PROBABILISTIC INFORMATION BASED CLUSTERING INVESTIGATION OF PROBABILITY DISTRIBUTION SIMILARITY

Mithilesh Kumar Singh[1], Tariq Siddiqui[2]

[1] *M.tech Scholar, Department of Computer Science & Engg., BERI,M.P.,India*
[2] *Assistant Professor, Department of Computer Science & Engg., BERI, M.P.,India*

ABSTRACT

Clustering is an essential task in data mining. The fundamental purpose of clustering is gathering the same object data in a massive dataset and identifying resemblances between the objects. Clustering of uncertain data is a more complicated task in both designing the similarity of data objects and implementing the efficient computational methods. Clustering uncertain data problems has been explained by utilizing many modern data mining techniques and numerous methods. Techniques have newly been convenient for clustering uncertain data based upon the conventional dividing clustering methods like k- means and density-based clustering methods like DBSCAN for uncertain data, they will be resolved by geometric distances between objects. Computing the resemblance between the data objects will be based upon a correlation distance measure and further clustered with occurrence based clustering or hierarchical clustering methods. Such methods cannot handle uncertain elements that are geometrically no conflict. In the recommended system, we could use probability that are the fundamental attributes of uncertain objects and are analyzed in measuring likeness between uncertain objects. The extremely suitable technique Kullback-Leibler divergence is employed to operations the distribution relationship between two uncertain data items. First the probability division method for a model, uncertain data object then thereafter estimate the similarity between data objects using distance metrics.

Keyword: *Clustering, Clustering uncertain data, density based clustering, partition clustering, KL- divergence.*

1. INTRODUCTION

Cloud Computing has been envisioned as the next generation architecture of IT enterprise, due to its long list of unprecedented advantages in the IT history: on-demand self-service, ubiquitous network access, location independent resource pooling, rapid resource elasticity, usage-based pricing and transference of risk [1]. As a disruptive technology with profound implications, Cloud Computing is transforming the very nature of how businesses use information technology. One fundamental aspect of this paradigm shifting is that data is being centralized or outsourced into the Cloud. From the user‘s perspective, including both individuals and IT enterprises, storing data remotely into the cloud in a flexible on-demand manner brings appealing benefits: relief of the burden for storage management, universal data access to independent sgeographical locations, and avoidance of capital expenditure on hardware, software, and personnel maintenances, etc. [2].

While Cloud Computing makes these advantages more appealing than ever, it also brings new and challenging security threats towards users‘ outsourced data. Since cloud service providers (CSP) are separate administrative entities, data outsourcing is actually relinquishing user‘s ultimate control over the fate of their data. As a result, the correctness of the data in the cloud are being put at risk due to the following reasons. First of all, although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. Examples of outages and security breaches of noteworthy cloud services appear from time to time [3]–[5].

Secondly, for the benefits of their own, there do exist various motivations for cloud service providers to behave unfaithfully towards the cloud 0users regarding the status of their outsourced data. Examples include cloud service providers, for monetary reasons, reclaiming storage by discarding data that have not been or is rarely

accessed, or even hiding data loss incidents so as to maintain a reputation [6]–[8]. In short, although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it does not offer any guarantee on data integrity and availability. This problem, if not properly addressed, may impede the successful deployment of the cloud architecture. As users no longer physically possess the storage of their data, traditional cryptographic primitives for the purpose of data security protection cannot be directly adopted. Thus, how efficiently verify the correctness of outsourced cloud data without the local copy of data files becomes a big challenge for data storage security in Cloud Computing. Note that simply downloading the data for its integrity verification is not a practical solution due to the expensiveness in I/O cost and transmitting the file across the network. Besides, it is often insufficient to detect the data corruption when accessing the data, as it might be too late to recover the data loss or damage. Considering the large size of the outsourced data and the user‘s constrained resource capability, the ability to audit the correctness of the data in a cloud environment can be formidable and expensive for the cloud users [8], [9].

