International Journal for Modern Trends in Science and Technology
Volume: 03, Special Issue No: 01, February 2017
ISSN: 2455-3778
http://www.ijmtst.com

Concealment Freeze Third Party Auditing for Secure Cloud Storage

E.Akhil[1] | G.K.Kishore Babu[2] | Ch.Jyosthna Devi[3]

[1]Assistant Professor, Department of CSE, Rise Krishna Sai Praksam Group of Colleges, Ongole, AP, India.
[2,3]Assistant Professor, Department of CSE, Chalapathi Institute of Engineering and Technology, Guntur, AP, India.

To Cite this Article
E.Akhil, G.K.Kishore Babu and Ch.Jyosthna Devi, "Concealment Freeze Third Party Auditing for Secure Cloud Storage", *International Journal for Modern Trends in Science and Technology*, Vol. 03, Special Issue 01, 2017, pp. 59-66.

ABSTRACT

Using cloud storage, users can remotely store their data and enjoy the on-demand high-quality applications and services from a shared pool of configurable computing resources, without the burden of local data storage and maintenance. However, the fact that users no longer have physical possession of the outsourced data makes the data integrity protection in cloud computing aformidable task, especially for users with constrained computing resources. Moreover, users should be able to just use the cloud storage as if it is local, without worrying about the need to verify its integrity. Thus, enabling public auditability for cloud storage is ofcritical importance so that users can resort to a third-party auditor (TPA) to check the integrity of outsourced data and be worry free. Tosecurely introduce an effective TPA, the auditing process should bring in no new vulnerabilities toward user data isolation, and introduceno additional online burden to user. In this paper, we propose a secure cloud storage system supporting isolation-preserving public auditing. We further extend our result to enable the TPA to perform audits for multiple users simultaneously and efficiently. Extensive security and performance analysis show the proposed schemes are provably secure and highly efficient. Our preliminary experiment conducted on Amazon EC2 instance further demonstrates the fast performance of the design.

KEYWORDS: *Data storage, isolation preserving, public auditability, cloud computing, delegation, batch verification, zero knowledge.*

I. INTRODUCTION

Cloud computing has been envisioned as the nextgenerationinformation technology architecture forenterprises, due to its long list of unprecedented advantagesin the IT history: on-demand self-service, ubiquitousnetwork access, location independent resource pooling,rapid resource elasticity, usage-based pricing and transferenceof risk as a disruptive technology with profoundimplications, cloud computing is transforming the verynature of how businesses use information technology. Onefundamental aspect of this paradigm shifting is that data arebeing centralized or outsourced to the cloud. From users'perspective, including both individuals and IT enterprises,storing data remotely to the cloud in a flexible on-demandmanner brings appealing benefits: relief of the burden forstorage management, universal data access with locationindependence, and avoidance of capital expenditure onhardware, software, and personnel maintenances, etc., While cloud computing makes these advantages moreappealing than ever, it also brings new and challengingsecurity threats toward users' outsourced data. Since cloudservice providers (CSP) are separate administrative

