VERIFICATION OF DATA INTEGRITY USING PUBLIC AUDITABILITY AND DATA DYNAMICS FOR STORAGE SECURITY IN CLOUD COMPUTING

Mr. Ramratan Rathore [1], Dr. P.S. Chowdhary [2], Dr. Nidhi Tyagi [3]

[1] Asst. Prof. , IT Deptt. Dr. K. N. Modi Institute of Engineering & Technology, (India)
[2] Professor Physics Deptt, C.M.D. P.G. College, (India)
[3]Associate Professor CSE Deptt, Shobhit University, (India)

ABSTRACT

Cloud computing is the Shared pool of resources, which can be requested on-demand to enjoy the services and application. The data owner remotely store their data services and application which can be retrieved when required. Some of the major challenging issues of the cloud computing are availability of service, data lock in, Data confidentiality and Auditability, Data transfer Bottleneck, performance unpredictability, scalable strong, scaling quickly, Reputation Fate Sharing, Software Licensing. The focus of my work is on Data Auditability.

The users do not fell protected while their data is on the remote storage. Various models to overcome over this fear are designed but they increase the burden of the client and thus they are not efficient. One of the solutions to overcome this burden of client is that they can assign their work to any third party who will verify the data on behalf of the client; hence public auditability came into picture. The public auditability permits a trusted entity called Third Party Auditor that has special capabilities as compared to data owner.

The various model which came into picture are the Provable Data possession (PDP), Proof of Retriveability (POR), Compact proof of Retriveability and then the POR and the PDP Model are extended with the Merkle hash tree to go for the Dynamic operations like block modification, insertion and deletion as cloud storage is not only meant for data backup or warehouse. Various clients can aggregate their auditing request and then the TPA will send the query. The Third Party Auditor is now able to do the multiple auditing operations simultaneously. The idea of Batch auditing not only helps multiple clients to simultaneously verify their data but also the TPA's auditing time is expected to save.

1. INTRODUCTION

The cloud computing is the internet based computing where virtual shared servers provides software, infrastructure, platform , devices and other resources hosting to customers on a pay-as-you-use basis. Users can access these services available on the "internet cloud" without having any previous know how on managing the resources involved. Thus, user can concentrate more on their core business processes rather than spending time and gaining knowledge on the resources needed to manage their business processes.

Cloud computing users do not own the physical infrastructure; rather they rent the usage from the third-party provider. The cloud Computing provides a remote architecture where the users can remotly access services, applications and data storage . One has to only establish account with CSP (Cloud Service Providers) like Amazon Web Services (AWS), Elastic Compute Cloud (EC2), Simple storage Services (S3), Virtual Private Network (VPN), Microsoft Azure and Google Apps etc. Cloud Computing is the Shared pool of resources, which can be requested on-demand to enjoy the services and application. They consume resources as a service and pay only for the resources that they use. Most cloud computing infrastructures consist of services delivered through common centers and built on servers. Sharing resources amongst can improve, as servers are not unnecessarily left idle, which can reduce cost significantly while increasing the speed of application development.

The data owner, in cloud computing, remotely stores their data services and applications which can be retrieved when needed. The clients save their data on the storage services and gets free from the burden of maintaining the data. But some of the incidences in the history of cloud computing show that they are not reliable at times Ex: Gmail disaster, Amazon S3 downfall. The data outsourcing free the data owner from local storage and maintenance of data. It also eliminates security of data. Some of the major challenging issues of the cloud computing are availability of service, data lock in, Data confidentiality and Auditability, Data transfer Bottleneck, performance unpredictability, scalable strong, scaling quickly, Reputation Fate Sharing, Software Licensing. The focus of my work is on Data Auditability.

The users do not fell protected while their data is on the remote storage. To overcome this fear and to maintain the efficiency of data, efficient method have to be deign which can assure the verification of data. Various models to overcome over this fear are designed but they increase the burden of the client and thus they are not efficient. One of the solutions to overcome this burden of client is that they can assign their work to any third party who will verify the data on behalf of the clients; hence public auditability came into picture. The public auditability permits a trusted entity called Third Party Auditor that has special capabilities as compared to data owner. This entity not only helps the data owner to reduce its computation resources but also helps to verify the on-demand data.

