ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 4, Issue 5, May 2017

Revocable ID-Based Proxy Re-Encryption and Data Uploading With Remote Data Integrity Checking

[1] Usharani J, [2] Dr. Usha Sakthivel
[1] PG Scholar, Department of Computer Science, VTU University, RRCE, Bengaluru-74
[2] Professor and HOD, Department of Computer Science, VTU University, RRCE, Bengaluru-74

***Abstract*— Many organizations have large amounts of data so, wants to store and process their data by using the remote cloud computing system. In public cloud, the clients store their massive data in the remote public cloud servers. Since the stored data is outside of the control of the clients, it entails the security risks in terms of confidentiality, integrity and availability of data and service. Remote data integrity checking is a primitive which can be used to convince the cloud clients that their data are kept intact. In some special cases, the data owner may be restricted to access the public cloud server, the data owner will delegate the task of data processing and uploading to the third party, for example the proxy. However, the major problem of cloud data storage is security. Therefore, cloud data storage need some mechanisms that should be able to specify storage correctness and integrity of data stored on a cloud. On the other side, the remote data integrity checking protocol must be efficient in order to make it suitable for capacity-limited end devices. Thus, based on revocable identity-based public cryptography and proxy public key cryptography, we will study RID-PREUIC protocol. Revocable ID-Based Proxy Re-Encryption and Data Uploading with Remote Data Integrity Checking is an attractive alternative for public key cryptography. RID-PREUIC eliminates the demands of public key infrastructure (PKI) and certificate administration in conventional public key settings. An RID-PREUIC consists of Client, Proxy, PCS (Public Cloud Server), End-user and a trusted third party (i.e. private key generator, PKG). The PKG is responsible to generate each user's private key by using the associated ID information (e.g. e-mail address, name or social security number). Therefore, no certificate and PKI are required in the associated cryptographic mechanisms under RID-PREUIC**

Keywords - Cloud computing, identity-based proxy re-encryption, Revocation, Key authority, remote data integrity checking

1. INTRODUCTION

Cloud computing has been envisioned as the next generation of distributed/utility computing. It is defined as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., Networks, servers, storage, applications, and services) that can be speedily provisioned and delivered with minimal management effort or less service provider interaction. Cloud computing is defined as five essential characteristics, three service models and four deployment models by the National Institute of Standards and Technology. The key features of cloud are, on-demand self-service, broad network access, location independent resource pooling, swift resource elasticity, and measured service. The main three service models are software as a service (SAAS), platform as a service (PAAS) and infrastructure as a service (IAAS), while the deployment models include private cloud, public cloud, community cloud and hybrid cloud. At present, cloud-computing model can offer any conceivable form of services, such as computational resources for high performance computing applications, web services, social networking, and telecommunications services. Many clients would like to store and process their data by using the remote cloud computing system.

In public cloud computing, the clients store their massive data in the remote public cloud servers. Since the stored data is outside of the control of the clients, it entails the security risks in terms of confidentiality, integrity and availability of data and service. Remote data integrity checking is a primitive which can be used to convince the cloud clients that their data are kept intact. In some special cases, the data owner may be restricted to access the public cloud server, the data owner will delegate the task of data processing and uploading to the third party, for example the proxy. Although, the major problem faced by cloud data storage is security. Therefore, cloud data storage need some mechanisms that should be able to specify storage correctness and integrity of data stored on a cloud. On the other side, the remote data integrity checking protocol must be efficient in order to make it suitable for capacity-limited end devices. Thus, based on revocable identity-based public cryptography and proxy public key cryptography, we will study RID-PREUIC protocol.

