 2^n$, where $2^n < y \le 2^{n+1}$ and $y_i = 0 or 1$. That is,

$$g^y = \prod_{i=0}^{i=n} (((g^{y_i})^2)^{\cdots})^2 \qquad (6).$$

Thus computing g^y needs at most n times square operations and n times multiplication operations, which in total needs $O(\ln p)$ multiplicative operations as $n \approx O(\ln y)$ is the same magnitude of $O(\ln p)$.

Lemma 2. For any element $\beta \in Z_p^*$ where p is a prime, given a number $x \in Z_p^*$, the computation overhead of $\beta^{\frac{1}{x}}$ is $O(\ln p)$ multiplicative operations.

Proof. Because p is a prime, x is relative prime to p.

According to the Euclid algorithm, there exists integer a and b satisfying $ax+bp=1$. Thus one easily deduces:

$$\beta=\beta^{ax+bp}=\beta^{ax},\beta^{\frac{1}{x}}=\beta^{a} \qquad (7)$$

From the above equation, we know the computation of $\beta^{\frac{1}{x}}$ approximately needs $O(\ln p)$ multiplicative operations.

• CSP's updating overhead: From the E.q.(3), every time CS updates the header file, he just needs to perform $|\omega'|\cdot O(\ln p)$ multiplication operations. Simultaneously CS deletes the original header file and increases the CR_{cloud} by one, the cost of which can be neglected.

• User's updating overhead: Once user wants to access file of his interest, he first makes the request of the CR_{cloud} from CS and compares the difference between CR_{cloud} and CR_{user}. Next he updates his private key based on the difference between the two recordings. Thus, according to lemma 1, user needs to perform $\{|\omega'|\cdot O(\ln p)\}$ multiplication operations.

Table 1. Complexity of our proposed scheme Operation Complexity

Operation	Complexity
File Access-CSP's Updating	$\|\omega'\|\cdot O(\ln p)$
File Access-User's Updating	$\|\omega'\|\cdot O(\ln p)$

6 Conclusions

In this paper, we investigate the key exposure problem in cloud's biometric based access control. To achieve protecting sensitive data along with private key confidential against malicious servers and other external attackers thus meeting the requirement of overcoming biometric noisy property, we proposed an updating FIBE scheme where there exists updating operation interaction between CS and user. Specifically, we rely on FIBE in the file creation to provide biometric identity based access control. In addition, by employing the key insulated encryption scheme, our scheme achieves secure key updating and resembles "one time pad" which is deemed definitely secure in key distribution and management. Through detailed security and performance analysis, our scheme is showed to be secure and lightweight.

Acknowledgement

The work is supported by the National Natural Science Foundation of China (No. 60372039).

References

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters. "Attribute–based encryption for fine–grained access control of encrypted data", In Proceedings of ACM CCS, pp. 89-98, 2006.

[2] A. Sahai and B. Waters.'Fuzzy, "Identity Based Encryption", In Proceedings of EUROCRYPT, volume 3494 of LNCS, pp. 457-473, 2005.

[3] B. Waters.:'Dual System Encryption, "Realizing Fully Secure IBE and HIBE under Simple Assumption". In Proceedings of CRYPTO, volume 5677 of LNCS, pp. 619-636, 2009.

[4] D. Boneh and X. Boyen, "Efficient Selective-ID Secure Identity Based Encryption Without Random Oracles", In Proceedings of EUROCRYPT, volume 3027 of LNCS, pp. 223-238, 2004 .

[5] G. Wang, Q. Liu, and J. Wu, "Hierarchical Attribute-Based Encryption for Fine-Grained Access Control in Cloud Storage Services". In Proceedings of ACM CCS 2010, P&D session, pp. 35-737.

[6] J. Horwitz, B. Lynn, "Toward Hierarchical Identity–based Encryption", In Proceedings of EUROCRYPT, volume 2332 of LNCS, pp. 466-481, 2002.

[7] C. Gentry and A. Silverberg, "Hierarchical ID-Based Cryptography". In Proceedings of ASIACRYPT, pp. 548-566, 2002.

[8] S. Muller, S. Katzenbeisser, and C. Eckert, "Distribute Attribute-Based Encryption". In Proceedings of ICISC, pp. 20-36, 2008.

[9] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing", In Proceedings of IEEE INFOCOM, pp. 534-542, 2010.

[10] Y. Dodis, J. Katz, S. Xu and M. Yung, "Key-Insulated Public-Key Cryptosystems", In Proceedings of EUROCRYPT, volume 2332 of LNCS, pp. 65-82, 2002.