Implementation of TPA and Data Integrity in Cloud Computing using RSA Algorithm

[1]Mr.Vinay Tila Patil, [2] Prof. Gajendra Singh Chandel

[1]Student, M. Tech. in software engineering, [2] HOD, Computer Science Dept.,

at SSSIST, Sehore (M.P.), India

Abstract— Cloud infrastructure has been envisioned as the next-generation construction of IT Initiative. It passages the application software and databases to the integrated large data hubs, where the administration of the data and services may not be fully trustworthy. This exceptional prototype brings about many new security challenges, which have not been well unwritten. This work studies the problem of ensuring the integrity of data storage in Cloud Computing. In certain, we consider the task of allowing a trusted third party (TPA), on behalf of the cloud client, to verify the data integrity of the data stored in the cloud server. By using TPA we eliminate the involvement of the cloud client through the auditing of whether his data integrity 0f stored data in the cloud server is to be sure integral. The support for data changing aspects via the most general forms of data operation, such as text modification, insertion and deletion, is also a significant step toward practicality, since services in Cloud Computing are not limited to archive or backup data only. While previous works on ensuring remote data integrity often lacks the support of either public auditability or dynamic data operations, but our work can be succeeds in both steps both. In this we first identify the difficulties and possible security problems of direct extensions with fully dynamic data updates from previous works and then show how to construct a smart verification scheme for the unified integration of these two outstanding features in our design. In particular, to achieve effective data dynamics, we improve the current proof of storage models by manipulating block tag authentication. To support efficient handling of multiple auditing tasks, we further explore the technique of signature to extend our main result into a multi-user setting, where TPA can perform multiple auditing tasks simultaneously. Extensive security and performance analysis show that the proposed schemes are highly efficient and provably secure.

Keywords- cloud computing , cloud security, TPA.

I. INTRODUCTION

Cloud computing is essentially a composition of a large-scale distributed and virtual machine computing infrastructure. This new paradigm delivers a large pool of virtual and dynamically scalable resources including computational power, storage, hardware platforms and applications to users via Internet technologies. All Internet users can make use of cloud systems and services, deriving many advantages when migrating all or some of their information to cloud computing environment.

However, just like real clouds, this virtual cloud is prone to unpredictability. Rain clouds harvest water through evaporation from one place and deliver this rain to distant lands. Similarly, cloud computing is a harvest of valuable data to be delivered from the Internet, possibly even to places where this data does not belong, which is the fear factor. Some may argue that the concept of distant land is made redundant by the concept of the Internet thus this fears is ill-based. One of the major challenges faced by cloud computing concept and its global acceptance is how to secure and protect the data and processes that are the property of the customers. The security of cloud computing environment is a new research area requiring further development by both the academic and industrial research communities. In fact, the migration process into the cloud is very simple. It starts by identifying what an organization needs to move to the cloud; finding a provider, negotiating the requirements to go to the cloud, and finally, signing off on the contract. Overall security may be considered to be based on trust and "keeping fingers crossed (hope)" alone. There is no guarantee that a cloud provider will always follow and meet contractual terms and conditions. Information Security Magazine asks [1]: "How do you perform an on-site audit when you have a distributed and dynamic multi-tenant computing environment spread all over the globe? It may be very difficult to satisfy auditors that your data is properly isolated and cannot be viewed by other customers."

Data security in cloud computing is a very important issue for various reasons. One of them is that in the cloud environment there is a financial contract between clients and the cloud provider. That is, the cloud clients should only pay for the services they use. The cloud providers should guarantee that and should compensate the customers for any loss that results from not fulfilling the service level agreement. Organizations are the main targeted customers for the cloud and they require a highly scalable access control for a large amount of stored data. Many users (both individuals and organizations) prefer to choose a cloud provider they trust and only inspect the SLA for standards compliance. They will most likely choose not to bother themselves with the complexity of using POS schemes with cloud storage services. Thus, it is up to the user whether to request using these POS with cloud storage or not. POS schemes have been around for some years and the question is: is there anybody who will use these POS? To the best of my knowledge no one uses them in commercial cloud systems. However, adopting these ideas could be simpler in

the future with all the advances in the ICT industry. This thesis focuses on introducing some solutions that allow the cloud customers to obtain assurance regarding the confidentiality, integrity, availability, fairness (or mutual non-repudiation), data freshness, geographic assurance and replication of the data stored in the cloud.