Revocable ID-Based Proxy Re-Encryption and Data Uploading with Remote Data Integrity Checking is an attractive alternative for public key cryptography. RID-PREUIC eliminates the demands of public key infrastructure (PKI) and certificate administration in conventional public key settings. An RID-PREUIC consists of Client, Proxy, PCS (Public Cloud Server), End-

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 5, May 2017

user and a trusted third party (i.e. private key generator, PKG). The PKG is responsible to generate each user's private key by using the associated ID information (e.g. e-mail address, name or social security number). Therefore, no certificate and PKI are required in the associated cryptographic mechanisms under RID-PREUIC. In such a case, ID-based proxy Re-encryption allows a sender to encrypt message directly by using a receiver's ID without checking the validation of public key certificate and send it to proxy, Proxy will again re-encrypt the client data and forward to cloud. Accordingly, the receiver uses the private key associated with her/his ID to decrypt such cipher text. Since a public key setting has to provide a user revocation mechanism, the research issue on how to revoke misbehaving/compromised users in an RID-PREUIC is naturally raised. Revoked user can't able to upload or download any file to/from the cloud.

Outsourcing data to cloud server implies that data is out control of clients. This may cause clients hesitation since the outsourced data usually contain valuable and sensitive information. Data sharing is often implemented in an open and hostile environment, and cloud server would become a target of attacks. Even worse, cloud server itself may reveal clients data for illegal profit. Also data sharing is not static. That is, when a client authorization gets expired, he/she should no longer possess the privilege of accessing the previously and subsequently shared data. Therefore, while outsourcing data to cloud server, clients also want to control access to these data such that only those currently authorized end-users can share the outsourced data. A natural solution to overcome the above problem is to use cryptographically enforced access control such as identity-based proxy re-encryption along with cloud revocation mechanism. Furthermore, to overcome the above security threats, such kind of identity-based access control placed on the shared data should meet some security goals, that includes data confidentiality, remote-integrity checking, and cloud revocation, backward and forward secrecy.

1.1 MOTIVATION

In public cloud, most clients upload their data to PCS and check their remote data's integrity by Internet. When the client is an individual manager, some practical problems will happen. If the manager is suspected of being involved into the commercial fraud, he will be taken away by the police. During the period of investigation, the manager will be restricted to access the network in order to guard against collusion. But, the manager's legal business will go on during the period of investigation. When a large of data is generated, who can help him process these data? If these data cannot be processed just in time, because of this the manager will face the loss of economic interest. In order to prevent the case happening, the manager has to delegate the proxy to process its data, for example, his secretary. But, the manager will not hope others have the ability to perform the remote data integrity checking. Public checking will incur some danger of leaking the privacy. For example, the stored data volume can be detected by the malicious verifiers. When the uploaded data volume is confidential, private remote data integrity checking is necessary. Although the secretary has the ability to process and upload the data for the manager, he still cannot check the manager's remote data integrity unless he is delegated by the manager. We call the secretary as the proxy of the manager.

In PKI (public key infrastructure), remote data integrity checking protocol will perform the certificate management. When the manager delegates some entities to perform the remote data integrity checking, it will incur considerable overheads since the verifier will check the certificate when it checks the remote data integrity. In PKI, the considerable overheads come from the heavy certificate verification, certificates generation, delivery, revocation, renewals, etc. In public cloud computing, the end devices may have low computation capacity, such as mobile phone, iPad, tablet etc. Revocable ID-based public key cryptography can eliminate the complicated certificate management. In order to increase the efficiency, revocable identity-based proxy-oriented data uploading and remote data integrity checking is more attractive. Thus, it will be very necessary to study the RID-PREUIC protocol.

The concept of revocable identity-based proxy re-encryption (RIPRE) might be a promising approach that fulfills the aforementioned security requirements for data sharing. RIPRE features a mechanism that enables a client to append the current time period to the ciphertext such that the receiver can decrypt the ciphertext only under the condition that he/she is not revoked at that time period. The client first decides the users who can share the data. Then, client will encrypts the data under the identities of end-user, and uploads the ciphertext of the shared data to the cloud server. When end-user wants to get the shared data, she or he can download and decrypt the corresponding ciphertext. However, for an unauthorized user and the cloud server, the plaintext of the shared data is not available. Such a data sharing system can provide confidentiality and backward secrecy. Furthermore, the method of decrypting and re-encrypting all the shared data

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 5, May 2017

can ensure forward secrecy.

