IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN : 2278-0661, p-ISSN : 2278-8727
PP 72-78
www.iosrjournals.org

Auditing of Cloud Data with Privacy Preserving using TPA

Rushikesh P. Dhanokar[1], Prof. Gitanjali S. Mate[2]
[1]*(Computer Engineering, VRISHANK IT SOLUTION PVT.LTD, India)*
[2]*(Dept.of Information Technology, JSPM's RSCOE, Tathawade, Pune ,India)*

Abstract: *Cloud computing is emerging Technology, economical and sensible option for all home users, IT companies and other organizations for storing their large amount of data remotely on Cloud Server. With the rapid rate at which data is being generated as well as the high costs of data storage devices, it costly for enterprises or individual users to regularly update their hardware. Cloud-based outsourced storage space reduces the user load of local storage and burden of maintenance. User can upload their data on cloud and can access those data anytime anywhere by internet with comparably low-cost. Data is maintained by the cloud service provider and user is billed based on usage. Even though benefits are higher but while storing data in cloud, security and integrity of data are major concerns as there are many chances for CSP to behave unfaithfully towards users about the status of their outsourced data. The data loss and data change events may take place because of network, software bugs as well as inner and outer threats. In order to overcome the threat of integrity, an independent auditing service which audit the data integrity of cloud is required. In this paper we allowed a third- party auditor to check the integrity of outsourced data with privacy preserving. In our proposed scheme we are using RSA based homomorphic linear authentication (HLA) and SHA-2 to verify integrity of data and prohibit the frame and collude attack. TPA audit multiple data files by batch auditing and give proof of missing or corruption of data.*

Keywords: *Batch auditing, Cloud Computing, Data Integrity, Privacy preserving, Homomorphic linear authentication.*

I. Introduction

Cloud computing has future of next generation information technology (IT) architecture due to it's a variety of advantages. Cloud computing world such as Amazon, Yahoo, Google, Microsoft, windows azure and Mozy.com which allows clients to store their data on remote storage. The data is stored remotely on remote storage and it can be accessed through the internet connection between client's machine and remote machine on cloud. Cloud computing provides a shared pool of resources, including data storage space containing spreadsheets, presentations, audio, photos, word processing documents, videos, records etc., webs, infinite computer processing control, and expert corporate and user presentations. But for sensitive and confidential data there should be some security mechanism, so as to provide protection for private data. Cloud gives number of advantages like Flexibility, Better Reliability and Security user do not have to maintain the data as it is maintained by the cloud service provider, pay only that they used, Portability user can access his data from anywhere with the help of internet and they do not need to carry the physical data storage devices, enabling ubiquitous, convenient, on-demand network access. Though these benefits make cloud storage a very economical option for storing data it has some drawbacks like the data loss incidents may take place. There are lots of inner and outer threats, for the benefits of their ownership, CSP to behave unfaithfully like data loss incident may be kept secret from client to maintain reputation, and there may be viruses in the network path or in the software [1]. Security and privacy issues of cloud storage are authentication, Correctness of data, availability, data leakage, data loss. So, it requires an auditing service to check the integrity of outsourced data.

As clients have limited capacity and they are only able to upload and download data from cloud storage. User downloads all data in order to check integrity of stored data. It is very costly and tedious task, especially when the user is set with a low computation device (e.g. smart phone) or is not always connected to the Internet. Therefore, it is necessary to offer an efficient audit service to check the integrity and availability of the stored data. In the proposed system a Third Party Auditor (TPA) is introduced who will verify the data integrity of the client's data stored on cloud storage. TPA audit data when user needed. TPA has more potential than user and beneficial for cloud provider too because audit result from TPA gives more values for Cloud base service platform and also they fulfill the cloud computing concerns [2]. Nevertheless, without appropriate implementation, public verifiable auditing would impose users a false perception that their data were undamaged in the cloud storage. This audit service is significantly important for digital forensics and data assurance in clouds. For some economic profit, TPA may not always be trustworthy may collude with the cloud service provider (CSP) to give the verification for hiding some CSP's corrupted incident i.e. collude attack. Opposite to this TPA may frame CSP for their some profit clash by means of intentionally inaccurate auditing this resulting in frame attack.

