Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

# Cloud Data Storage Services Considering Public Audit for Security

Swathi Sambangi

*Received: 13 April 2013 Accepted: 4 May 2013 Published: 15 May 2013*

## Abstract

The cloud computing is a computing technology that allows us to share the pool of configurable sources where the data of individuals or the organisations can be stored remotely those can be accessible on-demand high quality applications. Even though there are many advantages associated with cloud computing still it brings new challenges in terms of security provided for the data storage providers as sensitive information of individuals is stored in these cloud storage providers. The owners of data expect a cloud data storage provider to be ensured with high servicelevel requirements. To ensure the deployment of cloud data storage service with security levels, some efficient methods has to be designed for the verification of the correctness of data. This paper proposes architecture for cloud computing that has a trusted entity with expertise and capability to assess cloud storage security in assistance of data owner request. The main aim of this paper is to enable public risk auditing protocols with which data owners can gain trust in cloud.

***Index terms—***

## 1 Introduction

he cloud computing provides many of the services for IT enterprise like on demand self service, location independent resource pooling, ubiquitous network access, usage-based pricing, rapid resource elasticity and transference of risk. The primary concept of cloud computing is to provide a centralised data base or an outsourcing data from different individuals by using a cloud storage device.

As cloud computing allows individuals to store the data remotely, it has capability of enabling end users to use the resources provided in a rigid way. The advantage of cloud computing is it provides on demand self service methodology that authorizes users to request resources dynamically. Data owners store data on a cloud computing storage provider remotely and they can not directly use traditional cryptographic algorithm to ensure security for data. And downloading data for integrity verification costs high and even large data transmission through network frequently may support customers economically. Once if the data has been stored on cloud computing data storage provider data owner should not worry about security of data. In order to assure security for data and to enable data owner to use data without any worry about security for data, in this paper we propose publicly auditable cloud storage providers where data owners can rely on third party auditor to verify the data integrity of out sourced data to ensure security.

Representative network 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.

# 2 II.

# 3 Motivation

Cloud data storage can be affected by two different sources. The cloud data storage provider itself is untrusted and possibly malicious.

There are cases in corrupting the data that is stored by users on individual servers. An adversary can compromise an individual server pollute the original data files by modifying or even by introducing its own fraudulent data to prevent the original data from being retrieved by the user.

There is another case where all the servers can be compromised by an attacker so that they can modify the data in files. There are many adversaries to be considered because both malicious outsiders and semitrusted cloud storage service providers can be interrupted as the malicious outsiders can economically motivate by some others for their own benefit. Even though the cloud service providers can be some times semi trusted and most of the times they does not deviate from the executing a prescribed protocols as far as the security is being concerned, but still they might neglect to keep the files which are accessed frequently, they may cause the data corruption and even they may fail to execute the Byzantine protocols (which are meant to ensure security services.

The data owners could no longer physically possess the storage for their data; They cannot directly adopt cryptographic primitives to ensure the data security because downloading data every time for the purpose of integrity verification is not a feasible solution to be followed by the data owners as it is economically cannot be afforded by them. On the other side, detecting data corruption only when accessing data does not give assurance of the correctness of data, if the data size is too large. To overcome all of these challenges there is a need of third party auditing to ensure data security completely and save data of owners.

# 4 III.

# 5 Third Party Auditing

In Systematic point of view the auditing for cloud data storage should be real and the whole service architecture design should not be not only cryptographically strong. Below are the auditing concepts for Cloud data storage.

# 6 a) Support Batch Auditing

The prevalence of large-scale cloud storage service further demands auditing efficiency. When receiving multiple auditing tasks from different owners' delegations, a TPA should still be able to handle them in a fast yet cost-effective fashion. This property could essentially enable the scalability of a public auditing service even under a storage cloud with a large number of data owners.