1.2 RELATED WORK

There exist many different security problems in the cloud Computing. Proxy cryptography is a very important cryptography primitive. A new type of digital proxy signature is proposed, that allows a designated person, called a proxy signer, to sign on behalf of an original signer. When the bilinear pairings are brought into the identity-based cryptography, identity-based cryptography becomes efficient and practical. Since identity-based cryptography becomes more efficient because it avoids of the certificate management. An ID-based proxy signature scheme with message recovery is proposed. And proposed a proxy signature scheme and a threshold proxy signature scheme from the Weil pairing. By combining the proxy cryptography with encryption technique, some proxy re-encryption schemes are proposed. Also presented a non-interactive CPA (chosen-plaintext attack)-secure proxy re-encryption scheme, which is resistant to collusion attacks in forging re-encryption keys. Further proposed provable data possession (PDP) paradigm. In PDP model, the checker can check the remote data integrity without retrieving or downloading the whole data. PDP is a probabilistic proof of remote data integrity checking by sampling random set of blocks from the public cloud server, which drastically reduces I/O costs. The checker can perform the remote data integrity checking by maintaining small metadata. After that, some dynamic PDP model and protocols are designed. Later, proposed a proof of retrieve ability (POR) scheme. POR is a stronger model which makes the checker not only checks the remote data integrity but also retrieve the remote data.

By a cloud-aided service provider, an outsourcing computation technique is introduced into IBE to propose a revocable IBE scheme with a key-update cloud service provider (KU-CSP). They shifts the key-update procedures to a KU-CSP to alleviate the load of PKG. Also used the similar technique of revocable IBE, which partitions a user's private key into an identity key and a time update key. The PKG sends a user the corresponding identity key via a secure channel. Meanwhile, the PKG must generate a random secret value (time key) for each user and send it to the KU-CSP. Then the KUCSP generates the current time update key of a user by using the associated time key and sends it to the user via a public channel. To revoke a user, the PKG just asks the KU-CSP to stop issuing the new time update key of the user. IBE eliminates the need for providing a public key infrastructure (PKI). Regardless of the setting of IBE or PKI, there must be an approach to revoke users from the system when necessary, e.g., the authority of some user is expired or the secret key of some user is disclosed. Boneh and Franklin proposed a natural revocation way for IBE. They appended the current time period to the ciphertext, and non-revoked users periodically received private keys for each time period from the key authority. Unfortunately, such a solution is not scalable, since it requires the key authority to perform linear work in the number of non-revoked users. In addition, a secure channel is essential for the key authority and non-revoked users to transmit new keys. To conquer this problem, Boldyreva, Goyal and Kumar introduced a novel approach to achieve efficient revocation. They used a binary tree to manage identity such that their RIBE scheme reduces the complexity of key revocation to logarithmic (instead of linear) in the maximum number of system users. However, this scheme only achieves selective security. Subsequently, by using the aforementioned revocation technique, Libert and Vergnaud proposed an adaptively secure RIBE scheme based on a variant of Water's IBE scheme, Chen et al. constructed a RIBE scheme from lattices.

Seo and Emura proposed an efficient RIBE scheme resistant to a realistic threat called decryption key exposure, which means that the disclosure of decryption key for current time period has no effect on the security of decryption keys for other time periods. Inspired by the above work and, Liang et al. introduced a cloud-based revocable identity-based proxy re-encryption that supports user revocation and ciphertext update. To reduce the complexity of revocation, they utilized a broadcast encryption scheme to encrypt the ciphertext of the update key, which is independent of users, such that only non-revoked users can decrypt the update key.

1.3 PROPOSED SYSTEM

We proposed the novel security concept of RID-PREUIC in public cloud. We formalize RID-PREUIC's system model and security model. Then, the first concrete RID-PREUIC protocol is designed by using the bilinear pairings technique. The concrete ID-PUIC protocol is provably secure and efficient by using the formal security proof and efficiency analysis. On the other hand, the proposed RID-PREUIC protocol can also realize private remote data integrity checking, delegated remote data integrity checking and public remote data integrity checking based on the original client's authorization. RID-PREUIC enables a client to append the current time period to the

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 5, May 2017

ciphertext such that the receiver can decrypt the ciphertext only under the condition that he/she is not revoked at that time period. The client first decides the users who can share the data. Then, client will encrypts the data under the identities of end-user, and uploads the ciphertext of the shared data to the cloud server. When end-user wants to get the shared data, she or he can download and decrypt the corresponding ciphertext. However, for an unauthorized user and the cloud server, the plaintext of the shared data is not available. Such a data sharing system can provide confidentiality and backward secrecy.

