Survey Paper on Integrity Auditing of Storage

Ugale Santosh A[1]
1M.E. Computer AVCOE, Sangmner, India

ABSTRACT:
Cloud servers is a model for enabling convenient, on-demand network access to a shared pool of configurable server resources (networks, memory, storage, cpu, applications, and services) that can be rapidly provisioned and released with minimal management effort or cloud service provider interaction. The Cloud servers models offers the promise of massive cost savings combined with increased IT agility due to pay per consume. However, this technology challenges many traditional approaches to hosting provider and enterprise application design and management. Cloud servers are currently being used; however, data security cited as major barriers to adoption in cloud storage. Users can store data and used on demand or for the applications without keeping any local copy of the data. Users can able to upload data on cloud storage without worrying about to check or verify integrity. Hence integrity auditing for cloud data is more important task to ensure users data integrity. To do this user can resort the TPA (Third Party Auditor) to check the data on the cloud storage. TPA is the expertise and having knowledge and capabilities which users can unable to check. TPA audit the integrity of all files stored on the cloud storage on behalf of users and inform the results. Users should consider auditing process will not cause new vulnerability against the users data also ensures integrity auditing will not cause any resources problem.

Keywords: *Auditing, Cloud, Cloud servers, Data integrity, Data privacy, Security, Storage*

I. INTRODUCTION

Integrity auditing is something you need to have on cloud storage. Different threats imagine a hacker placing a backdoor on storage using applications; modify files, change permissions, or changing your order form to email him a copy of everyone's credit card and other information while leaving it appear to be functionally normally without any problem. By auditing process and setting up convenient period scan reporting, this notifies user within hours of when any file was changed, modified, added or removed. It also helps establish an audit trail in the event cloud storage is compromised. Cloud servers has been envisioned as the next-generation information technology (IT) architecture for government, research, and industry, due to configurable server resources and long list of advantages: on-demand self-service, dynamic resources allocation, Auto-Scaling technology, fast, secure, ubiquitous network access, location independent, resource elasticity, pay per consume, higher uptime and transference of risk [14].

Cloud Computing is remodeling the very nature of how businesses use information technology. One elementary side of this paradigm shifting is that data is being centralized or outsourced to the Cloud server. From users' perspective, including both user and enterprises, uploading data to the cloud server in a flexible on-demand manner brings appealing benefits: relief of the burden for storage and security management, global data access with independent geographical locations, and saving of capital expenditure on security [13], hardware resources and maintenance, etc. whereas Cloud storage makes these features more appealing than ever, it also brings new security vulnerability towards users' data. As a result, the integrity of the data in the cloud is being put at risk due to the different reasons. Although the infrastructures under the cloud provider are much more powerful and secure than local computing devices, they are still facing the different internal and external threats for data integrity. Secondly, there do exist various motivations for hosting provider to behave unfaithfully towards the cloud users regarding the status of their remote data. In short, although outsourcing data to the cloud servers is economically attractive for long-term huge data storage, cloud service provider does not provide any guarantee on data integrity and security. This drawback, if not properly addressed, could impede the successful deployment of the cloud server's design. As users data on remote storage, traditional cryptographic primitives for the purpose of data security protection cannot be adopted [10] directly specifically, downloading data on native system for its integrity verification is not a practical solution due to the transmission cost across the network and security reasons. Considering the large size of the outsourced data store and the user's constrained resources capability, the work of auditing the data correctness in a cloud server environment can be expensive for the cloud server users [7], [9]. Moreover, the overhead of using cloud server storage should be minimized as much as possible, such that cloud user does not need to perform huge operations to use the cloud server data. For example, it is desirable that cloud users do not need to worry about the need to verify the integrity of the data before or after the data retrieval. Besides, there may be multiple user's accesses the same cloud storage for different purpose and applications, say in an enterprise setting.

