IOP Conf. Series: Materials Science and Engineering **242** (2017) 012080 doi:10.1088/1757-899X/242/1/012080

A new data collaboration service based on cloud computing security

Ren Ying[1], Li Hua-Wei[2], Wang Li_na[1]

[1]Naval Aviation University, Shandong, China

[2]Shan dong Business Institute, Shan dong Yantai, 264001, China

Abstract: With the rapid development of cloud computing, the storage and usage of data have undergone revolutionary changes. Data owners can store data in the cloud. While bringing convenience, it also brings many new challenges to cloud data security. A key issue is how to support a secure data collaboration service that supports access and updates to cloud data. This paper proposes a secure, efficient and extensible data collaboration service, which prevents data leaks in cloud storage, supports one to many encryption mechanisms, and also enables cloud data writing and fine-grained access control.

1. Introduction

With the rapid development of cloud computing [1], the storage and usage of data have undergone revolutionary changes. Data owners can store data in the cloud and outsource data to the cloud. Users of cloud computing will inevitably worry about privacy and privacy protection for private data. Therefore, maintaining the availability and confidentiality of data becomes critical in high-quality data services provided by cloud service providers (CSPs).How to ensure secure data storage has become an important issue[2][3][4].At present, some people have done some work to solve the secure cloud storage and data access problems, such as the introduction of attribute based encryption (ABE) to achieve fine-grained access control. Programs based ABE are data read-only shared services, a one to one encryption mechanism, that is, encrypted data can only be decrypted by a particular recipient. Existing work does not consider multiple user collaboration operations (readable, writable) encrypted data, that is, data collaboration services. This paper proposes a scalable data collaboration service called -SECO.SECO enables secure data collaboration services and supports dynamic user/data operations[5][6].Using a multitier identity based encryption architecture (HIBE), the architecture includes a root private key generator (R-PKG),a series of layer key generators (L-PKG), and an independent working domain. R-PKG just generates the private key for L-PKG, and L-PKG generates the key for the next layer in turn. A domain consists of a domain key generator (D-PKG) and a number of users who collaborate to complete a project.

2. Problem description

Figure 1 depicts the system model of SECO, a multilayer HIBE architecture. It consists of a R-PKG, a series of L-PKG and D-PKG, and individual users. In this hierarchy, the R-PKG generates parameters for the system's all entities and generates the keys for the next layer's L-PKG. Then L-PKG generates the key for the next layer of entity. L-PKG can share the key generation and identity authentication loads for R-PKG. Therefore, the transmission of keys and authentication can be implemented locally. A domain consists of a D-PKG and some users who collaborate to complete a project。In each domain, D-PKG stores a user list UL_{dom} that records the public key of all legitimate users in the domain.