Research Motivation

• Many of the proposed protocols require the cloud customers to trust the cloud provider. Also, they see the security from the cloud provider perspective not from the cloud customer side [3, 4, and 2].

• Some of the service level agreements published by public cloud providers lack information on how a cloud customer can control his or her data when stored in the cloud. Also, in the event of not fulfilling the conditions, how the cloud provider will compensate the cloud customers is not specified.

• Some recent incidents have violated the data availability and scalability stated in the service level agreement.

Research Objectives

As defined in the prior section, the objectives that need to be addressed in this proposal are:

1. To design a secure storage architecture for data storage in the cloud. This architecture will focus on the security requirements including data confidentiality, integrity, availability, fairness, freshness.

2. To allow the cloud customers to check where their stored data is located, without relying on the word of the cloud provider.

3. To allow the cloud customers to verify that their stored data is replicated over multiple and diverse locations; again without relying on the provider's claim.

II. Literature Survey

Cloud computing is the collective term for a group of IT technologies which in collaboration are changing the landscape of how IT services are provided, accessed and paid for. Some of the supporting technologies have already been available for quite some time, but it is the combination of several technologies which enables a whole new way of using IT.

The NIST definition of cloud computing is:

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

A. Cloud service models:

• Software-as-a-Service (SaaS): The SaaS service model offers the services as applications to the consumer, using standardized interfaces. The services run on top of a cloud infrastructure, which is invisible for the consumer. The cloud provider is responsible for the management the application, operating systems and underlying infrastructure. The consumer can only control some of the user-specific application configuration settings.

• Platform-as-a-Service (PaaS): The PaaS service model offers the services as operation and development platforms to the consumer. The consumer can use the platform to develop and run his own applications, supported by a cloud-based infrastructure. "The consumer does not manage or control the underlying cloud infrastructure including network, servers, operating systems, or storage, but has control over the deployed applications and possibly application hosting environment configurations".

• Infrastructure-as-a-Service (IaaS): The IaaS service model is the lowest service model in the technology stack, offering infrastructure resources as a service, such as raw data storage, processing power and network capacity. The consumer can the use IaaS based service offerings to deploy his own operating systems and applications, offering a wider variety of deployment possibilities for a consumer than the PaaS and SaaS models. "The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components (e.g., host firewalls)".

Fig 1: Cloud service models

B. Cloud deployment models:

Regardless of which delivery model is utilized, cloud offerings can be deployed in four primary ways, each with their own characteristics.

The characteristics to describe the deployment models are;

1) Who owns the infrastructure?
2) Who manages the infrastructure?
3) Where is the infrastructure located?
4) Who accesses the cloud services?

• Public clouds: Public cloud computing is based on massive scale offerings to the general public. The infrastructure is located on the premises of the provider, who also owns and manages the cloud infrastructure. Public cloud users are considered to be untrusted, which means they are not tied to the organization as employees and that the user has no contractual agreements with the provider.

• Private clouds: Private clouds run in service of a single organization, where resources are not shared by other entities. "The physical infrastructure may be owned by and/or physically located in the organization's datacenters (on-premise) or that of a designated service provider (off-premise) with an extension of management and security control planes controlled by the organization or designated service provider respectively" (Bardin, Callas, Chaput et al. 2009). Private cloud users are considered as trusted by the organization, in which they are either employees, or have contractual agreements with the organization.

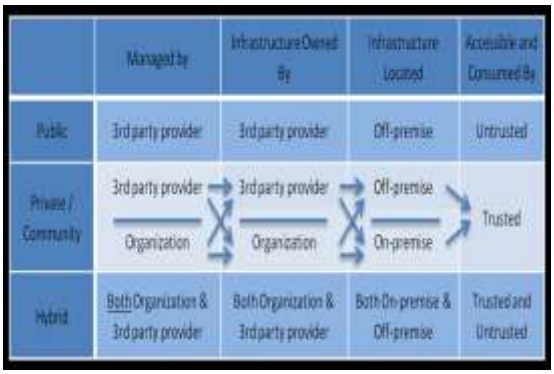


Fig 2: Cloud deployment models

• Community clouds: Community clouds run in service of a community of organizations, having the same deployment characteristics as private clouds. Community users are also considered as trusted by the organizations that are part of the community.