So to remove above problem Third party auditor (TPA), meets 1) TPA audit cloud data storage without the original copy of data, and should not put any extra on-line burden to the cloud user. 2) The third party auditing process should preserve user data privacy. To handle this problem, in this paper, we use RSA based homomorphic linear authentication (HLA). By mixing HLA with random masking our protocol guarantees that third party auditor could not learn anything about data content stored in cloud server during auditing processes. 3) TPA prohibit collude and frame attack by storing the CSP computed result on its side and give validation to user. The user can authenticate whether any TPA has cheated the date owner or indeed executed the designated computational audit task.

II. Related Work

Ateniese et al. [3] has first considered Public auditability in their model for provable data possession [PDP] for ensuring the storage correctness of the data files on the servers. They allow a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. They had proposed the schemes is based on public audit ability. It generates proof for possession by randomly sampling the blocks of data files, but this way the linear combination of the blocks may disclose the data to the third party auditor. So their protocol was not fully privacy preserving. Juels et al. [4], the simplest Proof of retrivability (POR) scheme has been made using a keyed hash function hk(F). In this scheme, before archiving the data file F in the cloud storage, the verifier pre-computes the cryptographic hash of F using hk(F), and stores this hash as well as the secret key K. Verifier releases the secret key K to the cloud archive to check the integrity of the file F is lost or valid and asks it to compute and return the value of hk(F). The verifier by storing multiple hash values for different keys can check the integrity of the file F for multiple times, each one being an independent proof. POR model by possession and retrievability of remote data files on archive servers are ensured by using spot-checking and error-correcting codes. Their scheme does not support public auditability and the user can perform fixed number of audit challenges.

Shacham and Waters [6] system to improve a proof-of retrievability in compact proof of retrievability should be possible to get the client's data from any checker that passes a verification check. The proof-of-retrievability schemes with full proofs of security has been presented by Juels and Kaliski [4]. By using BLS signatures the client's query and server's response are both extremely short. This scheme allows public variability, not only just the owner but anyone can act as a verifier. PRFs is secure in the standard model, allows only private verification. Schemes based on homomorphic properties to aggregate a proof into one small authenticator value. Homomorphic linear authenticators use to build from secure BLS signatures which are publicly verifiable. Their approach is not privacy preserving due to the linear combination of the blocks and this equation can be solved by third party auditor. **Shah** *et al.* proposed a protocol to allow a third party auditor to from time to time verify the data stored by a service and support in returning the data together to the customer. This protocol use HMAC method to ensure the data integrity in remote servers, the owner pre-computes some MACs of the data with different secret keys and sends all the MACs and keys to the auditor. However, the number of times a particular data item can be verified is limited by the number of secret keys that fixed beforehand. Besides, the auditor needs to store several MACs for each file.

Online storage integrity Shah et al. [5] [7] introduces a third party auditor. They are encrypting the whole data and taking keys to hide it from auditor, moreover, both this symmetric keyed hashes and encrypted data are store on cloud and auditor. Auditor checked the integrity of data file. This scheme requires the auditor to maintain state of every key, works on encrypted files and when all the keyed hashes are used is affected by online burden on users.

Further Wang et al. [1] [8], additional feature of partial dynamic data storage and combine BLS-based HLA with MHT for supporting full data dynamics has been proposed. In this scheme symmetric key cryptography is used but with limitation on number of audits. The protocols discussed above are not privacy preserving as both require the linear combination of sampled blocks as input.

TABLE I. Literature Survey

Period	IEEE References	Issues
2006 to 2008	G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, “Provable Data Possession at Un- trusted Stores,” Proc. 14th ACM Conf. Computer and Comm. Security, pp. 598- 609, 2007 [3].	1. Not privacy-preserving. 2. Its communication and computation complexity are high.
2009 to 2011	K. D. Bowers, A. Juels, and A. Oprea, “Proofs of Retrievability: Theory and Implementation,” Proc. ACM Workshop Cloud Comput- ing Security (CCSW 009), pp. 43-54, 2009. [10]	1. Secret keys that is Permanent priori. Once all possible secret keys are exhausted, then it0s problematic
2012 to 2014	C. Wang, Q. Wang, K. Ren, and W. Lou, “Towards Secure and Dependable Storage Services in Cloud Computing,” IEEE Trans. Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012. [9]	1. User need for auditing. 2. TPA gain knowledge about user data.

III. Problem Statement

There are mainly three different models involved that are cloud user, cloud service provider or cloud storage and third party auditor. Cloud storage and cloud service provider are same in this system. Cloud users contain data that has to be stored in the cloud by registering to particular cloud storage. CSP take data from users and it is hub of storage space and computational resources. Third party auditor is examiner which on behalf of user will do the verification of the data integrity of the data stored on cloud storage. CSP are responsible for data stored and maintained in cloud but some cloud server provider may damage user’s data.