1.4 OBJECTIVES

Revocable Identity-based proxy-re-encryption and data uploading with remote data integrity checking. By using identity-based public key cryptology, our proposed RID-PREUIC protocol is efficient since the certificate management is eliminated. RID-PREUIC is a novel proxy-oriented data uploading and remote data integrity checking model in public cloud. The formal system model and security model for RID-PREUIC protocol. Based on the bilinear pairings, we designed the first concrete RID-PREUIC protocol. Our protocol can realize private checking, delegated checking and public checking. The client first decides the users who can share the data by using revocable proxy re-encryption technique. Then, client will encrypts the data under the identities of end-user, and uploads the ciphertext of the shared data to the cloud server. When end-user wants to get the shared data, she or he can download and decrypt the corresponding ciphertext. However, for an unauthorized user and the cloud server, the plaintext of the shared data is not available. Such a data sharing system can provide confidentiality and backward secrecy. Once the user is revoked by the PCS, he can't able to access the stored file.

1.5 PAPER ORGANIZATION

The paper is organized as below. The system model and secure model of RID-PREUIC protocol are given in Section 2. The proposed RID-PREUIC protocol and its implementation are presented in 3. The performance analysis is given in section 4. Section 5 analyzes the proposed RID-PREUIC protocol's security. The proposed protocol is provably secure and more efficient. At the end of the paper, the conclusion is given in Section 6.

2 SYSTEM MODEL AND SECURITY MODEL OF RID-PREUIC

The system model and security model of RID-PREUIC protocol is shown below. Fig 2.1 shows system architecture of revocable identity-based proxy re-encryption and data uploading with remote data integrity checking.

An RID-PREUIC protocol consists of five different entities which are described below:

i . Original Client: An entity, which has massive data to be uploaded to PCS by the delegated proxy whenever he is not available, it can also perform the remote data integrity checking. Before sending any data to proxy, client will encrypt and send to the proxy.

ii. Public Cloud Server: PCS is an entity which has significant storage space and computation resource to maintain the clients' data. PCS is managed by cloud service provider. PCS can manage the cloud servers that have a large storage space available for any clients wants to store their data.

iii. Proxy: An entity, which is authorized to process the Original Client's data and upload them to PCS, is selected and authorized by Original Client. When Proxy satisfies the warrant which is signed and issued by Original- Client, it can process and upload the original client's data; otherwise, it cannot perform the procedure. Proxy will receive encrypted data from client, then it will re-encrypt and move to the PCS.

iv. KGC (Key Generation Centre): An entity, when receiving an identity, it generates the private key which corresponds to the received identity. KGC is the one who has expertise and capabilities that client/proxy may not have. We are using three different key authorities, we can't able to predict from which authority we are getting the key.

v. End-User: An entity, which requests PCS for stored data. After authorized by the PCS, and after getting download privilege from the client, end-user will download the data. End-User can retrieve the data from the cloud and he is under supervision of the client/proxy.