Therefore, to fully ensure the data security and save the cloud users‘ computation resources, it is of critical importance to enable public audit ability for cloud data storage so that the users may resort to a third party auditor (TPA), who has expertise and capabilities that the users do not, to audit the outsourced data when needed. Based on the audit result, TPA could release an audit report, which would not only help users to evaluate the risk of their subscribed cloud data services, but also be beneficial to the cloud service provider to improve their cloud based service platform [7]. In a word, enabling public risk auditing protocols will play an important role in this nascent cloud economy to become fully established; where users will need ways to assess risk and gain trust in Cloud. Recently, the notion of public audit ability has been proposed in the context of ensuring remotely stored data integrity under different systems and security models [6], [8], [10], [11]. Public audit ability allows an external party, in addition to the user himself, to verify the correctness of remotely stored data. However, most of these schemes [6], [8], [10] do not support the privacy protection of users‘ data against external auditors, i.e., they may potentially reveal user data information to the auditors, as will be discussed in Section III-C. This severe drawback greatly affects the security of these protocols in Cloud Computing. From the perspective of protecting data privacy, the users, who own the data and rely on TPA just for the storage security of their data, do not want this auditing process introducing new vulnerabilities of unauthorized information leakage towards their data security [12]. Moreover, there are legal regulations, such as the US Health Insurance Portability and Accountability Act (HIPAA) [13], further demanding the outsourced data not to be leaked to external parties [7]. Exploiting data encryption before outsourcing [11] is one way to mitigate this privacy concern, but it is only complementary to the privacy-preserving public auditing scheme to be proposed in this paper. Without a properly designed auditing protocol, encryption itself cannot prevent data from ―flowing away towards external parties during the auditing process.

Thus, it does not completely solve the problem of protecting data privacy but just reduces it to the one of managing the encryption keys. Unauthorized data leakage still remains a problem due to the potential exposure of encryption keys. Therefore, how to enable a privacy-preserving third-party auditing protocol, independent to data encryption, is the problem we are going to tackle in this paper. Our work is among the first few ones to support privacy-preserving public auditing in Cloud Computing, with a focus on data storage. Besides, with the prevalence of Cloud Computing, a foreseeable increase of auditing tasks from different users may be delegated to TPA.

2. PRESENT WORK

2.1 Definitions and Framework

We follow a similar definition of previously proposed schemes in the context of remote data integrity checking [9], [11], [13] and adapt the framework for our privacy-preserving public auditing system. A public auditing scheme consists of four algorithms (KeyGen, SigGen, GenProof, VerifyProof). KeyGen is a key generation algorithm that is run by the user to setup the scheme. SigGen is used by the user to generate verification metadata, which may consist of digital signatures. GenProof is run by the cloud server to generate a proof of data storage correctness, while VerifyProof is run by the TPA to audit the proof. Running a public auditing system consists of two phases, Setup and Audit:

2.1.1 Setup

The user initializes the public and secret parameters of the system by executing KeyGen, and pre-processes the data file F by using SigGen 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 pre-processing, the user may alter the data file F by expanding it or including additional metadata to be stored at server.

2.1.2 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 GenProof using F and its verification metadata as inputs. The TPA then verifies the response via VerifyProof. Our framework assumes the TPA is stateless, i.e., TPA does not need to maintain and update state between audits, which is a desirable property especially in the public auditing system [13]. Note that it is easy to extend the framework above to capture a stateful auditing system, essentially by splitting the verification metadata into two parts which are stored by the TPA and the cloud server respectively. Our design does not assume any additional property on the data file. If the user wants to have more error-resilience, he can first redundantly encodes the data file and then uses our system with the data that has error-correcting codes integrated.

2.2 The Basic Schemes

Before giving our main result, we study two classes of schemes as a warm-up. The first one is a MAC-based solution which suffers from undesirable systematic demerits bounded usage and stateful verification, which may pose an additional online burden to users, in a public auditing setting. This also shows that the auditing problem is still not easy to solve even if we have introduced a TPA. The second one is a system based on homomorphic linear authenticators (HLA), which covers much recent proof of storage systems. We will pinpoint the reason why all existing HLA- based systems are not privacy preserving. The analysis of these basic schemes leads to our main result, which overcomes all these drawbacks. Our main scheme to be presented is based on an implementation of ECC on a cloud.

