ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

A Comparative Study on Privacy-Preserving Public Auditing for Secure Cloud Storage

Vikram.J[1], M.Kalimuthu[2]

PG Scholar, Department of Information Technology, SNS College of Technology, Coimbatore, Tamil Nadu, India[1].

Associate Professor, Department of Information Technology, SNS College of Technology, Coimbatore, Tamil Nadu, India[2].

ABSTRACT: The Cloud computing is a latest technology which provides various services through internet. The Cloud server allows user to store their data on a cloud without worrying about correctness & integrity of data. Cloud data storage has many advantages over local data storage. User can upload their data on cloud and can access those data anytime anywhere without any additional burden. The User doesn't have to worry about storage and maintenance of cloud data. But as data is stored at the remote place how users will get the confirmation about stored data. Hence Cloud data storage should have some mechanism which will specify storage correctness and integrity of data stored on a cloud. The major problem of cloud data storage is security. The Log Records stored in log file of an organization may contain sensitive data which should be protected properly for proper working of an organization. Maintaining security of such log records is one of the important tasks. Also, over a long period of time maintaining integrity of such log data is very important. However, deploying such a system for security of log records is an overhead for an organization and also itrequires additional cost. Many researchers have proposed their work or new algorithms for security of log records or to resolve this security problem. This study also reveals about consistency rationing and various adaptive policies. In this work, propose a novel privacy-preserving mechanism that supports public auditing on shared data stored in the cloud. In particular, we exploit ring signatures to compute verification metadata needed to audit the correctness of shared data. With our mechanism, the identity of the signer on each block in shared data is kept private from public verifiers, who are able to efficiently verify shared data integrity without retrieving the entire file. In addition, our mechanism is able to perform multiple auditing tasks simultaneously instead of verifying them one by one. Our experimental results demonstrate the effectiveness and efficiency of our mechanism when auditing shared data integrity.

KEYWORDS: Cloud Computing, Privacy Preserving, Access Control, Public Auditing, Logging, Consistency, Data Anonymization

I. INTRODUCTION

Cloud Computing is the dreamed vision of computing as a public utility. It is a model for enabling convenient, on-demand network access to shared pool of configurable computing resources (e.g. networks, servers, storage, application and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. A cloud provider is a company which hosts the servers on its premises and makes the services available on-demand. The ever cheaper and more powerful processors, together with the "software as a service" (SaaS) computing architecture, are transforming data centers into pools of computing service on a huge scale. Meanwhile, the increasing network bandwidth and reliable yet flexible network connections make it even possible that clients can now subscribe high-quality services from data and software that reside solely on remote data centers. The construction of cloud and storing data in it has tremendous benefits. It facilitates the authenticated and authorized cloud users to access enormous resources that are outsourced and shared in the cloud.

Whenever required, the user can request and gain the access (only, if the users' credentials are validated [1]) in an easy way and at low cost, irrespective of the user location. Also, cloud computing takes away the expenses spent on installing all hardware and software, by allowing users to rent the resources based on their needs. Despite all these benefits, cloud computing still faces many challenges which forbid the successful implementation of the cloud. These

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

include both the traditional as well as cloud security challenges [2]. Specific to cloud computing, the issues are many, of which some are: identity management of cloud users, multi-tenancy support, securing the security of applications, preserving privacy of the users, attaining control over the life cycle of outsourced data, etc. Among which, the issues related to privacy preserving are alone looked at in this survey. Preserving the privacy of user, his identity and data in the cloud is very mandatory. With the rise in growth of cloud computing, the concerns about privacy preserving are also getting increased [3, 4]. But reaching the peak in providing and assuring privacy-preserved data access in cloud is yet in progress and still needs much attention to attain the goal.

All activities performed in an organization are recorded in log files. Log files keep track of all user activities. Also, log files are used to troubleshoot the problems, to identify users violating the policies or performing malicious activities. Log file is the most important target for malicious attacks [5]. The reason behind this is that attacker wants to leave no traces of the activities performed by him at the time of attack. Therefore, the first target of attack is normally log files of an organization. Not only has this but log files also contained information about confidential transactions performed in the organization. This sensitive data must be protected [6]. Also log file data can be used for unauthorized access of the system. From this scenario, it is clear that security log data is one of the most important tasks of an organization. Actually, different applications have different consistency requirements. For example, mail services need monotonicread consistency and read-your-write consistency, but social network services need causal consistency [7].