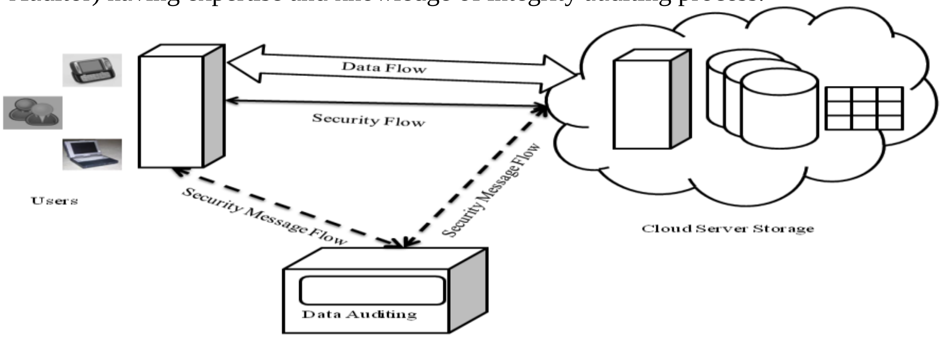
To make it ensure the data integrity and minimize the cloud server computation resources as well as online burden on cloud users', it is of critical importance to enable public auditing process for cloud data storage, so that cloud users may resort to an independent third party auditor (TPA) to audit the data stored on the cloud storage whenever necessary. The TPA, who has knowledge and capabilities that users do not, can check the data integrity of all the data stored in the cloud periodically on behalf of the cloud users, which provides a much more easier and affordable way for the users to ensure their cloud data storage integrity. Moreover, in addition to help users to evaluate the risk of their subscribed cloud data services, the audit result obtained from TPA would also be beneficial for the CSP or hosting provider to improve their security related to storage platform. In a word, auditing services will play an important role for this cloud economy to become fully established; where users will need ways to assess risk and gain trust in the cloud service provider or cloud storage. Currently, the notion of public auditability has been proposed in the context of ensuring remotely stored data integrity under different system and security models [8], [9], [10], [11], [12].

Auditability process allows a third party, in addition to the user himself, to verify the integrity of remotely stored cloud data. However, most of these schemes [8], [9], [11] do not consider the privacy protection of users' data against external auditors. Indeed, TPA may potentially reveal user data information to the auditors. This severe drawback greatly affects the security of these protocols in Cloud storage. From the perspective of protecting data privacy and integrity, the users, who own the data on cloud server and rely on TPA auditing process just for the storage security and integrity of their data, do not want TPA auditing process introducing new vulnerabilities of unauthorized data leakage towards their data security [12].

Also there are some legal regulations on outsourced data that is, data not to be leaked to external parties. Without properly designed auditing protocols, encryption itself cannot prevent data from "flowing away" towards TPA during the public auditing process. The reason, it does not completely solve the problem of protecting data privacy from external parties but just reduces it to the key management. Vulnerability of unauthorized data leakage still remains a problem due to the potential exposure of decryption keys. Therefore, how to enable an auditing protocol keeping data private, independent to data encryption is the problem I am going to tackle in this paper.

II. PROBLEM STATEMENT

The system model I have considered cloud data storage or files storage involving three different entities. As illustrated in figure 1 [1], the cloud users who store the huge amounts of data in the form of files on the cloud storage. Files may be in different types such as binary files, data files, logs files, hidden files. The cloud servers, which fully managed by the hosting or cloud service provider for the data storage space and different resources like network connection, backup facilities and different level security. Third entity is TPA (Third Party Auditor) having expertise and knowledge of integrity auditing process.



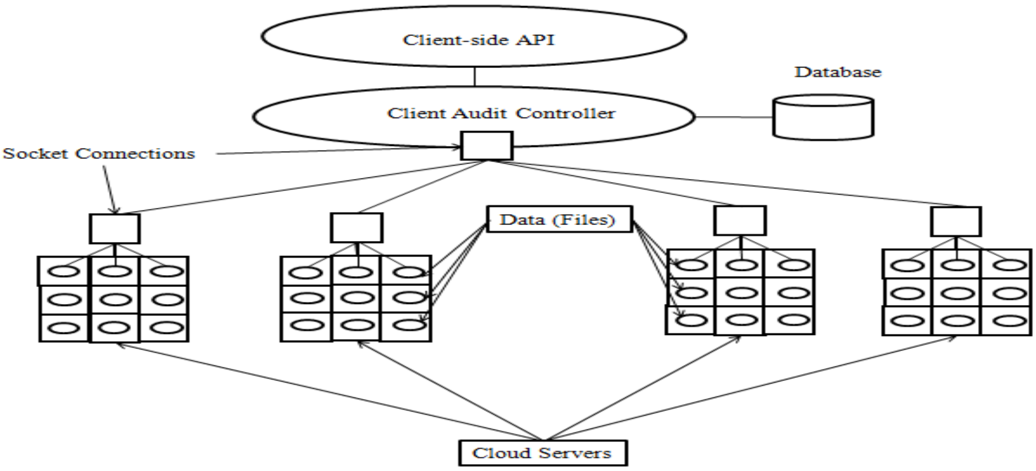
"Figure 1. Cloud architecture"

Cloud service provider is responsible for storage management, maintenance, scalable, pay per consume, location independent, higher availability and low cost data storage. Users upload and download data dynamically from storage space on the cloud server for its own application purpose. Users always need to ensures, data stored on the server is correct and maintained properly. To avoid computational resources and ensure data integrity and security users resort to TPA to audit the data on behalf of user on cloud server.

