National Journal of Multidisciplinary Research and Development
ISSN: 2455-9040
Impact Factor: RJIF 5.22
www.nationaljournals.com
Volume 3; Issue 1; January 2018; Page No. 137-139

A review on identity-based proxy-oriented data uploading and remote data integrity checking in public cloud

[1] Mohammed Aameruddin Mohammed Akbaruddin, [2] Shital Y Gaikwad
[1] Computer Science and Engineering, Matoashri Prathistan Group of Institutions Vishnupuri, Nanded, Maharashtra, India
[2] Assistant Professor, Department of Computer Science and Engg. Matoashri Prathistan Group of Institutions Vishnupuri, Nanded, Maharashtra, India

Abstract
More and more clients would like to store their data to public cloud servers (PCSs) along with the rapid development of cloud computing. New security problems have to be solved in order to help more clients process their data in public cloud. When the client is restricted to access PCS, he will delegate its proxy to process his data and upload them. On the other hand, remote data integrity checking is also an important security problem in public cloud storage. It makes the clients check whether their outsourced data are kept intact without downloading the whole data. From the security problems, we propose a novel proxy-oriented data uploading and remote data integrity checking model in identity-based public key cryptography: identity-based proxy-oriented data uploading and remote data integrity checking in public cloud (ID-PUIC). The proposed ID-PUIC protocol is provably secure based on the hardness of computational Diffie–Hellman problem. Our ID-PUIC protocol is also efficient and flexible. Based on the original client's authorization, the proposed ID-PUIC protocol can realize private remote data integrity checking, delegated remote data integrity checking, and public remote data integrity checking.

Keywords: cloud computing, identity-based cryptography, proxy public key cryptography, remote data integrity checking

1. Introduction
Along with the rapid development of computing and communication technique, a great deal of data are gen-erated. These massive data needs more strong computation resource and greater storage space. Over the last years, cloud computing satisfies the application requirements and grows very quickly. Essentially, it takes the data processing as a service, such as storage, computing, data security, etc. By using the public cloud platform, the clients are relieved of the burden for storage management, universal data access with independent geographical locations, etc. Thus, more and more 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. 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 identity-based public cryptography and proxy public key cryptography, we will study ID-PUIC protocol.

2. Related Work
There exist many different security problems in the cloud computing [1]. This paper is based on the research results of proxy cryptography, identity-based public key cryptography and remote data integrity checking in public cloud. In some cases, the cryptographic operation will be delegated to the third party, for example proxy. Thus, we have to use the proxy cryptography. Proxy cryptography is a very important cryptography primitive. In 1996, Mambo *et al*. proposed the notion of the proxy cryptosystem [2]. 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, more and more experts are apt to study identity-based proxy cryptography. In 2013, Yoon *et al*. proposed an ID-based proxy signature scheme with message recovery [3]. Chen *et al*. proposed a proxy signature scheme and a threshold proxy signature scheme from the Weil pairing [4]. By combining the proxy cryptography with encryption technique, some proxy re-encryption schemes are proposed. Liu *et al*. formalize and construct the attribute-based proxy signature [5]. Guo *et al*. presented a non-interactive CPA (chosen-plaintext attack)-secure proxy re-encryption scheme, which is resistant to collusion attacks in forging re-encryption keys [6]. Many other concrete proxy re-encryption schemes and their applications are also proposed [7]. In public cloud, remote data integrity

checking is an important security problem. Since the clients' massive data is outside of their control, the clients' data may be corrupted by the malicious cloud server regardless of intentionally or unintentionally.

In cloud computing, the third party auditing is indispens- able [H]. By using cloud storage, the clients can access the remote data with independent geographical locations. The end devices may be mobile and limited in computation and storage. Thus, efficient and secure ID-PUIC protocol is more suitable for cloud clients equipped with mobile end devices.

An ID-PUIC protocol consists of four different entities which are described below:

- Original Client: an entity, which has massive data to be uploaded to PCS by the delegated proxy, can perform the remote data integrity checking.
- PCS (Public Cloud Server): an entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain the clients' data.
- Proxy: an entity, which is authorized to process the Original Client's data and upload them, is selected and authorized by Original Client. When Proxy satisfies the warrant mω which is signed and issued by Original-Client, it can process and upload the original client's data; otherwise, it cannot perform the procedure.
- KGC (Key Generation Center): an entity, when receiving an identity, it generates the private key which corresponds to the received identity.