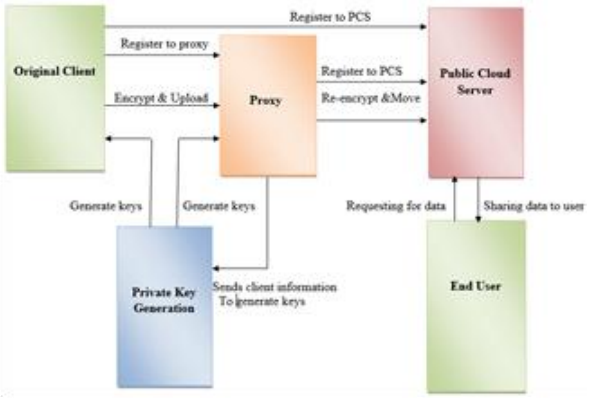

Fig 2.1 : RID-PREUIC protocol Architecture

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 5, May 2017

3. RID-PREUIC PROPOSED PROTOCOL

3.1 PRELIMINARIES

We propose an efficient RID-PUIC protocol for more secure data uploading and storage service in public clouds. Bilinear pairings technique makes identity-based cryptography. Our proposed RID-PREUIC protocol is built on the bilinear pairings. We first show the bilinear pairings. Then, the concrete RID-PREUIC protocol is designed from the bilinear pairings. At last, based on the computation cost and communication cost, we give the performance analysis from two aspects: theoretical analysis and prototype implementation. Also we are going to analyze the performance in terms of time complexity.

i. Bilinearity
ii. Non-degeneracy
iii. Computability

2.1.1 BILINEAR PAIRINGS

First we define several notations of bilinear pairings as follows:

- G is an additive cyclic group of a prime order q.
- GT is a multiplicative cyclic group of the same prime order q.
- P is a generator of G.

We say that eˆ: G * G ->GT is a bilinear map if it satisfies the following three properties:
(i) Bilinearity: for all g1, g2 ϵ G and a; b ϵ Zq*, we have eˆ(ag1, bg2) = eˆ(g1,g2)^{ab}.
(ii) Non-degeneracy: eˆ(P, P) generates GT .
(iii) Computability: For practical purposes, eˆ has to be computable in an efficient manner.
Note that an bilinear map eˆ is symmetric since,
eˆ(aP , bP) = eˆ(P, P)^{ab} = eˆ(bP , aP).

2.1.2 COMPLEXIY ASSUMPTION

The concrete bilinear pairings e can be constructed by using the modified Weil pairing or Tate pairings on elliptic curves. Our RID-PREUIC protocol construction takes use of the easiness of DDH (Decisional Diffie-Hellman) problem while its security is based on the hardness of CDH (Computational
Diffie-Hellman) problem. Let g be a generator of G1. CDH problem and DDH problem are defined below.
CDH Problem on G: Given P, aP, bP ϵ G3 with unknown a , b ϵ Zq*, compute gab ∈ G.
DDH Problem on G: Given the quadruple, (g, ga, gb, gˆ) ∈ G4, with unknown a, b ∈ Zq*, verify whether or not the formula gab = gˆ holds.
We choose the group G1 which satisfies the condition, CDH problem is difficult but DDH problem is easy. On the group G1, DDH problem is easy by using the bilinear pairings. (G1, G2) are also called GDH (Gap Diffie-Hellman) groups. On the groups G1 and G2, the basic requirement is that the DLP (Discrete Logarithm Problem) is difficult. Let g2 be a generator of G2. It is given below:
DLP on G1: Given (g, ga) where a ∈ Zq* is unknown, it is difficult to calculate a.
DLP on G2: Given (g2, g2a) with unknown a ∈ Zq*, it is difficult to calculate a. We choose the groups G1 and G2 that satisfy that DLP, CDH are difficult while DDH is easy.

3.2 RID-PREUIC PROTOCOL DESCRIPTION

We present the system operations of the proposed RID-PREUIC protocol. Our system has six roles, namely, Setup, private key generator (PKG), proxy Re-encryption, cloud revocation, users (client, proxy and end-user) and proof.

i. SETUP

The organization manager (client/proxy) initializes the public and secret parameters of the system by executing KEYGEN algorithm. In this step entity identity is input to PKG, then PKG generates entity's secret key. Especially it will generate the secret key for client and proxy. The client delegates the data to proxy, proxy will move the client data to CSP to stores the data at the cloud server. As part of preprocessing, the user may alter the data files by expanding it or including additional metadata to be stored at server.

ii. Private key Generator (PKG)