In cloud storage, consistency not only determines correctness but also the actual cost per transaction. In this work to solve the above privacy issue on shared data, we propose Oruta, a novel privacy-preserving public auditing mechanism. More specifically, we utilize ring signatures [8] to construct homomorphic authenticators [9] in Oruta, so that a public verifier is able to verify the integrity of shared data without retrieving the entire data—while the identity of the signer on each block in shared data is kept private from the public verifier. In addition, we further extend our mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks. Meanwhile, Oruta is compatible with random masking [10], which has been utilized in WWRL and can preserve data privacy from public verifiers. Moreover, we also leverage index hash tables from a previous public auditing solution [11] to support dynamic data.

II. RELATED WORK

A cloud is essentially a large-scale distributed system where each piece of data is replicated on multiple geographically distributed servers to achieve high availability and high performance. Thus, we first review the consistency models in distributed systems. In [10], as a standard textbook, proposed two classes of consistency models: data-centric consistency and client-centric consistency. Data-centric consistency model considers the internal state of a storage system, i.e., how updates flow through the system and what guarantees the system can provide with respect to updates. However, to a customer, it really does not matter whether or not a storage system internally contains any stale copies. As long as no stale data is observed from the client's point of view, the customer is satisfied. Therefore, client-centric consistency model concentrates on what specific customers want, i.e., how the customers observe data updates. Their work also describes different levels of consistency in distributed systems, from strict consistency to weak consistency. High consistency implies high cost and reduced availability. In [11] states that strict consistency is never needed in practice, and is even considered harmful. In reality, mandated by the CAP protocol [14, 15], many distributed systems sacrifice strict consistency for high availability. Then, we review the work on achieving different levels of consistency in a cloud.

In [16] investigated the consistency properties provided by commercial clouds and made several useful observations. Existing commercial clouds usually restrict strong consistency guarantees to small datasets (Google's MegaStore and Microsoft's SQL Data Services), or provide only eventual consistency (Amazon's simpleDB and Google's BigTable). In [17] described several solutions to achieve different levels of consistency while deploying database applications on Amanzon S3. In [18], the consistency requirements vary over time depending on actual availability of the data, and the authors provide techniques that make the system dynamically adapt to the consistency level by monitoring the state of the data. In [19] proposed a novel consistency model that allows it to automatically adjust the consistency levels for different semantic data. Finally, we review the work on verifying the levels of consistency provided by the CSPs from

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

the users' point of view. Existing solutions can be classified into trace-based verifications and benchmark-based verifications [20, 21]. Trace-based verifications focus on three consistency semantics: safety, regularity, and atomicity, which are proposed by Lamport [22], and extended by Aiyer et al. [23].

A register is safe if a read that is not concurrent with any write returns the value of the most recent write, and a read that is concurrent with a write can return any value. A register is regular if a read that is not concurrent with any write returns the value of the most recent write, and a read that is concurrent with a write returns either the value of the most recent write, or the value of the concurrent write. A register is atomic if every read returns the value of the most recent write. Misra [24] is the first to present an algorithm for verifying whether the trace on a read/write register is atomic. Following his work, In [25] proposed offline algorithms for verifying whether a key-value storage system has safety, regularity, and atomicity properties by constructing a directed graph. In [26] proposed an online verification algorithm by using the GK algorithm and used different metrics to quantify the severity of violations. The main weakness of the existing trace-based verifications is that a global clock is required among all users. Our solution belongs to trace-based verifications. However, we focus on different consistency semantics in commercial cloud systems, where a loosely synchronized clock is suitable for our solution.

Benchmark-based verifications focus on benchmarking staleness in a storage system. In [27] used multiple geographically-distributed users to read data, and found that S3 frequently violates monotonic-read consistency. The results of justify our two-level auditing structure. In [28] presents a client-centric benchmarking methodology for understanding eventual consistency in distributed keyvalue storage systems. In [29] assessed Amazon, Google, and Microsoft's offerings, and showed that, in Amazon S3, consistency was sacrificed and only a weak consistency level known as, eventual consistency, was achieved.