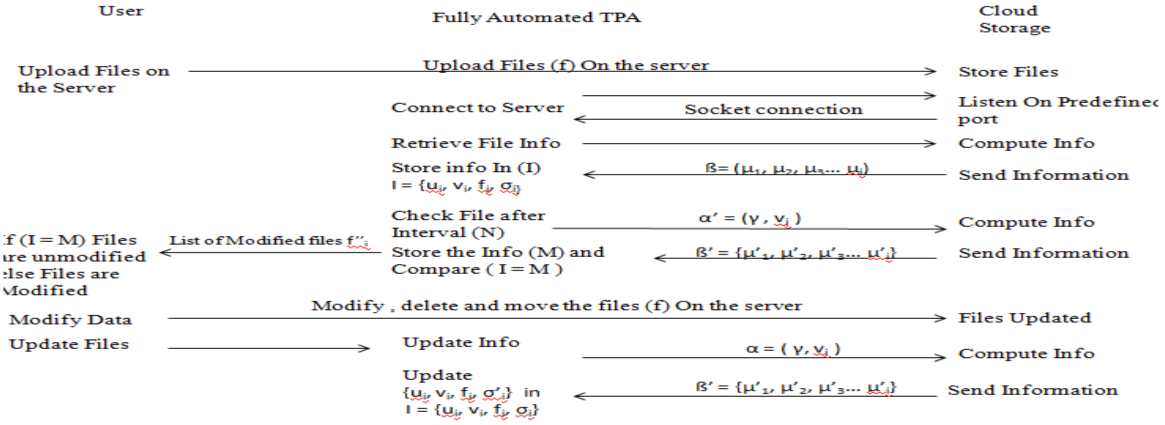
User's data could be hack, modified or changed by internal or external entities. It may includes software bugs, backdoors in different applications, outdated applications versions, plug-in, themes, templates, bugs in system or economically motivated hackers, malicious code and different upload forms. Cloud servers always provide better security but due to different integrity threats towards data like vulnerable functions used in application, outdated applications versions, plugins, themes, templates, bugs in system backdoors in application, applications from the un trusted sources which come with preloaded outdoors, hardware failure, network issue there is changes of data loss. Cloud service provider always try to hide these details from users to their own benefits as well as maintain industry reputation the reason cloud users cannot completely trust on the cloud service provider. With the help of auditing procedure user can gain trust as well as audit his data more efficiently.

III. PROPOSED WORK

This section presents integrity auditing scheme which provides a complete outsourcing solution of data. After introducing notations considered and brief preliminaries, I have started from an overview of proposed Integrity auditing scheme. Then, I am presenting main scheme and show how to extent proposed scheme to support integrity auditing for the TPA upon delegations from multiple users. Finally, I have proposed how to generalize integrity auditing keeping data privacy scheme and its support of dynamic data. Figure 2 illustrate the overview of integrity auditing structure.



"Figure 2. Integrity auditing block diagram"


User
Fully Automated TPA
Cloud Storage
Upload Files on the Server
Upload Files (f) On the server
Store Files
Connect to Server
Socket connection
Listen On Predefinec port
Retrieve File Info
Compute Info
Store info In (I)
I = {uᵢ, vᵢ, fᵢ, σᵢ}
ß= (μ₁, μ₂, μ₃... μᵢ)
Send Information
Check File after Interval (N)
α' = (γ , vⱼ)
Compute Info
:f (I = M) Files ıre unmodified :lse Files are Modified
List of Modified files f''ᵢ
Store the Info (M) and Compare (I = M)
ß' = {μ'₁, μ'₂, μ'₃... μ'ᵢ}
Send Information
Modify Data
Modify , delete and move the files (f) On the server
Files Updated
Update Files
Update Info
α = (γ, vⱼ)
Compute Info
Update {uᵢ, vᵢ, fᵢ, σ'ᵢ} in I = {uᵢ, vᵢ, fᵢ, σᵢ}
ß' = {μ'₁, μ'₂, μ'₃... μ'ᵢ}
Send Information

