Hindawi
Security and Communication Networks
Volume 2018, Article ID 8932325, 12 pages
https://doi.org/10.1155/2018/8932325

Research Article

Adaptive Secure Cross-Cloud Data Collaboration with Identity-Based Cryptography and Conditional Proxy Re-Encryption

Qinlong Huang, Yue He, Wei Yue, and Yixian Yang

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Qinlong Huang; longsec@bupt.edu.cn

Received 27 April 2018; Accepted 16 September 2018; Published 1 October 2018

Guest Editor: Wei Wang

Data collaboration in cloud computing is more and more popular nowadays, and proxy deployment schemes are employed to realize cross-cloud data collaboration. However, data security and privacy are the most serious issues that would raise great concerns from users when they adopt cloud systems to handle data collaboration. Different cryptographic techniques are deployed in different cloud service providers, which makes cross-cloud data collaboration to be a deeper challenge. In this paper, we propose an adaptive secure cross-cloud data collaboration scheme with identity-based cryptography (IBC) and proxy re-encryption (PRE) techniques. We first present a secure cross-cloud data collaboration framework, which protects data confidentiality with IBC technique and transfers the collaborated data in an encrypted form by deploying a proxy close to the clouds. We then provide an adaptive conditional PRE protocol with the designed full identity-based broadcast conditional PRE algorithm, which can achieve flexible and conditional data re-encryption among ciphertexts encrypted in identity-based encryption manner and ciphertexts encrypted in identity-based broadcast encryption manner. The extensive analysis and experimental evaluations demonstrate the well security and performance of our scheme, which meets the secure data collaboration requirements in cross-cloud scenarios.

1. Introduction

Cloud computing which benefits individual and enterprise users in the aspect of convenient access, rich computation, and storage resources is becoming more and more popular. Nowadays, many enterprises have built their own cloud platform on one or multiple public cloud systems (e.g., Dropbox, Google Drive, and Baidu PCS) for data storage and sharing [1]. More recently, they have been widely used for more advanced, user-desired functionalities, in particular data collaboration among multiple users, such as collaborative document or paper editing.

However, the desirable functionality is yet restricted to the "walled-garden" of each cloud storage service, since data collaboration happens inside each cloud service, but not cross the cloud when these cloud services are unavailable in certain regions. For example, a transnational corporation has its headquarter in United States that uses Google Drive to backup data, as well as branch office in China that uses Baidu PCS (Google Drive have been banned in China). Employees in headquarter or branch office can enjoy the data collaboration service by uploading and downloading the data in Google Drive or Baidu PCS, respectively. But data collaboration between these two cloud services is inconvenient and inefficient. Researchers have proposed proxy deployment schemes to realize the cross-cloud data collaboration and designed inter-proxy transfer protocols to improve efficiency.

As promising as it is, cloud service is also facing many challenges and may impede its fast growth if not well resolved. Data security and privacy are the most serious issues in cloud computing when handling sensitive information (e.g., project schedule and commercial data). Since the cloud servers are honest but curious, the collected private information may be directly revealed. A common belief on the security and privacy of cloud system is that the data should be encrypted. There are many proposals on how to use modern cryptographic techniques such as identity-based cryptography (IBC), or attribute-based encryption (ABE) to

achieve secure data collaboration inside the cloud service [2]. Identity-based encryption (IBE) is the most commonly used encryption technique, through which data owner could encrypt data with authorized user's identity. Then identity-based broadcast encryption (IBBE) is proposed by allowing data owner to grant access permission to a group of users at one time. Recent works also utilize ABE to achieve fine-grained access control over encrypted data using access policies and ascribed attributes among private keys and ciphertexts. However, ABE brings heavy overhead of data encryption, data decryption, and key management. It may be not suitable for data collaboration directly, since the data collaboration service involves multiple parties and results in a large number of encryption and decryption operations. Thus, IBC is a prevalent technique to enable users to protect data confidentiality in cloud computing. Hence, by using an identity, the user can share data with collaborative users in a secure manner, which motivates more users to enjoy the benefits of cloud collaboration.

In fact, different cryptographic techniques may be deployed in cloud services to protect data confidentiality. It makes cross-cloud data collaboration to be a deeper challenge. A simple solution is to re-encrypt the data and upload the ciphertext, while another appropriate approach is to introduce a semitrusted proxy to transform a ciphertext into another ciphertext which can be decrypted by other users. To solve this problem, some schemes based on proxy re-encryption (PRE) are proposed. Corresponding to above data collaboration scenario, existing PRE based schemes only can achieve coarse-grained control over ciphertexts, since it only allows the re-encryption procedure to be executed in an all-or-nothing manner. If a project leader in department A wonders to discuss detailed project schedule with collaborative users in department B, so he only asks the proxy to re-encrypt the ciphertext of project schedule to collaborative users, not other ciphertexts that he has encrypted. A recent concept referred to as conditional PRE (CPRE) could address this issue, in which data owner can enforce re-encryption control over the initial ciphertexts and only the ciphertexts meeting the specified condition can be re-encrypted by the proxy [3]. However, how to achieve conditional and cross-cloud ciphertext collaboration between different cloud services is still a challenging problem.

In this paper, we propose an adaptive secure cross-cloud data collaboration scheme with IBC and PRE technique. The main contribution of this paper can be summarized as follows:

(1) We propose a secure cross-cloud data collaboration framework, which protects data confidentiality with IBC technique and transfers the collaborated data in an encrypted form by deploying a proxy close to the clouds.

(2) We provide an adaptive CPRE (ACPRE) protocol with four types of ciphertext collaboration, which can achieve flexible and conditional data re-encryption among ciphertexts in IBE and IBBE manners.

(3) We conduct extensive security and performance analysis and also experimental evaluations, which demonstrate that our scheme is secure and efficient for data collaboration.

The rest of this paper is organized as follows. Sections 2 and 3 discuss the related work and preliminaries, respectively. Section 4 provides the system framework, security model, and system definition. In Section 5, we present our detailed construction. Then, we give the security and performance analysis of our scheme in Sections 6 and 7, respectively. We discuss the experimental results in Section 8 and conclude this paper in Section 9.

2. Related Work

2.1. Data Collaboration. The demand for cloud-based data collaboration is rising along with the tremendously increased popularity in cloud computing. At present, many researchers have been devoted to achieve flexible and scalable data collaboration in cloud environment. The common approach is the data owners use the encryption key to encrypt the whole document into the ciphertext before uploading it to the cloud for storage. While the collaborative users download the document and recover the text with the decryption key [4, 5]. However, distributing the decryption key to the collaborative users can be challenging. These schemes cannot support cross-cloud data collaboration where data are shared and collaboratively used among different parties who adopt different cryptographic techniques involved in the multiple clouds.

Nepal et al. [9] provided a data collaboration solution in the hybrid cloud by fragmenting, encrypting and signing the data before uploading it to the cloud storage. Unfortunately, this method does not give the detailed construction of designed algorithms. Ahmadian et al. [10] discussed a hybrid clouds based architecture to provide an ideal environment for cooperation of multiple organizations. In this scheme, trusted private cloud processes sensitive data and public cloud adopts order preserving encryption to process data. However, the security of this scheme relies on trusted private cloud and it only considers the case that all the organizations adopt the same public cloud.

Bessani et al. [11] designed a DepSky system, which can provide dependable and secure storage service on diverse public clouds for users through the encryption, encoding, and replication techniques. However, this scheme assumes all clouds adopt the same symmetric encryption. Based on it, Fabian et al. [12] presented an improved architecture for medical data sharing between different cooperating organizations in multiple clouds, which uses different cryptographic techniques. But it focuses on splitting and reconstructing the encrypted data based on secret sharing. E et al. [13] implemented an efficient cross-cloud file collaboration system, called CoCloud, which includes an inter-proxy transfer protocol and a cross-cloud data transfer optimization algorithm. Although CoCloud supports file collaborations among four popular cloud storage services in the United States and China, it ignores the security and privacy concerns of users that cannot support data collaboration on ciphertexts.