Jiang Wang et al. put forward an Anonymity-based method to achieve and preserve privacy in cloud computing [30]. The anonymity algorithm processes the data and anonymises all or some information before releasing it in the cloud milieu. When required, the cloud service provider makes use of the background knowledge it has and incorporates the details with the anonymous data to mine the needed knowledge. This approach differs from the traditional cryptography technology for preserving user's privacy as it gets rid of key management and thus it stands simple and flexible. While anonymising is easier, the attributes that has to be made anonymous varies and it depends on the cloud service provider. This approach will be suitable only for limited number of services. Thus, the method has to be bettered by automating the anonymisation.Architecture for database storage [31] in cloud is proposed in this paper, which preserves the privacy of users' data. This approach prevents the risk of both external and internal attacks to the outsourced data. The main architectural elements are the user interface, user engine, rule engine and the cloud database. Through user interface, the request for accessing database is obtained, which is sent as an XML/ RPC request to the user engine, rule engine and finally to the cloud database.

By means of encrypting and assigning secured identities for each request and response at each stage, together with the maintenance of machine readable usage /access rights, privacy is preserved. While it is easier to carry out the encryption schemes, there exists a difficulty in providing machine readable access rights. This problem of effective right expressions generation is the future work that has to be carried out. Miao Zhou et al. [32] considered the privacy of users in the cloud environment and proposed a flexible method of access control. Each cloud user is linked with certain attributes, which determines their access rights. The paper propounded a two-tier encryption model in which the base phase and surface phase builds up the two tiers of the model respectively. At the first phase, the data owner performs local attribute-based encryption on the data that has to be outsourced. The surface phase on the other hand is performed by the cloud servers, after the initialization done by the cloud data owner. This phase implements the T. JothiNeela and N. Saravanan Server re-encryption mechanism (SRM).

The SRM dynamically re-encrypts the encrypted data in the cloud, when the owner of that data requests. The request for SRM arises either when a new user has to be created or an existing user has to be repealed. Though the re-encryption takes place in cloud server, the privacy of users data is not compromised as the access policies remains hidden to the cloud servers. Thus, in this paper privacy of data is preserved by providing full access control to the owner of the data and by disallowing the cloud provider to gain knowledge about the data. David W. Chadwick et al. explained a policy based authorization infrastructure for the cloud with the intention of preserving privacy of user's

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

data [33]. Users can define their access policies and cling it to their data. This assures the controlled access of data in the cloud. Policy Decision Points (PDPs) and Policy Enforcement Points (PEPs) are used for making authorization decisions and enforcing these decisions respectively. Master PDP is launched which figures out and solves the conflicts among various decisions of different PDPs. Obligation service is provided as a part of the authorisation infrastructure, by means of which the data owner is intimated about the authorized or unauthorized access of their data.

This authorisation infrastructure trusts the cloud providers and considers only the threats that come from outsiders. As the cloud provider is trusted, encryption of outsourced data is not done. The enhancement of this approach could be done by focusing on security threats from cloud providers and also by partitioning the infrastructure into separate services, each running in a distinct virtual machine. This would step-up the performance of the system. In [34], one another method is constructed for preserving the confidentiality of users' data. The privacy constraints are illustrated by means of a graph. The nodes and links represent the attributes and confidentiality between the corresponding nodes respectively. Sensitive attributes are the subset of the entire group of attributes. These attributes should not be leaked out to the external party. A relation is drawn over such attributes, which is then vertically fragmented. While the owner has one fragment, the other fragment is placed at the external server. By making use of a common id, the relation can then be reconstructed. A graph coloring algorithm is used to perform fragmentation and placing the fragments at the appropriate location, as well. While fragmenting, it is necessary to check that the workload is kept minimized at the source and also the confidentiality constraints not been breached by the server fragment.

