ANALYSIS OF AN OPTIMIZED SECURE AUDITING PROTOCOL FOR STORING DATA DYNAMICALLY IN CLOUD COMPUTING

Gurpreet Singh and Raman Kumar
Department of Computer Science and Engineering
D A V Institute of Engineering and Technology, Jalandhar, Punjab, India.
er.ramankumar@aol.in

ABSTRACT: The studies show that the use of internet is endless. In these days information flows across the internet is not as secure as we think. As the cloud computing is getting attention the risks to the data which flows over cloud and stored over cloud are also rises. We focus on the pseudonym based auditing and saving auditing storage and avoid different attacks. We will process the operations based on the traces generation but storing the trace initial data in header so that it will not create overhead to running processes. Network Simulator may be used for experimentation with dense network. Cloud server structure will be opted for experimentation with various users.

1 INTRODUCTION

The studies show that the use of the internet is endless. In these days, information flows across the internet is not as secure as we think. As the cloud computing is getting attention, the risks to the data which flows over the cloud and stored over cloud are also rises. We focus on the pseudonym based auditing and saving auditing storage and avoid different attacks. We will process the operations based on the traces generation but storing the trace initial data in a header so that it will not create overhead to running processes. Network Simulator may be used for experimentation with a dense network. Cloud server structure will opt for experimentation with various users.

1.1 Essential Characteristics:

On-demand self-service. On-demand service is the methodology which gives rights to the user to request recourses on the run time and this type of transition happens immediately, although it depends on the cloud provider, what type of architecture and recourses they have.

Broad network access. Broad network access means that access to the private cloud can be enhanced to the other level like employees can use Smartphone, tablets and other devices to access the company recourses and work upon them through those devices.

Resource pooling. The cloud provider pooled the lot of resources on their side over the cloud. They authorize the uses to request those available resources at any time like on-demand, self-service, customer could alter their level of service at any time without even interacting with the cloud provider.

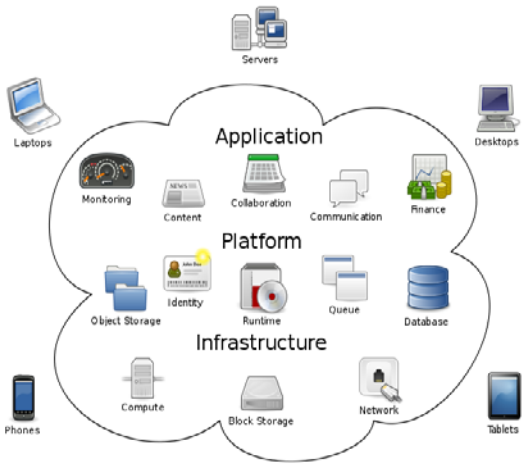

Figure-1: Overall view of cloud computing

Rapid elasticity. Rapid elasticity means consumers can request additional space in the cloud and other type of services as per their requirements and administering the multiple requests is a demanding and require precise administration.

Measured service. The services which are provided by the cloud provider and the services requested by the customer on on-demand self-service bases can be measured by the mechanism. Which maintain the transparency between the service provider and the user

1.2 Service Models:

Software as a Service (SaaS). This capability provided to the user and user can access the programs like browser or other type of programs which are at the cloud side. The applications are accessible from various client devices through either a thin client interface, such as a web browser (e.g., web-based email), or a program interface.

Platform as a Service (PaaS). The capability provided to the consumer is to deploy onto the cloud infrastructure consumer-created or acquired applications created using programming languages, libraries, services, and tools supported by the provider.

Infrastructure as a Service (IaaS). This capability provided to the customer to access the resources like operating systems, storage, and deployed applications; and possibly limited control of select networking components (e.g., host firewalls).

1.3 CLOUD COMPUTING COMMUNICATION

Cloud computing is a promising computing model that gives convenient and on-demand network access to a computing resources those are pooled over the cloud. Cloud storage service in cloud computing allow users to move their local computing system data and store over the cloud. More and more data owners start choosing to host their data in the Cloud [3]. The local management of such huge amount of data is problematic and costly due to the requirements of high storage capacity and qualified personnel. Therefore, Storage-as-a-Service offered by cloud service providers (CSPs) emerged as a solution to mitigate the burden of large local data storage and reduce the maintenance cost by means of outsourcing data storage [4]. Cloud Storage Providers like Microsoft with Sky Drive, Google Documents, and Drop Box etc have successfully dropped rates of storage available on internet. They promise availability of the data from different systems/locations/networks. Basic security like User based authentication access of data and maintaining offline data to the client's machine is also supported [4].