In the Proxy-key generation phase, the original client creates the warrant and helps the Proxy to get the proxy key from key authority. In this paper, we user three random key authority's, so, we can't predict from which authority we are getting the key. Before generating secret key to the user, key authority will validate the user identity. PKG sends the secret key to client/proxy by secure channel. In order to generate proxy key, the original client will interact with proxy ID. After verifying the proxy ID, PKG will generate secret key to the proxy.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 5, May 2017

iii. Proxy Re-encryption

In the proxy Re-encryption phase, proxy will receive the encrypted data from client and it will get the secret key from key authority, later re-encrypt the encrypted client data and move the re-encrypted data to PCS for storage. Later whenever client wants to access the data, he can get it from PCS by using proxy key and his identity.

iv. Cloud Revocation

In the cloud revocation phase, whenever user wants to access the data from CSP, they will request the secret key from the respected client, after that users will download the data. Before downloading the data users must get the privilege from the client. But some malicious users try to get the data by using wrong key, that time CSP will treat them as attacker and it will block the malicious users from accessing the data.

v. Users

In this phase, three different types of users are involved, like client, proxy and end-users. Client will encrypt the data, select the delegated proxy and upload the data to proxy, later proxy will re-encrypt the encrypted data by using secret key and move to CSP. Finally, whenever end-user wants to access the data from CSP, they will use their identity and get key from respected clients and download the data.

vi. Proof

In the Proof phase, the original client interacts with PCS. Through the interaction, client checks its remote data integrity. It is a two-step interaction procedure between client and PCS. If client authorizes the remote data integrity checking to some user, it sends to authorized user. The authorized user may be third party like proxy. Since, client can play a role as the verifier. Client also verify the valid end-user in order to give privilege for downloading.

4. PERFORMANCE ANALYSIS

We present the performance of our proposed RID-PREUIC protocol in terms of computation and communication overhead. At the same time, we implement the prototype of our RID-PREUIC protocol and evaluate its time cost. Then, we give the flexibility of remote data integrity checking in the phase Proof of our RID-PREUIC protocol. Finally, we compare our RID-PREUIC protocol with the other up-to-date remote data integrity checking protocols.

4.1 Computation cost: We calculate the computation overhead of Proposed RID-PREUIC protocol based on different phases. For the proxy, the computation overhead mainly comes from the phase Proxy re-encryption. In the proof phase, the original client interacts with PCS. Through the interaction, client checks its remote data integrity. In order to show our proposed protocol practical computation overhead, we have simulated the proposed RID-PREUIC protocol by using java programming language with 3.2 GHz CPU, GB RAM, physical memory of 5GB. In proxy re-encryption, when small file is re-encrypted, proxy's time cost is 0.102572s. When big size file is re-encrypted, proxy's time cost is 25.560061s. Proxy's time cost increases almost linearly with the increase in the size of the file. In Proof, the original client interacts with PCS and performs the remote data integrity checking. In Proof, when the warrant period is less, the original client's time cost is 0.567881s. When the warrant period is more, the original client's time cost is 3.342359s.

Protocols	Proxy Re-encryption	Proof cost of PCS	Proxy Data processing & Uploading	Integrity Checking	Certificate Management	Revocation
PPDP, Wang [1]	3.254301s	4.254612s	No	No	Required	No
POR, Zhang [2]	2.914250s	3.25412s	No	No	Required	No
RID-PREUIC	0.102572s	0.567881s	Yes	Yes	Not Required	Yes

Table 3.1: Comparison of Computation cost & Security parameters

We know that most computation will be performed by PCS. The other entities only performs a little computation. Our RID-PREUIC protocol can be used in the environment where the computation resource is limited, for example mobile phone.

4.2 Communication Overhead: We analyze our RID-PREUIC protocol's communication cost based on shortest key length of RSA algorithm i.e 1024 bits. After the data processing, the data is uploaded to PCS. Thus, we only consider the communication cost which is incurred in the remote data integrity checking. In Proof, the communication cost gives the request and response. In proof phase, the original client will interact with PCS periodically.