• Hybrid clouds: Hybrid clouds are a combination of public, private, and community clouds. Hybrid clouds leverage the capabilities of each cloud deployment model. Each part of a hybrid cloud is connected to the other by a gateway, controlling the applications and data that flow from each part to the other. Where private and community clouds are managed, owned, and located on either organization or third party provider side per characteristic, hybrid clouds have these characteristics on both organization and third party provider side. The users of hybrid clouds can be considered as trusted and untrusted. Untrusted users are prevented to access the resources of the private and community parts of the hybrid cloud.

C. *Cloud security issues:*

1. System Complexity

Compared to traditional data center the cloud architecture is much more complex. Therefore while considering security, security of all these components and interaction of these components with each other needs to be addressed [9].

2. Shared Multi-tenant Environment

Since the cloud need to provide service to millions of client, a logical separation of data is done at different level of the application stack [9]. Because of which a 8 attacker in the face of client can exploit the bugs gaining access to data from other organizations [9].

3. Internet-facing Services

The cloud service which is accessed over the internet via browser, the quality of service delivered on the network is another concern [9].

4. Loss of control

As the data of client is stored anywhere across the world control loss over physical, logical of system, and alternative control to clients' assets, mismanagement of assets are some additional concerns [9].

D. *Third Party Auditor:*

The third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service security on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. The users may resort to TPA for ensuring the storage security of their outsourced data, while hoping to keep their data private from TPA. We consider the existence of a semi-trusted CS as does. Namely, in most of time it behaves properly and does not deviate from the prescribed protocol execution. However, during providing the cloud data storage based services, for their own benefits the CS might neglect to keep or deliberately delete rarely accessed data files which belong to ordinary cloud users. Moreover, the CS may decide to hide the data corruptions caused by server hacks or Byzantine failures to maintain reputation. We assume the TPA, who is in the business of auditing, is reliable and independent, and thus has no incentive to collude with either the CS or the users during the auditing process. TPA should be able to efficiently audit the cloud data storage without local copy of data and without bringing in additional on-line burden to cloud users

III. PROPOSED SYSTEM

A. *Existing System*

Although the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public auditability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing.

Disadvantages:

Especially to support block insertion, which is missing in most existing schemes.

B. *Proposed System*

• 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.

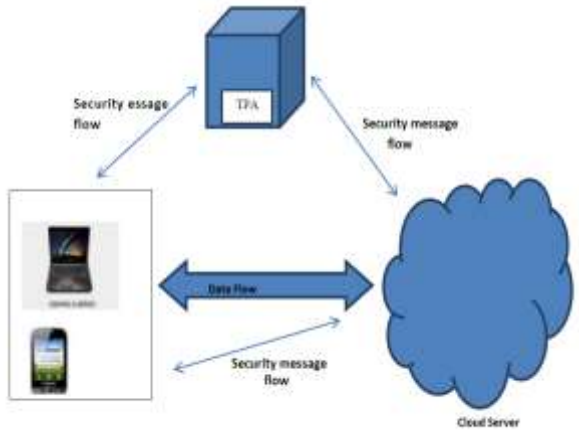


Fig 3: Architecture of storage of data on cloud using TPA

Advantages:

1) We motivate the public auditing system of data storage security in Cloud Computing, and propose a protocol supporting for fully dynamic data operations, especially to support block insertion, which is missing in most existing schemes;

2) We extend our scheme to support scalable and efficient public auditing in Cloud Computing. In particular, our scheme achieves auditing tasks from different users can be performed simultaneously by the TPA.

3) We prove the security of our proposed construction and justify the performance of our scheme through concrete implementation and comparisons.

IV. IMPLEMENTATION

Algorithm Techniques

- Setup Phase
- Audit Phase

The client's public key and private key are generated by invoking KeyGen (•). By running SigGen (•), the data file F is pre-processed, and the homomorphic authenticators together with metadata are produced.

KeyGen(1k). The client generates a random signing key pair (spk, ssk). Choose a random α ← Zp and compute v ← g α . The secret key is sk = (α , ssk) and the public key is pk = (v, spk).

SigGen (sk, F). Given F = (m1, m2 . . . , mn), the client chooses a random element u ← G. Let t = name||n||u||SSigssk (name||n||u) be the file tag for F.