2.3 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 the corresponding secret key sk to the TPA. Later, the TPA can randomly retrieve blocks with their MACs and check the correctness via sk. Apart from the high (linear in the sampled data size) communication and computation complexities, the TPA requires the knowledge of the data blocks for verification. To circumvent the requirement of the data in TPA verification, one may restrict the verification to just consist of equality checking. However, it suffers from the following severe drawbacks:

1) The number of times a particular data file can be audited is limited by the number of secret keys that must be fixed a priori. Once all possible secret keys are exhausted, the user then has to retrieve data in full to re-compute and re-publish new MACs to TPA;
2) The TPA also has to maintain and update state between audits, i.e., keep track on the revealed MAC keys. Considering the potentially large number of audit delegations from multiple users, maintaining such states for TPA can be difficult and error prone;
3) It can only support static data, and cannot efficiently deal with dynamic data at all. However, supporting data dynamics is also of critical importance for cloud storage systems. For the reason of brevity and clarity, our main protocol will be presented based on static data.

2.4 HLA-based Solution

To effectively support public auditability without having to retrieve the data blocks themselves, the HLA technique [9], [13], [8] can be used. HLAs, like MACs, are also some unforgivable verification metadata that authenticates the integrity of a data block. The difference is that HLAs can be aggregated. It is possible to compute an aggregated HLA, which authenticates a linear combination of the individual data blocks.

2.5 Privacy-Preserving Public Auditing Scheme Overview

To achieve privacy-preserving public auditing, we propose to uniquely integrate the homomorphic linear authenticator with random masking technique. In our protocol, the linear combination of sampled blocks in the server‘s response is masked with randomly generated by the server. With random masking, the TPA 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, no matter how many linear combinations of the same set of file blocks can be collected. On the other hand, the correct validation of the block-authenticator pairs can still be carried out in a new way which will be shown shortly, even with the presence of the randomness. Our design makes use of a public key based ECDHA, to equip the auditing protocol with public auditability. Specifically, we use the ECC proposed in [2], which is based on the short signature scheme.

2.6 Support for Data Dynamics

In Cloud Computing, outsourced data might not only be accessed but also updated frequently by users for various application purposes [21], [8], [22], [23]. Hence, supporting data dynamics for privacy-preserving public auditing is also of paramount importance. Now we show how to build upon the existing work [8] and adapt our main scheme to support data dynamics, including block level operations of modification, deletion and insertion

2.7 Generalization

As mentioned before, our protocol is based on the ECC]. One may apply the random masking technique we used to construct the corresponding zero knowledge proof for different homomorphic identification protocols. Therefore, our privacy- preserving public auditing system for secure cloud storage can be generalized based on other complexity assumptions, such as factoring [25].

2.8 Digital Signature

The Digital signature scheme is designed to provide the digital counterpart to handwritten signatures A digital signature is a number depending on some secret known only to the signer (the signer‘s private key), and, additionally, on the contents of the message being signed. Signatures must be verifiable -- if a dispute arises as to whether an entity signed a document, an unbiased third party should be able to resolve the matter equitably, without requiring access to the signer‘s private key. An application generates a digital signature for a message by first applying a hash of the message to the digital signature generates callable service. For implementing the digital signatures will use the Hash Algorithm called SHA1.

2.8.1 SHA-1:

A hash function is simply an algorithm that takes a string of any length and reduces it to a unique fixed length string. Hashes are used to ensure data and message integrity, password validity as well as the basis of many other cryptographic systems. The SHA-1 is known as a one-way hash function, meaning there is no known mathematical method of computing the input given only the output. The specification of the SHA-1, as defined by Federal Information Processing Standards (FIPS) Publication 180-2, states that the input consists of 512 bit blocks with a total input length less than 264 bits. Inputs which do not conform to integer multiples of 512 bit blocks are padded before any block is an input to the hash function. The SHA-1 algorithm outputs 160 bits, referred to as the digest. The full SHA-1 specification A hash is not _encryption‘ – it cannot be decrypted back to the original text (it is a _one-way‘ cryptographic function, and is a fixed size for any size of source text). This makes it suitable when it is appropriate to compare _hashed‘ versions of texts, as opposed to decrypting the text to obtain the original version. Such applications include storing passwords, challenge handshake authentication, and digital signatures.