1.4 EFFICIENT STORAGE DYNAMICALLY

Storage in cloud is an important service of cloud computing, which allows data owners to move data from their local computing systems to the cloud. More and more owners start to store the data in the cloud [2]. However, this new paradigm of data hosting service also introduces new security challenges [3]. Owners would worry that the data could be lost in the cloud. This is because data loss could happen in any infrastructure, no matter what high degree of reliable measures cloud service providers would take. Sometimes, cloud service providers might be dishonest. They could discard the data that have not been accessed or rarely accessed to save the storage space and claim that the data are still correctly stored in the cloud. Therefore, owners need to be convinced that the data are correctly stored in the cloud. Recently, several remote integrity checking protocols were proposed to allow the auditor to check the data integrity on the remote server. Many authors proposed a dynamic auditing protocol that can support the dynamic operations of the data on the cloud servers, but this method may leak the data content

to the auditor because it requires the server to send the linear combinations of data blocks to the auditor.

2 PREVIOUS WORK

Kan Yang et al. (2013) [1] In cloud computing, data owners host their data on cloud servers and users (data consumers) can access the data from cloud servers. Due to the data outsourcing, however, this new paradigm of data hosting service also introduces new security challenges, which requires an independent auditing service to check the data integrity in the cloud. Some existing remote integrity checking methods can only serve for static archive data and, thus, cannot be applied to the auditing service since the data in the cloud can be dynamically updated. Thus, an efficient and secure dynamic auditing protocol is desired to convince data owners that the data are correctly stored in the cloud. In this paper, They first design an auditing framework for cloud storage systems and propose an efficient and privacy-preserving auditing protocol. Then, they extend their auditing protocol to support the data dynamic operations, which is efficient and provably secure in the random oracle model. They further extend their auditing protocol to support batch auditing for both multiple owners and multiple clouds, without using any trusted organizer. The analysis and simulation results show that their proposed auditing protocols are secure and efficient, especially it reduce the computation cost of the auditor.

M. Lillibridge et al. (2003) [2] They present a novel peer-to-peer backup technique that allows computers connected to the Internet to back up their data cooperatively: Each computer has a set of partner computers, which collectively hold its backup data. In return, it holds a part of each partner's backup data. By adding redundancy and distributing the backup data across many partners, a highly-reliable backup can be obtained in spite of the low reliability of the average Internet machine. Because their scheme requires cooperation, it is potentially vulnerable to several novel attacks involving free riding (e.g., holding a partner's data is costly, which tempts cheating) or disruption. They defend against these attacks using a number of new methods, including the use of periodic random challenges to ensure partners continue to hold data and the use of disk space wasting to make cheating unprofitable. Results from an initial prototype show that their technique is feasible and very inexpensive: it appears to be one to two orders of magnitude cheaper than existing Internet backup services.

Y. Deswarte et al. (2004) [3] This paper analyzes the problem of checking the integrity of files stored on remote servers. Since servers are prone to successful attacks by malicious hackers, the result of simple integrity checks run on the servers cannot be trusted. Conversely, downloading the files from the server to the verifying host is impractical. Two solutions are proposed, based on challenge response protocols.

M. Naor et al. (2009) [4] They consider the problem of storing a large file on a remote and unreliable server. To verify that the file has not been corrupted, a user could store a small private (randomized) "finger- print" on his own computer. This is the setting for the well-studied authentication problem in cryptography, and the required fingerprint size is well understood. They study the problem of sublinear authentication: suppose the user would like to encode and store the file in a way that allows him to verify that it has not been corrupted, but without reading the entire file. If the user only wants to read q bits of the file, how large does the size s of the private fingerprint need to be? They define this problem formally, and show a tight lower bound on the relationship between s and q when the adversary is not computationally bounded, namely: $s \times q = \Omega(n)$, where n is the file size. This is an easier case of the online memory checking problem, introduced by Blum et al. in

1991, and hence the same (tight) lower bound applies also to that problem. It was previously shown that when the adversary is computationally bounded, under the assumption that one-way functions exist, it is possible to construct much better online memory checkers. T he same is also true for sub-linear authentication schemes. They show that the existence of one-way functions is also a necessary condition: even slightly breaking the $s \times q = \Omega(n)$ lower bound in a computational setting implies the existence of one-way functions..