Then the client computes signature σ i for each block mi (i = 1, 2 . . . n) as σ i ← (H(mi) • umi) α . Denote the set of signatures by _ = { σ i}, 1 ⩽ i ⩽ n. The client then generates a root R based on the construction (pk, sk) ← KeyGen (1k). This probabilistic algorithm is run by the client. It takes as input security parameter 1k, and returns public key pk and private key sk. (_, sigsk (H(R))) ← SigGen (sk, F).

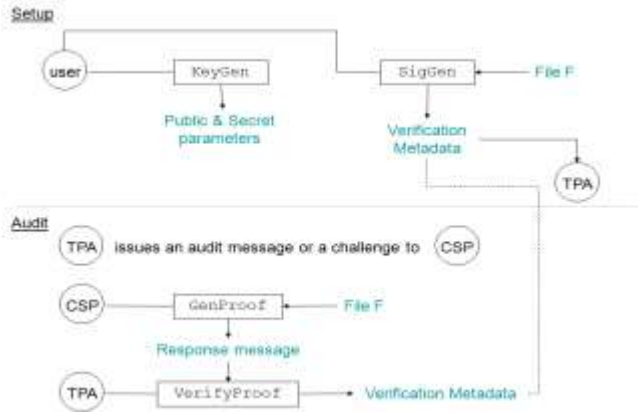


Fig 4: Implementation Flow

This algorithm is run by the client. It takes as input private key sk and a file F which is an ordered collection of blocks {mi}, and outputs the signature set _, which is an ordered collection of signatures {σi} on {mi}. It also outputs metadata-the signature sigsk (H(R)) of the root R of a Merkle hash tree. In our construction, the leaf nodes of the hashes of H (mi). (P) ← GenProof (F,_, chal). This algorithm is run by the server. It takes as input a file F, its signatures _, and a challenge chal. It outputs a data integrity proof P for the blocks specified by chal.

Module description

A. User secure identification

This module is the first level of the process, where the users are going to register themselves to the server with help of the third party authority. So this level of security will has more efficiency then the normal security issues. In additional the normal authentication process will be accomplished with it. After the registration there will be a secure key provided by the TPA and this will be used in further transaction.

Fig 5: The user login is used for the login into the cloud interface.

B. Metadata key generation

Let the verifier V wishes to the store the file F. Let this file F consist of n file blocks. Initially preprocess the file and create metadata to be appended to the file. Let each of the n

data blocks have m bits in them. A typical data file F which the client wishes to store in the cloud.
Each of the Meta data from the data blocks is encrypted by using a RSA algorithm to give a new modified Meta data Mi. Without loss of generality Show this process. The encryption method can be improvised to provide still stronger protection for Client's data. All the Meta data bit blocks that are generated using the procedure are to be concatenated together. This concatenated Meta data should be appended to the file F before storing it at the cloud server. The file F along with the appended Meta data with the cloud.

RSA Algorithm

RSA includes a public-key and a private-key. The public-key can be acknowledged to everyone and is used for encrypting messages. Messages encrypted with the public-key can only be decrypted using the private-key. The keys for the RSA algorithm are produced the following way:
Choose two separate prime statistics p and q.
For security purposes, the integer's p and q should be chosen at arbitrary, and should be of related bit-length.
Compute n = pq.
n is used as the modulus for both the public and private keys
Compute $\varphi(n) = (p-1)(q-1)$, where φ is Euler's totient function.
Choose an whole number e such that $1 < e < \varphi(n)$ and GCD of $(e, \varphi(n)) = 1$; i.e., e and φ(n) are coprime.
e is released as the public key supporter.
Determine d as:

$$d \equiv e^{-1} \pmod{\varphi(n)}$$

i.e., d is the multiplicative inverse of e mod φ(n).
Encryption
Encryption is the process of translating plain text into cipher text.

$$c = m^e \pmod{n}.$$

Decryption

Decryption is the process of translating cipher text into plain text

$$m = c^d \pmod{n}.$$

Fig 6: Key is assigned privately to the user for performing the data operations on the uploaded data.

C. Secure upload and download

In this module the user will upload the data which they need to send other or which they need to store in the cloud storage. These process are depends on the secure key which is provided to the user while the registration process held. After that if users want to verify the data they need this key.