# 7 b) Minimize Auditing Overhead

First and foremost, the overhead imposed on the cloud server by the auditing process just not outweigh its benefits. Such overhead may include both the I/O cost for data access and the bandwidth cost for data transfer. Any extra online burden on a data owner should also be as low as possible. Ideally, after auditing delegation, the data owner should just enjoy the cloud storage service while being worry-free about storage auditing correctness.

# 8 c) Support Data Dynamics

As a cloud storage service is not just a data warehouse, owners are subject to dynamically updating their data via various application purposes. The design of auditing protocol should incorporate this important feature of data dynamics in Cloud Computing.

# 9 d) Protect Data Privacy

Data privacy protection has always been an important aspect of a service level agreement for cloud storage services. Thus, the implementation of a public auditing protocol should not violate the owner's data privacy. In other words a TPA should be able to efficiently audit the cloud data storage without demanding a local copy of data or even learning the data content.

# 10 a) Utilizing Homomorphic Authenticators

To significantly reduce the arbitrarily large communication overhead for public auditability without introducing any online burden on the data owner, we resort to the homomorphic authenticator technique. Homomorphic authenticators are unforgettable metadata generated from individual data blocks, which can be securely aggregated in such a way to assure a verifier that a linear combination of data blocks is correctly computed by verifying only the aggregated authenticator.

Using this technique requires additional information encoded along with the data before outsourcing. Specifically, a data file is divided into n blocks mi (i =1, ?, n), and each block mi has a corresponding homomorphic authenticator ?i computed as its metadata to ensure the integrity. Every time it must be verified that the cloud server is honestly storing the data, the data owner or TPA can submit challenges chal = {(i, ?i)} for sampling a set

of randomly selected blocks, where {?i} can be arbitrary weights. Due to the nice property of the homomorphic authenticator, server only needs to response a linear combination of the sampled data blocks ? = ?i?i ? mi, as well as an aggregated authenticator ? = ?i?i ?i, both computed from {mi, ?i, ?i}i?chal. Once the response of ? and ? is verified by TPA, then high probabilistic guarantee on large fraction of cloud data correctness can be obtained.1 Because off-the-shelf error-correcting code technique can be adopted before data outsourcing [6,10], large fraction of current cloud data would be sufficient to recover the whole data. Note that for typical choices of block size mi and file block number n, where mi » log(n), the response ? and ? are (essentially) about the same size as individual block mi and ?i. This means almost constant communication overhead, independent of file size, for each auditing can be achieved. Moreover, since the TPA could regenerate the fresh random sampling challenges, unbounded auditing is achieved too, which means no additional online burden would be incurred towards data owner. However, despite the desirable properties, this approach only works well for encrypted data. When directly applied to unencrypted data, it still leaks bits information towards TPA, as discussed next.

# 11 b) Protecting Data Privacy

The reason that linear combination of sampled blocks may potentially reveal owner data information is due to the following fact about basic linear algebra theory: if enough linear combinations of the same blocks are collected, the TPA can simply derive the sampled data content by solving a system of linear equations.

This drawback greatly affects the security of using homomorphic-authenticator-based techniques in a publicly auditable cloud data storage system. From the perspective of protecting data privacy, the owners, who own the data and rely on the TPA just for the storage security of their data, do not want this auditing process introducing new vulnerabilities of unauthorized information leakage into their data security. Moreover, there are legal regulations, such as the U.S. Health Insurance Portability and Accountability Act (HIPAA) **??**17], further demand the outsourced data not to be leaked to external parties. Exploiting data encryption before outsourcing is one way to mitigate this privacy concern, but it is only complementary to the privacypreserving public auditing scheme to be deployed in cloud. Without a properly designed auditing protocol, encryption itself cannot prevent data from flowing away toward external parties during the auditing process. Thus, it does not completely solve the problem of protecting data privacy but just reduces it to the one of managing the encryption keys. Unauthorized data leakage still remains a problem due to the potential exposure of encryption keys. To address this concern, a proper approach is to combine the homomorphic authenticator with random masking. This way, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server. With random masking, the TPA no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the owner's data content, no matter how many linear combinations of the same set of file blocks can be collected. Meanwhile, due to the algebraic property of the homomorphic authenticator, the correctness validation of the block-authenticator pairs (? and ?) can still be carried out in a new way, even in the presence of randomness.