4.3 Protocol Comparison: In order to show our RID-PREUIC protocol's performance, we compare our protocol with the two recent protocols, i.e., Wang's protocol [1] and

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol4, Issue 5, May 2017

Zhang et al.'s protocol [2]. The simplified comparison of protocol is given in Table 3.1. From the computation comparison, we know that our protocol has very less computation cost in the phases proxy re-encryption and PCS also has the less computation cost in the phase Proof. On the other hand, our protocol can realize the three security properties: proxy data processing and uploading, remote data integrity checking with flexibility and not required certificate management and revocation feature. Flexibility means that our proposed protocol can realize the private checking, delegated checking and public checking according to the original client's authorization.

5. SECURITY ANALYSIS

We prove the security of our RID-PREUIC protocol in terms of correctness, proxy-protection and unforgeability. Proxy-protection means that the original client cannot pass himself off as the proxy to re-encrypt and move the re-encrypted data to PCS. Unforgeability means that when some malicious users are blocked by the PCS, then users cannot access the valid response which can pass the integrity checking. In order to show the security of proposed protocol, the basic information of documents are inevitably leaked to the honest-but-curious cloud server since all the data are stored at the server.

5.1 Correctness: All the documents returned from servers are originally uploaded by the client and remain unmodified. Whenever client is not available, proxy will move the data to PCS. We can verify the correctness of our proposed system by making use of user validation phase, key generation phase and proof phase.

5.2 Proxy protection: Our proposed protocol is proxy protective based on the difficulty of CDH problem. Consider the identity of proxy IDp and no of data files to be uploaded to PCS to be Fn. Where Fi ϵ Zp*. Suppose the original client delegate the data along with warrant to proxy, proxy will re-encrypt the data and get the keys from key authority and move to CSP if it satisfies the warrant. In this case we need to give security to proxy as well as clients identity along with the proxy/clients data.

We evaluate the success probability of C1's (request/response). In order to break CDH problem, the following 3 condition must hold simultaneously:

1) E1: C1 does not abort in A1's Proxy-key Generation query. Where A1's is the adversary of client.
2) E2: C1 does not abort in A1's keyGen & re-encryption.
3) E3: A1 generates a valid block-key forgery (Ff , Tf) which satisfies c f = 0.

C1 succeeds if the above three conditions hold. We calculate the following probability:

$$\Pr[E1 \wedge E2 \wedge E3] = \Pr[\epsilon 1] \Pr[E\epsilon 2|E\epsilon 1] \Pr[E\epsilon 3|E\epsilon 1 \wedge E\epsilon 2] = 1/n(qT/\ qT + 1\ )qT\ \ (1/qT + 1)\epsilon$$

5.3 Unforgeability: The proposed RID-PREUIC protocol is existentially unforgeable in the random oracle model if CDH problem on G1 is hard. The proof process is almost the same as Shacham-Waters's protocol [3]. In the phase blockgen & keygen, the proxy-key Pk is used in RID-PREUIC protocol while the data owner's secret key Sk is used in Shacham-Waters's protocol [3]. For PCS, Pk and Sk has the same function to generate the block keys. When PCS is dishonest, since Shacham-Waters's protocol is existentially unforgeable in random oracle model, our proposed RID-PREUIC protocol is also existentially unforgeable in the random oracle model.

Let the uploaded block-key pairs number be n. When d. block-key pairs are broken and c block-tag pairs are requested to upload, the broken block-key pairs can be found out with probability at least 1-(n−d / n)c and at most

1-(n−c+1-d / n−c+1)c. Thus, our RID-PREUIC protocol is (d / n , 1-(n-d / n)c)- is secure.

Combined with the performance analysis, when the stored block-key pairs number be 1000 and d. = 50, c = 7, the probability is 97.35%, the original client's time cost is 1.901102s and PCS's time cost is 6.251231s. When n = 1000, d. = 50, c = 5, the probability is 92.41%, the original client's time cost is 1.685642s and PCS's time cost is 5.496549s. From our experiments, our RID-PREUIC protocol is more efficient and secure.