Fig 7: The data is uploaded onto the server in the form of data blocks which are distributed on the cloud server.

Fig 8: The files on the server are downloaded using the above dialog box.

D. Verifiability

This module is the main process of the project, when a user need to check the data integrity and dynamics they have post a query to the TPA those are valid for the verification process. After the third party auditor receives the query of the user , it retrieve the corresponding key of the respective users and verify the data dynamics from the cloud storage.

Fig 9: File is verified for data integrity on performing the data operations on the data blocks.

E. Server

Server module is the controller process and it also the data source module. In this project the cloud storage are the main servers of the users. Each data transfer to the particular data source which the users depends on.

Fig 10: Cloud server stores the uploaded data blocks and creates the log in context to the data operations performed on the data blocks.

F. Third Party Auditor

The Third Party Auditor is a module which is used to the audit the data that are uploaded by the Data Owner in the Server of the Cloud Service Provider. So that they will audit the data based on the Data Owner's request. Once it received the request from data owner, it checks the data integrity stored in cloud server .In this auditing process [9] first it received the original Top Hash Value from data owner. Then it request particular part P1 along with its hash value h1 and also request h2 value from cloud server. Once it received this, it merges this two hash value and generates h12 and request h34 value from cloud server. Once it received this, it merges this two hash value and generates h1234 and request h5678 value from cloud server. Once it received this, it merges this two hash value and generates h12345678 value which is the new top hash value. Already TPA have original top hash value and now check this new top hash value with original top hash value, if both values are same means then it sends message to data owner that the data is in correct, consistent manner .If both values are different means then it sends message to data owner that the some data are lost from the original information.

Fig 11: File info creates the log for the data operation on the cloud servers like download and upload operations of the text files as in the above snapshots.

G. Reports

This module is last module for the process which will helps to show detail about the users, third party auditors and the data transactions and etc… usually the server only manages the reports module for each and every transactions.

Fig 12: Report

V. Security Analysis

In this section, we evaluate the security of the proposed scheme under the security model defined in Section 2.2. Following [4], we consider a file F after Reed-Solomon coding.

Definition 1: (CDH Problem) The Computational Diffie-Hellman problem is that, given g, gx, gy ∈ G for unknown x, y ∈ Zp, to compute gxy. We say that the (t, ϙ)-CDH assumption holds in G if no t-time algorithm has the non-negligible possibility ϙ in resolving the CDH problem. A proof-of-retrievability protocol is sound if any cheating proved that convinces the verification algorithm that it is storing a file F is actually storing that file, which we define in saying that it yields up the file F to an extractor algorithm which interacts with it using the proof-of-retrievability protocol. We say that the adversary (cheating server) is ϙ-admissible if it convincingly answers a ϙ-fraction of verification challenges. We formalize the notion of an extractor and then give an accurate explanation for reliability.

Scheme / Metric	2	4	12*	14	My scheme
Data dynamics	**No**		**Yes**		
Public auditability	**Yes**	**Yes**	**No**	**No+**	**Yes**
Sever comp. complexity	O(1)	O(1)	O(1)	O(log n)	O(log n)
Verifier comp. complexity	O(1)	O(1)	O(1)	O(log n)	O(log n)
Comm. complexity)	O(1)	O(1)	O(1)	O(log n)	O(log n)
Verifier storage complexity	O(1)	O(1)	O(1)	O(1)	O(1)

TABLE 1: Comparisons of different remote data integrity checking schemes. The security parameter λ is eliminated in the costs estimation for simplicity.

Where;

∗ the scheme only supports bounded number of integrity challenges and partially data updates, i.e., data insertion is not supported.

+ No explicit implementation of public auditability is given for this scheme.

	BLS-based instantiation		RSA-based instantiation		[14]
Metric\ Rate-ϙ	99%	97%	99%	97%	99%
Sever comp. time (ms)	6.45	2.11	9.81	4.55	14.9
Verifier comp. time (ms)	806.01	284.17	779.10	210.47	782.56
Comm. cost (KB)	239	80	223	76	280

TABLE 2: Performance comparison under different tolerance rate ϙ of file corruption for 1GB file. The block size for RSA-based instantiation and scheme in [14] is chosen to be 4KB.

