IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727,
PP 71-76
www.iosrjournals.org

Privacy Preserving Secure auditing method For integrity with non-repudiation support

Jinsha K[1], Anver S R[2]
1(CSE Department, LBS institute of technology, Thiruvananthapuram, India)
2(CSE Department, LBS institute of technology, Thiruvananthapuram, India)

ABSTRACT: *Cloud computing is a well emerging technology with wide use. Its IaaS, PaaS, SaaS etc properties will attract users to use Cloud facility. Several security factors are made in to consider in this area. There may be possibility of integrity loss of outsourced data in cloud. So the need for the security of confidential data is increasing day by day. Here we introduce a new technique which will check the integrity of data with the help of a trusted third party auditor. It also provides the mechanism for verifying against non-repudiation attack.*

Keywords- *Bilinear mapping, Cloud Computing, Integrity, Non-repudiation, Public auditing*

I. INTRODUCTION

Cloud computing is a fast growing computing area, where users can remotely store their data into the cloud. The remote data enables the users to enjoy the on-demand high quality applications and services from a shared pool of configurable computing resources. Cloud computing provide several functionalities such as : on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [2] . So it is considered as a blooming technology in IT sector. One fundamental benefit of cloud computing is that Storage management and safety.ie. The huge amount of data in user system may cause difficulty in storage and safety management of data. In outsourced cloud data enable provision for solution for all these problems. It also benefits users by universal data access by storage data in geographically different areas and thus it avoids expenditure of hardware cost software cost and security cost etc. The NIST defined cloud computing as "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, and services) that can be rapidly provisioned and released with minimal management effort of service provider interaction".[1] Cloud storage denotes a family of increasingly popular on-line services for archiving, backup, and even primary storage of files. Amazon S3 [3] is a well-known example. Cloud-storage providers offer users clean and simple file-system interfaces, abstracting away the complexities of direct hardware management. At the same time, though, such services eliminate the direct oversight of component reliability and security that enterprises and other users with high service-level requirements have traditionally expected. It is efficient and cost effective i.e. the customer put certain money for the service and the service granted according to cost and service rule. The user need not worry about the data that he has been stored in cloud because the service provider handle the storage capacity. It allows users to store and access their data stored in another location without worry of loss from their system. The CSP offers adequate hardware, software and network resources to host owners' personal needs. It also provide mechanisms for efficiently create, update and access outsourced data. Here the different registered authenticated users can share and use the data. If the users system crashes then the data is safe in the cloud so no need of worry about losing data.

The characteristics of cloud computing include [4]

Elasticity is a main feature for cloud systems which defines the underlying infrastructure capability to adapt to changing requirements such as amount and size of data used in an application. Cloud computing involves two types of vertical and horizontal scalability. The vertical scalability refers to the size of the instances and implicit to the amount of resources which are required for maintaining the size. But, horizontal scalability denotes the amount of instances to satisfy changing amounts of requests.

Reliability is the capability of ensuring the continuity of the system operation without disruption such as loss of data or code reset during execution.

Quality of Service (QoS) support is vitally important for specific requirements which should be met through the provided services or resources.

Agility and adaptability are two key features of great concern to cloud systems relevant to the elastic capabilities. They refer to on-time reaction to changes in the size of resources and the amount of requests as well as adaptation to changes according to the conditions of environment.

Availability of services lies in the ability of providing redundant services and data to mask failures transparently. Fault tolerance also needs this ability to introduce new redundancy such as fresh or previously failed nodes, in an online fashion without or with a little performance penalty. With the increase of simultaneous access, availability is attained through replication of services or data and dis- emanating them across various resources.

Cloud can be served as Infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and/or software-as-a-service (SaaS) etc. It has several advantages include on demand service, ubiquitous network access, location independent resource pooling rapid resource elasticity and usage-based pricing etc.

1.2 Cloud and Security

While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users' outsourced data. The Difference in infrastructure lead to inability of traditional security methods for cloud safety.

For the issue of data integrity, an obvious mechanism is checking on retrieval which is impossible to assure integrity for the whole data because huge amount of data are outsourced to the cloud and only few are frequently accessed. Since most of the data are accessed less frequently (or not at all), this implies the integrity check of such data is bypassed by this technique. Another technique is to download the whole data and perform the integrity check which is impractical due to heavy I/O overhead on the cloud server and high communication cost for the owner.

Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user's ultimate control over the fate of their data. As a result, the correctness of the the cloud is being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time [5]. Secondly, there do exist various motivations for CSP to behave unfaithfully towards the cloud users regarding the status of their outsourced data. For examples, CSP might reclaim storage for monetary reasons by discarding data that has not been or is rarely accessed, or even hide data loss incidents so as to maintain a reputation [5]. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale data storage, it does not immediately offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture.