entities,data outsourcing is actually relinquishing user's ultimatecontrol over the fate of their data. As a result, thecorrectness of the data in the cloud is being put at riskdue to the following reasons. First of all, although theinfrastructures under the cloud are much more powerfuland reliable than personal computing devices, they are stillfacing the broad range of both internal and external threatsfor data integrity Examples of outages and securitybreaches of noteworthy cloud services appear from time totime. Second, there do exist various motivationsfor CSP to behave unfaithfully toward the cloud usersregarding their outsourced data status. For examples,SPmight reclaim storage for monetary reasons by discardingdata that have not been or are rarely accessed, or even hidedata loss incidents to maintain a reputation. Inshort, although outsourcing data to the cloud is economicallyattractive for long-term large-scale storage, it does notimmediately offer any guarantee on data integrity andavailability. This problem, if not properly addressed, mayimpede the success of cloud architecture.As users no longer physically possess the storage of theirdata, traditional cryptographic primitives for the purposeof data security protection cannot be directly adopted.In particular, simply downloading all the data for itsintegrity verification is not a practical solution due to theexpensiveness in I/O and transmission cost across thenetwork. Besides, it is often insufficient to detect the datacorruption only when accessing the data, as it does not giveusers correctness assurance for those un-accessed data andmight be too late to recover the data loss or damage.Considering the large size of the outsourced data and theuser's constrained resource capability, the tasks of auditingthe data correctness in a cloud environment can beformidable and expensive for the cloud users.Moreover, the overhead of using cloud storage should beminimized as much as possible, such that a user does notneed to perform too many operations to use the data. In particular, users maynot want to go through the complexity in verifying the dataintegrity. Besides, there may be more than one user accessesthe same cloud storage, say in an enterprise setting. Foreasier management, it is desirable that cloud only entertainsverification request from a single designated party.To fully ensure the data integrity and save the cloud users'computation resources as well as online burden, it is ofcritical importance to enable public auditing service for clouddata storage, so that users may resort to an independentthird-party auditor (TPA) to audit the outsourced data whenneeded. The TPA, who has expertise and capabilities thatusers do not, can periodically check the integrity of all thedata stored in the cloud on behalf of the users, whichprovides a much more easier and affordable way for the usersto ensure their storage correctness in the cloud. Moreover, inaddition to help users to evaluate the risk of their subscribedcloud data services, the audit result from TPA would also bebeneficial for the cloud service providers to improve theircloud-based service platform, and even serve for independentarbitration purposes. In a word, enabling publicauditing services will play an important role for this nascentcloud economy to become fully established, where users willneed ways to assess risk and gain trust in the cloud.Recently, the notion of public auditability has beenproposed in the context of ensuring remotely stored dataintegrity under different system and security models. Public auditability allows an external party, inaddition to the user himself, to verify the correctness ofremotely stored data. However, most of these schemes do not consider the isolation protection of users'data against external auditors. Indeed, they may potentiallyreveal user's data to auditors, as will be discussed. This severe drawback greatly affects thesecurity of these protocols in cloud computing. From theperspective of protecting data isolation, the users, who ownthe data and rely on TPA just for the storage security oftheir data, do not want this auditing process introducingnew vulnerabilities of unauthorized information leakagetoward their data security. Moreover, there arelegal regulations, such as the US Health Insurance Portabilityand Accountability Act (HIPAA), furtherdemanding the outsourced data not to be leaked to externalparties. Simply exploiting data encryption beforeoutsourcing could be one way to mitigate thisisolation concern of data auditing, but it could also be anoverkill when employed in the case of unencrypted/publiccloud data, due to the unnecessary processing burden for cloudusers. Besides, encryption does not completely solve theproblem of protecting data isolation against third-partyauditing but just reduces it to the complex key managementdomain. Unauthorized data leakage still remains possibledue to the potential exposure of decryption keys.Therefore, how to enable aisolation-preserving thirdpartyauditing protocol, independent to data encryption, isthe problem we are going to tackle in this paper. Our workis among the first few ones to

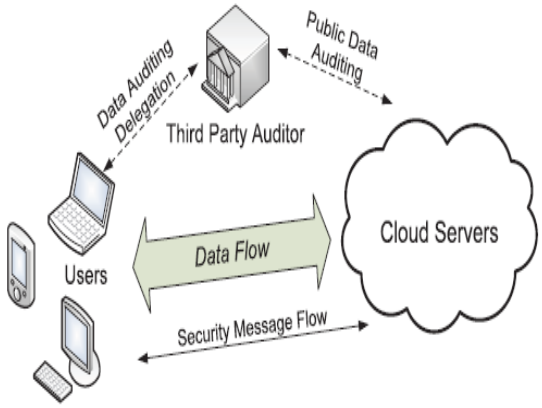
support isolation-preservingpublic auditing in cloud computing, with a focus on datastorage. Besides, with the prevalence of cloud computing, aforeseeable increase of auditing tasks from different usersmay be delegated to TPA. As the individual auditing ofthese growing tasks can be tedious and cumbersome, anatural demand is then how to enable the TPA to efficientlyperform multiple auditing tasks in a batch manner, i.e.,simultaneously.To address these problems, our work utilizes thetechnique of public key-based homomorphic linear authenticator, which enables TPA toperform the auditing without demanding the local copy ofdata and thus drastically reduces the communication andcomputation overhead as compared to the straightforwarddata auditing approaches. By integrating the HLA withrandom masking, our protocol guarantees that the TPAcould not learn any knowledge about the data contentstored in the cloud server (CS) during the efficient auditingprocess. The aggregation and algebraic properties of theauthenticator further benefit our design for the batchauditing. Specifically, our contribution can be summarizedas the following three aspects:

1. We motivate the public auditing system of datastorage security in cloud computing and provide aisolation-preserving auditing protocol. Our schemeenables an external auditor to audit user's cloud datawithout learning the data content.
2. To the best of our knowledge, our scheme is the firstto support scalable and efficient isolation-preservingpublic storage auditing in cloud. Specifically, ourscheme achieves batch auditing where multipledelegated auditing tasks from different users canbe performed simultaneously by the TPA in aisolation-preserving manner.