VI. Performance Analysis

From Table 4, it can be observed that the overall performance of the three schemes is comparable to each other. Due to the smaller block size (i.e., 20bytes) compared to RSA-based instantiation, our BLS-based instantiation is more than 2 times faster than the other two in terms of server computation time. However, it has larger computation cost at the verifier side as the pairing operation in BLS scheme consumes more time than RSA techniques. Note that the communication cost of DPDP scheme is the largest among the three in practice. This is because there are 4-tuple values associated with each skip list node for one proof, which results in extra communication cost as compared to our constructions.

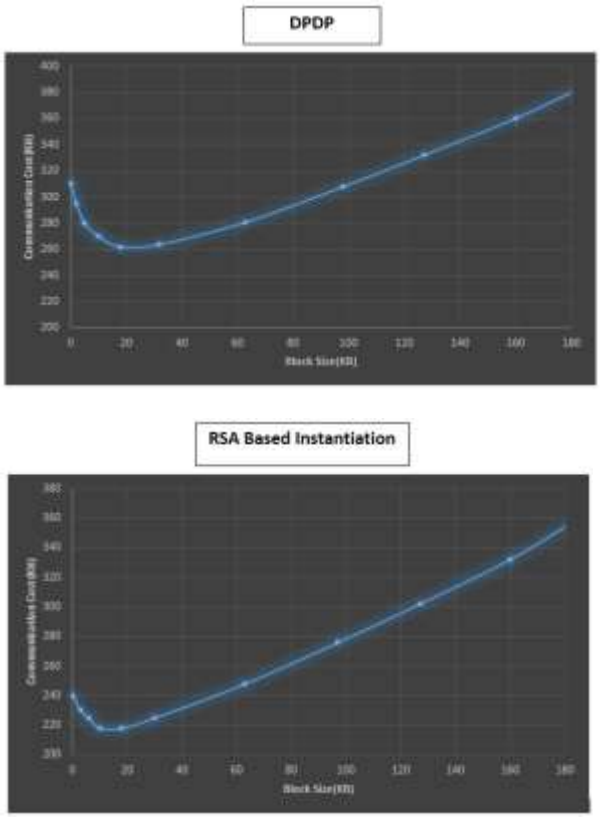


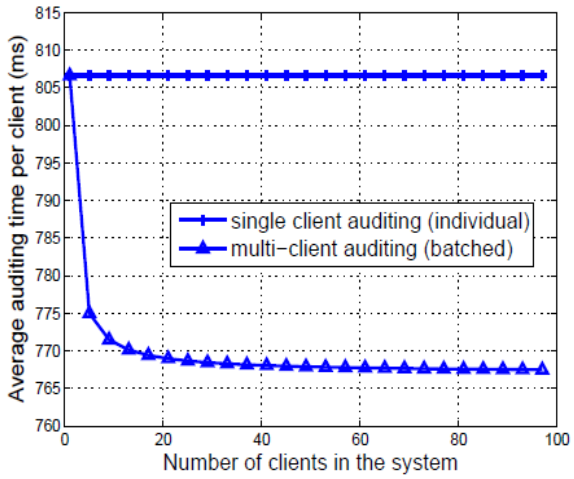
Fig.6: Comparison of communication complexity between our RSA-based instantiation and DPDP [14],

For 1 GB file with variable block sizes. The detection probability is maintained to be 99%.

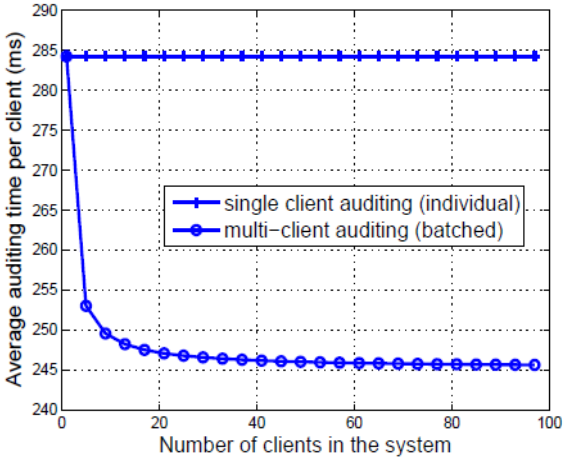
The communication overhead (server's response to the challenge) of our RSA-based instantiation and DPDP scheme [14] under different block sizes is illustrated in Fig. 6. We

can see that the communication cost grows almost linearly as the block size increases, which is mainly caused by the increasing in size of the verification block. However, the experiments suggest that when block size is chosen around 16KB, both schemes can achieve an optimal point that minimizes the total communication cost.