The fragmentation is carried out based on certain metrics like Min-Attr, Min-Query and Min-Cond. These metrics combined with the appropriate fragmentation guarantees that the outsourced data will always be protected from third party attacks, thus ensuring keeping up of the privacy. Thus this work adopted only fragmentation to attain privacy effectively and efficiently, keeping the cryptographic techniques aside. The effectiveness can still be improved, by constructing a hyper graph rather than a two-dimensional graph. Preserving cloud computing Privacy (PccP) model is explained in [35], which is one another approach to attain privacy. The Consumer Layer forms the basement of the model, where the cloud users have to submit their request for accessing cloud services. The second layer is the Network Interface or Address Mapping layer. The function of this layer is to modify the original IP address related to the access request. Thus, it assures the privacy of users' IP address. Next layer is the Privacy Preserved Layer, which is the topmost layer of the model. This layer has an associated Unique User Cloud Identity Generator.

Hence, this layer preserves the privacy of users' sensitive information by implementing the Privacy check mechanism. This mechanism enables the user to specify the access control and the amount of data transparency in the cloud. If a particular Personal Data Attribute (PDA) of a user has to be specified with the transparency level, then a Boolean function of the attribute is to be carried out, which is named as Transparency Purpose in Cloud (TPC). Thus, PccP forecloses both the access of user identification and data content. AdeelaWaqar et al. [36] focused on the possibility of metadata exploitation in the cloud. By gaining knowledge of the metadata, the attacker could compromise users' privacy. As a solution, a framework is proposed to preserve the data privacy. First, the metadata that has to be put in cloud's database are segregated. The segregated attributes are then grouped as exclusively private, partially private and nonprivate depending on the sensitivity of data. Following this data classification, the next phase called table splitting comes up, where the database tables are divided both horizontally and vertically. The splitting of the database table ensures the database normalization. Next is to perform metadata reconstruction as and when required by the cloud.

This phase is called ephemeral referential consonance. This phase guarantees that data is not leaked from the cloud database both before and after splitting. These steps are illustrated by considering the possible attacks on metadata kept in Eucalyptus database files and ensuring the prevention of attacks by the proposed framework. Thus, the method proves to be efficient. C. Wang et al. proposed a privacy-preserving method to carry out public auditing on the cloud information [37]. In case of cloud computing, it is not sufficient to adopt the traditional cryptographic measures to achieve security. The reason is due to data outsourcing and the ubiquitous nature of the data. So, in this paper they opt the concept of Third Party Auditing (TPA). Homomorphic authenticator and random masking ensures that TPA could not gain any knowledge during the process of auditing. Thus, TPA is trusted and capable of accessing the cloud storage to perform auditing. The audit report brings out the risks, if any is present in the data. The public auditing system is built using four algorithms and two phases.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

KeyGen and SigGen algorithms make up the first phase called Setup, in which initialization of secret parameters and generation of verification metadata are done. Following this, the Audit phase carries out the auditing process and ascertains the correctness of data in the cloud server. This is done in this second phase using GenProof and VerifyProof algorithms. The approach guarantees the correctness of data in cloud server, preservability of privacy and security for batch auditing (simultaneous auditing for multi-user setup). C. Wang et al. in [38] enhanced their previous proposal by improving the security strength of data storage. A new protocol for privacy-preserving public auditing is designed for this purpose. Public auditing with zero-knowledge leakage is also achieved. Batch auditing is also enhanced with the improvement in main auditing scheme. As an extension to the previous work, the authors put forward the support for data dynamics and generalization of the auditing scheme in this paper.

An experiment is conducted on an instance of Amazon EC2 and the better performance of the proposed design is proved. Boyang Wang et al. [39] analyzed the work of Wang et al. and propounded another public auditing mechanism. Oruta provides data privacy, identity privacy. Also it ensures correctness and unforgeability while carrying out the public auditing. However, the identity privacy is not achieved. The approach considers three main entities: the cloud server, TPA and the users. Users are statically grouped as the original user (owner of the outsourced data) and group users. The original user can control their data and its flow in the cloud. All users request and depend on the TPA to carry out auditing for verifying the rightness of data. Homomorphic Authenticable Ring Structures (HARS) scheme, comprising three algorithms: KeyGen, RingSign and RingVerify are constructed here for achieving the privacy-preserving auditing. Still, the approach can be empowered by focusing on an efficient auditing approach to ascertain the integrity of shared data in dynamically grouped users' environment. Appointing the logs record to the cloud environment saves the cost In this paper we are suggesting the homomorphic encryption scheme that provides a strong security.

