GNITED MINDS
Journals

*International Journal of Information Technology and Management*

*Vol. VI, Issue No. I, February-2014, ISSN 2249-4510*

# EXTENSIVE SECURE CLOUD STORAGE SYSTEM SUPPORTING PRIVACY-PRESERVING PUBLIC AUDITING

www.ignited.in

*International Journal of Information Technology and Management*
*Vol. VI, Issue No. I, February-2014, ISSN 2249-4510*

# Extensive Secure Cloud Storage System Supporting Privacy-Preserving Public Auditing

**Mounica Doosetty[1] Keerthi Kodakandla[2] Ashok R[3] Shoban Babu Sriramoju[4]**

[1]Student, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India
[2]Student, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India
[3]Student, Department of Computer Science and Engineering, Varadha Reddy College of Engineering, Warangal, India
[4]Associate Professor, Department of Computer Science and Engineering, Varadha Reddy college of Engineering, Warangal, India

***Abstract – Several tendencies are aperture up the era of cloud computing which is an Internet-based expansion and use of computer technology. The still cheaper and more controlling processors, together with the Software as a Service (SaaS) computing architecture, are changing data centres into pools of computing service on an enormous scale. The mounting network bandwidth and reliable yet bendy network connections make it even potential that users can now pledge high quality services from data and software that inhabit solely on distant data centres. In organize to achieve the assurance of cloud data reliability and availability and impose the quality of cloud storage service, we believe the problem of build a secure cloud storage service on top of a communal cloud infrastructure where the check provider is not finally trusted by the customer. We illustrate, at a high level contract signing protocol that join modern and non-standard cryptographic primitives in order to accomplish our goal.***

***Keywords: Dependable distributed storage, Data integrity, error localization, data dynamics, cloud computing.***

---------------------------♦-----------------------------

## I. INTRODUCTION

More than a few trends area unit fissure up the time of Cloud Computing, that is connect degree Internet-based progress and use of engineering. The ever cheaper and additional powerful processors, beside the software system as a service (SaaS) computing design, area unit remodelling data centres into pools of computing service on a large scale. The increasing network information measure and reliable nevertheless flexible network connections build it even potential that users will currently subscribe top quality services from knowledge and software system that reside exclusively on remote knowledge centres. Moving knowledge into the cloud offers nice convenience to users since they don’t need to care regarding the complexities of direct hardware management. Whereas these internet-based on-line services do offer vast amounts of space for storing and customizable computing resources, this computing platform transfer, however, is eradicate the responsibility of native machines for information maintenance at a comparable time. As a result, users’ area element at the mercy of their cloud service suppliers for the contribute and integrity of their information. On the one hand, although the cloud infrastructures area unit rather more controlling and reliable than private computing devices internal and external threats for data truthfulness still exist samples of outages and data loss incidents of noteworthy cloud storage services appear from time to time. Therefore, although outsourcing data in to the cloud is economically enticing for the cost and complexity of long-run large-scale data storage, its lacking of providing sturdy declaration of data integrity and convenience could impede its wide adoption by both enterprise and entity cloud users. In order to attain the assurances of cloud data integrity and convenience and impose the quality of cloud storage service, economical ways that enable on demand data correctness verification on behalf of cloud users have to be planned. However, the fact that users no longer have physical possession of data within the cloud prohibits the direct adoption of conventional science primitives for the aim of data integrity protection. Hence, the authentication of cloud storage correctness must be conducted without specific in sequence of the full data files. Thus, it's additionally essential to support the integration of this dynamic feature into the cloud storage correctness assertion that makes the system style even more difficult. As an complementary approach, researchers have conjointly planned distributed protocols for making certain storage correctness across multiple servers or peers. In this paper, we have a tendency to propose an efficient and versatile distributed storage verification theme with specific dynamic data support to make sure the correctness and handiness of users' information within the cloud. we have a tendency to consider erasure correcting code within the file distribution preparation to supply redundancies and guarantee the information dependableness against Byzantine servers , wherever a storage server may fail in absolute ways that. This construction drastically reduces the communication and storage overheads compared to the normal replication-based file distribution techniques. By utilization of homomorphism token with distributed verification of erasure-coded information, our scheme achieves the storage correctness insurance as well as information error localization. So as to save lots of the time, computation resources, and even the connected online burden of users, we conjointly offer the extension of the planned main scheme to support third-party auditing, wherever users will safely delegate the integrity checking tasks to TPA and be free to use the services. Our work is among the primary few ones during this field to contemplate distributed information storage security in Cloud Computing.