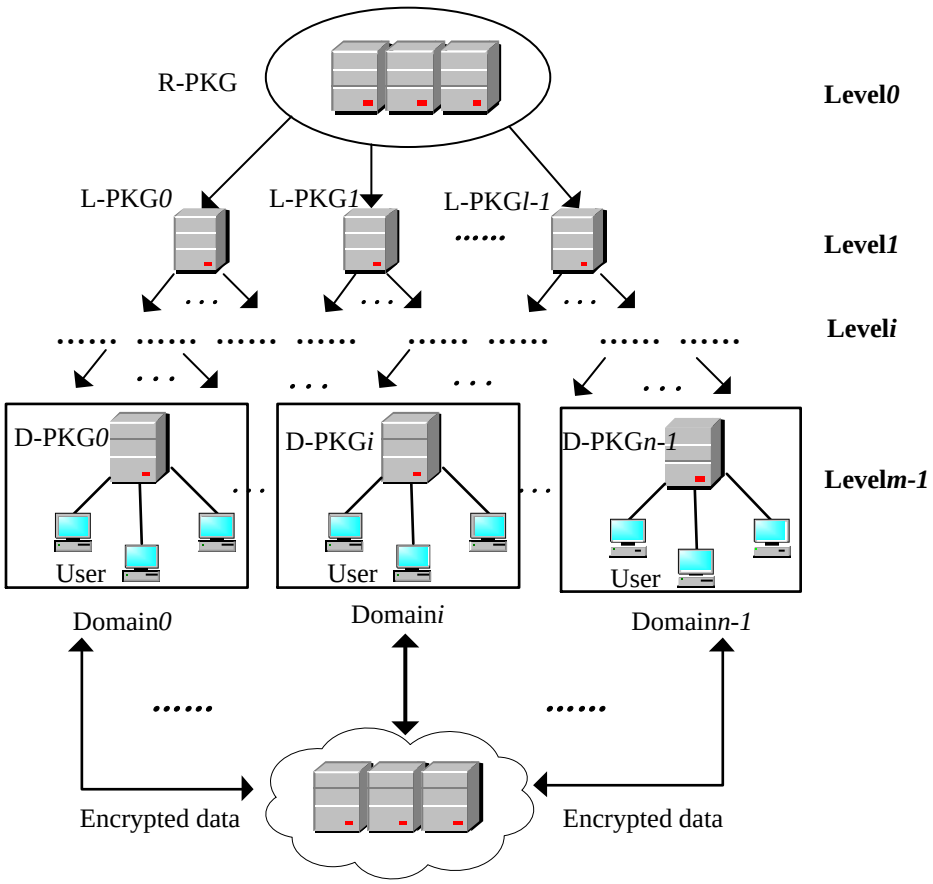


Figure 1 system model

D-PKG sends the latest list of users to all legitimate users in the domain. All entities in the domain store their data on the server side of the cloud. Users use their keys to decrypt data stored on the cloud server. All entities in the domain can dynamically access (read, write, update, etc.) data stored on the server side of the cloud.

In order to implement secure data collaboration services in cloud computing, this paper mainly aims to ensure domain data cannot be obtained by cloud server or an attacker. All attacks can either be active or passive. In order to implement secure data collaboration services, we implement the following security requirements:

1.Fine grained access control: Each user can access only own authorized data and cannot access unauthorized data.

2. Collusion attack: Users cannot conspire to access unauthorized data with other users or cloud servers by sharing their keys

3.Backward secrecy: The data access control policy must ensure that the cancelled user cannot access the data in the cloud server

3. Secure Cloud Data Collaboration Scheme

In order to implement secure cloud data collaboration services, this paper proposes a three layer SECO scheme.

3.1 System model

The system model of the three layer SECO scheme includes the following four entities:

1. Cloud Servers：Manage a large number of servers, have considerable storage space and computing power, and use these servers to provide high quality cloud computing services for the outside.

2.Root key generator (R-PKG)：Have a master key and generate the corresponding key for D-PKG.

3.Domain key generator (D-PKG)：Requests the key from the R-PKG and generates the key for all the entities in that domain.
4.User：Work together to complete a project, receive their private keys from the D-PKG, and rely on the cloud server to store their data.
Figure 2 depicts a three tier SECO system model, which is a three layers HIBE architecture. From the system model, we can see that it consists of a R-PKG and a range of work domains. A domain consists of a D-PKG, and some users who collaborate to complete a project. In practice, R-PKG is a trusted third party, and D-PKG is a team leader to manage all domain users.