This improved technique ensures the privacy of owner data content during the auditing process, regardless of whether or not the data is encrypted, which definitely provides more flexibility for different application scenarios of cloud data storage. Besides, with the homomorphic authenticator, the desirable property of constant communication overhead for the server's response during the audit is still preserved.

# 12 c) Handling Multiple Concurrent Tasks

With the establishment of privacy-preserving public auditing in cloud computing, a TPA may concurrently handle auditing delegations on different owners' requests. The individual auditing of these tasks in a sequential way can be tedious and very inefficient for a TPA. Given K auditing delegations on K distinct data files from K different owners, it is more advantageous for a TPA to batch these multiple tasks together and perform the auditing one time, saving computation overhead as well as auditing time cost. Keeping this natural demand in mind, we note that two previous works. Can be directly extended to provide batch auditing functionality by exploring the technique of bilinear aggregate signature. Such a technique supports the aggregation of multiple signatures by distinct signers on distinct messages into a single signature and thus allows efficient verification for the authenticity of all messages. Basically, with batch auditing the K verification equations (for K auditing tasks) corresponding to K responses {?, ?} from a cloud server can now be aggregated into a single one such that a considerable amount of auditing time is expected to be saved. A very recent work gives the first study of batch auditing and presents mathematical details as well as security reasoning's.

# 13 V. results

# 14 Global

[1] [2]

[1] BCloud Data Storage Services Considering Public Audit for Security

[2] © 2013 Global Journals Inc. (US)