To fully ensure the data integrity, availability and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third party auditor (TPA) to audit the outsourced data when needed.

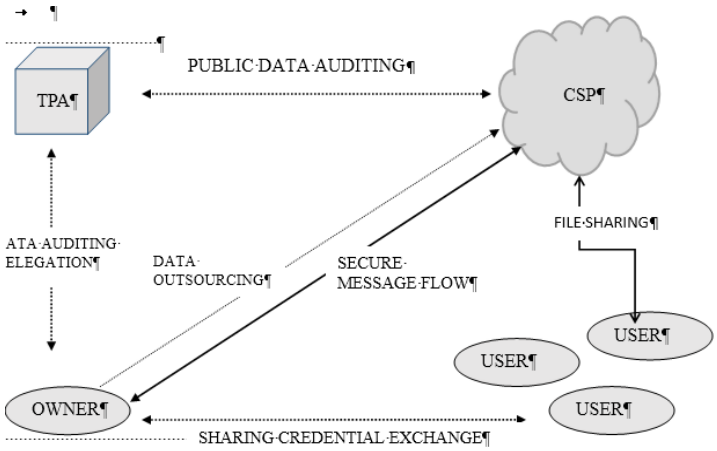


Figure 1: Architecture of Cloud system

II. LITERATURE SURVEY

Recently there are a lot of work has been conducted to maintain the security of outsourced data. Atenieseet 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.

The PDP method is developed for checking how frequently, effectively and securely verifying that the CSP is faithfully storing its client's potentially very large amount of data – is not cheating the client. In this context, cheating means that the server might delete some of the data or it might not store all data in fast storage, e.g., place it on CDs or other tertiary off-line media. It is important to note that a storage server might not be malicious; instead, it might be simply unreliable and lose or inadvertently corrupt hosted data. An effective PDP technique must be equally applicable to malicious and unreliable servers. The problem is further complicated by the fact that the client might be a small device (e.g., a PDA or a cell-phone) with limited CPU, battery power and communication facilities. Hence, the need to minimize bandwidth and local computation overhead for the client in performing each verification.

The provable data possession method is take the advantage of reducing communication overhead by sending small amount of data. It uses public key based technique. Public key- based technique allowing any verifier (not just the client) to query the server and obtain an interactive proof of data possession. It uses a RSA based homomorphic verifiable tag for each block. The homomorphic verifiable tag means tag for a single block m1 is t1 and t2 for another m2 then t1+t2 be the tag for m1+m2.

Here initially client computes the tag for each single block and outsource the data along with the tag set to the server. Since each client can verify the data possession all calculations are made in client itself. No privacy is guaranteed for user data and the data checked for integrity is also limited for certain random blocks. They do not consider dynamic data storage and direct extension of their scheme from their static data storage to dynamic data storage may suffer security and design problems.

Schacham and waters proposed compact proof of retrievability method. In a proof-of-retrievability system, a data storage center convinces a verifier that he is actually storing all of a client's data. The central challenge is to build systems that are both client and provably secure that is, it should be possible to extract the client's data from any prover that passes a verification check.

It uses spot-checking and error-correcting codes are used to ensure both "possession" and "retrievability" of data files on archive service systems. Specifically, some special blocks called "sentinels" are randomly embedded into the data file F for detection purpose, and F is further encrypted to protect the positions of these special blocks. The sentinals are random message block encrypted by a secret key. During the verification phase, the client asks for randomly picked sentinels and checks whether they are intact. If the server modifies or deletes parts of the data, then sentinels would also be affected with a certain probability. However, sentinels should be in distinguishable from other regular blocks; this implies that blocks must be encrypted. Thus, unlike the PDP scheme in [6], POR cannot be used for public databases, such as libraries, repositories, or archives. In other words, its use is limited to confidential data the number of queries a client can perform is also a fixed priori, This is because sentinels, and their position within the database, must be revealed to the server at each query – a revealed sentinel cannot be reused and the introduction of pre-computed "sentinels "prevents the development of realizing dynamic data updates. In addition, public auditability is not supported in their scheme.

An improved PoR scheme with full proofs of security in the security model defined in [7]. They use publicly verifiable homomorphic authenticators built from BLS signatures ,based on which the proofs can be aggregated into a small authenticator value, and public retrievability is achieved. Still, the authors only consider static data files and also it creates computational overhead and communication overhead to user.

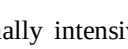
Wang's Enabling Public auditability and Data dynamics for Storage security in cloud explains the problem of ensuring the integrity of data storage in Cloud Computing. In particular, it consider the task of allowing a third party auditor (TPA), on behalf of the cloud client, to verify the integrity of the dynamic data stored in the cloud. The introduction of TPA eliminates the involvement of the client through the auditing of whether his data stored in the cloud is indeed intact, which can be important in achieving economies of scale for Cloud Computing. The support for data dynamics via the most general forms of data operation, such as block 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.