2.2. Identity-Based Proxy Re-Encryption. IBE and IBBE techniques are both widely adapted to share data and guarantee data confidentiality in cloud [14]. In their approaches, data

owner publishes encrypted data to the service provider with one or multiple recipients, while only the intended users who can derive the decryption key would be able to decrypt and access the data owner's private data.

Combined with IBBE and IBE, PRE is another technique which is commonly adopted to achieve data sharing and disseminating. The first PRE scheme was proposed by Blaze [15]. It can transform the ciphertext under Alice to be another ciphertext under Bob by using semitrusted proxy. Following this seminal work, identity-based PRE [16] was proposed, which allows any recognizable string to serve as a public key. Since then, PRE became a hot research topic in cryptography field. Liang et al. [17] proposed a cloud-based revocable identity-based PRE scheme, which supports user revocation by making a proxy to re-encrypt all ciphertexts every once in a while. Li et al. [18] designed a secure identity-based PRE scheme with multi-hop construction. Wang et al. [19] designed a health cloud system framework based on IBE, called IBPRE, in which doctor can access health data with authorization from patient based on PRE. Zhou et al. [6] proposed an identity-based PRE construct named IBBPRE to convert the user's IBBE-encrypted data into IBE-encrypted data without leaking any sensitive information. Combining the broadcast encryption system and PRE, Sun et al. [20] constructed chosen-ciphertext secure PRE scheme IBPBRE, which enables the proxy to transfer IBE-encrypted data into IBBE-encrypted data.

2.3. Conditional Proxy Re-Encryption. The above PRE schemes only allow data sharing in a coarse-grained manner. If the user delegates a re-encryption key to the proxy, either all ciphertexts can be re-encrypted and then be accessible to the intended users, or none of the ciphertexts can be re-encrypted or accessed by others. This issue is addressed in the CPRE schemes[21, 22], which is first proposed by Weng et al. [23]. In this scheme, proxy can successfully re-encrypt data only if the prescribed conditions are met. Shao et al. [7] proposed an identity-based CPRE (IBCPRE), in which a proxy is allowed to transform a subset of ciphertexts under an identity to other ciphertexts under another identity. However, it cannot authorize decryption right to a group of users. Chu et al. [24] introduced a more generalized notion of conditional proxy broadcast re-encryption (IBCPBRE), which allows a user to delegate the decryption rights of ciphertexts to a group of users, restricted to a certain condition. Xu et al. [8] proposed an efficient conditional identity-based broadcast PRE scheme for cloud email, which is referred to as IBBCPBRE, and transformed an IBBE ciphertext into another IBBE ciphertext if the conditions are satisfied.

3. Preliminaries

3.1. Bilinear Map. Let $\mathbb{G}_0$ and $\mathbb{G}_T$ be two multiplicative groups with the same prime order p. A map $e : \mathbb{G}_0 \times \mathbb{G}_0 \longrightarrow \mathbb{G}_T$ with the following properties is said to be bilinear:

(1) Computability. There is a polynomial time algorithm to compute $e(g, h) \in \mathbb{G}_T$ for any $g, h \in \mathbb{G}_0$.

(2) Bilinearity. For all $g, h \in \mathbb{G}_0$ and $a, b \in \mathbb{Z}_p$, we have $e(g^a, h^b) = e(g, h)^{ab}$.

(3) Nondegeneracy. There exists $g, h \in \mathbb{G}_0$ such that $e(g, h) \neq 1$.

3.2. Identity-Based Encryption. The IBE scheme consists of the following algorithms.

(1) $(PK, MK) \longleftarrow Setup(1^\lambda)$. It takes a security parameter λ as input and outputs public parameters PK which are distributed to users, and the master secret key MK which is kept private.

(2) $SK \longleftarrow KeyGen(MK, PK, ID)$. It takes the master secret key MK and an identity ID as input, and outputs a secret key SK corresponding to ID.

(3) $CT \longleftarrow Enc(ID, PK, M)$. It takes an identity ID, the public parameters PK and a plaintext M as input, and outputs a ciphertext CT.

(4) $M/\perp \longleftarrow Dec(SK, PK, CT)$. It decrypts the ciphertext CT using the secret key SK, and outputs M or $\perp$.

3.3. Conditional Proxy Re-Encryption. The CPRE scheme which allows encrypting and re-encrypting with a condition is comprised of the following algorithms.

(1) $GK \longleftarrow Setup(1^\lambda)$. It takes a security parameter λ as input and generates the global public parameter GK.

(2) $(PK, SK) \longleftarrow KeyGen(GK, u)$. It generates the public key PK and secret key SK for a user u.

(3) $RK \longleftarrow ReKeyGen(SK_A, PK_B, c)$. It takes as input a secret key SK_A of the delegator, a public key PK_B of the delegatee, a conditional set c, and outputs a re-encryption key RK.

(4) $CT \longleftarrow Enc\text{-}1(PK, M)$. It takes as input a public key PK, a message M, and outputs a first-level ciphertext CT under public key PK.

(5) $CT' \longleftarrow Enc\text{-}2(PK, M, c)$. It takes as input a public key PK, a message M and a conditional set c, and outputs a second-level ciphertext CT'.

(6) $CT \longleftarrow ReEnc(CT', RK)$. It takes as input a second-level ciphertext CT' associated with c, a re-encryption key RK, and outputs a first-level ciphertext CT if the conditional set is matched.

(7) $M \longleftarrow Dec\text{-}1(CT, SK)$. It takes as input a first-level ciphertext CT and a secret key SK, and outputs a message M or an error symbol.

(8) $M \longleftarrow Dec\text{-}2(CT', SK)$. It takes as input a second-level ciphertext CT' and a secret key SK, and outputs a message M or an error symbol.

4. Scheme Overview

4.1. System Model. Our scheme applies IBC and PRE techniques to accomplish secure cross-cloud data collaboration. As shown in Figure 1, the system model of our scheme consists of trust authority, cloud service provider (CSP), proxy server, data owner, and user.

(1) Trust authority. The trust authority is a fully trusted party that generates system parameters to initialize the system. It also generates secret keys with users' identity.

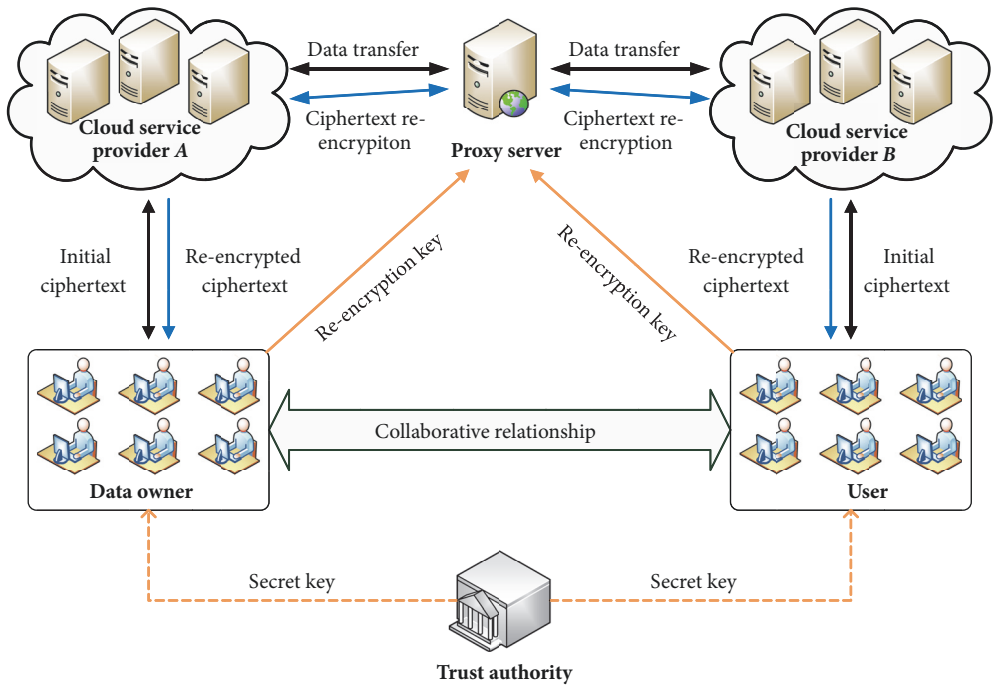

Figure 1: System model of our scheme.