The various model which came into picture are the Provable Data possession (PDP), Proof of Retriveability (POR), Compact proof of Retriveability and then the POR and the PDP Model are extended with the Merkle hash tree to go for the Dynamic operations like block modification, insertion and deletion as cloud storage is not only meant for data backup or warehouse. The Merkle Hash Tree (MHT) is used to provide block tag authentication in data dynamic operations. The extended model of the POR uses the BLS (Boneh-Lynn-Shacham) Signature for block tag authentication and proves to be better than the RSA. The implementations of the Merkle Hash Tree with the BLS Signature prove out to do the public auditability and also support Data Dynamic. But, after achieving this one of the great challenge in front is the aggregation of the multiple client request which will minimize the latency of the query. Thus my works focus on the technique of bilinear aggregate signature which is used to support multiple auditing operations. Various clients can aggregate their auditing request and then the TPA will send the query. The Third Party Auditor is now able to do the multiple auditing operations simultaneously. The technique of multiple clients auditing is also extended in the POR model.

II PROPOSED SYSTEMS

2.1 Merkle Hash Tree (MHT)

Merkle hash tree: Merkle hash tree are the one which provide authentication to the data and proves that the set of data is undamaged and secure. In cryptography this tree acts as a data structure that contains the summary information for a larger piece of data. The name of the tree is named upon a scientist Ralph Markle. The Markle hash tree can be constructed as a binary tree in which the leaf nodes are the hashes of the authenticated data value.

The leaves are read from left to right and the hashes are forms towards the root. Today Markle hash tree are used to verify the data in peer t munication. The tree is used to check whether the sender is sent the correct data or not.

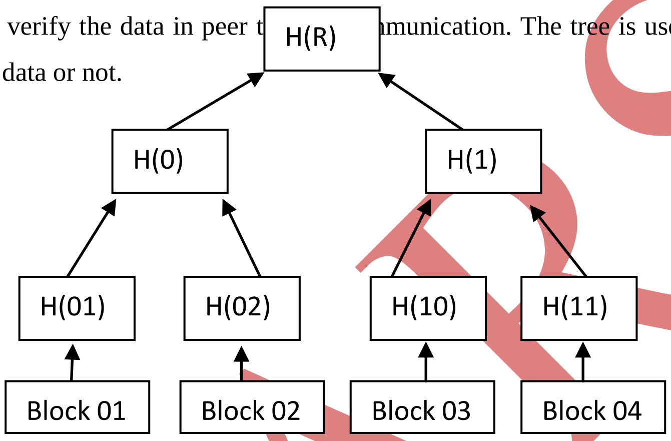

Fig1. Markle Hash Tree

A hash tree is a binary tree consists of the leaves which are hashes of data blocks, e.g. a file or set of files. Nodes further up in the tree are the hashes of their respective children. For example, in the diagram hash 0 is the result of concatenating and hashing hash 0-0 and then hash 0-1. That is, hash 0 = hash (hash 0-0 || hash 0-1) where || denotes concatenation. And finally the root of the tree consist the hash of the root.

2.2 Extending C-POR System with MHT

In compact proof of Retriveability, various security models proves to be efficient in achieving the integrity of the data stored in the cloud thus through we can easily verify the integrity of remotely stored data. But it fails when it comes to deal with the dynamic data verification. The Compact proof of Retriveability is a strong model to for verifying the integrity and extracting the file from the prover. It provides a strong model using BLS signature to verify the integrity of file publicly. But in this model the data dynamics is again a challenge to overcome this. The data dynamic operations in this models leads to security loopholes. The updating operation cannot be made in this model because any updating lead to change in the index name and thus the index of every other file has to be changed thus the signature have to recalculated again. Thus data dynamic is not possible. Hence one of the challenging issues in storage security is to verify the data under data dynamics.

2.3 Batching Of Multiclient Auditing Request

The implementations of the Merkle Hash Tree with the BLS Signature prove out to do the public auditability and also support Data Dynamic. But, after achieving this one of the great challenge in front is the aggregation of the multiple client request which will minimize the latency of the query. Thus my work focuses on the technique of bilinear aggregate signature which is used to support multiple auditing operations. Various clients can aggregate their auditing request and then the TPA will send the query.

The Cloud Computing always require strong handling of multi-client request as the services of it are simultaneously enjoyed by the multi users. The cloud services now need the services to be enhanced by serving the multi-auditing task. The key idea behind this bilinear aggregate signature which has the following property:

For any $u_{1:}$

$u_2 \epsilon$ G, $e(u_1u_2, v) = e(u_1, v) \cdot e(u_2, v)$ and

For any u:

$v \in$ G, $e(\psi(u), v) = e(\psi(v), u)$.