II. Problem Statement

2.1 The System and Threat Model

We consider a cloud data storage service involving threedifferent entities, as illustrated in Fig. 1: the cloud user, whohas large amount of data files to be stored in the cloud; thecloud server, which is managed by the cloud service provider toprovide data storage service and has significant storagespace and computation resources the third-party auditor, who hasexpertise and capabilities that cloud users do not have and
is trusted to assess the cloud storage service reliability onbehalf of the user upon request. Users rely on the CS forcloud data storage and maintenance. They may alsodynamically interact with the CS to access and update theirstored data for various application purposes. As users nolonger possess their data locally, it is of critical importancefor users to ensure that their data are being correctly storedand maintained. To save the computation resource as wellas the online burden potentially brought by the periodicstorage correctness verification, cloud users may resort toTPA for ensuring the storage integrity of their outsourceddata, while hoping to keep their data private from TPA.We assume the data integrity threats toward users' datacan come from both internal and external attacks at CS.These may include: software bugs, hardware failures, bugsin the network path, economically motivated hackers,malicious or accidental management errors, etc. Besides,CS can be self-interested. For their own benefits, such as tomaintain reputation, CS might even decide to hide thesedata corruption incidents to users. Using third-partyauditing service provides a cost-effective method for usersto gain trust in cloud. We assume the TPA, who is in thebusiness of auditing, is reliable and independent. However,it may harm the user if the TPA could learn the outsourceddata after the audit.Note that in our model, beyond users' reluctance to leakdata to TPA, we also assume that cloudservers have noincentives to reveal their hosted data to external parties. Onthe one hand, there are regulations, e.g., HIPAA,requesting CS to maintain users' data isolation. On the otherhand, as users' data belong to their business asset, therealso exist financial incentives for CS to protect it from anyexternal parties. Therefore, we assume that neither CS norTPA has motivations to collude with each other during theauditing process. In other words, neither entity will deviatefrom the prescribed protocol execution in the followingpresentation.To authorize the CS to respond to the audit delegated toTPA's, the user can issue a certificate on TPA's public key,and all audits from the TPA are authenticated against sucha certificate. These authentication handshakes are omitted
in the following presentation.



2.2 Design Goals

To enable isolation-preserving public auditing for cloud datastorage under the aforementioned model, our protocoldesign should achieve the following security and performanceguarantees:

1. Public auditability: to allow TPA to verify thecorrectness of the cloud data on demand withoutretrieving a copy of the whole data or introducingadditional online burden to the cloud users.
2. Storage correctness: to ensure that there exist nocheating cloud servers that can pass the TPA's auditwithout indeed storing users' data intact.
3. Isolation preserving: to ensure that the TPA cannotderive users' data content from the informationcollected during the executing KeyGeneration, and preprocesses the data file F by using Signature Generation to generate the verification metadata. The user then stores the data file F and the verification metadata at the cloud server, and delete its local copy. As part of preprocessing, the user may alter the auditing process.
4. Batch auditing: to enable TPA with secure andefficient auditing capability to cope with multipleauditing delegations from possibly large number ofdifferent users simultaneously.
5. Lightweight: to allow TPA to perform auditingwith minimum communication and computationoverhead.

III. THE PROPOSED SCHEMES

This section presents our public auditing scheme whichprovides a complete outsourcing solution of data—not onlythe data itself, but also its integrity checking. Afterintroducing notations and brief preliminaries, we start froman overview of our public auditing system and discuss twostraightforward schemes and their demerits. Then, wepresent our main scheme and show how to extent ourmain scheme to support batch auditing for the TPA upondelegations from multiple users. Finally, we discuss how togeneralize our isolation-preserving public auditing schemeand its support of data dynamics.