6. CONCLUSION

We motivated by the organization application needs, we proposes the novel security concept of revocable ID-based proxy re-encryption with data uploading and remote integrity checking [RID-PREUIC] protocol in public cloud. In which the revocation procedure is performed by the PCS to alleviate the load of the PKG. The paper formalizes RID-PREUIC's system model and security model. Then, the first concrete RID-PREUIC protocol is designed by using the bilinear pairings technique. Based on the original client's authorization, the proposed RID-PREUIC protocol can also realize private remote data integrity checking, delegated remote data integrity checking and public remote data integrity checking. For security analysis, we demonstrated our RID-PREUIC protocol is semantically secure against malicious attacks under the decisional bilinear Decisional Diffie-Hellman and conditional Diffie-Hellman assumption. The concrete

RID-PREUIC protocol is provably secure and efficient by using the formal security proof, probability analysis and efficiency analysis. The performance of RID-PREUIC protocol can be increased by reducing the computation and communication overhead, our proposed scheme is well suited for mobile devices. We also implemented the RID-PREUIC protocol on real cloud.

REFERENCES

[1] H. Wang, "Proxy provable data possession in public clouds," IEEE Trans. Services Comput., vol. 6, no. 4, pp. 551–559, Oct./Dec. 2013.

[2] J. Zhang, W. Tang, and J. Mao, "Efficient public verification proof of retrievability scheme in cloud," Cluster Comput., vol. 17, no. 4, pp. 1401–1411, 2014.

[3] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. ASIACRYPT, vol. 5350. 2008, pp. 90–107.

[4] Z. Fu, X. Sun, Q. Liu, L. Zhou, and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," IEICE Trans. Commun., vol. E98-B, no. 1, pp. 190–200, 2015.

[5] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, "Mutual verifiable provable data auditing in public cloud storage," J. Internet Technol., vol. 16, no. 2, pp. 317–323, 2015.

[6] M. Mambo, K. Usuda, and E. Okamoto, "Proxy signatures for delegating signing operation," in Proc. CCS, 1996, pp. 48–57.

[7] E.-J. Yoon, Y. Choi, and C. Kim, "New ID-based proxy signature scheme with message recovery," in Grid and Pervasive Computing (Lecture Notes in Computer Science), vol. 7861. Berlin, Germany: Springer- Verlag, 2013, pp. 945–951.

[8] B.-C. Chen and H.-T. Yeh, "Secure proxy signature schemes from the weil pairing," J. Super comput., vol. 65, no. 2, pp. 496–506, 2013.

[9] X. Liu, J. Ma, J. Xiong, T. Zhang, and Q. Li, "Personal health records integrity verification using attribute based proxy signature in cloud computing," in Internet and Distributed Computing Systems (Lecture Notes in Computer Science), vol. 8223. Berlin, Germany: Springer- Verlag, 2013, pp. 238–251.

[10] H. Guo, Z. Zhang, and J. Zhang, "Proxy re-encryption with un forgeable re-encryption keys," in Cryptology and Network Security (Lecture Notes in Computer Science), vol. 8813. Berlin, Germany: Springer-Verlag, 2014, pp. 20–33.

[11] E. Kirshanova, "Proxy re-encryption from lattices," in Public-Key Cryptography (Lecture Notes in Computer Science), vol. 8383. Berlin, Germany: Springer-Verlag, 2014, pp. 77–94.

[12] P. Xu, H. Chen, D. Zou, and H. Jin, "Fine-grained and heterogeneous proxy re-encryption for secure cloud storage," Chin. Sci. Bull., vol. 59, no. 32, pp. 4201–4209, 2014.

[13] S. Ohata, Y. Kawai, T. Matsuda, G. Hanaoka, and K. Matsuura, "Re-encryption verifiability: How to detect malicious activities of a proxy in proxy re-encryption," in Proc. CT-RSA Conf., vol. 9048. 2015, pp. 410–428.

[14] G. Ateniese et al., "Provable data possession at un trusted stores," in Proc. CCS, 2007, pp. 598–609.

[15] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, "Scalable and efficient provable data possession," in Proc. Secure Comm, 2008, Art. ID 9.