So to give a proof of integrity of storage data on cloud and minimize the overload of user we use a third party auditor. Third party auditability, privacy preserving with integrity check, proof of cloud data storage correctness, batch auditing, and lightweight processing for mobile user are the design goals to be completed.

IV. Challenges

Following security and performance challenges should be achieved:-

1) Third party auditability: To allow TPA to verify the correctness of the cloud data, flexible on demand without regaining a copy of the whole data or bring additional online burden to the cloud users.
2) Efficiency: Data uploading and auditing of data Communication and computation must be low.
3) Storage accuracy: To confirm that there is no cheating cloud server that can pass the TPA’s audit without really storing users’ data undamaged.
4) Privacy preserving: To ensure that the TPA cannot originate users’ data content from the information collected during the auditing process.
5) Prohibit Attacks: - To insure that the frame and collude attack should not take place.

V. Implementation Details

A. System Structure

Architecture of system shown in fig. 1.

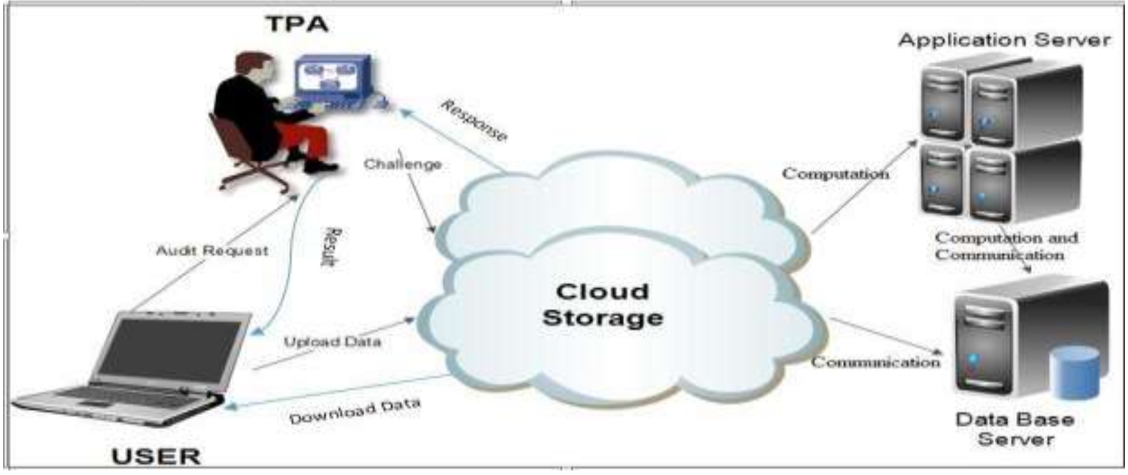

Fig. 1 System Architecture.

Client, TPA and cloud server task are:-
Client:

1) The Client will login to cloud system through the client login id and password.
2) The clients generate signature and metadata of uploading file.
3) Then the client will be allowed to upload the file and its metadata.
4) The client can download its data from cloud server at anytime from anywhere.

TPA:

1) The TPA will login to the system through the TPA login.
2) The TPA will then select the Client for whom the verification needs to be carried out.
3) Then the files will be selected for the verification purpose and regarding challenge send to cloud sever.
4) Taking challenge response from cloud server the TPA verify more than one file at same time.

Cloud Server:

1) Get data from registered users then store user file and its encrypted meta-data.
2) Give the verification proof of user storage data to TPA.
3) Update and maintain the user's data.

B. Algorithms

Input: User data (File).
Output: Data (File) Integrity Proof. (Lost or valid)

1. Generation of key is initiated by the user and compute Pk and Sk. // Public and secret key.
2. By using SHA-2 generate verification metadata (M) i.e. Digest by dividing file into blocks.
3. Verification metadata (M) encrypted by using Pk and file Store on cloud server. // (RSA)
4. On user request TPA Challenged to cloud server containing number of position of blocks.
5. Cloud compute storage correctness when it challenged.
6. Cloud server mask bit with PRF and aggregated authenticator send to TPA.
7. Third party auditor generate aggregated authenticator form response mask blocks.
8. TPA with help of verification of proof Verify CSP response and calculated aggregated authenticator.
9. TPA Store cloud response for further user verification.// limitation up to 10.