3.2 Definitions and Framework

We follow a similar definition of previously proposedschemes in the context of remote data integrity checkingand adapt the framework for our isolationpreservingpublic auditing system.A public auditing scheme consists of four algorithmsKey-Gen isa key generation algorithm that is run by the user to setupthe scheme. Sig-Gen is used by the user to generateverification metadata, which may consist of digital signatures.Gen-Proof is run by the cloud server to generate aproof of data storage correctness, while VerifyProof isrun by the TPA to audit the proof.Running a public auditing system consists of two phases,Setup and Audit:

- **Setup:** The user initializes the public and secret parameters of the system by data file by expanding it or including additional metadata to be stored at server.
- **Audit:** The TPA issues an audit message or challenge to the cloud server to make sure that the cloud server has retained the data file F properly at the time of the audit. The cloud server will derive a response message by executing General Proof using file and its verification metadata as inputs. The TPAthen verifies the response via VerifyProof.

3.3 The Basic Schemes

Before giving our main result, we study two classes ofschemes as a warmup. The first technique is a MAC-based solutionwhich suffers from undesirable systematic demerits bounded usage and statefull verification, which may poseadditional online burden to users, in a public auditingsetting. This also shows that the auditing problem is still noteasy to solve even if we have introduced a TPA. The secondone is a system based on homomorphic linear authenticators,which covers much recent proof of storage systems. Wewill pinpoint the reason why all existing HLA-basedsystems are not isolation preserving. The analysis of thesebasic schemes leads to our main result, which overcomes allthese drawbacks. Our main scheme to be presented is basedon a specific HLA scheme.

MAC based solution.

There are two possible ways to make use of MAC to authenticate the data. A trivial way is just uploading the data blocks with their MACs to the server, and sends thecorresponding secret key to

the TPA. Later, the TPA can randomly retrieve blocks with theirMACs and check the correctness via sk. Apart from thehigh communication andcomputation complexities, the TPA requires the knowledgeof the data blocks for verification.

HLA-based solution.

To effectively support publicauditability without having to retrieve the data blocksthemselves, the HLA techniquecan be used.HLAs, like MACs, are also some unforgettable verificationmetadata that authenticate the integrity of a data block. Thedifference is that HLAs can be aggregated. It is possible tocompute an aggregated HLA which authenticates a linearcombination of the individual data blocks.At a high level, an HLA-based proof of storage systemworks as follow. The user still authenticates each element ofF ={m_i} by a set of HLAs ϕ. The TPA verifies the cloudstorage by sending a random set of challenge {v_i}. Thecloud server then returns and its aggregatedauthenticator _ computed from ϕ. Though allowing efficient data auditing and consumingonly constant bandwidth, the direct adoption of these HLAbasedtechniques is still not suitable for our purposes. Thisis because the linear combination of blocks,may potentially reveal user data information to TPA, andviolates the isolation-preserving guarantee. Specifically, bychallenging the same set of c block m1;m2; m3 ;mc usingc different sets of random coefficients TPA canaccumulate c different linear combinations TPA can derive the user's data m1;m2; . . .;mcby simply solving a system of linear equations.

3.5 Support for Batch Auditing

With the establishment of isolation-preserving publicauditing, the TPA may concurrently handle multipleauditing upon different users' delegation. The individualauditing of these tasks for the TPA can be tedious and veryinefficient. Given K auditing delegations on K distinctdata files from K different users, it is more advantageousfor the TPA to batch these multiple tasks together andaudit at one time. Keeping this natural demand in mind,we slightly modify the protocol in a single user case, andachieve the aggregation of K verification equations into a single one, as shown. As aresult, a secure batch auditing protocol for simultaneousauditing of multiple tasks is obtained.

IV. EVALUATION

4.1 Security Analysis

We evaluate the security of the proposed scheme byanalyzing its fulfillment of the security guarantee described in namely, the storage correctness and isolationpreservingproperty. We start from the single user case,where our main result is originated. Then, we show thesecurity guarantee of batch auditing for the TPA inmultiuser setting.