To validate a password,-you can store a hash of the password, then when the password is to be authenticated, you

hash the password the user supplies, and if the hashed versions match, the password is authenticated; but the original password cannot be obtained from the stored hash challenge handshake authentication (or_challenge hash authentication') avoids transmissions passwords in _clear' – a client can send the hash of a password over the internet for validation by a server without risk of the original password being intercepted

Anti-tamper – link a hash of a message to the original, and the recipient can re-hash the message and compare it to the supplied hash: if they match, the message is unchanged; this can also be used to confirm no data-loss in transmission.

Digital signatures are rather more involved, but in essence, you can sign the hash of a document by encrypting it with your private key, producing a digital signature for the document. Anyone else can then check that you authenticated the text by decrypting the signature with your public key to obtain the original hash again, and comparing it with their hash of the text.

3. METHEDOLOGY

3.1 Problem Statement

3.1.1 The System and Threat Model:- We consider a cloud data storage service involving three different entities, as illustrated in Fig. 1: has significant storage space and computation resources (we will not differentiate CS and CSP hereafter); the third party auditor (TPA), who has expertise and capabilities that cloud users do not have and is trusted to assess the cloud storage service reliability on behalf of the user upon request. Users rely on the CS for cloud data storage and maintenance. They may also dynamically interact with the CS to access and update their stored data for various application purposes. As users no longer possess their data locally, it is of critical importance for users to ensure that their data are being correctly stored and maintained. To save the computation resource as well as the online burden potentially brought by the periodic storage correctness verification, cloud users may resort to TPA for ensuring the storage integrity of their outsourced data, while hoping to keep their data private from TPA. We assume the data integrity threats towards users' data can come from both internal and external attacks at CS. These may include: software bugs, hardware failures, bugs in the network path, economically motivated hackers, malicious or accidental management errors, etc. Besides, CS can be self-interested. For their own benefits, such as to maintain reputation, CS might even decide to hide these data corruption incidents to users. Using third-party auditing service provides a cost-effective method for users to gain trust in Cloud. We assume the TPA, who is in the business of auditing, is reliable and independent. However, it may harm the user if the TPA could learn the outsourced data after the audit. Note that in our model, beyond users' reluctance to leak data to TPA, we also assume that cloud servers has no incentives to reveal their hosted data to external parties. On the one hand, there are regulations, e.g. HIPAA [16], requesting CS to maintain users' data privacy. On the other hand, as users' data belong to their business asset [10], there also exist financial incentives for CS to protect it from any external parties. Therefore, we assume that neither CS nor TPA has motivations to collude with each other during the auditing process. In other words, neither entities will deviate from the prescribed protocol execution in the following presentation. To authorize the CS to respond to the audit delegated to TPA's, the user can issue a certificate on TPA's public key, and all audits from the TPA are authenticated against such a certificate. These authentication handshakes are omitted in the following presentation.

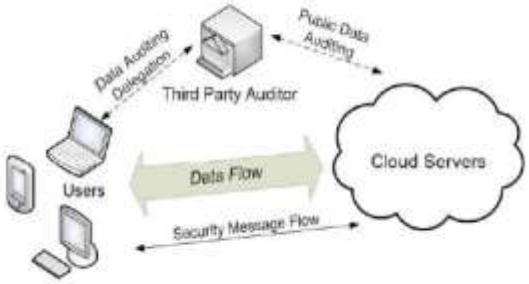

Figure 1:- The Architecture of cloud data storage

3.2 Design Goals

To enable privacy-preserving public auditing forcloud data storage under the aforementioned model, our protocol design should achieve the following security and performance guarantees.