From the role of the remote data integrity checker, all the remote data integrity checking protocols are classified into two categories: private remote data integrity checking and public remote data integrity checking. In the response checking phase of private remote data integrity checking, some private information is indispensable. On the contrary, private information is not required in the response checking of public remote data integrity checking. Specially, when the private information is delegated to the third party, the third party can also perform the remote data integrity checking. In this case, it is also called delegated checking.

3. System Model and Security Model of ID-PUIC

In this section, we give the system model and security model of ID-PUIC protocol. An ID-PUIC protocol consists of four different entities which are described below:

1) Original Client: an entity, which has massive data to be uploaded to PCS by the delegated proxy, can perform the remote data integrity checking.
2) PCS (Public Cloud Server): an entity, which is managed by cloud service provider, has significant storage space and computation resource to maintain the clients' data.
3) Proxy: an entity, which is authorized to process the Original Client's data and upload them, is selected and authorized by Original Client. When Proxy satisfies the warrant mω which is signed and issued by Original-Client, it can process and upload the original client's data; otherwise, it can not perform the procedure.
4) KGC (Key Generation Center): an entity, when receiving an identity, it generates the private key which corre- sponds to the received identity.

In our proposed ID-PUIC protocol, Original Client will interact with PCS to check the remote data integrity. Thus, we give the definition of interactive proof system. Then, we give the formal definition and security model of ID- PUIC protocol.

Definition 1 (Interactive Proof System [31]): Let c, s: $N \rightarrow R$ be functions satisfying $c(n)>s(n) + 1$ p(n) for some polynomial $p(\cdot)$. An interactive pair (P,V) is called an interactive proof system for the language L, with completeness bound $c(\cdot)$ and soundness bound $s(\cdot)$, if

1) Completeness: for every $x \in L$, $Pr [< P,V >(x) =1]\geq c(|x|)$.
2) Soundness: for every $x \in L$ and every interactive machine B, $Pr [< B,V >(x)=1]\leq s(|x|)$.

In the definition of ID-PUIC, i.e., Definition 2, we will take use of the interactive proof system.

Definition 2 (ID-PUIC): An ID-PUIC protocol is a collec- tion of four phases (Setup, Extract, Proxy-key Generation, TagGen) and an interactive proof system (Proof). The detailed phases are described below.

1) Setup: When the security parameter k is input, the algorithm outputs the system public parameters and the master secret key. The system public parameters are made public and the master secret key msk is made confidential by KGC.
2) Extract: When the system public parameters, the master secret key msk, and an identity ID are input, KGC outputs the private key skID which corresponds to the identity ID.
3) Proxy-Key Generation: Original Client generates the warrant mω and signs mω. Then, it sends the warrant-signature pair to the proxy. Upon receiving the warrant-signature pair from Original Client, the proxy generates the proxy-key by using its own private key.
4) TagGen: Input the file block Fi and the proxy-key, the proxy generates the corresponding tag Ti. Then, it uploads the block-tag pair to PCS.
5) Proof: It is an interactive proof system between PCS and Original Client. At the end of the interactive proof protocol, Original Client outputs a bit {0|1} denoting "success" or "failure".

A practical ID-PUIC protocol must be efficient and prov- ably secure. Based on the communication and computation overheads, efficiency analysis can be given. On the other hand, a secure ID-PUIC protocol must satisfy the following security requirements:

1) Original Client can perform the ID-PUIC protocol without the local copy of the file(s) to be checked.
2) Only if the proxy is authorized, i.e., it satisfies the warrant mω, the proxy can process the files and upload the block-tag pairs on behalf of Original Client.
3) Original Client cannot counterfeit the proxy to generate block-tag pairs, i.e., the proxy-protection property is satisfied.
4) If some challenged block-tag pairs are modified or lost, PCS's response cannot pass Original Client's integrity checking.

To capture the above security requirements, we formalize the security definition of an ID-PUIC protocol. First, we give the formal definition of proxy-protection.