D. New and M. Rose declared in their work as: The BSD Syslog Protocol [40] defines a number of service associated options and also relate to inseminating event messages. This message also describes the two mappings of the syslog protocol to TCP connections, both which are helpful for transmitting trustworthy delivery of event messages. This administers a trivial mapping maximizing backward compatibility and also helps in supplying a more entire mapping. Both provide a degree of sturdiness and security in message delivery that is engaged to the usual UDP-based syslog protocol, by providing encryption and authentication over a connection-oriented protocol. M. Bellare and B. S. Yee[41] explaining as: Applications incorporate more secure audit logs (e.g., syslogd data) for intrusion exposure or accountability, communications security, and authenticating incomplete results of computation for mobile agents. Computer audit logs including descriptions of notable events which crashes of system programs, system resource consumption, and failed login attempts, etc. Many of these events are serious for investigating analysis after a break-in.

The aim of an experienced attacker will be the audit log system: the attacker desires to remove traces of the compromise, to avoid detection as well as to maintain the method of attack undisclosed so that the security gap broken will not be detected by the system administrator. To construct the audit log secure, we must avoid the attacker from modifying the audit log data.J. E. Holt [42] predictable as: The famous application Tripwire manages cryptographic fingerprints of all files on a computer granting the administrators to identify when attackers compromise the system and modify the essential system files. But Tripwire is incompatible for system logs and other file that alters often, since the fingerprints generates affect to files in their intactness. A number of peoples have projected cryptographic techniques which permit each new log entry to be fingerprinted, blocking attackers from discarding proof of their attacks from system logs. B. Schneier and J. Kelsey [43] discussed on the following: In many real-world applications, sensitive data are put in logs which are fewer on an untrusted machine.

When an incident takes place as an attacker controls this machine, it is assured that the attacker will achieve tiny or no information from the log less and to bound his ability to damage the log files. Here the projected system shows a computationally inexpensive method for making all log entries generated earlier to the logging machine's compromise unfeasible for the attacker to read and also unfeasible to unnoticeably change or destroy. A computer that uses logs of different kinds of network activity wishes to have log entries of an attack undeletable and not alterable, even in the occasion that an attacker takes over the logging machine over the network. An intrusion-detection system that logs the starting access and exit of people into a secured region desires to oppose attempts to scrub out or alter logs, even after the machine on which the logging takes place has been taken over by an attacker.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

D. Ma and G. Tsudik [44] stated as: The necessity for secure logging is well-understood by the security professionals, counting both researchers and practitioners. The capability to professionally validate all log entries is more essential to any purpose handling secure logging techniques. In this paper, we start by investigating state-of-the-art in secure logging and recognize some troubles inborn to systems based on trusted third-party servers. They suggest a very dissimilar approach to secure logging based upon newly developed Forward-Secure Sequential Aggregate authentication techniques. D. Dolev and A. Yao [45] explained as: newly the use of public key encryption was to offer a secure network communication which has established a significant attention. Such public key systems are frequently useful in providing against the passive eavesdroppers, who regularly try to strike the lines and try to decode the message. It has been pointed out, that an inappropriate designed procedure could be susceptible to an active behavior like that, and one who may imitate another user or change the message being transmitted. Numerous models have been prepared in which the security of protocols are discussed accurately. Algorithms and characterizations are used in finding protocol security in these models which have been given. The use of public key encryption was to offer a secure network communication which has established substantial attention.