A. Juels et al. (2007) [5] In this paper, they define and explore proofs of retrievability(PORs). A POR scheme enables an archive or back-up service (prover) to produce a concise proof that a user (verifier) can retrieve a target file F, that is, that the archive retains and reliably transmits file data sufficient for the user to recover F in its entirety. A POR may be viewed as a kind of cryptographic proof of knowledge (POK), but one specially designed to handle a large file (or bitstring) F. they explore POR protocols here in which the communication costs, number of memory accesses for the prover, and storage requirements of the user (verifier) are small parameters essentially independent of the length of F. In addition to proposing new, practical POR constructions, they explore implementation considerations and optimizations that bear on previously explored, related schemes. In a POR, unlike a POK, neither the prover nor the verifier need actually have knowledge of F. PORs give rise to a new and unusual security definition whose formulation is another contribution of their work. They view PORs as an important tool for semi-trusted online archives. Existing cryptographic techniques help users ensure the privacy and integrity of files they retrieve. It is also natural, however, for users to want to verify that archives do not delete or modify files prior to retrieval. The goal of a POR is to accomplish these checks without users having to download the files themselves. A POR can also provide quality of service guarantees, i.e., show that a file is retrievable within a certain time

T.J.E. Schwarz et al. (2006) [6] The emerging use of the Internet for remote storage and backup has led to the problem of verifying that storage sites in a distributed system indeed store the data; this must often be done in the absence of knowledge of what the data should be. They use m/n erasure correcting coding to safeguard the stored data and use algebraic signature-hash functions with algebraic properties for verification. Their scheme primarily utilizes one such algebraic property: taking a signature of parity gives the same result as taking the parity of the signatures. To make their scheme collusion resistant, They blind data and parity by XORing them with a pseudo-random stream. Their scheme has three advantages over existing techniques. First, it uses only small messages for verification, an attractive property in a P2P setting where the storing peers often only have a small up- stream pipe. Second, it allows verification of challenges across random data without the need for the challenger to compare against the original data. Third, it is highly resistant to coordinated attempts to undetectably modify data. These signature techniques are very fast, running at tens to hundreds of megabytes per second. Because of these properties, the use of algebraic signatures will permit the con- struction of large scale distributed storage systems in which large amounts of storage can be verified with minimal network bandwidth.

D.L.G Filho et al. (2006) [7] They observe that a certain RSA-based secure hash function is homomorphic. They describe a protocol based on this hash function which prevents ‘cheating’ in a data transfer transaction, while placing little burden on the trusted third party that oversees the protocol. They also describe a cryptographic protocol based on similar principles, through which a prover can

demonstrate possession of an arbitrary set of data known to the verifier. The verifier isn't required to have this data at hand during the protocol execution, but rather only a small hash of it. The protocol is also provably as secure as integer factoring.

F. Sebe et al. (2008) [8] Checking data possession in networked information systems such as those related to critical infrastructures (power facilities, airports, data vaults, defence systems, and so forth) is a matter of crucial importance. Remote data possession checking protocols permit checking that a remote server can access an uncorrupted file in such a way that the verifier does not need to know beforehand the entire file that is being verified. Unfortunately, current protocols only allow a limited number of successive verifications or are impractical from the computational point of view. In this paper, they present a new remote data possession checking protocol such that

1) it allows an unlimited number of file integrity verifications and

2) its maximum running time can be chosen at set-up time and traded off against storage at the verifier.

C. Wang et al. (2010) [9] Cloud computing is the long dreamed vision of computing as a utility, where data owners can remotely store their data in the cloud to enjoy on-demand high-quality applications and services from a shared pool of configurable computing resources. While data outsourcing relieves the owners of the burden of local data storage and maintenance, it also eliminates their physical control of storage dependability and security, which traditionally has been expected by both enterprises and individuals with high service-level requirements. In order to facilitate rapid deployment of cloud data storage service and regain security assurances with outsourced data dependability, efficient methods that enable on-demand data correctness verification on behalf of cloud data owners have to be designed. In this article they propose that publicly auditable cloud data storage is able to help this nascent cloud economy become fully established. With public auditability, a trusted entity with expertise and capabilities data owners do not possess can be delegated as an external audit party to assess the risk of outsourced data when needed. Such an auditing service not only helps save data owners' computation resources but also provides a transparent yet cost-effective method for data owners to gain trust in the cloud. They describe approaches and system requirements that should be brought into consideration, and outline challenges that need to be resolved for such a publicly auditable secure cloud storage service to become a reality.