Proof of integrity setup into two steps
SETUP: In set up phase user sets public and secret parameters of the system by executing key generation algorithm and preprocess the data file F by using SHA-2 algorithm to generate the verification metadata. By using RSA metadata get encrypted. User upload the data file F and its encrypted metadata on cloud server and delete local copy.

AUDIT: In audit phase on request of user TPA send audit message or challenge to the cloud server to check the stored data integrity. Random masking Homomorphic linear authenticator technique is used by cloud sever to mask the blocks. Cloud server give the proof by retained the data file F as it is and TPA will then verify the proof. TPA cannot derive the user's data content due to lack of all the necessary information to build up a correct group of linear equations, so we preserve privacy. TPA also store cloud response up to 10 response for further user verification and prohibit collude and frame attack.

C. Privacy Preserving Public Auditing Scheme

RSA based Homomorphic linear authenticator with random masking technique is used to achieve privacy preserving with third party auditing. The linear combination of sampled blocks in the server's response is masked with randomness generated by the server using PRF. Even though many linear combinations of the same set of file blocks can be collected by using random masking, the third party auditor no longer has all the necessary information to build up a correct group of linear equations and therefore cannot derive the user's data content. The design makes use of public key based homomorphic linear authenticator to equip auditing protocol with public auditability.

D. Batch Auditing

Third party auditor can concurrently handle multiple auditing from different users delegation and thus supports batch auditing. If TPA audit the tasks individually it will be tedious and inefficient communication and computation time get increase. It is always advantageous for the TPA to batch multiple batches together and audit at one time. By aggregating K verification equations into single one, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

E. Mathematical Modelling

Let the system S is represented as: S = {U, K, P}.

A. Setup Phase let U contains user0s data, which users want to verify. U = {l1, l2, l3, ... ,ln} Where, l1, l1, ... are the different files of data.

B. Audit Phase Let K be the set of request for auditing from different users. K = {t1, t2, t3, ... ,tn} Where, t1, t2, t3, ... ,tn are batch TPA send to CSP.

C. Auditing Phase Let P is a Proof of integrity. P = {p1, p2, p3, ... ,pn} Where, p1, p2, p3, ... Aggregate authenticator to the one received from CSP to TPA. I.e. Proof of integrity (lost or valid).

States of accountability mechanism in cloud are:-

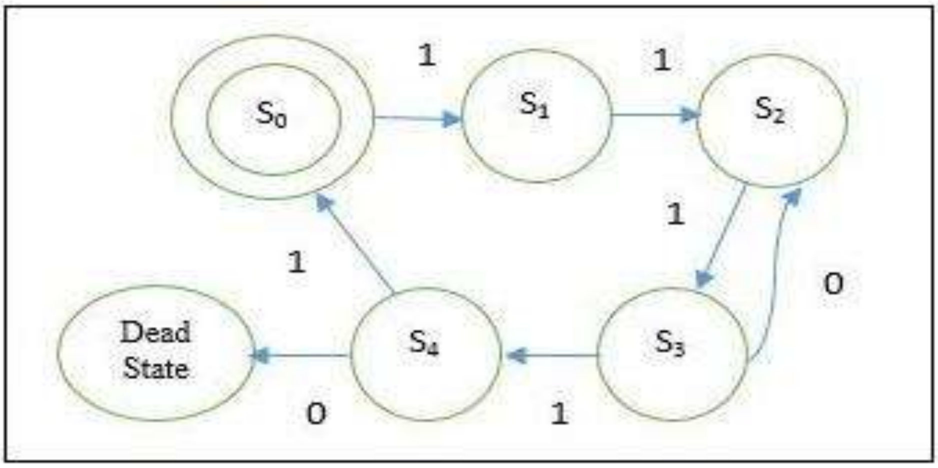

Fig. 2 State Transition Diagram

Where,
1: Successful
0: Unsuccessful

Transition are:
S_0: User will send File to cloud storage.
S_1: Get metadata from File and encrypt it.
S_2: Stored File on cloud and cloud challenged by TPA.
S_3: TPA get response from cloud server.
S_4: Verify the integrity and send to user also store the cloud response.