(2) CSP. The CSP is a semitrusted party which offers data storage service and enables the authorized users to access ciphertexts stored on it.

(3) Proxy server. The proxy server provides data transfer service based on web APIs between CSPs. Moreover, it is in charge of generating re-encrypted ciphertext to achieve cross-cloud data collaboration among multiple users.

(4) Data owner. The data owners encrypt data with IBE or IBBE algorithm and then upload ciphertext to the CSP. When sharing data stored in one CSP with the user who cannot access directly, the data owners send the re-encryption key to the proxy server which transforms the ciphertext into a re-encrypted ciphertext that can be accessed by user.

(5) User. The user can decrypt the ciphertext stored in the cloud with his or her secret key. Simultaneously, the users can also decrypt the data owners' re-encrypted ciphertext with their secret keys, which achieves data collaboration between two CSPs.

4.2. Security Requirements. We assume the trust authority to be a trusted party which will not collude with unauthorized users. However, we assume the CSP and proxy server are honest but curious, which means they may collude to get unauthorized data. Specifically, the security requirements cover the following aspects.

(1) Data confidentiality. The unauthorized users who are not the intended receivers defined by data owner should be prevented from accessing the data; unauthorized access from the semitrusted CSP should also be prevented.

(2) Re-encryption secrecy. The re-encryption keys with unauthorized data users should be prevented from transforming the ciphertexts successfully.

4.3. System Definitions. Our ACPRE protocol includes the following four types of ciphertext collaboration.

Type I Individual-to-Individual Re-Encryption. The initial ciphertext associated with one authorized user is transformed to a re-encrypted ciphertext of one user by proxy server.

Type II Individual-to-Group Re-Encryption. The initial ciphertext associated with one authorized user is transformed to a re-encrypted ciphertext of a set of users by proxy server.

Type III Group-to-Individual Re-Encryption. The initial ciphertext associated with a set of authorized users is transformed to a re-encrypted ciphertext of one user by proxy server.

Type IV Group-to-Group Re-Encryption. The initial ciphertext associated with a set of authorized users is transformed to a re-encrypted ciphertext of a set of users by proxy server.

To implement the above protocol, we define our scheme with the following algorithms.

(1) $Setup(1^{\lambda}, N)$. The trust authority takes as input a security parameter λ and the maximal size of receiver set N and outputs a system public key PK and a master secret key MK.

(2) $KeyGen(PK, MK, ID)$. The trust authority takes as input PK and MK and an identity ID, and outputs the secret key SK.

(3) $IBE.Enc(PK, M, ID, C)$. The CSP takes as input PK, data M, an identity ID, and a condition C and then outputs an initial ciphertext CT_{IBE}.

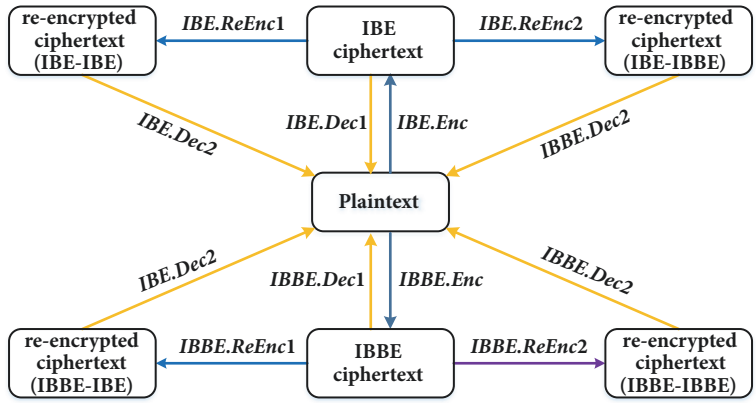

Figure 2: System workflow of our scheme.

(4) *IBBE.Enc*(PK, M, U, C). The CSP takes as input PK, data M, a set U of identities, and a condition C and then outputs an initial ciphertext CT_{IBBE}.

Type I Re-Encryption

(5) *IBE.ReKeyGen*1(PK, ID, SK, ID', C). The user takes as input PK, his or her identity ID and secret key SK, a user's identity ID' and a condition C, then outputs a re-encryption key RK.

(6) *IBE.ReEnc*1(PK, RK, CT_{IBE}). The proxy server takes as input PK, a re-encryption key RK, and an initial ciphertext CT_{IBE} and then outputs a re-encrypted ciphertext CT'_{IBE}.

Type II Re-Encryption

(7) *IBE.ReKeyGen*2(PK, ID, SK, U', C). The user takes as input PK, his or her identity ID and secret key SK, a set U' of users' identities, and a condition C and then outputs a re-encryption key RK.

(8) *IBE.ReEnc*2(PK, RK, CT_{IBE}). The proxy server takes as input PK, a re-encryption key RK and an initial ciphertext CT_{IBE} and then outputs a re-encrypted ciphertext CT'_{IBBE}.

Type III Re-Encryption

(9) *IBBE.ReKeyGen*1(PK, ID, SK, U, ID', C). The user takes as input PK, his or her identity ID and secret key SK, a set U of authorized users' identities, an identity ID' of user and a condition C, then outputs a re-encryption key RK.

(10) *IBBE.ReEnc*1(PK, ID, RK, U, CT_{IBBE}). The proxy server takes as input PK, a user's identity ID and re-encryption key RK, a set U of authorized users' identities, and an initial ciphertext CT_{IBBE} and then outputs a re-encrypted ciphertext CT'_{IBE}.

Type IV Re-Encryption

(11) *IBBE.ReKeyGen*2(PK, ID, SK, U', C). The user takes as input PK, his or her identity ID and secret key SK, a set U' of users' identities, and a condition C and then outputs a re-encryption key RK.

(12) *IBBE.ReEnc*2(PK, ID, RK, U, CT_{IBBE}). The proxy server takes as input PK, a user's identity ID and re-encryption key RK, a set U of authorized users' identities, and an initial ciphertext CT_{IBBE} and then outputs a re-encrypted ciphertext CT'_{IBBE}.

(13) *IBE.Dec*1(PK, SK, CT_{IBE}). The user takes as input PK, a secret key SK, and initial ciphertext CT_{IBE} and then outputs data M.

(14) *IBE.Dec*2(PK, SK, CT'_{IBE}). The user takes as input PK, a secret key SK, and re-encrypted ciphertext CT'_{IBE} and then outputs data M.

(15) *IBBE.Dec*1(PK, ID, SK, U, CT_{IBBE}). The user takes as input PK, his or her identity ID and secret key SK, a set U of authorized users' identities, and initial ciphertext CT_{IBBE} and then outputs data M.

(16) *IBBE.Dec*2(PK, ID', SK, U', CT'_{IBBE}). The user takes as input PK, his or her identity ID' and secret key SK, a set U' of authorized users' identities, and re-encrypted ciphertext CT'_{IBBE} and then outputs data M.

4.4. System Workflow. The system workflow is described as Figure 2. In the system initialization phase, trust authority runs *Setup* algorithm to generate system public key and master secret key. Concurrently, it also uses *KeyGen* algorithm to generate secret keys for the users in the system. At first, data owner runs *IBE.Enc* or *IBBE.Enc* algorithm to encrypt data with its identity and a keyword as condition. Then data owner outsources the encrypted data to the CSP. The user, which can access the same CSP with the data owner, would send a request of accessing the ciphertext to the CSP. After receiving the request, the CSP will send the ciphertext to the user. If it is the intended user, it could run *IBE.Dec*1 or *IBBE.Dec*1 algorithm to decrypt the ciphertext with his or her secret key. Furthermore, an authorized user could run re-encryption key generation algorithm to generate the re-encryption key containing new identities and keyword condition and then upload it into the proxy server. The re-encryption key decides which users can have the ability to access the data by enforcing the new identities on it. With the re-encryption key, the proxy server would run re-encryption algorithm to transform the initial ciphertexts to the re-encrypted ciphertexts. When the user sends an access request to the CSP, the CSP would return the re-encrypted

ciphertexts. The user can run the *IBE.Dec*2 or *IBBE.Dec*2 algorithm to decrypt the re-encrypted ciphertexts.