Definition 3 (Proxy-Protection): An ID-PUIC protocol satisfies the property of proxy-protection if for the probabilistic polynomial time adversary A1, the probability that A1 wins the ID-PUIC game-1 is negligible. The ID-PUIC game-1 between A1 and the challenger C1 is given below:

1) Setup: The challengerC1 runs Setup and gets the system public parameters and master secret key. By running Extract, C1 gets OriginalClient IDo's private key skID o and the proxy IDp's private key skIDp. It sends the public parameters and OriginalClient IDo's private key skID o toA1 while it keeps confidential the master secretkey msk and the proxy IDp's private key skIDp.
2) Oracle queries: A1 adaptively queries the oracles Extract, Hash, Proxy-key Generation, TagGen to C1 below:
   - Extract queries. A1 queries the entity ID's private key to C1. For the identity ID, C1 runs Extract and gets the private key skID. Then, it forwards skID to A1. Denote S as the queried identity set in the phase. The restriction is that IDp cannot be queried in the phase, i.e., IDp ∈ S.
   - Hash queries. A1 queries hash oracle to C1 adaptively. C1 responds A1 the hash values.
   - Proxy-key Generation queries. A1 sends (ID o, ID p) to C1 and queries the proxy-key where the original client is ID o and the proxy is ID p. Denote S as the set which is composed of all the queried original client identity and proxy identity pairs. The restriction is that (ID o, IDp)∈ S.
   - TagGen queries. A1 makes block-tag pair queries adaptively. For the block Fi, C1 computes its tag Ti and responds A1 with Ti.

At the end of game-1, A1 outputs the forged block-tag pair (F, T) with non-negligible probability, where F has not been queried to TagGen oracle. If (F, T) is valid block-tag pair, then A1 wins the above game with non-negligible probability.
The security definition 3 gives the formal security defini- tion for proxy-protection property of ID-PUIC protocol. Let OriginalClient be the adversary. If OriginalClient cannot win the ID-PUIC game-1 with non- negligible probability, then the ID-PUIC protocol satisfies the proxy-protection property.

4. Conclusion

This paper proposes the novel security concept of ID-PUIC in public cloud. The paper formalizes ID-PUIC's system model and security model. Then, the first concrete ID-PUIC 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 ID-PUIC 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.
In this paper its followed Two layer security i.e. at Original Clients to Pkg & Proxy to Pkg. We have trying to provide three layer security. We have added User instead of Original Clients & Transaction Manager instead of Proxy, pkg & Admin instead of PCS. In above paper it is provided Original Client can also get & upload file to the PCS but I am doing little bit change in that User can only get or download files because in public cloud End user can only give request for the file or data and get files or data as per the request granted by Transaction Manager and Transaction Manager should give the permission to access file. So I think it's good as per previous one.

5. References

1. Ren Y, Shen J, Wang J, Han J, Lee S. Mutual verifiable provable data auditing in public cloud storage, J Internet Technol. 2015; 16(2):317-323.
2. Mambo M, Usuda K, Okamoto E. Proxy signatures for delegating signing operation, in Proc. CCS, 1996, 48-57.
3. Yoon EJ, Choi Y, Kim C. New ID-based proxy signature scheme with message recovery, in Grid and Pervasive Computing Lecture Notes in Computer Science, Berlin, Germany: Springer- Verlag, 2013; 7861:945-951.
4. Chen BC, Yeh HT. Secure proxy signature schemes from the weil pairing, J. Supercomput. 2013; 65(2):496-506.
5. Liu X, Ma J, Xiong J, Zhang T, Li Q. Personal health records integrity verification using attribute based proxy signature in cloud computing, in Internet and Distributed Computing Systems Lecture Notes in Computer Science, Berlin, Germany: Springer- Verlag, 2013; 8223:238-251.
6. Guo H, Zhang Z, Zhang J. Proxy re-encryption with unforgeable re-encryption keys, in Cryptology and Network Security Lecture Notes in Computer Science, Berlin, Germany: Springer-Verlag, 2014; 8813:20-33.
7. Xu P, Chen H, Zou D, Jin H. Fine-grained and heterogeneous proxy re-encryption for secure cloud storage, Chin. Sci. Bull., 2014; 59(32):4201-4209.
8. Wang C, Wang Q, Ren K, Lou W. Privacy-preserving public auditing for data storage security in cloud computing, in Proc. IEEE INFOCOM, 2010, 1-9.