Input: = {0, 1}

Representation of
A = ({S_0, S_1, S_2, S_3, S_4} {0, 1}, δ, S_0, S_4)

Input given:- 1110101
Expected output
$\delta(S_0,1) = S_1$
$\delta(S_1,1) = S_2$
$\delta(S_2,1) = S_3$
$\delta(S_4,1) = S_4$
$\delta(S_4,1) = S_0$

F. Properties of System

Public auditability is achieved in the protocol and it does not pose any potential online burden on users. It supports privacy of user data by employing a random masking and linear combination of data blocks. Given the huge volume of data outsourced in the cloud, checking a portion of the data file is more affordable and practical for both the TPA and the cloud server than checking all the data, as long as the sampling strategies provides high-probability assurance. Data storage validation, Auditing Support, Efficient, Preserve Privacy property, prohibit attacks, Batch audit: TPA has capacity to verify multiple files at a time.

VI. Results

A. Data Set

The User side process is implemented on workstation with Intel core i5 processer running at 2.6 GHz, 4 MB RAM and 500 GB HDD. Cloud server side process is implemented on Windows Azure compute and storage instance Standard_A2 (2 cores processer, 3.5 GB RAM) and 489 GB HDD. The evaluation includes security analysis and performance analysis. The efficiency analysis on the batch auditing, is done by considering only the total number of operations. In performance analysis the bandwidth of user uploading data is low.

B. Results

To get a complete view of batching efficiency, we conduct a timed batch auditing test, where the number of auditing tasks is increased from 1 to approximately 200 with intervals of 10. The performance of the corresponding non batched (individual) auditing is provided as a baseline for the measurement. It can be shown that compared to individual auditing, batch auditing indeed helps reducing the TPA's computation cost, as more than 16 percent of per task auditing time is saved. Cost for data storage on server is reduced.

Server setup phase time, server computing time to compute aggregated authenticator and TPA computing time for verification of file with block size (300, 500, 1000) are shown in table 2.

TABLE 2: Computation on Different block

Block Size	300	500	1000
Server Setup Time (ms)	290.91	491.43	788.16
Server Comp. Time(ms)	225.33	346.56	534.23
TPA Comp. Time (ms)	359.80	537.34	710.76

Figure 2 shows comparisons of computation time required for block size 300, 500 and 1000. X-axis contain Time in millisecond and Y axis contain Block size.

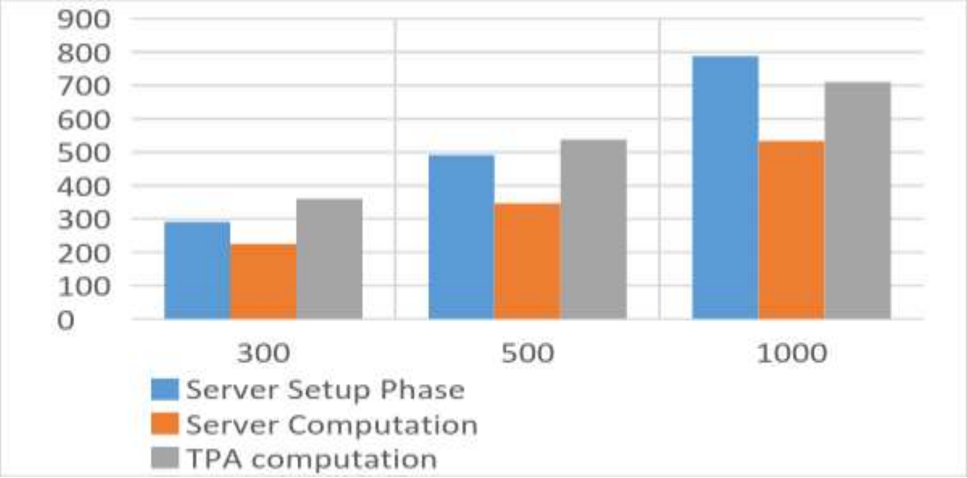

Fig. 3 Block size Vs. Computation time

The following figure shows that Windows Azure overview where computation and storage take place.