5. Construction

5.1. System Setup. The trust authority first runs *Setup* algorithm to select a bilinear map $e : \mathbb{G}_0 \times \mathbb{G}_0 \longrightarrow \mathbb{G}_T$, where $\mathbb{G}_0$ and $\mathbb{G}_T$ are two multiplicative groups with prime order p. Then the trust authority chooses a maximum number of receivers N, chooses $g, h, u \in \mathbb{G}_0$, $\gamma \in \mathbb{Z}_p$ randomly, cryptographic hash functions $H_1 : \{0,1\}* \longrightarrow \mathbb{Z}_p^*$, $H_2 : \{0,1\}* \longrightarrow \mathbb{G}_0$ and $H_3 : \mathbb{G}_T \longrightarrow \mathbb{G}_0$, and finally outputs a system public key $PK = (g^\gamma, e(g,h), e(g,h)^\gamma, h, h^\gamma, ..., h^{\gamma^N}, u, u^\gamma, ..., u^{\gamma^N}, t, t^\gamma, ..., t^{\gamma^N})$ and a master secret key $MK = (g, \gamma)$.

5.2. Key Generation. For each user with identity *ID*, the trust authority runs *KeyGen* algorithm to generate the secret key $SK = g^{1/(\gamma+H_1(ID))}$.

5.3. Data Encryption. The data owner encrypts data M with a set of identities, and then outsources the result to the CSP. First, the data owner chooses a random DK and encrypts M based on symmetric encryption algorithm SE, that is $C_0 = SE_{DK}(M)$.

(1) If the data owner chooses to share data with a single user, then he runs *IBE.Enc* algorithm to pick $k \in \mathbb{Z}_p$, and compute the following with a condition $\alpha \in \mathbb{Z}_p$.

$$\begin{aligned} CT_{IBE} &= \left(C_0, C_1 = DK \cdot e(g,h)^k, C_2 \right. \\ &= h^{k\cdot(\gamma+H_1(ID))}, C_3 = \left(ut^\alpha\right)^{k\cdot((\gamma+H_1(ID))/H_1(ID))}, C_4 \\ &\left. = \left(ut^\alpha\right)^{-k}\right) \end{aligned} \tag{1}$$

(2) If the data owner chooses to share data with a group of users, then he runs *IBBE.Enc* algorithm to pick $k \in \mathbb{Z}_p$, and choose a set U of users' identities, and compute the following with a condition $\alpha \in \mathbb{Z}_p$.

$$\begin{aligned} CT_{IBBE} &= \left(C_0, C_1 = DK \cdot e(g,h)^k, C_2 \right. \\ &= h^{k\cdot\prod_{ID_i\in U}(\gamma+H_1(ID_i))}, C_3 \\ &= \left(ut^\alpha\right)^{k\cdot\prod_{ID_i\in U}((\gamma+H_1(ID_i))/H_1(ID_i))}, C_4 = g^{-\gamma k}, C_5 \\ &\left. = \left(ut^\alpha\right)^{-k}\right) \end{aligned} \tag{2}$$

5.4. Data Re-Encryption. Suppose a user with identity *ID* needs to collaborate with other users, he can generate a re-encryption key and send it to the CSP for ciphertext re-encryption according to the chosen collaboration type.

Type I Re-Encryption

*IBE.ReKeyGen*1: The user chooses a single user with identity ID' and picks a random $s \in \mathbb{Z}_p$ and computes the following re-encryption key RK with a condition $\alpha \in \mathbb{Z}_p$.

$$\begin{aligned} R_1 &= SK \cdot \left(ut^\alpha\right)^{k'} \cdot u^s = g^{1/(\gamma+H_1(ID))} \cdot \left(ut^\alpha\right)^{k'} \cdot u^s, \\ R_2 &= h^{k'\cdot(\gamma+H_1(ID))}, \\ R_3 &= IBE.Enc\left(ID', s\right) \end{aligned} \tag{3}$$

*IBE.ReEnc*1: The CSP takes RK as input and first computes

$$\begin{aligned} C_1' &= \frac{C_1}{\left(e\left(C_4, R_2\right) \cdot e\left(R_1, C_2\right)\right)} \\ &= DK \cdot e\left(u^s, h^{-k}\right)^{\gamma+H_1(ID)} \end{aligned} \tag{4}$$

Then the CSP generates the re-encrypted ciphertext.

$$\begin{aligned} CT_{IBE}' &= \left(C_0' = C_0, C_1' = DK \cdot e\left(u^s, h^{-k}\right)^{\gamma+H_1(ID)}, C_2' \right. \\ &\left. = C_2 = h^{k\cdot(\gamma+H_1(ID))}, C_3' = R_3 = IBE.Enc\left(ID', s\right)\right) \end{aligned} \tag{5}$$

Type II Re-Encryption

*IBE.ReKeyGen*2: The user chooses a set U' of users' identities and picks a random $s \in \mathbb{Z}_p$ and then computes the following re-encryption key RK with a condition $\alpha \in \mathbb{Z}_p$.

$$\begin{aligned} R_1 &= SK \cdot \left(ut^\alpha\right)^{s/H_1(ID)} = g^{1/(\gamma+H_1(ID))} \cdot \left(ut^\alpha\right)^{s/H_1(ID)}, \\ R_2 &= h^{k'\cdot\prod_{ID_i\in U'}(\gamma+H_1(ID_i))}, \\ R_3 &= H_3\left(e\left(g,h\right)^{k'}\right) \cdot h^s, \\ R_4 &= g^{-\gamma k'} \end{aligned} \tag{6}$$

*IBE.ReEnc*2: The CSP takes RK as input and first computes

$$C_1' = \frac{C_1}{e\left(R_1, C_2\right)} = DK \cdot e\left(\left(ut^\alpha\right)^{-s}, h^k\right)^{(\gamma+H_1(ID))/H_1(ID)} \tag{7}$$

Then the CSP generates the re-encrypted ciphertext.

$$\begin{aligned} CT_{IBBE}' &= \left(C_0' = C_0, C_1' = DK \right. \\ &\cdot e\left(\left(ut^\alpha\right)^{-s}, h^k\right)^{(\gamma+H_1(ID))/H_1(ID)}, C_2' = R_2 \\ &= h^{k'\cdot\prod_{ID_i\in U'}(\gamma+H_1(ID_i))}, C_3' = C_3 \\ &= \left(ut^\alpha\right)^{k\cdot((\gamma+H_1(ID))/H_1(ID))}, C_4' = R_4 = g^{-\gamma k'}, C_5' = R_3 \\ &\left. = H_3\left(e\left(g,h\right)^{k'}\right) \cdot h^s\right) \end{aligned} \tag{8}$$

Type III Re-Encryption

*IBBE.ReKeyGen*1: The user chooses a single user with identity ID' and picks a random $s \in \mathbb{Z}_p$ and then computes the following re-encryption key RK with a condition $\alpha \in \mathbb{Z}_p$.

$$\begin{aligned} R_1 &= SK \cdot \left(ut^{\alpha}\right)^{k'} \cdot u^{s/H_1(ID)} \\ &= g^{1/(\gamma+H_1(ID))} \cdot \left(ut^{\alpha}\right)^{k'} \cdot u^{s/H_1(ID)}, \\ R_2 &= h^{k' \cdot \prod_{ID_i \in U}(\gamma+H_1(ID_i))}, \\ R_3 &= IBE.Enc\left(ID', s\right) \end{aligned} \tag{9}$$

*IBBE.ReEnc*1: The CSP takes RK as input and first computes

$$\begin{aligned} C_1' = C_1 &\cdot \left(e\left(C_4, h^{\Delta_\gamma(ID,U)}\right) \cdot e\left(R_1, C_2\right)\right. \\ &\left. \cdot e\left(C_5, R_2\right)\right)^{-1/\prod_{IID_i \in U \wedge ID_i \neq ID} H_1(ID_i)} = DK \\ &\cdot e\left(u^s, h^{-k}\right)^{\prod_{ID_i \in U}((\gamma+H_1(ID_i))/H_1(ID_i))} \end{aligned} \tag{10}$$