4.2 Performance Analysis

We now report some performance results of our experiments.We consider our auditing mechanism happensbetween a dedicated TPA and some cloud storage node,where user's data are outsourced to. In our experiment, theTPA/user side process is implemented on a workstationwith an Intel Core 2 processor running at 1.86 GHz,2,048 MB of RAM, and a 7,200 RPM Western Digital250 GB Serial ATA drive. The cloud server side process isimplemented on Amazon Elastic Computing Cloud (EC2)with a large instance type [27], which has 4 EC2 ComputeUnits, 7.5 GB memory, and 850 GB instance storage. Therandomly generated test data is of 1 GB size. All algorithmsare implemented using C language. Our code uses thePairing-Based Cryptography (PBC) library version 0.4.21.The elliptic curve utilized in the experiment is an MNTcurve, with base field size of 159 bits and the embeddingdegree 6. The security level is chosen to be 80 bit, whichmeans $|v_i|$ =80 and |P|=160. All experimental resultsrepresent the mean of 20 trials.Because the cloud is a pay-per-use model, users have topay both the storage cost and the bandwidth cost when using the cloud storage auditing. Thus,when implementing our mechanism, we have to take intoconsideration both factors. In particular, we conduct theexperiment with two different sets of storage/communicationtradeoff parameters.When the mechanism incurs extra storage cost as large asthe data itself, but only takes very small auditingbandwidth cost. Such a mechanism can be adopted whenthe auditing has to happen very frequently because theresulting data transfer charge could be dominant in the use-model. On the other hand, we also choose aproperly larger s=10, which reduces the extra storage costto only 10 percent of the original data but increases theauditing bandwidth cost roughly 10 times larger than thechoice of s ¼ 1. Such a case is relatively more desirable if theauditing does not need to happen frequently. In short, userscan flexibly choose the storage/communication tradeoffparameter s for their different system application scenarios.On our not-so-powerful workstation, the measurementshows that the user setup phase achieves a throughput of around 9.0 KB/s and 17.2 KB/s. when s ¼ 1 and s ¼ 10,

respectively. These results are notvery fast due to the expensive modular exponentiationoperations for each 20 byte block sector in the authenticatorcomputation.Note that for each data file to be outsourced, such setup phase happens once only. Further, since the authenticator generationon each block is independent, these one-time operationscan be easily parallelized by using multithreading techniqueon the modern multicore systems. Therefore, variousoptimization techniques can be applied to speedup the userside setup phase. As our paper focuses on isolation-preservingstorage auditing performance, in the following, we willprimarily assess the performance of the proposed auditingschemes on both TPA side and cloud server side, and showthey are indeed lightweight. We will focus on the cost of theisolation-preserving protocol and our proposed batch auditingtechnique.

4.2.1 Cost of Isolation-Preserving Protocol

We begin by estimating the cost in terms of basiccryptographic operations. Suppose there are c random blocks specified in thechallenge message ramp during the Audit phase. Underthis setting, we quantify the cost introduced by the isolationpreservingauditing in terms of server computation, auditorcomputation as well as communication overhead. Since thedifference for choices on s has been discussed previously, inthe following isolation-preserving cost analysis we only givethe atomic operation analysis for the case s = {1} forsimplicity. The analysis for the case of s = {10} followssimilarly and is thus omitted.

s = 1	Our Scheme		[13]	
Sampled blocks *c*	460	300	460	300
Sever comp. time (ms)	335.17	219.27	333.23	217.33
TPA comp. time (ms)	530.60	357.53	526.77	353.70
Comm. cost (Byte)	160	160	40	40
s = 10	Our Scheme		[13]	
Sampled blocks *c*	460	300	460	300
Sever comp. time (ms)	361.56	242.29	342.91	223.64
TPA comp. time (ms)	547.39	374.32	543.35	370.29
Comm. cost (Byte)	1420	1420	220	220

Fig-1:Performance under Different Number of Sampled Blocks for High Assurance (≥95%) Auditing

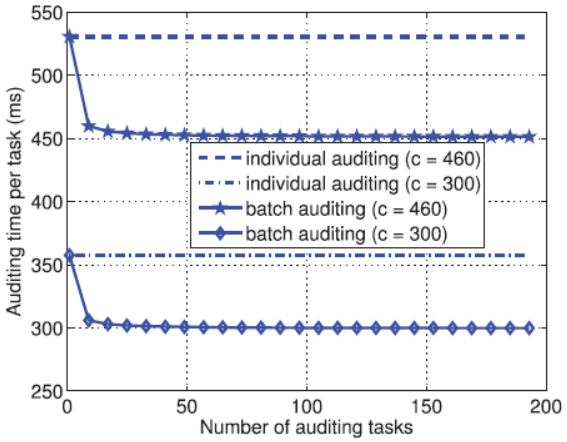


Fig-2: Comparison on auditing time between batch and individual auditing: per task auditing time denotes the total auditing time divided by the number of tasks.