"Figure 3. Auditing protocol"

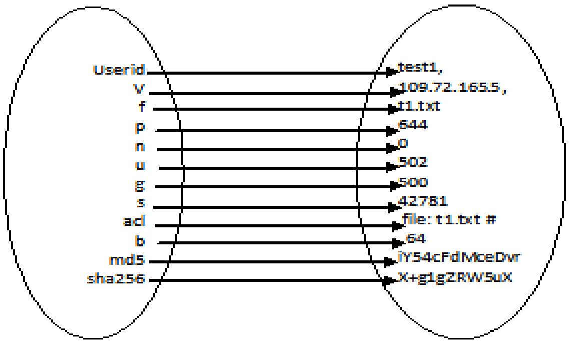
IV. IMPLEMENTATION DETAILS

4.1 Mathematical Model

S={x, e, i, o, f, DD, NDD, success, failure}
Let S be the solution perspective of the class
x= Initial state of the class Initialize ()
x= {Initialize ()} sets the default values for all variables.
Input i =(I1,I2)
I1= {{U}{V}{F}{σ}}
DD=deterministic data it helps identifying the load store functions or assignment functions.
NDD=Non deterministic data of the system S to be solved.
Success-desired outcome generated.
Failure-Desired outcome not generated or forced exit due to system error.
Set of 'k' cloud Users U={u_1,u_2,u_3, ……. u_k}
Set of 'm' cloud servers V={v_1, v_2, v_3, ……. v_m}
Set of files on cloud storage F={f_1, f_2, f_3, ……. f_n}
Set of file tags $\sigma_{i=\{}$ f+p+n+u+g+s+acl+b+selinux+md5+sha256}$_{,}$ i ϵ (1, n)
p= File permissions, t= File type, i= File Inode number
u=File User ID, g= File Group ID, s= File Size
b= File Block count, m= File Modified time
a= File Access Time (when the file was last read, c= is the inode change time, n= Number of links For file S= Check for growing size, md5: md5 hash, sha1= sha1 hash, f = File name, I = Initial Values in Database, N = Interval of auditing process, M = New Value database , LI= List Of files, ST= Detail info of modified files, Set of file tags σ calculated based on the file types,γ= directory path , α = query v=cloud IP address, ß= set of results µ= consist of file stats.
[Data DATA = f+p+n+u+g+s+acl+b+selinux+md5+sha256]
[Growing files GROW=p+u+g+i+n+S+acl+selinux] [Password and shadow files IMP =A+sha256]
[Binary and Configuration files**.** FIXF =A+sha256] [Hidden file PERM = p+u+g+i+acl+selinux]
[Directories DIR = p+n+ i+u+g+acl+ selinux]
Where A= p + n + i + u + g + b + s +m + c +acl + selinux + md5 H=sha1+sha256+sha512

4.1.1 Initialize ()

TPA send Query initialize()
α = (γ, v_j) where, γ ϵ n and v_j is a j^{th} cloud server.
(γ is set or path of (n) files and v_j is cloud IP address)
v_j cloud server produces ß= (μ_1, μ_2, μ_3… μ_i)
Where, μ_i comes from (f_1, f_2, f_3…f_n) consists of pair (f_i, σ_i). TPA store the received values in database (I) Figure 4 show sets of variables and values. I = {u_i, v_j, f_i, σ_i} Where, u_i is user, v_j cloud server and σ_i consist of signature tag of file f_i



"Figure 4. Sets of variables"

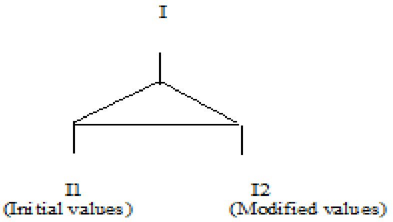
f= {update ( ), Check integrity ( )}

4.1.2 Update()

A step after user uploads/modified the files on cloud server.
TPA send Query Update $\alpha = (\gamma , v_j)$ where $\gamma \in$ n' and v_j is a j^{th} cloud server. n' updated files.
Set of tags $\sigma'_{i\,=}$ {f+p+n+u+g+s+acl+b+selinux+md5+sha256}$_{,}$
$i \in (1, n')$ where σ' updated files tags Number of files F = {f1, f_2, f_3, f_4, ……. f'_n}
Cloud server produces ß' = {$\mu'_1, \mu'_2, \mu'_3\ldots \mu'_i$} Where μ_i comes from ($f_1, f_2, f_3\ldots f_{n'}$) consists of pair ($f_i, \sigma'_i$)
TPA add/replace the ß' values {u_i, v_i, f_i, σ'_i} in I = {u_i, v_i, f_i, σ_i}
I = {u_i, v_i, f_i, σ_i} where u_i is user, v_i cloud server and σ_i consist of signature tag of file f_i