Then the CSP generates the re-encrypted ciphertext.

$$\begin{aligned} CT'_{IBE} = \Big(& C_0' = C_0, C_1' = DK \\ & \cdot e\left(u^s, h^{-k}\right)^{\prod_{ID_i \in U}((\gamma+H_1(ID_i))/H_1(ID_i))}, C_2' \\ & = \left(C_2\right)^{1/\prod_{ID_i \in U} H_1(ID_i)} \\ & = h^{k \cdot \prod_{ID_i \in U}((\gamma+H_1(ID_i))/H_1(ID_i))}, C_3' = R_3 \\ & = IBE.Enc\left(ID', s\right)\Big) \end{aligned} \tag{11}$$

Type IV Re-Encryption

*IBBE.ReKeyGen*2: The user chooses a set U' of users' identities and picks a random $s \in \mathbb{Z}_p$ and then computes the following re-encryption key RK with a condition $\alpha \in \mathbb{Z}_p$.

$$\begin{aligned} R_1 &= SK \cdot \left(ut^{\alpha}\right)^{s/H_1(ID)} = g^{1/(\gamma+H_1(ID))} \cdot \left(ut^{\alpha}\right)^{s/H_1(ID)}, \\ R_2 &= h^{k' \cdot \prod_{ID_i \in U'}(\gamma+H_1(ID_i))}, \\ R_3 &= H_3\left(e\left(g,h\right)^{k'}\right) \cdot h^s, \\ R_4 &= g^{-\gamma k'} \end{aligned} \tag{12}$$

*IBBE.ReEnc*2: The CSP takes RK as input and first computes

$$\begin{aligned} C_1' = C_1 &\cdot \left(e\left(C_4, h^{\Delta_\gamma(ID,U)}\right) \cdot e\left(R_1, C_2\right)\right)^{-1/\prod_{ID_i \in U \wedge ID_i \neq ID} H_1(ID_i)} \\ &= DK \cdot e\left(\left(ut^{\alpha}\right)^s, h^{-k}\right)^{\prod_{ID_i \in U}((\gamma+H_1(ID_i))/H_1(ID_i))} \end{aligned} \tag{13}$$

Then the CSP generates the re-encrypted ciphertext.

$$\begin{aligned} CT'_{IBBE} = \Big(& C_0' = C_0, C_1' = DK \\ & \cdot e\left(\left(ut^{\alpha}\right)^s, h^{-k}\right)^{\prod_{ID_i \in U}((\gamma+H_1(ID_i))/H_1(ID_i))}, C_2' = R_2 \\ & = h^{k' \cdot \prod_{ID_i \in U'}(\gamma+H_1(ID_i))}, C_3' = C_3 \\ & = \left(ut^{\alpha}\right)^{k \cdot \prod_{ID_i \in U}((\gamma+H_1(ID_i))/H_1(ID_i))}, C_4' = R_4 \\ & = g^{-\gamma k'}, C_5' = R_3 = H_3\left(e\left(g,h\right)^{k'}\right) \cdot h^s\Big) \end{aligned} \tag{14}$$

5.5. Data Decryption. (1) If the ciphertext is an initial ciphertext CT_{IBE} generated by *IBE.Enc* algorithm, the user with identity ID' runs *IBE.Dec*1 algorithm to compute the following and generates $DK = C_1/K$.

$$\begin{aligned} K &= e\left(SK, C_2\right) = e\left(g^{1/(\gamma+H_1(ID))}, h^{k \cdot (\gamma+H_1(ID))}\right) \\ &= e\left(g,h\right)^k \end{aligned} \tag{15}$$

(2) If the ciphertext is a Type I or Type III re-encrypted ciphertext CT'_{IBE}, the user with identity ID' runs *IBE.Dec*2 algorithm to compute the following.

$$s = IBE.Dec\left(SK_{ID'}, C_3'\right) \tag{16}$$

Then the user generates $DK = C_1' \cdot e(u^s, C_2')$.

(3) If the ciphertext is an initial ciphertext CT_{IBBE} generated by *IBBE.Enc* algorithm, the user with identity ID' runs *IBBE.Dec*1 algorithm to compute the following if $ID' \in U$.

$$\begin{aligned} K &= \left(e\left(C_4, h^{\Delta_\gamma(ID,U)}\right) \cdot e\left(SK, C_2\right)\right)^{1/\prod_{ID_i \in U \wedge ID_i \neq ID} H_1(ID_i)} \\ &= e\left(g,h\right)^k \end{aligned} \tag{17}$$

Then the user generates $DK = C_1/K$.

(4) If the ciphertext is a Type II or Type IV re-encrypted ciphertext CT'_{IBBE}, the user with identity ID' runs *IBBE.Dec*2 algorithm to compute the following if $ID' \in U$.

$$\begin{aligned} K' = &\left(e\left(C_4', h^{\Delta_\gamma(ID',U')}\right)\right. \\ &\left. \cdot e\left(SK', C_2'\right)\right)^{1/\prod_{ID_i \in U' \wedge ID_i \neq ID'} H_1(ID_i)} = e\left(g,h\right)^{k'} \end{aligned} \tag{18}$$

Then the user computes h^s.

$$Z = \frac{C_5'}{H_3\left(K'\right)} = \frac{H_3\left(e\left(g,h\right)^{k'}\right) \cdot h^s}{H_3\left(e\left(g,h\right)^{k'}\right)} = h^s \tag{19}$$

Finally, the user generates $DK = C_1' \cdot e(Z, C_3')$.

6. Security Analysis

6.1. Correctness. The authorized user can generate DK from the re-encrypted ciphertext CT' according to the collaboration type.

(1) If CT' is a Type I re-encrypted ciphertext CT_{IBE}, it computes

$$\begin{aligned} DK &= C_1' \cdot e\left(u^s, C_2'\right) \\ &= DK \cdot e\left(u^s, h^{-k}\right)^{\gamma + H_1(ID)} \cdot e\left(u^s, h^{k\cdot(\gamma + H_1(ID))}\right) \end{aligned} \tag{20}$$

(2) If CT' is a Type III re-encrypted ciphertext CT'_{IBE}, it computes

$$\begin{aligned} DK &= C_1' \cdot e\left(u^s, C_2'\right) \\ &= DK \cdot e\left(u^s, h^{-k}\right)^{\prod_{ID_i \in U}((\gamma + H_1(ID_i))/H_1(ID_i))} \\ &\quad \cdot e\left(u^s, h^{k\cdot\prod_{ID_i \in U}((\gamma + H_1(ID_i))/H_1(ID_i))}\right) \end{aligned} \tag{21}$$

(3) If CT' is a Type II re-encrypted ciphertext CT_{IBBE}, it computes

$$\begin{aligned} DK &= C_1' \cdot e\left(Z, C_3'\right) \\ &= DK \cdot e\left(\left(ut^{\alpha}\right)^s, h^{-k}\right)^{\prod_{ID_i \in U}((\gamma + H_1(ID_i))/H_1(ID_i))} \\ &\quad \cdot e\left(h^s, \left(ut^{\alpha}\right)^{k\cdot\prod_{ID_i \in U}((\gamma + H_1(ID_i))/H_1(ID_i))}\right) \end{aligned} \tag{22}$$

(4) If CT' is a Type IV re-encrypted ciphertext CT'_{IBBE}, it computes

$$\begin{aligned} DK &= C_1' \cdot e\left(Z, C_3'\right) \\ &= DK \cdot e\left(\left(ut^{\alpha}\right)^{-s}, h^k\right)^{(\gamma + H_1(ID))/H_1(ID)} \\ &\quad \cdot e\left(h^s, \left(ut^{\alpha}\right)^{k\cdot((\gamma + H_1(ID))/H_1(ID))}\right) \end{aligned} \tag{23}$$

6.2. Scheme Security. The shared data in our scheme is encrypted with IBE and IBBE techniques, which are secure against chosen plaintext attack (CPA) since the decisional bilinear Diffie–Hellman (DBDH) assumption holds [25]. Our scheme is CPA secure in the random oracle model with the game among adversary $\mathscr{A}$ and challenger $\mathscr{C}$.