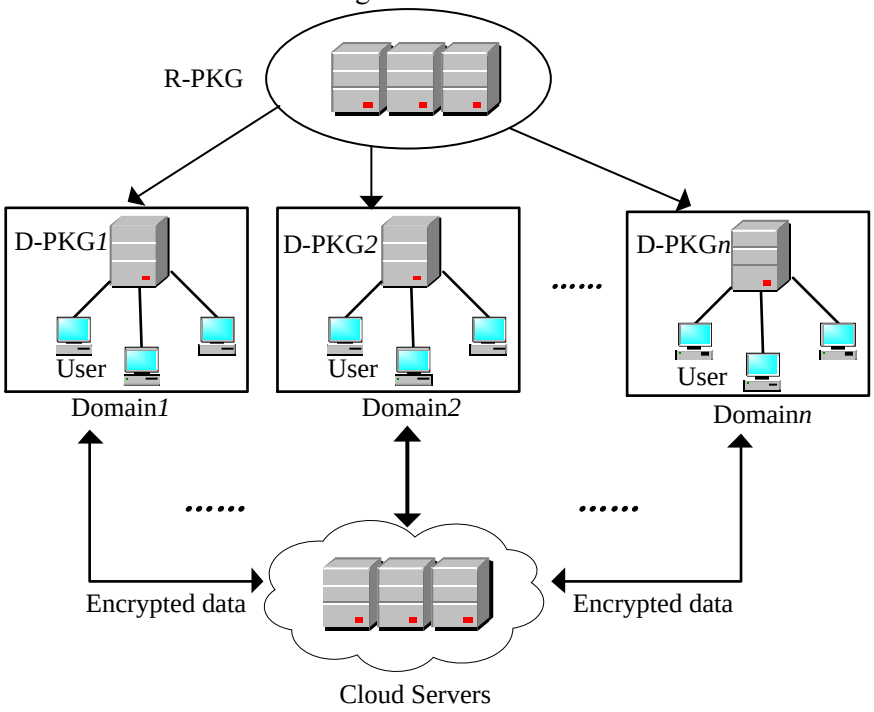


Figure 2 three tier SECO system model

3.2 Design and Implementation
In the three layers of the SECO scheme, *Level*0 = {R-PKG} and *Level*1 = {D-PKGs}.RPKG generates keys for D-PKGs, and then D-PKG generates keys for domain users. D-PKG has two keys: one private key and one master key. D-PKG uses these two keys to generate the private key for all users in its domain. Each user selects a random seed as their master key. In a domain, D-PKG has a unique ID with each user, and ID is an arbitrary string that can be the user's identity card number or mailbox. The user's public key is a ID group that consists of the D-PKG of the domain and the user's own ID, such as (D-PKG's ID, User's ID)[7]. In addition, R-PKG will also disclose some system parameters for encrypting and decrypting cloud data.
3.2.1System initialization. Set K as the security parameter used by the BDH parameter generator IG .
BDH(Bilinear Diffie-Hellman)Parameter generator[8][9]:if the random algorithm IG in polynomial time, by entering a security parameter $K > 0$, the output of two orders of prime q Multiplication cycle group $\mathcal{G}_1$ *and* $\mathcal{G}_2$ and a bilinear mapping: $\hat{e}:\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.Then the random algorithm is the BDH parameter generator. The public key of the user E_i is (ID_{dom}, ID_i), where ID_{dom} is ID of D-PKG, andID_i is the ID. of the userE_i.
The R-PKG runs the BDH parameter generator I$\mathcal{G}$, takes the security parameter K as input, and outputs params (system parameters) and a root master key s_0.System parameters include clear text space M and cipher text space C description, and other parameters. The system parameter is open, but the root master key is only R-PKG aware.

R-PKG first executes BDH parameter generator $I\mathcal{G}$ to generate two order of prime q multiplication cyclic group $\mathcal{G}_1$and $\mathcal{G}_2$,and a bilinear mapping $\hat{e}:\mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$.

The mapping has the properties of bilinear, computability and non degeneracy.

R-PKG then chooses an arbitrary generators $P_0 \epsilon \mathbb{G}_1$ and a random seed $s_0 \in \mathbb{Z}_q$,and make $Q_0 = s_0 P_0$.Among them,$\mathbb{Z}_q = \{0,1,2,\ldots\ldots q-1\}$.Finally,R-PKG defines four hash functions:$H_1:\{0,1\}^* \to \mathbb{G}_1$, $H_2:\mathbb{G}_2 \to \{0,1\}^n$, $H_3:\{0,1\}^n \times \{0,1\}^n \to \mathbb{Z}_q$ and $H_4:\{0,1\}^n \to \{0,1\}^n$, The four hash functions will be treated as random oracles.