G. Ateniese et al. (2007) [10] They introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely distributed storage systems. They present two provably secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using their implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

M.A. Shah et al. (2008) [11A growing number of online service providers offer to store customers' photos, email, file system backups, and other digital assets. Currently, customers cannot make

informed decisions about the risk of losing data stored with any particular service provider, reducing their incentive to rely on these services. They argue that third-party auditing is important in creating an online service-oriented economy, because it allows customers to evaluate risks, and it increases the eficiency of insurance based risk mitigation. They describe approaches and system hooks that support both internal and external auditing of online storage services, describe motivations for service providers and auditors to adopt these approaches, and list challenges that need to be resolved for such auditing to become a reality.

C.C. Erway et al. (2009) [12] As storage outsourcing services and resource sharing networks have become popular, the problem of efficiently proving the integrity of data stored at untrusted servers has received increased attention. In the provable data possession (PDP) model, the client pre-processes the data and then sends it to an untrusted server for storage, while keeping a small amount of metadata. The client later asks the server to prove that the stored data has not been tampered with or deleted (without downloading the actual data). However, the original PDP scheme applies only to static (or append-only) files. They present a definitional framework and efficient constructions for dynamic provable data possession (DPDP), which extends the PDP model to support provable updates to stored data. They use a new version of authenticated dictionaries based on rank information. The price of dynamic updates is a performance change from O(1) to O(logn) (or O(nǫ logn)), for a file consisting of n blocks, while maintaining the same (or better, respectively) probability of misbehavior detection. Their experiments show that this slowdown is very low in practice (e.g., 415KB proof size and 30ms computational overhead for a 1GB file). They also show how to apply their DPDP scheme to outsourced file systems and version control systems (e.g., CVS).

G. Ateniese et al. (2008) [16] Storage outsourcing is a rising trend which prompts a number of interesting security issues, many of which have been extensively investigated in the past. However, Provable Data Possession (PDP) is a topic that has only recently appeared in the research literature. The main issue is how to frequently, efficiently and securely verify that a storage server is faithfully storing its client (potentially very large) has outsourced data. The storage server is assumed to be untrusted in terms of both security and reliability. (In other words, it might maliciously or accidentally erase hosted data; it might also relegate it to slow or off-line storage.) The problem is exacerbated by the client being a small computing device with limited resources. Prior work has addressed this problem using either public key cryptography or requiring the client to outsource its data in encrypted form. In this paper, they construct a highly efficient and provably secure PDP technique based entirely on symmetric key cryptography, while not requiring any bulk encryption. Also, in contrast with its predecessors, their PDP technique allows outsourcing of dynamic data, i.e, it efficiently supports operations, such as block modification, deletion and append.

3 PROPOSED WORK

Related scheme has developed a method for secure and optimal auditing in storage while using cloud services. There are many attacks that can breach into storage credentials and some of the dangerous attack are reply and forge attack and to prevent the replay attack, related paper introduce an index table to record the abstract information of the data. The Index denotes the current block number of data block m_i in the data component M.Dat tags are generated based on time stamps provided by solution. This ITable is created by the owner during the owner initialization and managed by the auditor. When the owner completes the data

dynamic operations, it sends an update message to the auditor for updating the ITable that is stored on the auditor. After the confirmation auditing, the auditor sends the result to the owner for the confirmation that the owner's data on the server and the abstraction information on the auditor are both up-to-date. This completes the data dynamic operation. This work is very much prevent the attack but a slighter limitation is the storage while performing operations which are critical so in this research will purpose a lighter detect based TagGen and Itable process which will be carry in packet header itself with 2 bit of information.