## II. RELATED WORK

Juels and Kaliski Jr. [10] described a formal "proof of irretrievability" (POR) model for ensuring the remote data integrity. Their scheme combines error correcting code and spot-checking to ensure both possession and irretrievability of files on archive service systems. Waters [17] built on this model and constructed a random linear function-based homomorphism authenticator which enables unlimited number of challenges and requires less communication overhead due to its usage of relatively small size of BLS signature. Bowerset al.[18] proposed an improved framework for POR protocols that generalizes both Juelsand Shacham's work. In their sequence work, Bowers et al. [23] extended POR model to distributed systems. However all these schemes are focusing on static data and the effectiveness of their schemes rests primarily on the pre-processing steps that the user conducts before outsourcing the data file F but any change to the contents of F and few bits must propagate through the error correcting code and the corresponding random shuffling process and thus introducing significant computation and communication complexity. Recently, Dodis et al. [20] gave theoretical studies on generalized framework for different variants of existing POR work. Ateniese et al. [11] defined the Provable Data Possession (PDP) model for ensuring possession of file on untrusted storages. There scheme utilized public key-based holomorphic tags for auditing the data file.

The Storage and Computation Cost of Token Pre computation for 1 GB Data File under Different System Settings In other related work, Curtmola et al. [19] ensured that data possession of multiple replicas across the distributed storage system. They extend the PDP scheme to cover multiple replicas without encoding each replica separately, providing a guarantee that multiple copies of data are actually maintained. Lillibridge et al. [25] presented aP2P backup scheme in which blocks of a data file are dispersed across m þ k peers using an erasure code. Peers can make a request for random blocks from their backup peers and verify the integrity using separate keyed cryptographic hashes attached on each block. This scheme can detect data loss from free-riding peers, but it does not ensure entire data is unchanged. Filho and Barreto [37] proposed to verify data integrity using RSA-based hash to demonstrate unchea table data possession in peer-to-peer file sharing networks. But their proposal requires exponentiation over the entire information file, which is clearly impractical for the server whenever the file is large. Shahet al. [12], [13] proposed allowing a TPA to keep online storage honest by first encrypting the data then sending a number of pre computed symmetric-keyed hashes over the encrypted data to auditor. However, their scheme works for only encrypted files and auditors must maintain long-term state. Schwarz and Miller [24] proposed to ensure static file integrity across multiple distributed servers using erasure coding and block-level file integrity checks. We have adopted some ideas of their distributed storage verification protocol. Our scheme further supports data dynamics and explicitly studies the problem of misbehaving server identification. Very recently Wang et al.[31] gave a study on many existing solutions on checking remote data integrity and discussed their pros and cons under different design scenarios of secure cloud storage services. Portions of the work presented in this paper have previously appeared as an extended abstract in [1]. In this paper, we utilize the public key based holomorphic authenticator and uniquely integrate it with random mask technique to achieve a privacy-preserving public auditing system for cloud data storage security while keeping all above requirements in view. To support efficient handling of multiple auditing tasks, we explore the technique of bilinear aggregate signature to extend our result into a multiuser setting, where multiple auditing tasks can be simultaneously by TPA.

## EXPLORATION OF THE PROPOSED CONTRACT SIGNING PROTOCOL

A fair contract signing protocol allows two potentially mistrusted parities to exchange their commitments. Contract signing is truly simple due to the existence of "simultaneity". That is two parties generally sign two hard copies of the same contract at the same place and at the same time based on the RSA signature scheme. Here the fair exchange, between two (or multiple) potentially mistrusted parities exchanging digital items over computer networks in a fair way, so that each party gets the other's item, or neither party does. In the fair exchange will contain:

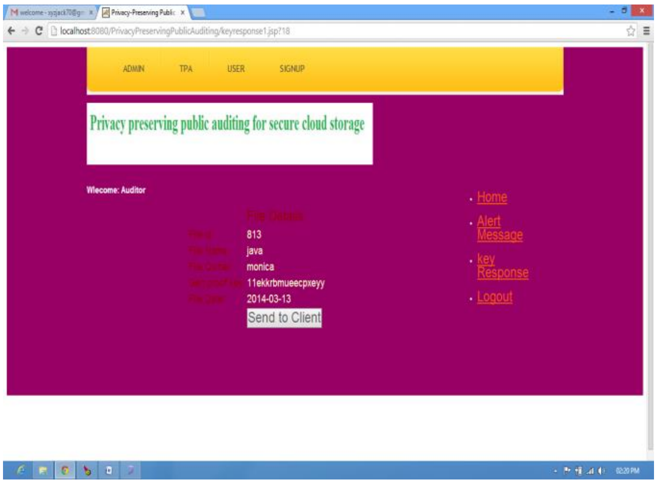
1) Contract Signing Protocol.