The BLS based construction allows the aggregate signature to create signatures on arbitrary distinct messages. It also enables the aggregation of multiple signature from different clients on distinct message to make a single short signature, thus greatly reduces the communication cost while providing efficient verification for the authenticity of all messages.

Considering that there are K clients in the system, and each client k has data files

$F_i = (m_{k,1}, . . . ,m_{k,n})$, where $k \epsilon \{1, . . . ,K\}$.

The procedure is as follows.

For a particular client k,

pick random $x_k \leftarrow Zp$, and compute $v_k = g^{xk}$.

(1) The client's public key is $v^k \epsilon$ G and private key is $x_k \epsilon$ Zp.

(2) In the SigGen phase, given the file F_k = (mk,1, . . . ,mk,n), client k chooses a random

element $u_k \leftarrow$ G and computes signature

$\sigma_{k,i} \leftarrow [H(m_{k,i}) \cdot u_m^{k,i}\ k\ ]^{xk} \epsilon$ G.

(3) Challenge phase, the verifier sends the query $Q = \{(i, \nu i)\}_{s1\leq i\leq sc}$ to the server to verify all K clients.

(4) GenProof phase, upon receiving the challenge, for each client k ($k \epsilon \{1, . . . ,K\}$), the server computes

$\mu_k = \sum_{\{(i,\nu i)\}s1\leq i\leq sc} \nu_i\ m_{k,i} \in Zp$ and

$\sigma = \prod_{k=1}^{k} (\prod_{\{(i,\nu i)\}s1\leq i\leq sc} \sigma_{k.i}{}^{\nu i})$

$= \prod_{k=1}^{k} e(\prod_{\{(i,\nu i)\}s1\leq i\leq sc} [H(m_{k,i}) \cdot u_k^{mk,i}\ ]^{xkvi}\ )$.

The prover then responses the verifier with $\{\sigma, \{\mu_k\}_{1\leq k\leq K,}\{k,i\}, \{H(m_{k,i})\}\}$. Similar to verify proof phase in single client case, the verifier first authenticates tags $H(m_{k,i})$ by verifying signatures on the roots (for each k client's file). If the authentication succeeds, then, with the help of bilinear map, the verifier can check if the following equation holds:

$$e(\sigma, g) = \prod^{K}_{k=1} e\left(\prod_{\{(i,vi)\}s1\leq i\leq sc} [H(m_{k,i})]^{vi} \cdot (u_k)^{\mu k} , v_k\right).$$

The equation is similar to the checking equation in the single-client case, and it holds because:

$$\begin{aligned} e(\sigma, g) &= e\left(\prod_{k=1}^{K} \left(\prod_{\{(i,vi)\}s1\leq i\leq sc} \sigma^{vi}{}_{k,i}\right), g\right) \\ &= e\left(\prod_{k=1}^{K} \left(\prod_{\{(i,vi)\}s1\leq i\leq sc} [H(m_{k,i}) \cdot u_k{}^{mk,i}]^{xkvi}\right), g\right) \\ &= \prod_{k=1}^{K} e\left(\left[\prod_{\{(i,vi)\}s1\leq i\leq sc} [H(m_{k,i})]^{vi} \cdot (u_k)^{\mu k} \right]^{xk} , g\right) \\ &= \prod_{k=1}^{K} e\left(\prod_{\{(i,vi)\}s1\leq i\leq sc} [H(m_{k,i})]^{vi} \cdot (uk)^{\mu k} , g^{xk}\right). \end{aligned}$$

Hence the multi-client request can be aggregated and efficiently executed this reduces the computing cost of the TPA.