2013

Figure 1: T © 2013

1

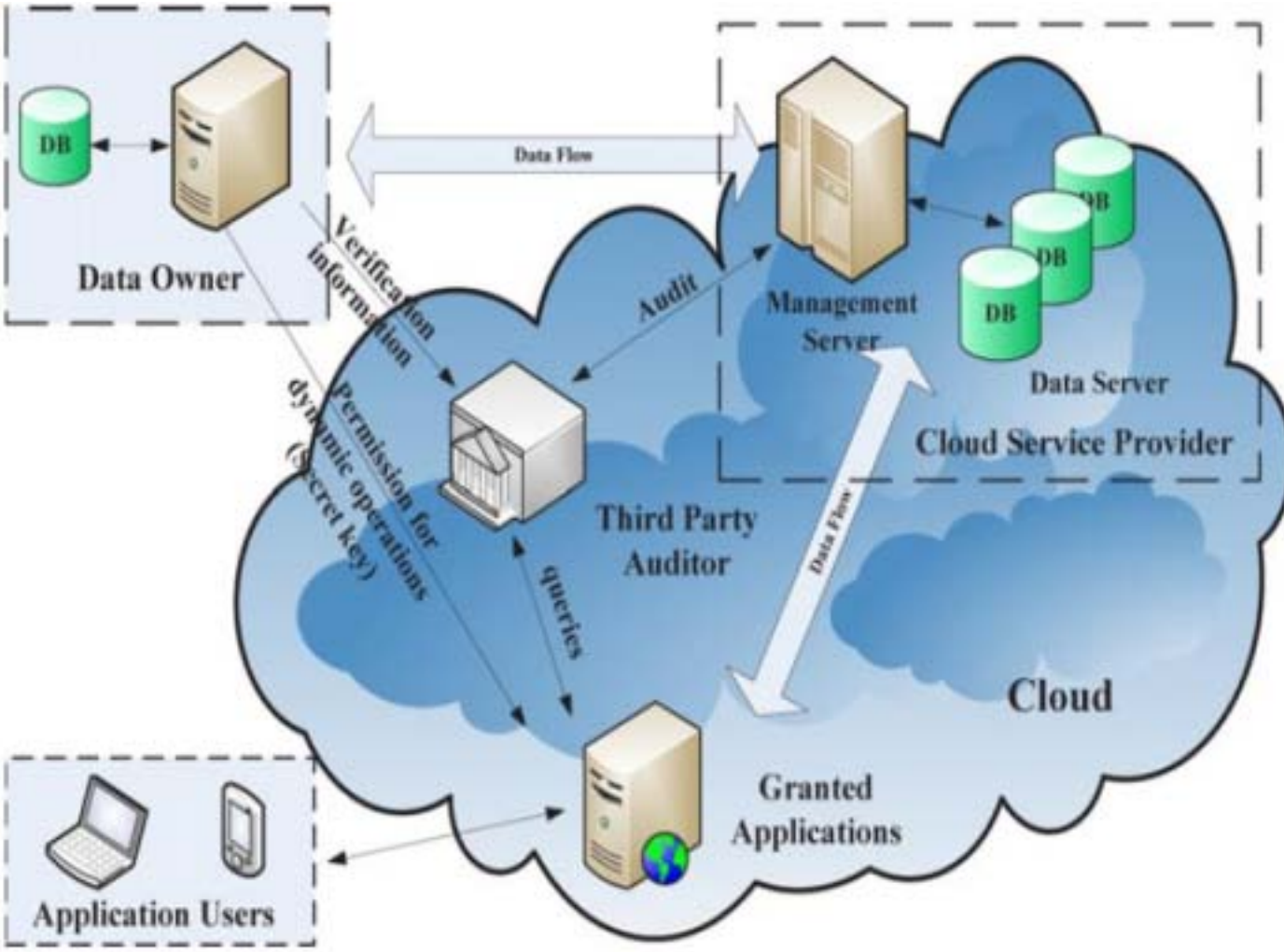


Figure 2: Figure 1 :

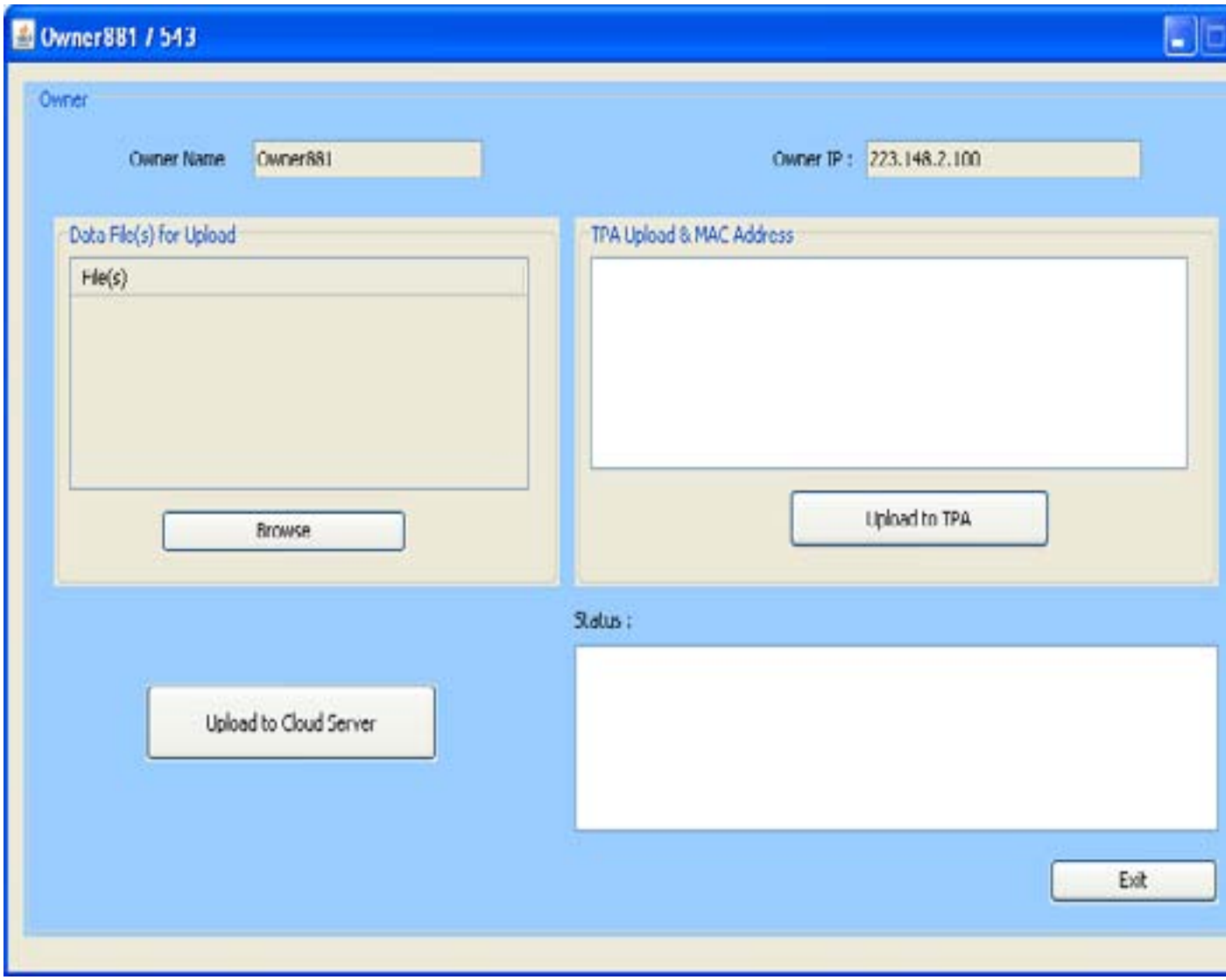
Owner881 / 543
Owner
Owner Name
Owner881
Owner IP :
223.148.2.100
Data File(s) for Upload
File(s)
Browse
TPA Upload & MAC Address
Upload to TPA
Status :
Upload to Cloud Server
Exit