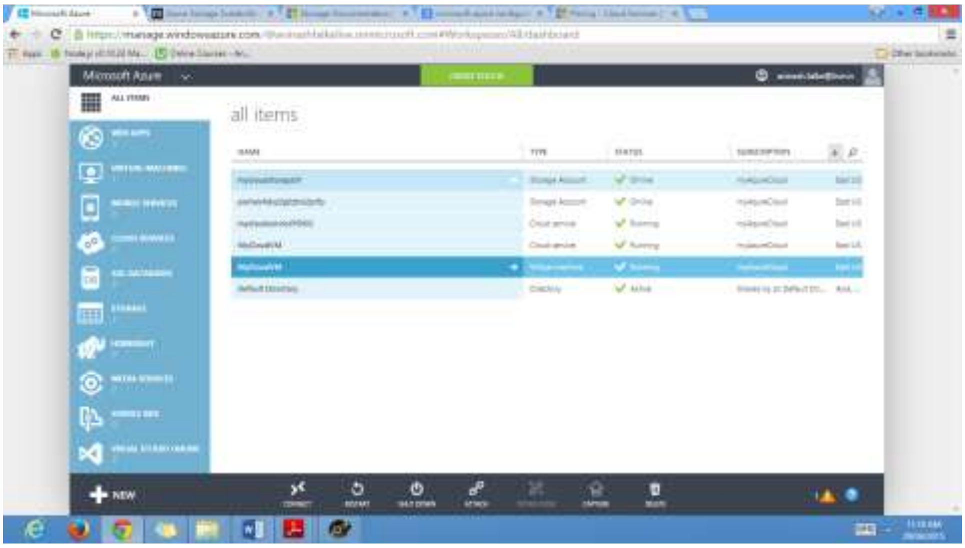

Fig. 4 Windows Azure Cloud

VII. Conclusion

In this paper, we propose a privacy-preserving TPA auditing system for data storage integrity in cloud computing. We use the RSA based homomorphic linear authenticator and random masking to assure that the TPA would not learn any knowledge about the data content stored on the cloud server during the efficient auditing process, which not only eliminates the burden of cloud user from the tedious and possibly expensive auditing task, but also reduce the users fear of their outsourced data leakage. SHA-2 produces Meta-data that is only encrypted so store cost is reduced. Security of the scheme is done by privacy preserving and storage correctness property. In addition TPA may simultaneously handle several audit sessions from different users for their outsourced data files, where the TPA can perform multiple auditing tasks in a batch manner for better performance efficiency. The system is totally secure and highly efficient. The RSA algorithm is partially homomorphic encryption so using fully homomorphic encryption can be a future enhancement.

Acknowledgement

I would like to thank my guide Prof. G. S. Mate for her help and guidance throughout this project, without them this would not have been possible.

References

[1] Cong Wang, Member, IEEE, Sherman S.M. Chow, Qian Wang, Member, IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE, ”Privacy- Preserving Public Auditing for Secure Cloud Storage” IEEE Transactions on Computers, vol. 62, no. 2, February 2013.R. Caves, Multinational Enterprise and Economic Analysis, Cambridge University Press, Cambridge, 1982.

[2] CloudSecurityAlliance,“TopThreatstoCloudComputing,”http://www.cloudsecurityalliance.org, 2010

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, ”Provable Data Possession at Untrusted Stores,” Proc.14th ACM Conf. Computer and Comm. Security (CCS ’07), pp. 598-609, 2007.

[4] A. Juels and J. Burton, S. Kaliski, ”PORs: Proofs of Retrievability for Large Files,” Proc. ACM Conf. Computer and Comm. Security (CCS’07), pp. 584-597, Oct. 2007.008.

[5] M.A. Shah, R. Swaminathan, and M. Baker, ”Privacy- Preserving Audit and Extraction of Digital Contents,” Cryptology ePrint Archive, Report 2008.

[6] H. Shacham and B. Waters, ”Compact Proofs of Retrievability,” Proc. Int’l Conf. Theory and Application of Cryptology and Information Security: Advances), in Cryptology (Asiacrypt vol. 5350, pp. 90-107, Dec. 2008.

[7] M.A. Shah, M. Baker, J.C. Mogul, and R. Swaminathan, ”Auditing to Keep Online Storage Services Honest,” Proc. 11th USENIXWorkshop Hot Topics in Operating Systems (HotOS ’07), pp. 1-6, 2007.

[8] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, ”Enabling Public Auditability and Data Dynamics for Storage Security in Cloud Computing,” IEEE Trans. Parallel and Distributed Systems, vol. 22,no. 5, pp. 847-859, May 2011.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou, “Towards Secure and Dependable Storage Services in Cloud Computing,” IEEE Trans. Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012.

[10] K.D. Bowers, A. Juels, and A. Oprea, “Proofs of Retrievability: Theory and Implementation,” Proc. ACM Workshop Cloud Comput- ing Security (CCSW 009), pp. 43-54, 2009.

[11] 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 and Data Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.