It provide data error localization which means it can identify the data errors easily. It uses BLS(Boneh-Lynn-Schacham) based signature scheme as tags. BLS is a scheme which selects x ε [0, n] as a private key and public key as g^x . Then sign is generated as hashing h = H(m) and the sign σ = h^x .The verification process is done as, $e(\sigma, g) = e(H(m), g^x)$. It also uses HLA homomorphic linear authentication and Merkle Hash Tree for data dynamics.

It does not provide privacy of data. Later they modified the paper by providing privacy of data from third party auditor [9]. Then they further modified the paper [10] with privacy protection but it has several drawback [11]. The cryptanalysis of the paper is done in [11].It finds that this protocol is vulnerable to attacks from a malicious cloud server and an outside attacker. The malicious cloud server can modify the outsourced data as he wants when he possesses it. Moreover, the malicious cloud server can pass the auditing from the TPA when he loses the outsourced data. Besides the attacks from the malicious cloud server, find that this protocol is vulnerable to attacks from an outside attacker. Even if the cloud server is trusted, the outside attacker can intercept the data sent from the user to the cloud server in TagBlock step and modify it arbitrarily. Furthermore, the outside attacker can just eavesdrop on that data and forge a great deal of data.

First consider Data modification Tag forging attack. We assume that the malicious cloud server wants to modify mj in F to mj*which is chosen as his wish. The malicious cloud server can modify the data mj and forge its corresponding tag σ j so that they will be able to pass the auditing from the TPA. After the user finishes KeyGen and TagBlock, the malicious cloud server computes σj*=σj. $W^{mj^*-mj}=((H(j).u^{mj^*})^x$ using the data mj he possesses, tag σ j and public key w. Then the malicious cloud server modifies mj to mj∗ and σ jto σj*. Upon receiving challenge chal={(i,vi)}i∈I , the malicious cloud server computes r = fk (chal)and R = $(w)^r= (u^x)^r$ as normal. However, he computes μ ′* = $\Sigma_{i\in I}$ vimi*,σ* =$\Pi_{i\in I}\sigma^{vi^*}$, μ * =μ ′* + rh(R) differently. Finally, he sends {μ *,σ *,R} to the TPA as a new response to the challenge. Upon receiving the response {μ *,σ *,R} , it is clear that

the TPA verifies equation, equal.

From the above attack scheme, we can see that the malicious cloud server can modify the outsourced data as he wants when he possesses the outsourced data and the TPA can't find this modification. Similarly data lost auditing pass attack, data interception and modification attack and data eavesdropping and forgery are also possible [11]. Later Wang proposed a paper saying that it overcome all the problems. But it again lack of dynamic data manipulation and complexity in signature calculation. Because it need to calculate and publish  a computationally intensive task [12] as part of its public parameter and this make the scheme inefficient.

III. PROPOSED SCHEME

3.1 Problem Statement

System model

- This proposed scheme consists of three different entities and interactions among them.
- Cloud server which is owned by CSP, has the infrastructure and used to host outsourced storage, and provides efficient mechanisms for its users to create, store, update and request for retrievability.
- User (client), who has data to be stored on the cloud, can operate on hi/her outsourced data.
- Third party auditor, another entity who has better expertise and capabilities than the user, is trusted to measure the cloud storage reliability and validity on behalf of users when needed

The proposed system has several advantages over the paper reviewed in the case of integrity protection and possibility of non -repudiation service. Our contribution basically relay on the Wangs' concept. But his paper has several drawback regarding the security and computation and it will not give any provision for the non-repudiation service. So here we introduce a new technique which reduce the computation overhead of the sender while calculating the tag and support non-repudiation.

It uses basic mathematical functions such as group, abelian group, and bilinear mapping function.

Group(G,*) is a set of elements together with a binary operation defined on it and satisfies the property of closure, associatively , identity and inverse.

Abelian Group(G, *) is a group which satisfies commutative property in operation *.
Bilinear Mapping (G1, *), (G2, *) and (GT , *) are three cyclic groups of prime order p; g1 is a generator of G1 and g2 is a generator of G2; e is a bilinear map e : G1 ×G2→GT , i.e., a map satisfying the following properties:
Bilinearity : ∀u ∈ G1, ∀v ∈ G2, ∀a, b ∈ Z, $e(u^a, v^b) = e(u, v)^{ab;}$
Non-degeneracy: e(g1, g2) ≠ 1 and is thus a generator of GT .

F—the data file to be outsourced, denoted as a sequence of n blocks m1,m2 . . .m_n € Zp for some large prime p and uses basic hash functions H(.)