Proof. The adversary $\mathscr{A}$ chooses a set U^* of challenge identities. The challenger $\mathscr{C}$ randomly runs the *Setup* algorithm to generate a system public key PK and a master secret key MK, and models the hash functions H_1, H_2 and H_3 as three random oracles. The adversary $\mathscr{A}$ can issue hash query, key generation query, and re-encryption key generation query to challenger $\mathscr{C}$. In the challenge phase, the adversary sends two challenge messages m_0 and m_1 to challenger $\mathscr{C}$, then the challenger $\mathscr{C}$ runs encryption algorithm to generate the challenge ciphertext CT^*, where b is chosen randomly in $\{0, 1\}$. Finally, the challenger $\mathscr{C}$ sends the challenge ciphertext CT^* to adversary $\mathscr{A}$. In the guess phase, the adversary $\mathscr{A}$ outputs a guess $b' \in \{0, 1\}$. As proved in [8], we can find that if the adversary $\mathscr{A}$ successfully breaks our scheme, except it can break the CPA security of the IBE and IBBE scheme, or can solve the DBDH problem in re-encryption key generation query.

Specially, the data is encrypted with a random symmetric key DK, and then DK is protected by IBE or IBBE technique. Since the symmetric encryption and IBC scheme are secure, the confidentiality of outsourced data can be guaranteed against unauthorized users whose identities are not in the set of receivers' identities. Moreover, the re-encryption requests from unauthorized users who are not included in the identity set or do not own the same condition will not be executed successfully, and the CSP cannot get any useful information about the DK from the ciphertext and re-encryption key. □

7. Performance Analysis

We analyze the performance efficiency of data encryption, re-encryption and decryption by comparing our scheme with several secure data collaboration schemes. The result is shown in Table 1. Let T_{pair} be the computation cost of a single pairing, T_{exp} be the computation cost of an exponent operation, and N_u be the total number of users. We ignore simple multiplication, hash, and symmetric encryption and decryption operations.

First, we discuss the computation cost of data encryption. The encryption process can be divided into two types: IBBE encryption and IBE encryption. IBBPRE [6], IBBCPBRE [8] and *IBBE.Enc* algorithm in our scheme belong to the former, and their computation costs are T_{pair}+$(N_u$+5$)T_{exp}$, $(3N_u$+8$)T_{exp}$, $(3N_u$+10$)T_{exp}$, respectively, which grow linearly with the number of users. In addition, the *IBE.Enc* algorithm in our scheme cost $9T_{exp}$ to encrypt the data, which is constant and the same as IBCPRE [7], and also less than these compared schemes. However, compared with other three schemes, we can see that encryption algorithms of our scheme cost a little more time since initial ciphertext will be transformed into two types of re-encrypted ciphertext in subsequent operation, while other schemes will be converted into only one type of re-encrypted ciphertext.

In the re-encryption phase, IBBPRE [6], IBCPRE [7], and IBBCPBRE [8] correspond to Type III, Type I, and Type IV of our scheme. The computation cost of IBBPRE [6] is more than Type III in our scheme and the other two schemes are the same as Type I and Type IV in our scheme. It should be noted that these re-encryption computations are performed by cloud server or proxy server.

Further, in the initial ciphertext decryption phase, IBBPRE [6], IBBCPBRE [8] and *IBBE.Dec*1 algorithm in our scheme all decrypt ciphertext for sharing data with a group, so their decryption computation costs are the same as $2T_{pair}$+N_uT_{exp}, which are related to the number of users. IBCPRE [7] and *IBE.Dec*1 algorithm in our scheme decrypt the ciphertext which is encrypted to an individual, and they only need to perform one pairing operation to decrypt. In the re-encrypted ciphertext decryption phase, IBBCPBRE

Table 1: Comparison of computation in secure data collaboration.

Schemes		*Enc*		*ReEnc*	*Dec*-1	*Dec*-2
IBBPRE [6]		$T_{pair}+(N_u+5)T_{exp}$		$(3N_u+3)T_{exp}$	$2T_{pair}+N_uT_{exp}$	$3T_{pair}+T_{exp}$
IBCPRE [7]		$T_{pair}+4T_{exp}$		$2T_{pair}$	T_{pair}	$2T_{pair}$
IBBCPBRE [8]		$(3N_u+8)T_{exp}$		$2T_{pair}+N_uT_{exp}$	$2T_{pair}+N_uT_{exp}$	$3T_{pair}+N_uT_{exp}$
Our scheme	IBE	$9T_{exp}$	Type-I	$2T_{pair}$	T_{pair}	$2T_{pair}$
			Type-II	T_{pair}		
	IBBE	$(3N_u+10)T_{exp}$	Type-III	$3T_{pair}+(N_u+1)T_{exp}$	$2T_{pair}+N_uT_{exp}$	$3T_{pair}+N_uT_{exp}$
			Type-IV	$2T_{pair}+N_uT_{exp}$		

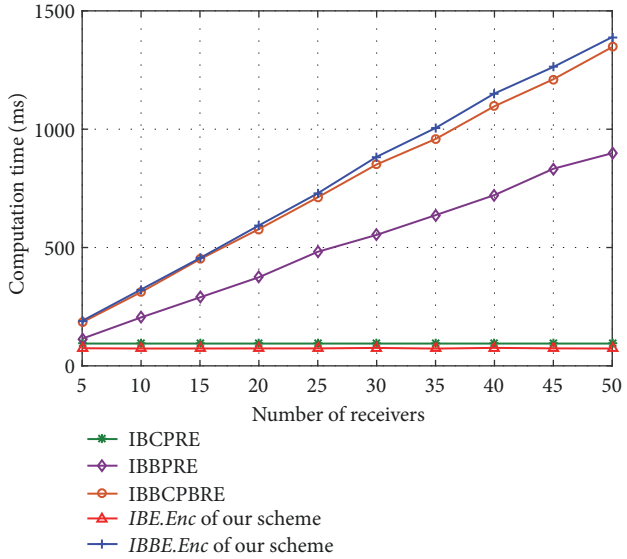

Figure 3: Computation cost of data encryption.

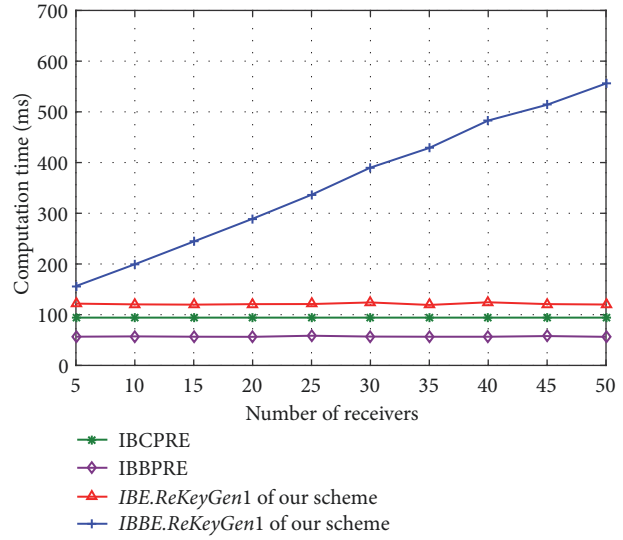

Figure 4: Computation cost of IBE re-encryption key generation.

[8] and *IBBE.Dec*2 algorithm in our scheme both cost $3T_{pair}+N_uT_{exp}$ to decrypt the ciphertext, which also increase linearly with the number of users.

8. Experimental Evaluations

We conduct experiments on a Windows system with an Intel Core i7-6700 CPU with 3.4 GHz processor and 8 GB memory. The experimental prototype is written in Java language with Java pairing-based cryptography (jpbc) library [26]. We accomplish several relative schemes including IBBPRE [6], IBCPRE [7] and IBBCPBRE [8] in the same experimental environment, and use a pairing type A 160-bit elliptic curve group based on the super-singular curve over a 512-bit finite field.