Inference:

This paper analyses and discusses various methods like adopting cryptographic methods, writing access rights and policies, anonymising data, segregating or fragmenting and then reconstructing the data, secure logging, consistency. All these approaches would preserve the privacy of user and data and while performing public auditing on the cloud data. In this a comprehensive system to securely contract out log records to a cloud service provider. Still there is a need to protect the log records for the proper functioning of any organization and security threat hinders the success of Cloud Computing. The comparative study of table is given as follows:

Table.1. Comparative study

Method	Parameters Used	Advantages	Disadvantages
A novel consistency model	Number of data, Dimensionality of data space	High correlation among the tuples	More Number of dimensions would be violated
Anonymity-based method	Distinct datas, Maximum transaction size and Average transaction size on data	Global Anonymization to ensure privacy	Loss of utility
Homomorphic Authenticable Ring Structures (HARS) scheme	Single, Complete and Average link	Partition the records into equivalence Classes	Utility was still not achieved
Public auditing mechanism	Dimensionality of data space	ensures correctness and unforgeability	the identity privacy is not achieved
Preserving cloud computing Privacy	Number of data	preserves the privacy of users' sensitive information	It is not sufficient to adopt the traditional cryptographic measures to achieve security

III. CONCLUSION

In this paper, we have reviewed several existing techniques for secure logging and consistency. Some of the privacy threats are addressed and the techniques to overcome them are surveyed. While some approaches utilized traditional cryptographic methods to achieve privacy, some other approaches kept them away and focused on alternate methodologies in achieving privacy. Also, approaches to preserve privacy at the time of public auditing are also discussed. Thus, to conclude it is necessary that every cloud user must be guaranteed that his data is stored, processed, accessed and audited in a secured manner at any time. The user must be given complete access control over the published data. Also, powerful security mechanisms must always supplement every cloud application. Design

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

architecture that reduces cost and overhead to secure log records of an organization. To attain all these needs a privacy-preserving public auditing mechanism for shared data in the cloud. We utilize ring signatures to construct homomorphic authenticators, so that a public verifier is able to audit shared data integrity without retrieving the entire data, yet it cannot distinguish who is the signer on each block. To improve the efficiency of verifying multiple auditing tasks, we further extend our mechanism to support batch auditing.