For proposed work, focus will be on pseudonym based auditing and saving auditing storage against different attacks. In auditing process, the auditing protocol only involves two-way communication: Challenge and Proof. During the confirmation auditing phase, the owner requires the auditor to check whether the owner's data are correctly stored on the server. Further auditing data will be updated according to the required storage solutions. The auditor then sends the auditing result to the owner. If the result is true, the owner is convinced that its data are correctly stored on the server, and it may choose to delete the local version of the data. The pseudonyms will compose of public key, private key, and a certificate will be used for efficient preservation of privacy. Users can be assured of their anonymity through pseudonym and authenticated as normal users through a certificate. The TTP stores pseudonyms and actual ID of users to reveal the anonymity in case of a problem and it is also used for storing results for whole auditing process. Then process the operations based on the traces like Itable generation and TagGen generation but storing the trace initial data in header so that it will not creat overhead to running processes. Network Simulator will be used for experimentation with dense network. Cloud server structure will be opted for experimentation with various users.

5. CONCLUSION

Focus will be on pseudonym based auditing and saving auditing storage against different attacks. We will process the operations based on the traces generation but storing the trace initial data in header so that it will not create overhead to running processes. Network Simulator will be used for experimentation with dense network. Cloud server structure will be opted for experimentation with various users.

REFERENCES

[1] Kan Yang, Student Member, IEEE, and Xiaohua Jia, Fellow, IEEE "An Efficient and Secure Dynamic Auditing Protocol for Data Storage in Cloud Computing," IEEE Transaction on Parallel And Distributed Systems, VOL. 24, NO. 9, Sep 2013

[2] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," Proc. USENIX Ann. Technical Conf., pp. 29-41, 2003.

[3] Y. Deswarte, J. Quisquater, and A. Saidane, "Remote Integrity Checking," Proc. Sixth Working Conf. Integrity and Internal Control in Information Systems (IICIS), Nov. 2004.

[4] M. Naor and G.N. Rothblum, "The Complexity of Online Memory Checking," J. ACM, vol. 56, no. 1, article 2, 2009.

[5] A.Juelsand B.S. KaliskiJr., "Pors:Proofs of Retrievabilityfor Large Files," Proc. ACM
Conf. Computer and Comm. Security, P. Ning, S.D.C. di Vimercati, and P.F. Syverson,

eds., pp. 584-597, 2007.

[6] T.J.E. Schwarz and E.L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," Proc. 26th IEEE Int'l Conf. Distributed Computing Systems, p. 12, 2006.

[7] D.L.G. Filho and P.S.L.M. Barreto, "Demonstrating Data Posses- sion and Uncheatable Data Transfer," IACR Cryptology ePrint Archive, vol. 2006, p. 150, 2006.

[8] F. Sebe ´, J. Domingo-Ferrer, A. Martı ´nez-Balleste ´, Y. Deswarte, and J.-J.Quisquater, "Efficient Remote Data Possession Checking in Critical Information Infrastructures," IEEE Trans. Knowledge Data Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.

[9] C. Wang, K. Ren, W. Lou, and J. Li, "Toward Publicly Auditable Secure Cloud Data Storage Services," IEEE Network, vol. 24, no. 4, pp. 19-24, July/Aug. 2010.

[10] G. Ateniese, R.C. Burns, R. Curtmola, J. Herring, L. Kissner, Z.N.J. Peterson, and D.X. Song, "Provable Data Possession at Untrusted Stores," Proc. ACM Conf. Computer and Comm. Security, P. Ning, S.D.C. di Vimercati, and P.F. Syverson, eds., pp. 598-609, 2007.

[11] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, J. Pieprzyk, ed., pp. 90-107, 2008.

[12] C.C. Erway, A. Ku ¨pc¸u ¨, C. Papamanthou, and R. Tamassia, "Dynamic Provable Data Possession," Proc. ACM Conf. Computer and Comm. Security, E. Al-Shaer, S. Jha, and A.D. Keromytis, eds., pp. 213-222, 2009.

[13] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage from Homomorphic Identification Protocols," Proc. Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology, M. Matsui, ed., pp. 319-333, 2009.

[14] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," technical report, Nat'l Inst. of Standards and Technology, 2009.

[15] Dhiraj Soni and Raman Kumar, "Multimedia Cloud Computing Security an Evolving Trend: Rudimentary Essentials Computing", International Journal of Advanced Trends in Computer Applications, IJATCA, Vol. 2, Issue 7, pp. 19-25, 20th January 2016.