4.2.2 Batch Auditing Efficiency

Discussion in Section 3.5 gives an asymptotic efficiencyanalysis on the batch auditing, by considering only the totalnumber of pairing operations. However, on the practicalside, there are additional less expensive operations requiredfor batching, such as modular exponentiations and multiplications.Thus, whether the benefits of removing pairingssignificantly outweighs these additional operations remainsto be verified. To get a complete view of batching efficiency,we conduct a timed batch auditing test, where the numberof auditing tasks is increased from 1 to approximately 200with intervals of 8. Note that we only focus on the choice of s = {1} here, from which similar performance results can bedirectly obtained for the choice of s = {10}. The performance of the corresponding individualauditing isprovided as a baseline for the measurement. Following thesame settings c = {300} and c = {460}, the average per taskauditing time, which is computed by dividing total auditingtime by the number of tasks, is given in Fig. 2 for both batchand individual auditing. It can be shown that compared toindividual auditing, batch auditing indeed helps reducingthe TPA's computation cost, as more than 15 percent of per-taskauditing time is saved.

V. CONCLUSION

In this paper, we propose aisolation-preserving publicauditing system for data storage security in cloud computing.We utilize the hemimorphic linear authenticator andrandom masking to guarantee that the TPA would not learnany knowledge about the data content stored on the cloudserver during the efficient auditing process, which not

onlyeliminates the burden of cloud user from the tedious andpossibly expensive auditing task, but also alleviates theusers' fear of their outsourced data leakage. ConsideringTPA may concurrently handle multiple audit sessions fromdifferent users for their outsourced data files, we furtherextend our isolation-preserving public auditing protocol intoa multiuser setting, where the TPA can perform multipleauditing tasks in a batch manner for better efficiency.Extensive analysis shows that our schemes are provablysecure and highly efficient. Our preliminary experimentconducted on Amazon EC2 instance further demonstratesthe fast performance of our design on both the cloud and theauditor side. Weleave the full-fledged implementation of themechanism on commercial public cloud as an importantfuture extension, which is expected to robustly cope withvery large scale data and thus encourage users to adopt cloudstorage services more confidently.

REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Isolation-PreservingPublic Auditing for Storage Security in Cloud Computing," Proc.IEEE INFOCOM '10, Mar. 2010.

[2] P. Mell and T. Grance,"Draft NIST Working Definition ofCloud Computing," http://csrc.nist.gov/groups/SNS/cloudcomputing/ index.html, June 2009.

[3] 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.

[4] Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing,"Technical Report UCB-EECS-2009-28, Univ. of California,Berkeley, Feb. 2009.WANG ET AL.: ISOLATION-PRESERVING PUBLIC AUDITING FOR SECURE CLOUD STORAGE 373

[5] Cloud Security Alliance, "Top Threats to Cloud Computing,"http://www.cloudsecurityalliance.org, 2010.

[6] M. Arrington, "Gmail Disaster: Reports of Mass Email Deletions,"http://www.techcrunch.com/2006/12/ 28/gmail-disasterreportsof-mass-email-deletions/, 2006.

[7] J. Kincaid,"MediaMax/TheLinkup Closes Its Doors," http://www.techcrunch.com/2008/07/10/mediam axthelinkup-closesits-doors/, July 2008.

[8] Amazon.com, "Amazon s3 Availability Event: July 20, 2008,"http://status.aws.amazon.com/s 30080720.html, July 2008.[8] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling PublicAuditability and Data Dynamics for Storage Security in CloudComputing," IEEE Trans. Parallel and Distributed Systems, vol. 22,no. 5, pp. 847-859, May 2011.

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

[10] M.A. Shah, R. Swaminathan, and M. Baker, "Isolation-PreservingAudit and Extraction of Digital Contents," Cryptology ePrintArchive, Report 2008/186, 2008.

[11] A. Juels and J. Burton, S. Kaliski, "PORs: Proofs of Retrievabilityfor Large Files," Proc. ACM Conf. Computer and Comm. Security(CCS '07), pp. 584-597, Oct. 2007.