Our system involves basic algorithms include keygeneration(), Challenge(), Verify() where the key generation is used both auditability and non repudation verification. KeyGen(1k). The client first generates a random signing key pair (ssk, spk). Then chooses x in Zp, u in G and computes $v = g^x$. User then states, sk = (x, ssk) as his/her secret key and pk = (u, v, g, spk) as public parameters. SigGen(sk, F), for file naming, client chooses a random element name in Z_p for file F = {mi}1<i<n and computes the file tag as t = name||$Sign_{ssk}$(name) with signature on name. Next, for each block mi in Zp, user generates a signature as follows:

$$\sigma_i = (H(W_i) \cdot u^{m_i})^x \quad (1)$$

Where Wi =name||i .Then finally, user sends {F,{ σ_i }, t} to the server for storage and deletes the file and its corresponding set of signatures from local storage. Any time when TPA wants to start the auditing protocol, first s/he retrieves the file tag t for F and checks its validity using spk, and quits if fail. If the proof on t is correct, the next step for client or TPA is to construct and send an challenge chal to server. That is, TPA picks random elements c, k1, k2 in Zp and sends chal = (c, k1, k2) to the server where k1 and k2 are pseudorandom permutation keys chosen randomly by user for each auditing.

After receiving the challenge, the server first determines the subset I, Sj using pseudorandom permutation and it also determines value v , R^{μ} , for random masking μ* changed to μ by adding random values as μ = μ* + r h(R) .It also calculates summation of selected signature. Then the TPA request server to verify and generate proof. The TPA after receiving the proof check for validity.

$$e(\sigma, g) \Leftrightarrow e(\prod_{i \in I} H(Wi)^{vi} u^{\mu} R^{-h(R)}, v) \quad (2)$$

The same also used for repudiation attack checking and confirming the user and receiver that no non-repudiation event has occurred. The method used public key cryptography by which both sever and user share a public key.

IV. EXPERIMENT RESULTS

The experiment is conducted to find whether it is working correctly. The results shows the well performance and result with great detection of modification in data. The new theory can be proved theoretically by using public key scheme. The keys used are very secure and it provide better protection against the most common attacks. The formal verification of the method shows an improved computation overhead. The security can be analyzed by using Computational DH theory and properties of discrete logarithm. The verification with the help of this mathematical theories shows the same security of data as before but with improved performance. The repudiation attack is prevented with the help of unique identity signature attached with each request. The non-repudiation technique also shows better security.

V. CONCLUSION

Cloud Computing as the on-demand and remote provision of computational resources has been eagerly waited for a long time as a computing utility. It helps users to store their data in the cloud and enjoy the high quality service. However, users do not have physical possession on their own data, hence it is indispensable to create mechanisms on how to protect the security of the data stored. Thus, some auditing protocols are introduced to ensure authenticity and integrity of the outsourced data. There are several mechanisms are proposed to ensure the security of data. Here we further try to improve mechanism for auditing with security against non-repudiation attack as our future work.

REFERENCES

[1]. 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.
[2]. 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. UCB-EECS-2009-28, Feb 2009
[3]. J. Hendricks, G. R. Ganger, and M. K. Reiter. Verifying distributed erasure-coded data. In 26th ACM Symposium on Principles of Distributed Computing(PODC), 2007.
[4]. Ateniese G, Burns R, Curtmola R, Herring J, Khan O, Kissner L, et al. Remote data checking using provable data possession. ACM Transaction in Information System Security 2011;14:1–34
[5]. Cong Wang, Kui Ren, Jin Li, "Toward publicly Auditable Secure cloud data storage Service", IEEE Network august 2010
[6]. H. Shacham and B. Waters, "Compact proofs of retrievability,"in Proceedings. of Asiacrypt 2008, vol. 5350, Dec 2008, pp. 90–107
[7]. Juels and B. S. Kaliski, Jr., "Pors: proofs of retrievability for large files," in Proceedings of CCS'07. New York, NY, USA: ACM, 2007,pp. 584–597.
[8]. Wang C, Chow S, Wang Q, Ren K, Lou W. "Enabling public auditability and Data dynamics for storage security in cloud computing". IEEE Transaction in cloud-computing 2010.
[9]. Cong Wang, Kui Ren, Jin Li, "Privacy Preserving public Auditing for data storage Security in Cloud Computing", IEEE INFOCOM august 2011
[10]. Wang C, Chow S, Wang Q, Ren K, Lou W. "Privacy-preserving public auditing for secure cloud storage". IEEE Transaction in cloud-computing 2013.
[11]. Chunxiang X, Xiaohu H, Daniel-Abraha W. "Cryptanalysis of Wang's auditing protocol for data storage security in cloud computing". In: Liu C et al.,editors. Information computing and applications, vol. 308. Berlin Heidelberg: Springer; 2012. p. 422–8
[12]. John Bethencourt, "Bilinear Maps",Computer Sciences Department, Carnegie Mellon University, http://courses.csail.mit.edu/6.897/springer04/L25.pdf.