2) Certified e-mail systems.

3) Non-reputation Protocol.

## I. CERTIFIED E-MAIL SYSTEMS:

Certified electronic mail enables two mutually suspicious users to exchange a receipt for electronic mail. One family of protocols, the believers' protocols, use a trusted third party. The second family, the sceptics' protocols, uses no third party. Our protocols are secure in a very strong sense; the probability of one party cheating can be made arbitrarily small. The protocols provide a practical example of the use of various innovative cryptographic techniques, including digital signatures, bit commitment, and zero-knowledge interactive proofs. These protocols can be implemented in modern communication networks.


welcome - syjack70@g
Privacy-Preserving Public
localhost:8080/PrivacyPreservingPublicAuditing/keyresponse1.jsp?18
ADMIN
TPA
USER
SIGNUP
Privacy preserving public auditing for secure cloud storage
Wlecome: Auditor
Home
Alert Message
key Response
Logout
02:20 PM

File Details
813
java
monica
11ekkrbmueecpxeyy
2014-03-13
Send to Client

## II. NON-REPUTATION PROTOCOL:

The goal of a non-repudiation service is to collect and validate irrefutable evidence regarding the transfer of a message from the originator to recipient possibly involving the service of a trusted third party called the Delivery Agent. We differentiate between the following non-repudiation services.

## III. CONTRACT SIGNING PROTOCOL:

Contract signing protocol is essentially implied by fair exchange of digital signatures between two potentially mistrusted parities with the Trusted Third Party (TTP).Two Parties a and b want to sign a contract cover a communication network. They must simultaneously exchange their commitments to c since simultaneous exchange is usually impossible, protocols are needed to approximate simultaneity by exchanging partial commitments in piece by piece manner. During this protocol, one party or another may have a slight advantages a fair protocol keeps this advantage within acceptable limits

Contract signing Protocol was divided into:

1. Gradual Exchange without any TTP.
2. Protocol with an online TTP.
3. Protocol with off line TTP.

### 1) Gradual Exchange without any TTP:

Gradual Exchange protocols to meet computational fairness: Both parties exchange their commitments or secrets bit-by-bit then if one party stops prematurely then both parties have the same fraction of the peer's secret, which means that they complete the contract off-line by investing about the same amount of computing work. The advantage of this approach is that no TTP is involved. However, this is unrealistic for most real-world applications due to the following reasons. First, it is assumed that the two parties have equivalent computation resources. Otherwise, this protocol is favourable to the party with stronger computing power who may conditionally force the other party to commit the contract by its interest. At the same time, these protocols are inefficient because the costs of computation and communication are extensive.

### 2) Protocol with an online TTP:

An on-line TTP is always involved in every change. In this scenario a TTP is essentially a mediator.

- Each party first sends item to the TTP
- The TTP checks the validity of these items
- If all expected items are correctly received, the TTP forwards each item to the party

### 3) Protocol with off line TTP:

This protocol is optimistic in the sense that the TTP is not invoked in the execution of exchange unless one of the two parties misbehaves or the communication channel is out of order. Trusted Third Party (TTP) is not invoked when the two involved parties perform the protocol correctly. This kind of protocol is more practical than those in which TTP mediates all transactions.

In the above said protocols are very difficult to manage. So new contract signing protocol for two mutually distrusted parties. Our protocol is based on an RSA multisignature, which is formally proved to be secured and optimistic because.

- Fairness
- Optimism
- Abuse freeness

- Provable Security
- Timely Termination
- Compatibility
- TTP' Statelessness
- High Performance

To exchange the signatures we use:

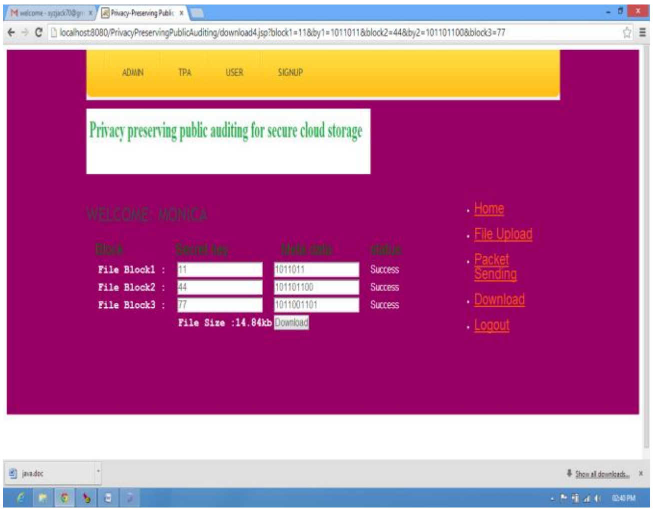
1) Registration protocol

2) Signature exchange protocol

3) Dispute resolution protocol

1) **Registration Protocol**: At first parties should Register at TTP and get certificate from the TTP. Registration protocol is a little bit complicated, we remark that this stage needs to be executed only once for a sufficiently long period.