III EXPECTED OUTCOME

Public Auditability allows the third party auditor to do integrity verification in place of owner. Thus, public auditability helps in reducing the computation of the data owner by assigning its computation to the TPA. Also there are challenge of dynamic data operation and designing the protocols that can accommodate the verification of dynamic data files. In cloud the client wants to simultaneously do data auditability and data dynamics. The designing of protocol to verify the integrity with the help of Merkle Hash Tree allows for the dynamic data operation. The use authenticating the Block tag instead of original data makes the verifier to authenticate only the block tag and hence makes the scheme a block less approach. Also in the setup phase the verifier stores the metadata R and hence makes the scheme stateless, but this scenario will make the verifier to cheat the client as the server may keep the old data and its corresponding signature. Since the data and signature are reliable thus the client may not be able to check for up to date data. Thus to overcome this difficulty the metadata R can be kept public. The scenario of multiple client data auditability is also considered and with the help of bilinear aggregate signature the TPA is able to batch the multi client auditing and also reduce its computing cost. The idea of batch auditing is also feasible with the PDP and POR model. Hence the capability of public verification of the two models can be increased and can be applied efficiently for multiple public verifications. The batch auditing in POR helps the TPA to compute multi client request in less time and reduce the computation overhead. Thus computation cost of TPA can be reduced with the help of batch auditing.

IV CONCLUSION

Public Auditability allows the third party auditor to do integrity verification in place of owner. Thus, public auditability helps in reducing the computation of the data owner by assigning its computation to the TPA. Also there are challenge of dynamic data operation and designing the protocols that can accommodate the verification of dynamic data files. In cloud, the clients want to simultaneously do data auditability and data dynamics. The designing of protocol to verify the integrity with the help of Merkle Hash Tree allows for the dynamic data operation. The use authenticating the Block tag instead of original data makes the verifier to authenticate only the block tag and hence makes the scheme a block less approach. Also in the setup phase the verifier stores the metadata R and hence makes the scheme stateless, but this scenario will make the verifier to cheat the client as the server may keep the old data and its corresponding signature. Since the data and signature are reliable thus the client may not be able to check for up to date data. Thus to overcome this difficulty the metadata can be kept public. The scenario of multiple client data auditability is also considered and with the help of bilinear aggregate signature the TPA is able to batch the multi-client auditing and also reduce its computing cost. The idea of batch auditing is also feasible with the PDP and POR model. Hence the capability of public verification of the two models can be increased and can be applied efficiently for multiple public verifications. The batch auditing in POR helps the TPA to compute multi-client request in less time and reduce the computation overhead. Thus computation cost of TPA can be reduced with the help of batch auditing.

REFRENCES

[1] P. Mell and T. Grance, 2009, "*Draft NIST Working Definition of Cloud Computing*", [online] Available at: http://csrc.nist.gov/groups/SNS/cloud-computing/index.html

[2] G.Boss, P. Malladi et al., October 2007; **"**Cloud Computing*", High Performance On Demand Solutions (HiPODS),* version 1.1.

[3] M. Armbrust et al., Feb. 2009 ; "*Above the Clouds: A Berkeley View of Cloud Computing,"*Univ. California, Berkeley, Tech. Rep. UCBEECS-2009-28.

[4] Janakiram, August 2010; "Demystifying the Cloud An introduction to Cloud Computing ",Version 1.1.

[5] "Top 30 Cloud Service Providers Gaining Mind Share in 3Q 2010", [online] Available at: http://cloudcomputing.sys-con.com/node/1513491

[6] Ragib HasanJohns,"Security and privacy in cloud computing", *Hopkins Universityen*.600.412 Spring 2010.

[7] Amazon.com, July 2008; "Amazon s3 Availability Event: July 20, 2008,"; [online] Available at : http://status.aws.amazon.com/s3-20080720.html

[8] M. Arrington, Dec. 2006;"Gmail Disaster: Reports of Mass Email Deletions", [online] Available at: http://www.techcrunch.com/2006/12/28/gmail-disaster-reports-of-massemail- deletions/

[9] M. Krigsman, July 2008 "Apple's MobileMe Experiences Post-Launch Pain," [online] Available at: http://blogs.zdnet.com/projectfailures/?p=908

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, 2007; "Provable data possession at untrusted stores," in Proc. of CCS'07. New York, NY, USA: ACM, pp. 598–609.

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

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

[13]H. Shacham and B. Waters, 2008 "Compact proofs of retrievability," in Proc. of ASI-ACRYPT'08. Springer-Verlag, , pp. 90–107.

[14] M. A. Shah et al., May 2007, "Auditing to keep Online Storage Services Honest," Proc. USENIX HotOS '07.

[15] G. Ateniese et al., Sept. 2008. ; "Scalable and Efficient Provable Data Possession," Proc. SecureComm '08.

[16] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, 2009; "Enabling public verifiability and data dynamics for storage security in cloud computing," Cryptology ePrint Archive, Report 2009/281.