International Journal of Electronic and Electrical Engineering.
ISSN 0974-2174, Volume 7, Number 1 (2014), pp. 7-12

http://www.irphouse.com

Data Integrity for Secure Dynamic Cloud Storage System Using TPA

Aishath Sania[1], Asst. Prof. Mohammed Hussain K[2], Dr. Jose Alex Mathew[3]

[1,2,3]P.A. College of Engineering, Near Mangalore University, Nadupadav, Mangalore–574153, Karnataka, India
E-mail: [1]claansania@gmail.com, [2]mdhk10@yahoo.co.in, [3]aymanamkuzhy@gmail.com

Abstract

Using dynamic cloud storage system, users can remotely store their data and can enjoy on-demand high quality applications and services from shared pool of configurable computing services, without the burden of local data storage and maintenance. As clients no longer possess their data locally the clients should be equipped with certain security means so that they can periodically verify the correctness of the remote data even without the existence of local copies. TPA only has the authority to check the integrity of client's data without revealing the original data and cloud server saves encrypted data and users passwords while client can modify, delete and download their data.

Keywords: Data Integrity, Cloud Storage, TPA.

1. Introduction

Cloud Computing has been envisioned as the next-generation information technology (IT) architecture for enterprises, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced to the Cloud. From users' perspective, including both individuals and IT enterprises, storing data remotely to the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access

with independent geographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances.

2. Literature Survey

Recently, much of growing interest has been pursued in the context of remotely stored data verification [2]-[4]. Ateniese et al. are the first to consider public auditability in their defined "provable data possession" (PDP) model for ensuring possession of files on untrusted storages. In their scheme, they utilize RSA based homomorphic tags for auditing outsourced data, thus public auditability is achieved. However, Ateniese et al. do not consider the case of dynamic data storage, and the direct extension of their scheme from static data storage to dynamic case may suffer design and security problems. In their subsequent work, Ateniese et al. propose a dynamic version of the prior PDP scheme. However, the system imposes a priori bound on the number of queries and does not support fully dynamic data operations, i.e., it only allows very basic block operations with limited functionality, and block insertions cannot be supported.

3. Proposed system

One of the important concerns that need to be addressed is to assure the customer of the integrity i.e. correctness of his data in the cloud. As the data is physically not accessible to the user the cloud should provide a way for the user to check if the integrity of his data is maintained or is compromised.TPA verifies the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users. Storage correctness to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact. Privacy preserving ensure that the TPA cannot derive users data content from the information collected during the auditing process. Here we provide a scheme which checks data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud. This can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA). It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted.

4. System Architecture

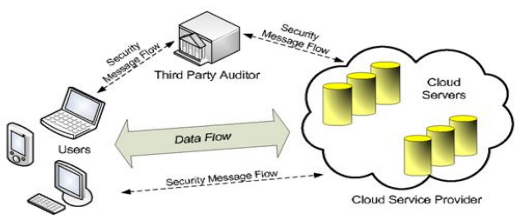

Fig. 1: System Architecture.

A representative architecture for cloud data storage is illustrated in Fig. 1 Three different network entities can be identified as follows:

Client: an entity, which has large data files to be stored in the cloud and relies on the cloud for data maintenance and computation, can be either individual consumers or organizations;

Cloud Storage Server (CSS): an entity, which is managed by Cloud Service Provider (CSP), has significant storage space and computation resource to maintain the clients' data.

Third Party Auditor (TPA): an entity, which has expertise and capabilities that clients do not have, is trusted to assess and expose risk of cloud storage services on behalf of the clients upon request.

Algorithm: Proposed scheme consists of four algorithms:
Key Generation Algorithm: RSA algorithm is used here for generating the keys for client and server those are public key and private key respectively. Encryption and decryption are both methods used to ensure the secure passing of messages and other sensitive documents and information. It generate Public Key (n, e) and Private Key d, Encryption with key (n, e) and Decryption with key d.

Signature Generation Algorithm: OPEN SSL is used to generate 'Signature' (from server side). It is an open-source implementation of the SSL and TLS protocols.