Clear text space is $M = \{0,1\}^n$,Cipher text space is $\mathbb{C} = \mathbb{G}_1^t \times \{0,1\}^n$,Where t is the number of receivers in the cipher text.System parameter is: $params = \langle \mathbb{G}_1, \mathbb{G}_2, \hat{e}, P_0, H_1, H_2, H_3, H_4 \rangle$ R-PKG's root master key is s_0.

3.2.2 Key generation algorithm. Each D-PKG receives the system parameter (params) from the R-PKG.Every D-PKG randomly selected a seed $s_{dom} \in Z_q$ as the main key. s_{dom} will be used to generate the private key of the domain user. In addition to s_{dom} ,each D-PKG does not allow to generate any other params. R-PKG uses his master key to generate the private key for the next layer of D-PKGs, while D-PKG uses the system params and its own private key to generate the private key for all domain users. Make s_0. the unit element for group $\mathbb{G}_1$.

For each D-PKG *Edom* $\in$ *Level*1, he first selects a random seed $s_{dom} \in Z_q$ as his master key.Given the public key ID_{dom} of E_{dom} ,R-PKG generates the private key SK_{dom} for E_{dom} . R-PKG at first calculated $P_{dom} = H_1(ID_{dom}) \in G_1$ and R-PKG then calculate the private key for D-PKG as follows:

$$SK_{dom} = S_0 + s_0 P_{dom} \quad (3.1)$$

After the private key is calculated, R-PKG sends the value $Q_{dom} = s_{dom} P_0$ to the E_{dom} .For each D-PKG, he has two keys: a master key s_{dom} and a private key SK_{dom}.The private key SK_{dom} is used to decrypt all data stored in the cloud. Each D-PKG uses its master key and private key to generate the private key for all users in the domain.For each user E_i which is E_{dom} in the domain D-PKG , its ID group is(ID_{dom}, ID_i). E_i randomly selects an element $s_i \in Z_q$ as the primary key.To generate a private key SK_i for E_i .

For each E_i ,D-PKG E_{dom} first calculated $P_i = H_1(ID_{dom}, ID_i) \in \mathbb{G}_1$,D-PKG then calculate the private key for E_i as follows:

$$SK_i = SK_{dom} + s_{dom} P_i \quad (3.2)$$

After the private key is calculated, D-PKG sends the values Q_{dom} and Q_i to the E_i ,where $Q_i = s_i P_0 \cdot E_i$ also has two keys: a master key s_i and a private key SK_i. E_i uses s_i and SK_i to decrypt authorized data in the cloud.

3.2.3 Encryption algorithm. In the process of encryption, user input system parameters params,clear text $M \in M$ and ID express group of potential recipients, and then calculate the ciphertext $C \in \mathcal{C}$. After modifying the data of *D*,users use ID(public key)(ID_{dom}, ID_i)group in the domain t receiversrs,and $1 \le i \le t$ to encrypt D. The user first for all $1 \le i \le t$ calculation *Pi* = *H*1(*IDdom, IDi*) and $P_{dom} = H_1(ID_{dom})$, then, the user selects a random number $\sigma \in \{0,1\}$ and $r = H_3(\sigma, M)$ .Therefore, ciphertext is calculated as follows:

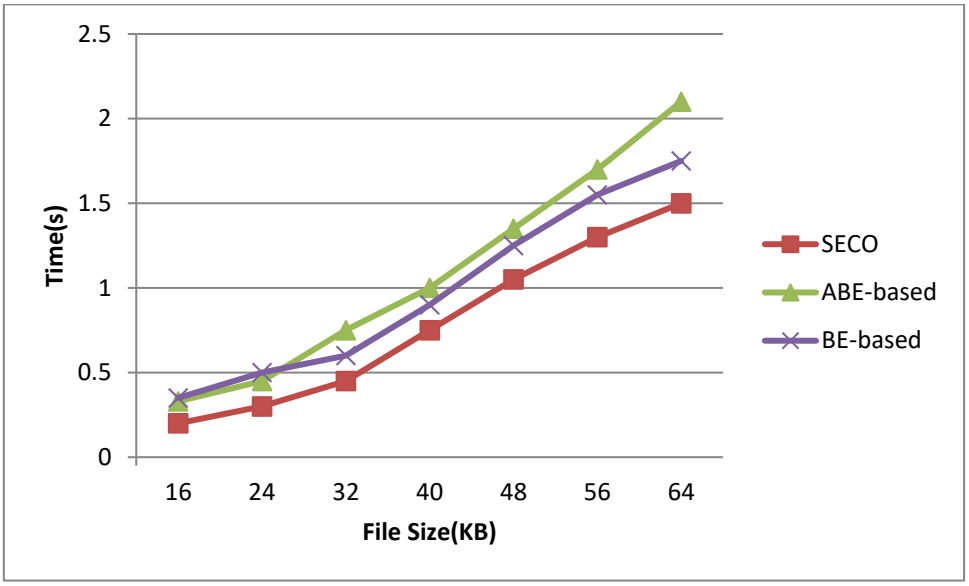
$$C = [rP_0, rP_1, \ldots, rP_t, \sigma \oplus H_2(g^r), M \oplus H_4(\sigma)] \quad (3.3)$$

among $g = \hat{e}(Q_0, P_{dom}) \in \mathbb{G}_2$。 From the formula 3-3, you can see that the user encrypts the data D with the public key of the t receiver in the same domain, and sends the ciphertext to the cloud server. It should be noted that since D-PKG manages all domain users, users can obtain the public key of the receiver from D-PKG or other users.

3.2.4 Decryption algorithm. In the process of decryption, D-PKG or user input system parameters params, cipher text $C \in \mathcal{C}$ and their encrypted private key *SK*, and then decrypt the clear text data $D \in \mathcal{M}$.D-PKG can decrypt all the cipher text data in the domain, but the user in the domain can only decrypt the authorized data.

4. Experimental result

We used a three layer SECO scheme to evaluate performance, where $Level_0 = \{RootPKG\}$ and $Level_1 = \{D-PKGs\}$ where all domain users are in *Level*2.We calculate the time cost of each algorithm, and all the experimental results are the average of the 100 experiments. We compare SECO and ABE based schemes, overhead based on the BE scheme, and then evaluate the scalability of the SECO scheme



Figures 3 the time overhead of the encryption algorithm in SECO and the scheme based ABE, the scheme based on BE

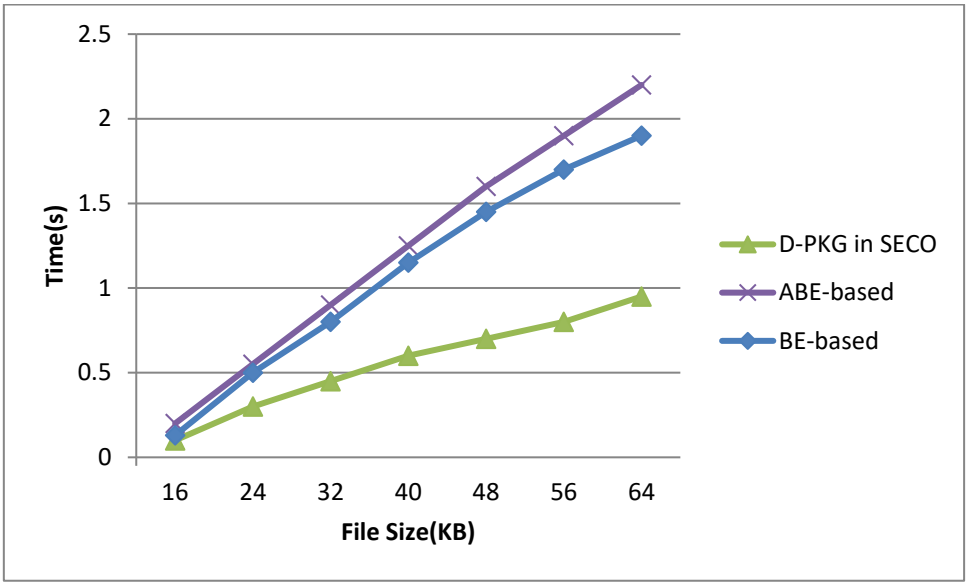


Figure 4 the time overhead of decryption algorithms SECO and scheme based on ABE, scheme based on BE