We analyze the computation cost of the data encryption by comparing our scheme with IBBPRE [6], IBCPRE [7] and IBBCPBRE [8]. In the data encryption phase, the data owner in these schemes encrypts a file with IBE algorithm or IBBE algorithm and then posts the encrypted file to the CSP. Figure 3 shows the computation cost on data owners during this phase. In the re-encryption key generation phase, we give the computation cost of IBE re-encryption key generation and IBBE re-encryption key generation respectively, and the results are shown in Figures 4 and 5. Figure 4 shows the computation cost on users for IBE re-encryption key generation versus the number of users in the ciphertext. The computation cost in IBCPRE [7], IBBPRE [6] and Type I re-encryption in our scheme is almost constant, while the result in Type III re-encryption in our scheme grows linearly with the number of users. The reason is that the former executes re-encryption algorithm for individual and the latter executes for group. Figure 5 focuses on the computation cost of IBBE re-encrypted key generation. We can see that the re-encryption key generation time of IBBCPBRE [8], Type II and Type IV of our scheme is almost the same, which increase mainly with the number of users. The experimental result of re-encryption phase is depicted in Figure 6, which shows the computational time of re-encryption on the proxy server versus the number of receivers associated with initial ciphertext in the CSP. Obviously, IBBCPBRE [8], Type III and Type IV re-encryption in our scheme vary linearly with the number of receivers, and IBCPRE [7], Type I and Type II re-encryption in our scheme remain constant. Moreover, the computation cost of IBBPRE [6] in this phase is much more than other schemes.

Furthermore, we evaluate the computation cost from three aspects in data decryption phase, and the results are

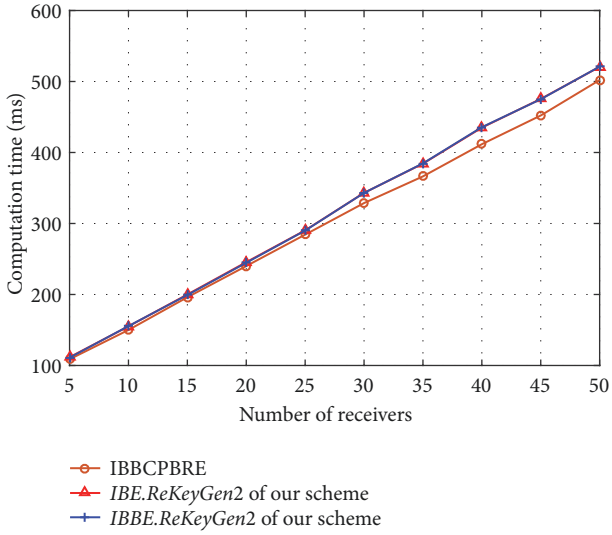

Figure 5: Computation cost of IBBE re-encryption key generation.

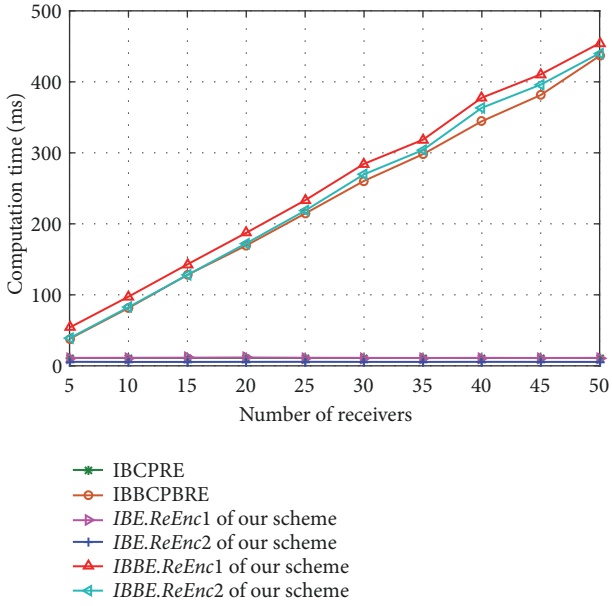

Figure 6: Computation cost of data re-encryption.

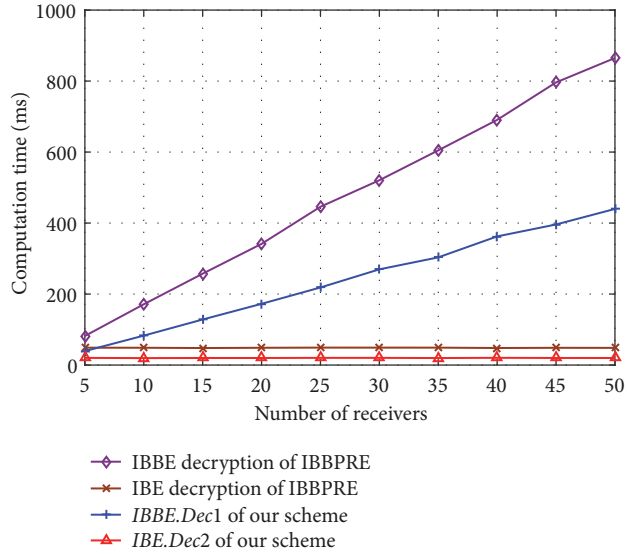

Figure 7: Computation cost of decryption in Type III.

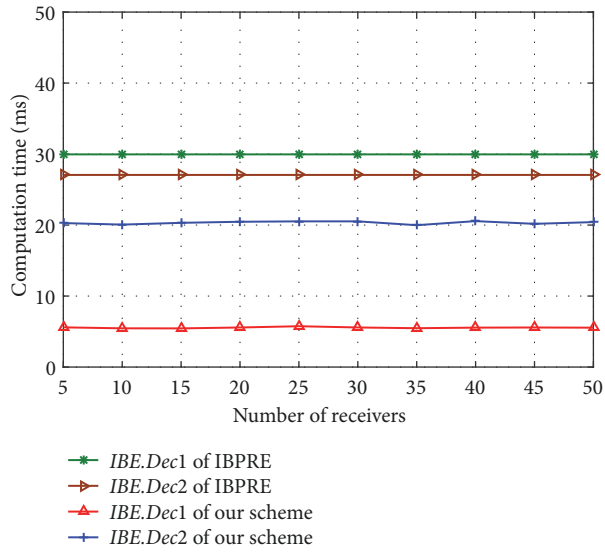

Figure 8: Computation cost of decryption in Type I.

shown in Figures 7, 8, and 9, respectively. Figure 7 reveals the computation cost on user side when decrypting IBBE initial ciphertext and re-encrypted ciphertext of IBE. Compared with IBBPRE [6], it is obviously that the computation cost of decrypting the initial ciphertext of IBBPRE [6] grows at faster pace than our scheme, while the time of decrypting re-encrypted ciphertext of IBBPRE [6] and our scheme is almost constant. The fact is that IBBPRE [6] takes about 49 ms that is higher than 20 ms in our scheme. The computation cost on IBPRE [19] and our scheme are shown in Figure 8. The computation time of decrypting initial ciphertext and re-encrypted ciphertext in IBPRE [19] is 30 ms and 27 ms, and the result in our scheme is 20 ms and 5 ms. The computation cost of IBPRE [19] is evidently more than that of our scheme since decryption algorithm in the former needs more pairing operation. Figure 9 indicates the computation cost of *IBBE.Dec2* algorithm in our scheme and re-encrypted ciphertext decryption in IBBCPBRE [8]. The experimental results show that our scheme is almost identical with IBBCPBRE [8].