2) **Signature exchange protocol:** the contract explicitly contains the following information: a predetermined but reasonable deadline , the identities of parties and the TTP. Our signature exchange protocol is briefly illuminated in Figure 1.



localhost:8080/PrivacyPreservingPublicAuditing/download4.jsp?block1=11&by1=1011011&block2=44&by2=101101100&block3=77

Privacy preserving public auditing for secure cloud storage

WELCOME: MONICA

| Block | Secret key | Meta data | status |
|---|---|---|---|
| File Block1 : | 11 | 1011011 | Success |
| File Block2 : | 44 | 101101100 | Success |
| File Block3 : | 77 | 1011001101 | Success |

File Size :14.84kb

**3) Dispute resolution protocol:** If party has sent his signature another party but does not receive the value before the deadline, then he sends the TTP to apply dispute resolution. Upon receiving application, the TTP performs.

1) TTP first verifies.

2) The TTP checks whether the deadline expires or not

3) If expires Get valid from the TTP directly by initiating dispute resolution protocol.

4) Run the Signature Exchange Protocol Again.

5) Exchange the Signatures.

## CONCLUSION

In this paper, we investigated the problem of data security in cloud data storage and data transmission. We propose a privacy-preserving public auditing system for data storage security in Cloud Computing. So we utilize the holomorphic linear authenticator and random masking to guarantee that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process which will not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also alleviates the user's fear of their outsourced data leakage and also TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend out privacy-preserving public auditing protocol into a multi-user setting where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and high efficient.

## REFERENCES

[1] Cong Wang, Qian WangKui Ren, Ning Cao, and Wenjing Lou "Toward Secure and Dependable Storage Services in Cloud Computing" IEEE transactions on services computing, vol. 5, no. 2, april-june 2012

[2] Qian Wang,Cong Wang, Kui Ren, Wenjing Lou Jin Li "Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing" IEEE transactions on parallel and distributed systems, vol. 22, no. 5, may 2011

[3] Boris Tomas1and Bojan Vuksic2 "Peer to Peer Distributed Storage and Computing Cloud System" International conference on information technology interfaces, june 25-28, 2012, cavtat, Croatia

[4] "Security and Privacy Challenges in Cloud Computing Environments" co-published by the IEEE computer and reliability ieee november/december 2010

[5] Subashini S, Kavitha V., "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications (2011) vol. 34 Issue 1, January 2011 pp. 1-11.

[6] Balachander R.K, Ramakrishna P, A. Rakshit, "Cloud Security Issues, IEEE International Conference on Services Computing (2010)," pp. 517-520.

[7] Kresimir Popovic, Željko Hocenski, “Cloud computing security issues and challenges,” MIPRO 2010, pp. 344-349.

[8] Amazon.com, “Amazon Web Services (AWS),” Online at http://aws. amazon.com, 2008.

[9] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres1, Maik Lindner, “A Break in Clouds: Towards a cloud Definition,” ACM SIGCOMM Computer Communication Review, vol. 39, Number 1, January 2009, pp. 50-55.

[10] Patrick McDaniel, Sean W. Smith, “Outlook: Cloudy with a chance of security challenges and improvements,” IEEE Computer and reliability societies (2010), pp. 77-80.

[11] Sameera Abdulrahman Almulla, Chan Yeob Yeun, “Cloud Computing Security Management,” Engineering systems management and its applications (2010), pp. 1-7.

[12] Steve Mansfield-Devine, “Danger in Clouds”, Network Security (2008), 12, pp. 9-11.

[13] Anthony T. Velte, Toby J.Velte, Robert Elsenpeter, Cloud Computing: A Practical Approach, Tata Mc Graw Hill 2010.

[14] Gary Anthes, “Security in the cloud,” In ACM Communications (2010), vol.53, Issue11, pp. 16-18.

[15] Lombardi F, Di Pietro R. Secure virtualization for cloud computing. Journal of Network Computer Applications (2010), doi:10.1016/j.jnca.2010.06.008.