4.1.3 Check integrity()

Initial values I = {u_i, v_i, f_i, σ_i} where, u_i user, v_i cloud sever IP, $\mu_i = (f_i, \sigma_i)$ file name with file stats.
Interval to check integrity (N)
Set of tags $\sigma'_{i\,=}$ { f+p+n+u+g+s+acl+b+selinux+md5+sha256}$_{,}$ $i \in (1, n')$ where σ' updated files tags
Number of files F = { f_1, f_2, f_3, f_4, ……. $f_{n'}$}
TPA to cloud server Query Check $\alpha' = (\gamma , v_j)$
Produces ß' = {$\mu'_1, \mu'_2, \mu'_3\ldots \mu'_i$} where μ_i comes from ($f'_1, f'_2, f'_3\ldots f'_n$)
TPA store the received ß' values {f'_i, σ'_i} in database (M) along with user and server details.
M = {$u_i, v_i, f'_i, \sigma'_i$ }
TPA Search M {$u_i, v_i, f'_i, \sigma'_i$ } in to the database I {u_i, v_i, f_i, σ_i}
If M {$u_i, v_i, f'_i, \sigma'_i$ } $\in$ I {u_i, v_i, f_i, σ_i}



"Figure 5. Results comparison"

As per the Figure 5 TPA system compares the values
Success- If M{$u_i, v_i, f'_i, \sigma'_i$ } ≠ Search result (I){u_i, v_i, f_i, σ_i}
Results:: Files modified lists (f'_i) Else M {$u_i, v_i, f'_i, \sigma'_i$ } = Search result (I){u_i, v_i, f_i, σ_i}
Results:: Files not modified
Failure-Desired results are not generated.
In this scheme, work based on the six phases includes Install client, connect, upload, initialize, check/compare and update

REFERENCES

[1] Cong Wang ,Sherman S.M Chow, Qian Wang, Kui Ren and wening Lou, "Privacy-Preserving Public Auditing for Secure cloud storage" in IEEE transaction on computers vol 62 No 2 February 2013.

[2] Cong Wang, Qian Wang, Kui Ren, Ning Cao, and Wenjing Lou "Toward Secure and Dependable Storage Services in Cloud Computing" IEEE Transaction on Services Computing vol 5 No 2 April-June 2012.

[3] Qian Wang, Cong Wang, Kui Ren , Wenjing Lou And Jin Li " Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" IEEE transaction Paper on Parallel and Distributed Systems vol 22 No 5, pp. 847-859, May 2011.

[4] Kan Yang and Xiaohua Jia "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing" IEEE transaction on parallel distributed system, Vol 24 No 9 September 2013.

[5] Yan Zhu, Hongxin Hu, Gail-Joon Ahn and mengyang Yu "Cooperative Provable Data possession for Integrity Verification in Multicloud Storage." IEEE Transactions on parallel and distributed system, Vol 23, No. 12, pp. 2231-2244,December 2012.

[6] Shucheng Yu, C. Wang, K. Ren, and Wenjing Lou, "Achieving secure, scalable, and fine-grained access control in cloud computing," in *Proc. of IEEE NFOCOM'10*, San Diego, CA, USA, March 2010.

[7] Cloud Security Alliance, "Security guidance for critical areas of focus in cloud computing," 2009, http://www.cloudsecurityalliance.org.

[8] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted Stores," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp. 598–609.

[9] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in *Proc. of ESORICS'09, volume 5789 of LNCS*. Springer-Verlag, Sep. 2009, pp. 355–370.

[10] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability or large files," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp. 584–597.

[11] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. of Asiacrypt 2008*, vol. 5350, Dec 2008, pp. 90–107.

[12] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, "Auditing to keep online storage services honest," in *Proc. Of HotOS'07*. Berkeley, CA, USA: USENIX Association, 2007, pp.1–6.

[13] 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," University of California, Berkeley, Tech. Rep.

[14] P. Mell and T. Grance, "Draft NIST working definition of cloud computing," Referenced on June. 3rd, 2009 Online at http://csrc.nist.gov/groups/SNS/cloud-computing/index.html, 2009.