- Public auditability: to allow TPA to verify the correctness of the cloud data on demand without retrieving a copy of the whole data or introducing additional online burden to the cloud users.
- Storage correctness: to ensure that there exists no cheating cloud server that can pass the TPA‘s audit without indeed storing users‘data intact.
- Privacy-preserving: to ensure that the TPA cannot derive users‘ data content from the information collected during the auditing process.
- Batch auditing: To enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously.
- Lightweight: to allow TPA to perform auditing with minimum communication and computation overhead.
- Cloud Computing Benefits: Cost including lower implementation and maintenance cost. Less hardware to purchase and support. Flexible and Agile computing Platform. Highly Scalable. High Performance resource. High Reliability. Cloud computing helps organizations to reduce power, cooling, storage and space usage. Better IT Resource management and Business Focus. Rapid Development, Deployment and change management. Better Performance. Improved Security.
- Architecture of Cloud Computing: NIST (National Institute of Standards and Technology) is a well accepted institution all over the world for their work in the field of Information Technology. NIST defines the Cloud Computing architecture by describing five essential characteristics, three cloud services models and four cloud deployment models As described above, there are 5 essential characteristics of Cloud Computing which explains there relation and difference from the traditional computing.
- On-demand-self-service: Consumer can provision or un-provision the services when needed, without the human interaction with the service provider.
- Broad Network Access: It has capabilities over the network and accessed through standard mechanism.
- Resource Pooling: The computing resources of the provider are pooled to serve multiple consumers which are using a multi-tenant model, with various physical and virtual resources dynamically assigned, depending on consumer demand.
- Rapid Elasticity: Services can be rapidly and elastically provisioned.
- Measured Service: Cloud computing systems automatically control and optimize resource usage by providing a metering capability to the type of services.

There are 3 Cloud Services Models and these 3 fundamental classifications are often referred to as software, platform or infrastructure as a service.

- Cloud Software as Service: This is a capability in which the consumer can use the provider‘s applications running on the cloud.
- Cloud Platform as Service: In this type of service, the consumer can deploy, the consumer created or acquired applications created by using programming languages or tools provided by provider, on the cloud infrastructure.
- Cloud Infrastructure as Service: This is a capability provided to the consumer by which, it can provision processing, storage, networks and other fundamental computing resources where the consumers can deploy and run the software.

Architecture is a layered model consisting of four layers such as Hardware layer, Infrastructure layer, Application layer and Platform layer

3.2.1 Hardware Layer:

This layer is responsible for managing the physical resources of the cloud, including physical servers, routers, switches, power and cooling systems. In practice, the hardware layer is typically implemented in data centers.

3.2.2 Infrastructure Layer:

Also known as the virtualization layer, the infrastructure layer creates a pool of storage and computing resources by partitioning the physical resource using virtualization technologies. The infrastructure layer is an essential

component of cloud computing, since many key features, such as dynamic resource assignment, are only made available through virtualization technologies.

3.2.3 Platform Layer:

Built on top of the infrastructure layer, the platform layer consists of operating systems and application frameworks. The purpose of the platform layer is to minimize the burden of deploying applications directly into VM containers.

3.2.4 Application Layer:

The application layer consists of the actual cloud applications.

.

4. RESULT

To perform the operation, steps are given below:

1. Login into Application
2. Select a dataset from desire destination
3. Import a dataset into Application
4. Read a Dataset values
5. Run a PMF on Dataset
6. Calculate a Density Estimation
7. Run a Divergence algorithm on Calculated Density
8. Calculate a Medoids of Dataset
9. Perform a K Means Clustering
10. Execute a Uncertain K-Medoids
11. Perform a Random K- Medoids Algorithm
12. Perform a Density Based Clustering

After performing the above steps of proposed method we will get the comparative graph of different clustering technique as shown in figure. 2.