Hash code generation algorithm: MD5 is used to generate Hash code in client and then it'll send to TPA(while uploading it'll generate hash code & send to TPA).

Password Encryption Algorithm: Base64 is used for encrypting the password of clients (while client is registering the password he entered will be encrypted & that encrypted password will store in database), while client is logging in then it'll decrypt the password and client will log in.

5. Cloud User Options in proposed system

The proposed system is developed using PHP software and MySQL database. Following options are available to the user:

Check Integrity: Client while storing encrypted data files in cloud server; the hash code of that data file is generated and stored in TPA. When user request for check integrity, digital signature for the sored data file is generated. Here our check integrity just checks the integrity of the data i.e. the data has been modified or deleted by the inside or outside attacker.

Initially the cloud client send the request for check integrity to TPA for the data file d(), TPA will forwards that request to the Server, Server will further fetches the respected data file d() from the Server database and generates the hash code for that file i.e. h(f), and that will be send to the TPA with file name, TPA will generates the signature i.e. SignGen() from the hash code sent by the Server, TPA will further fetches the old signature from the TPA database i.e. SigAvil(), finally TPA will does the equality check between the SignGen() and SignAvail(), Ack will be sent to the Client depend upon the equality checking. If the two compared signature matches,

message" the data d() is trusted" else "the data d() is untrusted" will be send to the client.

*Dynamic data storage assurance with check integrity:*Now we show how our scheme can explicitly and efficiently handle fully dynamic data operations like Data Updating including data edit (E) and data deletion (D) for cloud data storage. Note that in the following descriptions, we assume that the data file F and the signature Sign() have already been generated and properly stored at server. Initially client sends modify request to Server for data file d(), Server will fetches that respected data file d() and allows client to modify or update his file, after modification by client Server will update the file and generates the hash code for the updated file and sends it to TPA with file name. TPA will generates Signature for the hash code which is sent by the Server and finally TPA update the Signature in its database for future references.

*Download option for user:*Initially the cloud client send the request for download option to TPA for the data file d(), TPA will forwards that request to the Server, Server will further fetches the respected data file d() from the Server database. Server checks whether the user is genuine and the data file d() is download.

6. Result

Snapshots of the proposed system are given in figure 2, 3, 4 and 5.

Fig. 2: Cloud User Login Page

Fig. 3: User Registration Page

Fig. 4: Client option with cloud server

Fig. 5: Client check integrity option on demand

7. Conclusion

Here we have presented a dynamic cloud storage system model for secure data integrity and client can also modify, delete and download their data by using trusted Third Party Auditor. Here we addressed mainly two issues data integrity and public Auditability. Public Auditability ensures that original stored data in the cloud is not revealed to TPA during auditing process and also allows TPA to check integrity on demand without retrieving a copy of whole data with no additional online burden to cloud users. Data correctness to ensure the correctness of data stored in the cloud that is there exist no cheating in the cloud server, even the password of the cloud user is not known to the cloud server. Here we provide a scheme which checks data integrity in the cloud which the customer can employ to check the correctness of his data in the cloud.This can be agreed upon by both the cloud and the customer and can be incorporated in the Service level agreement (SLA).It is important to note that our data integrity scheme just checks the integrity of data that is, if the data has been illegally modified or deleted. Here we have done for only text data file.

References

[1] CongWang ;Chow,S.S.M. ; QianWang ; KuiRen ;WenjingLou "Privacy_preserving Public Auditing for Secure CloudStorage", IEEE Transactions on Computers Volume: 62, Issue: 2 2013 ,PP no : 362–375

[2] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Storage Security in Cloud Computing," Proc. IEEE INFOCOM '10, Mar. 2010.

[3] P. Mell and T. Grance, "Draft NIST Working Definition of Cloud Computing," http://csrc.nist.gov/groups/SNS/cloudcomputing/index.html, June 2009.

[4] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing," Technical Report UCB-EECS-2009-28, Univ. of California, Berkeley, Feb. 2009.