We also conduct experiments for multi-client batch auditing and demonstrate its efficiency in Figure 7, where the number of clients in the system is increased from 1 to approximately 100 with intervals of 4. As we can see, batch auditing not only enables simultaneously verification from multiple-client, but also reduces the computation cost on the TPA side. Given total K clients in the system, the batch auditing equation helps reduce the number of expensive pairing operations from 2K, as required in the individual auditing, to K + 1.



(a) Tolerance rate ρ is 99%.



(b) Tolerance rate ρ is 97%.

Fig. 7: Performance comparison between individual auditing and batch auditing. The average per client auditing time is computed by dividing total auditing time by the number of clients in the system. For both tolerance rate ρ = 99% and ρ = 97%, the detection probability is maintained to be 99%.

Thus, a certain amount of auditing time is expected to be saved. Specifically, following the same experiment setting as ρ = 99% and 97%, batch auditing indeed saves TPA's computation overhead for about 5% and 14%, respectively. Note that in order to maintain detection probability of 99%, the random sample size in TPA's challenge for ρ = 99% is quite larger than ρ = 97%: 460 versus 152. As this sample size is also a dominant factor of auditing time, this explains why batch auditing for ρ = 99% is not as efficient as for ρ = 97%.

VII. Conclusion

The cloud data security is much critical task. To ensure storage security in cloud computing; it is critical to enable Trusted third party to evaluate the service quality from an objective independent perspective.

In this paper, we explore the problem of providing trusted checking and integrity check in cloud storage. To ensure efficient data integrity, we improve existing proof of storage models by manipulating the MHT construction for authentication.

By using TPA we can easily secure all the data integrity operation on cloud server.

References

[1] Neil Roiter. How to secure cloud computing,Mar 2009. Available at: hTPA://searchsecurity.techtarget. com.

[2] Q. Zheng and S. Xu. Fair and dynamic proofs of retrievability. In Proceedings of the first ACM conference on Data and application security and privacy, pages 237–248. ACM, 2011.

[3] Karyn Benson, Rafael Dowsley, and Hovav Shacham. Do you know where your cloud files are? In Proceedings of the 3rd ACM workshop on Cloud computing security workshop, CCSW '11, pages 73–82, New York, NY, USA, 2011. ACM.

[4] Kevin D. Bowers, Marten van Dijk, Ari Juels, Alina Oprea, and Ronald L. Rivest. How to tell if your cloud files are vulnerable to drive crashes. In Proceedings of the 18th ACM conference on Computer and communications security, CCS '11, pages 501–514, New York, NY, USA, 2011. ACM.

[5] Jean-Philippe Aumasson, Aikaterini Mitrokotsa, and Pedro Peris-Lopez. A note on a privacy-preserving distance-bounding protocol. In Sihan Qing,Willy Susilo, GuilinWang, and Dongmei Liu, editors, Information and Communications Security, volume 7043 of Lecture Notes in Computer Science,pages 78–92. Springer Berlin - Heidelberg, 2011.

[6] 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. Saint Malo, France: Springer-Verlag, 2009, pp. 355–370.

[7] 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. New York, NY, USA: ACM, 2007, pp. 598–609.

[8] A. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in Proc. of CCS'07. New York, NY, USA: ACM, 2007, pp. 584–597.

[9] H. Shacham and B. Waters, "Compact proofs of retrievability," in Proc. of ASIACRYPT'08. Melbourne, Australia: Springer-Verlag, 2008, pp. 90–107.

[10] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: Theory and implementation," Cryptology ePrint Archive, Report 2008/175, 2008.

[11] M. Naor and G. N. Rothblum, "The complexity of online memory checking," in Proc. of FOCS'05, Pittsburgh, PA, USA, 2005, pp. 573–584.

[12] E.-C. Chang and J. Xu, "Remote integrity check with dishonest storage server," in Proc. of ESORICS'08. Berlin, Heidelberg: Springer-Verlag, 2008, pp. 223–237.

[13] M. A. Shah, R. Swaminathan, and M. Baker, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008.