Figure 3:

[Armbrust (2009)] *Above the Clouds: A Berkeley View of Cloud Computing*, M Armbrust . Feb. 2009. Univ. California, Berkeley (Tech. Rep. UCBEECS-2009-28)

[Amazon and Com (2008)] 'Amazon s3 Availability Event'. Amazon , Com . `http://status.aws.amazon.com/s3-20080720.html` *July* July 20. 2008. 2008.

[Krigsman ()] 'Apple's MobileMe Experiences Post-Launch Pain'. M Krigsman . `http://blogs.zdnet.com/projectfailures/?p=908` *July* 2008.

[Mell and Grance ()] *Draft NIST Working Definition of Cloud Computing*, P Mell , T Grance . 2009.

[Erway (2009)] 'Dynamic Provable Data Possession'. C Erway . *Proc. ACM CCS '09*, (ACM CCS '09) Nov. 2009. p. .

[Wang (2009)] 'Enabling Public Verifiability and Data Dynamics for Storage Security in Cloud Computing'. Q Wang . *Proc. ESORICS '09*, (ESORICS '09) Sept. 2009. p. .

[Arrington (2006)] *Gmail Disaster: Reports of Mass Email Deletions*, M Arrington . Dec. 2006.

[Juels et al. (2007)] 'PORs: Proofs of Retrievability for Large Files'. A Juels , J Burton , S Kaliski . *Proc. ACM CCS '07*, (ACM CCS '07) Oct. 2007. p. .

[Wang (2010)] 'Privacy-Preserving Public Auditing for Storage Security in Cloud Computing'. C Wang . *Proc. IEEE INFOCOM '10*, (IEEE INFOCOM '10) Mar. 2010.

[Merkle ()] 'Protocols for Public Key Cryptosystems'. R C Merkle . *Proc. IEEE Symp. Security Privacy*, (IEEE Symp. Security Privacy) 1980.

[Ateniese (2008)] 'Scalable and Efficient Provable Data Possession'. G Ateniese . *Proc. SecureComm '08*, (SecureComm '08) Sept. 2008.