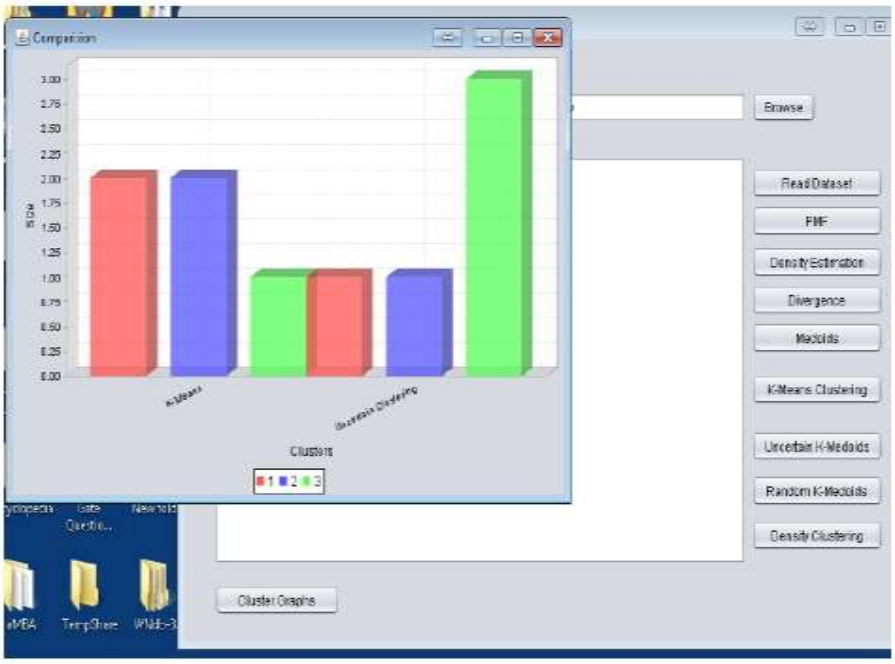

Figure. 2. Comparatively Graph between Different Clustering Algorithm

5. CONCLUSION

In this project, we delve into clustering uncertain data and propose a privacy-preserving public auditing system for data storage security in Cloud Computing. We utilize the Elliptical Curve Cryptography to guarantee 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 alleviates the users'fear of their outsourced data leakage. Considering TPA may concurrently handle multiple audit sessions from different users for their outsourced data files, we further extend our privacy preserving public auditing protocol into a multiuser setting, where the TPA can perform multiple auditing tasks in a batch manner for better efficiency. Extensive analysis shows that our schemes are provably secure and highly efficient. Now a day_s Cloud Computing facing security Challenges. User put their data in the cloud and data is being transferred from one Cloud to another and users are concerned about the security. We are concerned higher security of Data and therefore we proposed an Encryption Algorithm i.e. ECC which takes least time to encrypt the Data than others and will ensure about the faster retrieval of Data. Security related parameters such as Encryption, Authentication and Access Control, Separation of Duties for the security has been satisfied in this Algorithm in order to achieve the Security. The presented simulation results showed that ECC has a better performance and more secure than other Encryption Algorithms.

The data which is being transmitted is in encrypted form so that no third party user will be able to access the data and the entire data will gets Encrypted in the form of ECC Algorithm.

6. FUTURE SCOPE

- To newly propose a more secured system in which, if the users access data without permission must be blocked from the entire network.
- A Proxy Re-encryption scheme and also the parameters of higher bits which satisfy the ECC Algorithm has been taken into consideration for providing higher security of data.

7. REFERENCES

[1]. C. Wang, Q. Wang, K. Ren, and W. Lou, ―Privacy- preserving public auditing for storage security in cloud computing,‖ in Proc. of IEEE INFOCOM'13, Feb 2013.

[2]. P. Mell and T. Grance, ―Draft NIST working definition of cloud computing,‖ Referenced on June. 3rd, 2009. http://csrc.nist.gov/groups/SNS/cloud- computing/index.html.

[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. Zaharia, ―Above the clouds: A Berkeley view of cloud computing,‖ University of California, Berkeley, Tech. Rep. UCBEECS- 2009-28, Feb 2009.

[4]. Cloud Security Alliance,Top threats to cloud computing, 2010,http://www.cloudsecurityalliance.org.

[5]. M. Arrington, ―Gmail disaster: Reports of mass email deletions,‖2006, http://www.techcrunch.com/2006/12/28/gmail- disasterreports-of-mass-email-deletions/.

[6]. J. Kincaid, ―MediaMax/The Linkup closes its doors,‖ July 2008, http://www.techcrunch.com/2008/07/10/ mediamaxthelinkup-closes-its-doors/.

[7]. Amazon.com, ―Amazon s3 availability event: July 20, 2008,‖ http://status.aws.amazon.com/s3-20080720.html, 2008.

[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 Transactions on Parallel and Distributed Systems, vol. 22, no. 5, pp. 847–859, 2011.

[9]. G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, ―Provable data possession at untrusted stores,‖ in Proc. of CCS'07, 2007, pp. 598–609.