9. Conclusion

Recent years, data collaboration between different clouds becomes increasingly popular. In this paper, we propose an adaptive secure cross-cloud data collaboration scheme with IBC and CPRE techniques. Firstly, we present a cross-cloud data collaboration framework which protects the confidentiality of data in the semitrusted CSPs with IBE or IBBE technique. The framework deploys a proxy

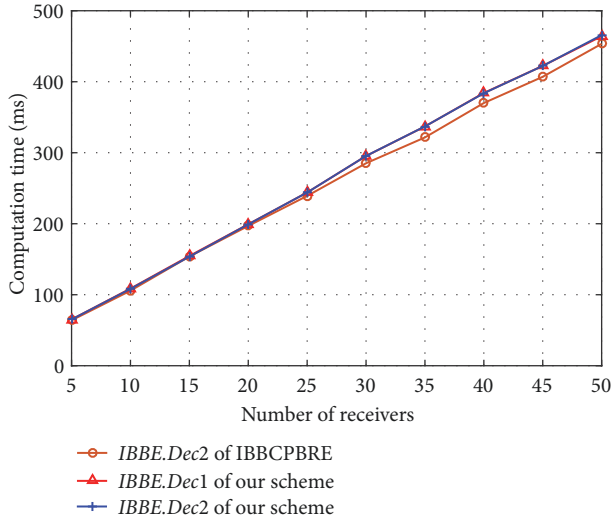

FIGURE 9: Computation cost of decrypting IBBE re-encrypted ciphertext.

server close to the clouds to transfer and re-encrypt the collaborated data. Secondly, we provide an ACPRE protocol, which not only addresses the issue of flexible ciphertext collaboration between CSPs, but also supports conditional data re-encryption. We further make comparisons with current schemes and conduct experiments with jpbc library. The results indicate that our cross-cloud data collaboration scheme is secure, efficient, and practical.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant No. 2016YFB0800605, the National Natural Science Foundation of China under Grant No. 61572080, Key Program of Joint Funds of the National Natural Science Foundation of China under Grant No. U1736212, and China Scholarship Council under Grant No. 201806475007.

References

[1] J. Li, M. Wen, C. Gu, and H. Li, "PSS: Achieving high-efficiency and privacy-preserving similarity search in multiple clouds," in *Proceedings of the ICC 2016 - 2016 IEEE International Conference on Communications*, pp. 1–6, Kuala Lumpur, Malaysia, May 2016.

[2] B. Albelooshi, E. Damiani, K. Salah, and T. Martin, "Securing Cryptographic Keys in the Cloud: A Survey," *IEEE Cloud Computing*, vol. 3, no. 4, pp. 42–56, 2016.

[3] D. Li and Y. Zhou, "A secure and reliable hybrid model for cloud-of-clouds storage systems," in *Proceedings of the 22th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 1157–1162, Wuhan, China, 2016.

[4] H. Wang, P. Shi, and Y. Zhang, "JointCloud: A Cross-Cloud Cooperation Architecture for Integrated Internet Service Customization," in *Proceedings of the 37th IEEE International Conference on Distributed Computing Systems, ICDCS 2017*, pp. 1846–1855, USA, June 2017.

[5] J. Li, D. Lin, A. C. Squicciarini, J. Li, and C. Jia, "Towards Privacy-Preserving Storage and Retrieval in Multiple Clouds," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 499–509, 2017.

[6] Y. Zhou, H. Deng, Q. Wu, B. Qin, J. Liu, and Y. Ding, "Identity-based proxy re-encryption version 2: making mobile access easy in cloud," *Future Generation Computer Systems*, vol. 62, pp. 128–139, 2016.

[7] J. Shao, G. Wei, Y. Ling, and M. Xie, "Identity-based conditional proxy re-encryption," in *Proceedings of the IEEE International Conference on Communications (ICC '11)*, pp. 1–5, Kyoto, Japan, June 2011.

[8] P. Xu, T. F. Jiao, Q. H. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email," *IEEE Transactions on Computers*, vol. 65, no. 1, pp. 66–79, 2016.

[9] S. Nepal, C. Friedrich, L. Henry, and S. Chen, "A secure storage service in the hybrid cloud," in *Proceedings of the 4th IEEE/ACM International Conference on Cloud and Utility Computing, UCC 2011*, pp. 334-335, Australia, December 2011.

[10] M. Ahmadian, A. Paya, and D. C. Marinescu, "Security of applications involving multiple organizations and order preserving encryption in hybrid cloud environments," in *Proceedings of the 28th IEEE International Parallel and Distributed Processing Symposium Workshops, IPDPSW 2014*, pp. 894–903, USA, May 2014.

[11] A. Bessani, M. Correia, B. Quaresma, F. André, and P. Sousa, "DepSky," in *Proceedings of the the sixth conference*, p. 31, Salzburg, Austria, April 2011.

[12] B. Fabian, T. Ermakova, and P. Junghanns, "Collaborative and secure sharing of healthcare data in multi-clouds," *Information Systems*, vol. 48, pp. 132–150, 2015.

[13] J. E, Y. Cui, P. Wang, Z. Li, and C. Zhang, "CoCloud: Enabling efficient cross-cloud file collaboration based on inefficient web APIs," in *Proceedings of the INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, Atlanta, GA, May 2017.

[14] F. Beato, S. Meul, and B. Preneel, "Practical identity-based private sharing for online social networks," *Computer Communications*, vol. 73, pp. 243–250, 2016.

[15] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," in *Advances in Cryptology—EUROCRYPT '98 (Espoo)*, vol. 1403 of *Lecture Notes in Computer Science*, pp. 127–144, Springer, Berlin, Germany, 1998.

[16] M. Green and G. Ateniese, "Identity-based proxy re-encryption," in *Proceedings of the 5thInternational Conference on Applied Cryptography and Network Security*, pp. 288–306, Zhuhai, China, 2007.

[17] K. Liang, J. K. Liu, D. S. Wong, and W. Susilo, "An Efficient Cloud-Based Revocable Identity-Based Proxy Re-encryption Scheme for Public Clouds Data Sharing," in *Computer Security - ESORICS 2014*, vol. 8712 of *Lecture Notes in Computer Science*, pp. 257–272, Springer International Publishing, Cham, 2014.

[18] Z. Li, C. Ma, and D. Wang, "Towards Multi-Hop Homomorphic Identity-Based Proxy Re-Encryption via Branching Program," *IEEE Access*, vol. 5, pp. 16214–16228, 2017.

[19] X. A. Wang, J. Ma, F. Xhafa, M. Zhang, and X. Luo, "Cost-effective secure E-health cloud system using identity based cryptographic techniques," *Future Generation Computer Systems*, vol. 67, pp. 242–254, 2017.

[20] J. Sun and Y. Hu, "Chosen-ciphertext secure bidirectional proxy broadcast re-encryption schemes," *International Journal of Information and Communication Technology*, vol. 8, no. 4, pp. 405–419, 2016.

[21] J. Weng, Y. Yang, Q. Tang, R. H. Deng, and F. Bao, "Efficient Conditional Proxy Re-encryption with Chosen-Ciphertext Security," in *Information Security Journal*, vol. 5735 of *Lecture Notes in Computer Science*, pp. 151–166, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

[22] K. Liang, Z. Liu, X. Tan, D. S. Wong, and C. Tang, "A CCA-Secure Identity-Based Conditional Proxy Re-Encryption without Random Oracles," in *Information Security and Cryptology – ICISC 2012*, vol. 7839 of *Lecture Notes in Computer Science*, pp. 231–246, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[23] J. Weng, R. H. Deng, X. Ding, C.-K. Chu, and J. Lai, "Conditional proxy re-encryption secure against chosen-ciphertext attack," in *Proceedings of the 4th International Symposium on ACM Symposium on Information, Computer and Communications Security, ASIACCS'09*, pp. 322–332, Australia, March 2009.

[24] C.-K. Chu, J. Weng, S. S. M. Chow, J. Zhou, and R. H. Deng, "Conditional proxy broadcast re-encryption," in *Information Security and Privacy*, C. Boyd and J. G. Nieto, Eds., vol. 5594 of *Lecture Notes in Computer Science*, pp. 327–342, Springer, Berlin, Germany, 2009.

[25] Q. Huang, Y. Yang, and J. Fu, "PRECISE: Identity-based private data sharing with conditional proxy re-encryption in online social networks," *Future Generation Computer Systems*, vol. 86, pp. 1523–1533, 2018.

[26] *The java pairing-based cryptography library*, http://gas.dia.unisa.it/proje%20cts/jpbc/.