We evaluated the SECO scheme, the overhead based on the ABE scheme[10][11],and the encryption and decryption algorithms based on the BE scheme. Figures 3 and 4 illustrate the overhead of SECO and ABE based scenarios, based on the BE scheme, as the data size increases. Figure 3 shows the time overhead of the encryption algorithm in SECO and the scheme based ABE, the scheme based on BE.As can be seen from Figure 3,the time overhead of the encryption algorithm increases linearly with the size of the data in all three scenarios. With the increase of data size, SECO spend less time on Expenses scheme based on ABE, scheme based on BE. Figure 4 shows the time overhead of decryption algorithms SECO and scheme based on ABE, scheme based on BE. From Figure 4 we find that the time overhead for all three scenarios varies with the size of the data

5. Summary

This paper studies one to many encryption mechanisms, data write operations and fine-grained access control issues, and proposes a secure Cloud Data Collaboration Scheme- SECO. SECO takes advantage of a multilayer HIBE scheme to ensure data security in the cloud. SECO also implements one to many encryption mechanisms and data write operations, and then implements secure data collaboration services in cloud computing. Security analysis shows that SECO is safe for IND-ID-CCA under BDH assumption, and fine grained access control, anti collusion attack and backward secrecy are also implemented. In addition, we evaluate the performance of SECO. The experimental results show that the SECO scheme is very inexpensive and efficient

Reference

[1] Armbrust M, Fox A, Griffith R,et al. Above the clouds: A Berkeley view of cloud computing[R].[S.l.]: EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2009-28, 2009.
[2] Vimercati S D C D,Foresti S,et al.Encryption policies for regulating access to outsourced data[J]. ACM Transactions on Database Systems (TODS), 2010, 35(2):12:1–12:46.
[3] Samarati P,di Vimercati S D C.Data protection in outsourcing scenarios: issues and directions[C]//Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS). .[S.l.]: [s.n.], 2010:1–14.
[4] Arora R, Parashar A, Transforming C C I.Secure User Data in Cloud Computing Using Encryption Algorithms [J]. International Journal of Engineering Research and Applications (IJERA), 2013, 3(4):1922–1926.

[5]Vimercat1 S D C D,Foresti S,Jajodia S,et al.Encryption policies for regulating access to outsourced data[J].ACM Transactions on Database Systems (TODS), 2010, 35(2):12:1–12:46.
[6]Samarati P,di Vimercati S D C.Data protection in outsourcing scenarios: issues and directions[C]//Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS).[S.l.]: [s.n.] ,2010:1–14.
[7]Horwitz J,Lynn B.Toward hierarchical identity-based encryption[C]//Advances in Cryptology–EUROCRYPT. .[S.l.]: [s.n.] , 2002:466–481.
[8]Gentry C,Silverberg A.Hierarchical ID-based cryptography[C]//Advances in Cryptology–ASIACRYPT. .[S.l.]: [s.n.] , 2002:149–155.
[9]Boneh D,Franklin M.Identity-based encryption from the Weil pairing[C]//Advances in Cryptology–CRYPTO..[S.l.]: [s.n.],2001:213–229.
[10]Delerablee C.Identity-based broadcast encryption with constant size ciphertexts and private keys [M]//Advances in Cryptology–ASIACRYPT.[S.l.]: Springer, 2007:200–215.
[11]Gentry C,Waters B.Adaptive security in broadcast encryption systems (with short ciphertexts)[M]//Advances in Cryptology–EUROCRYPT.[S.l.]: Springer, 2009:171–188.