[12] Cloud Security Alliance, "Security Guidance for Critical Areas ofFocus in Cloud Computing," http://www.cloudsecurityalliance.org, 2009.

[13] H. Shacham and B. Waters, "Compact Proofs of Retrievability,"Proc. Int'l Conf. Theory and Application of Cryptology and InformationSecurity: Advances in Cryptology (Asiacrypt), vol. 5350, pp. 90-107,Dec. 2008.

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

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

[16] 104th United States Congress, "Health Insurance Portability andAccountability Act of 1996 (HIPPA)," http://aspe.hhs.gov/admnsimp/pl104191.htm, 1996.

[17] R. Curtmola, O. Khan, and R. Burns, "Robust Remote DataChecking," Proc. Fourth ACM Int'l Workshop Storage Security andSurvivability (StorageSS '08), pp. 63-68, 2008.

[18] K.D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability:Theory and Implementation," Proc. ACM Workshop Cloud ComputingSecurity (CCSW '09), pp. 43-54, 2009.

[19] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from theWeil Pairing," J. Cryptology, vol. 17, no. 4, pp. 297-319, 2004.

[20] A.L. Ferrara, M. Green, S. Hohenberger, and M. Pedersen,"Practical Short Signature Batch Verification," Proc. Cryptographers'Track at the RSA Conf. 2009 on Topics in Cryptology (CT-RSA),pp. 309-324, 2009.

[21] G. Ateniese, R.D. Pietro, L.V. Mancini, and G. Tsudik,"Scalable and Efficient Provable Data Possession," Proc. Int'lConf. Security and Isolation in Comm. Networks (SecureComm '08),pp. 1-10, 2008.

[22] C. Wang,Q. Wang, K. Ren, and W. Lou, "Towards Secure andDependable Storage Services in Cloud

Computing," IEEE Trans.Service Computing, vol. 5, no. 2, 220-232, Apr.-June 2012.
[23] C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia,"Dynamic Provable Data Possession," Proc. ACM Conf. Computerand Comm. Security (CCS '09), pp. 213-222, 2009.
[24] R.C. Merkle, "Protocols for Public Key Cryptosystems," Proc. IEEESymp. Security and Isolation, 1980.
[25] G. Ateniese, S. Kamara, and J. Katz, "Proofs of Storage fromHomomorphic Identification Protocols," Proc. 15th Int'l Conf.Theory and Application of Cryptology and Information Security:Advances in Cryptology (ASIACRYPT), pp. 319-333, 2009.
[26] M. Bellare and G. Neven, "Multi-Signatures in the Plain Public-Key Model and a General Forking Lemma," Proc. ACM Conf.Computer and Comm. Security (CCS), pp. 390-399, 2006.
[27] Amazon.com, "Amazon Elastic Compute Cloud," http://aws.amazon.com/ec2/, 2009.[28] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. Yau, "EfficientProvable Data Possession for Hybrid Clouds," Cryptology ePrintArchive, Report 2010/234, 2010.
[28] Y. Dodis, S.P. Vadhan, and D. Wichs, "Proofs of Retrievability viaHardness Amplification," Proc. Theory of Cryptography Conf. Theoryof Cryptography (TCC), pp. 109-127, 2009.
[29] F. Sebe, J. Domingo-Ferrer, A. Martı´nez-Balleste, Y. Deswarte, andJ.-J. Quisquater, "Efficient Remote Data Possession Checking inCritical Information Infrastructures," IEEE Trans. Knowledge andData Eng., vol. 20, no. 8, pp. 1034-1038, Aug. 2008.
[30] T. Schwarz and E.L. Miller, "Store, Forget, and Check: UsingAlgebraic Signatures to Check Remotely Administered Storage,"Proc. IEEE Int'l Conf. Distributed Computing Systems (ICDCS '06),2006.
[31] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, "MR-PDP:Multiple-Replica Provable Data Possession," Proc. IEEE Int'l Conf.Distributed Computing Systems (ICDCS '08), pp. 411-420, 2008.
[32] K.D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availabilityand Integrity Layer for Cloud Storage," Proc. ACM Conf. Computerand Comm. Security (CCS '09), pp. 187-198, 2009.
[33] Cong Wang, Sherman S.M. Chow,Qian Wang,Kui Ren,Wenjing Lou,Isolation-Preserving Public Auditing for Secure Cloud Storage IEETOC,PP.62-2 Feb-2013.