[10]. M. A. Shah, R. Swaminathan, and M. Baker, ―Privacy-preserving audit and extraction of digital contents,‖ Cryptology ePrint Archive, Report 2008/186, 2008.

[11]. A. Juels and J. Burton S. Kaliski, ―PORs: Proofs of retrievability for large files,‖ in Proc. of CCS'07, October 2007, pp. 584–597.

[12]. Cloud Security Alliance, ―Security guidance for critical areas of focus in cloud computing,‖ 2009, http://www.cloudsecurityalliance.org.

[13]. H.Shacham and B.Waters, ―Compact proofs of retrievability,‖ in Proc. of Asiacrypt, vol. 5350, Dec 2008, pp. 90–107.

[14]. C.Wang,K. Ren, W. Lou, and J. Li, ―Towards publicly auditable secure cloud data storage services,‖ IEEE Network Magazine,vol. 24, no. 4, pp. 19–24, 2010.

[15]. M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, ―Auditing to keep online storage services honest,‖ in Proc. of HotOS‘07, 2007, pp. 1– 6.

[16]. 104th United States Congress ―Health Insurance Portability and Accountability Act of 1996 (HIPPA),‖Online at http://aspe.hhs.gov/admnsimp/pl104191.htm, 1996.

[17]. R. Curtmola, O. Khan, and R. Burns, ―Robust remote data checking,‖ in Proc. of the 4th ACM international workshop on Storage security and survivability (StorageSS‘08), 2008, pp. 63–68.

[18]. K. D. Bowers, A. Juels, and A. Oprea, ―Proofs of retrievability: Theory and implementation,‖ in Proc. of ACM workshop on Cloud Computing security (CCSW‘09), 2009, pp. 43–54.

[19]. D. Boneh, B.Lynn, and H.Shacham, ―Short signatures from the Weil 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,‖ in Proc. of CT-RSA, volume 5473 of LNCS. Springer-Verlag, 2009, pp. 309–324.

[21]. G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, ―Scalable and efficient provable data possession,‖ in Proc. of SecureComm‘08, 2008, pp. 1–10.

[22]. C. Wang, Q. Wang, K. Ren, and W. Lou, ―Towards secure and dependable storage services in cloud computing,‖ IEEE Transactions on Service Computing, 2011, to appear.

[23]. C. Erway, A. Kupcu, C. Papamanthou, and R. Tamassia, ―Dynamic provable data possession,‖ in Proc. of CCS‘09, 2009, pp. 213–222.

[24] R.C. Merkle, ―Protocols for public key cryptosystems,‖ in Proc. of IEEE Symposium on Security and Privacy, 1980.

[25]. G.Ateniese,S.Kamara, and J.Katz, ―Proofs of storage from homomorphic identification protocols,‖ in Proc. of ASIACRYPT, 2009, pp. 319–333.

[26]. M.Bellare and G. Neven, ―Multi-signatures in the plain public key model and a general forking lemma,‖ in Proc. of CCS, 2006, pp. 390–399.

[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, ―Efficient provable data possession for hybrid clouds,‖ Cryptology ePrint Archive, Report 2010/234, 2010.

[29]. Y. Dodis, S. P. Vadhan, and D. Wichs, ―Proofs of retrievability via hardness amplification,‖ in TCC, 2009, pp. 109–127.

[30]. F. Sebe, J. Domingo-Ferrer, A. Mart´ınez-Balleste, Y. Deswarte, and J.-J. Quisquater, ―Efficient remote data possession checking in critical information infrastructures,‖ IEEE Transactions on Knowledge and Data Engineering, vol. 20, no. 8, pp. 1034–1038, August 2008.

[31]. T. Schwarz and E. L. Miller, ―Store, forget, and check: Using algebraic signatures to check remotely administered storage,‖ in Proc. of ICDCS‘06, 2006.

[32]. R. Curtmola, O. Khan, R. Burns, and G. Ateniese, ―MR-PDP: Multiple-replica provable data possession,‖ in Proc. of ICDCS‘08. IEEE Computer Society, 2008, pp. 411–420.

[33]. K. D. Bowers, A. Juels, and A. Oprea, ―HAIL: A high-availability and integrity layer for cloud storage,‖ in Proc. of CCS‘09, 2009, pp. 187–198.