REFRENCES

[1] Takabi H (2010). Security and Privacy Challenges in Cloud Computing Environments, IEEE Security & Privacy, vol 8(6), 24–31.
[2] Rong C, Nguyen S T et al. (2013). Beyond lightning: A survey on security challenges in cloud computing, Computers & Electrical Engineering, vol 39(1), 47–54.
[3] Gellman, R. (2009). WPF REPORT: Privacy in the clouds: Risks to privacy and confidentiality from cloud computing. Released February, 23.
[4] Xiao, Z., & Xiao, Y. (2013). Security and privacy in cloud computing. Communications Surveys & Tutorials, IEEE, 15(2), 843-859.
[5] Indrajit Ray,K.Belyaev,‖Secure Logging As A Service-Delegating log management to the cloud ‖, IEEE Systems Journal,June 2013
[6] D. Ma and G. Tsudik, ―A new approach to secure logging,‖ ACM Trans. Storage, vol. 5, no. 1, pp. 2:1–2:21, Mar. 2009.
[7] W. Lloyd, M. Freedman, M. Kaminsky, and D. Andersen, "Don't settle for eventual: scalable causal consistency for wide-area storage with COPS," in Proc. 2011 ACM SOSP
[8] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," Proc. 22nd Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT'03), pp. 416-432, 2003.
[9] H. Shacham and B. Waters, "Compact Proofs of Retrievability," Proc. 14th Int'l Conf. Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '08), pp. 90107, 2008.
[10] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," Proc. IEEE INFOCOM, pp. 525-533, 2010.
[11] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S.S Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storages in Clouds," Proc. ACM Symp. Applied Computing (SAC'11), pp. 1550-1557, 2011
[12] A. Tanenbaum and M. Van Steen, Distributed Systems: Principles and Paradigms. Prentice Hall PTR, 2002.
[13] W. Vogels, "Data access patterns in the Amazon.com technology platform,"in Proc. 2007 VLDB.
[14] E. Brewer, "Towards robust distributed systems," in Proc. 2000 ACM PODC.
[15] Brewer, E, "Pushing the CAP: strategies for consistency and availability,"Computer, vol. 45, no. 2, 2012.
[16] Vogels, W., "Eventually consistent," Commun. ACM, vol. 52, no. 1, 2009.
[17] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska, "Building a database on S3," in Proc. 2008 ACM SIGMOD.
[18] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann, "Consistency rationing in the cloud: pay only when it matters," in Proc. 2009 VLDB.
[19] S. Esteves, J. Silva, and L. Veiga, "Quality-of-service for consistency of data geo-replication in cloud computing," Euro-Par 2012 Parallel Processing, vol. 7484, 2012.
[20] H. Wada, A. Fekete, L. Zhao, K. Lee, and A. Liu, "Data consistency properties and the trade-offs in commercial cloud storages: the consumers' perspective," in Proc. 2011 CIDR
[21] D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the cloud," in Proc. 2010 ACM SIGMOD.
[22] L. Lamport, "On interprocess communication," Distributed Computing, vol. 1, no. 2, 1986.
[23] A. Aiyer, L. Alvisi, and R. Bazzi, "On the availability of non-strict quorum systems," Distributed Computing, vol. 3724, 2005.
[24] J. Misra, "Axioms for memory access in asynchronous hardware systems," ACM Trans. Programming Languages and Systems, vol. 8, no. 1, 1986.
[25] E. Anderson, X. Li, M. Shah, J. Tucek, and J. Wylie, "What consistency does your key-value store actually provide," in Proc. 2010 USENIX HotDep.
[26] W. Golab, X. Li, and M. Shah, "Analyzing consistency properties for fun and profit," in Proc. 2011 ACM PODC.
[27] D. Bermbach and S. Tai, "Eventual consistency: how soon is eventual?" in Proc. 2011 MW4SOC.
[28] M. Rahman, W. Golab, A. AuYoung, K. Keeton, and J. Wylie, "Toward a principled framework for benchmarking consistency," in Proc. 2012 Workshop on HotDep.
[29] D. Kossmann, T. Kraska, and S. Loesing, "An evaluation of alternative architectures for transaction processing in the cloud," in Proc. 2010 ACM SIGMOD.
[30] Wang J, Zhao Y et al. (2009). Providing Privacy Preserving in cloud computing, International Conference on Test and Measurement, vol 2, 213–216.
[31] Greveler U, Justus b et al. (2011). A Privacy Preserving System for Cloud Computing, 11th IEEE International Conferenceon Computer and Information Technology, 648–653.
[32] Zhou M, Mu Y et al. (2011). Privacy-Preserved Access Control for Cloud Computing, International Joint Conference of IEEE TrustCom-11/IEEE ICESS-11/FCST-11, 83–90.
[33] Chadwick D W, and Fatema K (2012). A privacy preserving authorisation system for the cloud, Journal of Computer and System Sciences, vol 78(5), 1359–1373.
[34] Sayi T J V R K M K, Krishna R K N S et al. (2012). Data Outsourcing in Cloud Environments: A Privacy Preserving Approach, 9th International Conference on Information Technology- New Generations, 361–366.
[35] Rahaman S M, and Farhatullah M (2012). PccP: A Model for Preserving Cloud Computing Privacy, International Conference on Data Science & Engineering (ICDSE), 166–170.
[36] Waqar A, Raza A et al. (2013). A framework for preservation of cloud users' data privacy using dynamic reconstruction of metadata, Journal of Network and Computer Applications, vol 36(1), 235–248.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 11, November 2014

[37] Wang C, Wang Q et al. (2010). Privacy-Preserving Public Auditing for Storage Security in Cloud Computing, Proceedings IEEE INFOCOM'10.
[38] Wang C, Chow S S M et al. (2013). Privacy-Preserving Public Auditing for Secure Cloud Storage, IEEE Transactions on Computers, vol 62(2), 362–375.
[39] Wang B, Li B et al. (2012). Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud, IEEE Fifth International Conference on Cloud Computing, 295–302.
[40] D. New and M. Rose, Reliable Delivery for Syslog,Request for Comment RFC 3195, Internet Engineering TaskForce, Network Working Group, Nov. 2001.
[41] M. Bell are and B. S. Yee, ―Forward integrity for secureaudit logs,‖ Dept. Computer. Sci., Univ. California, SanDiego, Tech. Rep., Nov. 1997.
[42] J. E. Holt, ―Logcrypt: Forward security and public verification for secure audit logs,‖ in Proc. 4th AustralasianInform. Security Workshop, 2006, pp. 203 – 211.
[43] B. Schneier and J. Kelsey, ―Security audit logs to support computer forensics,‖ ACM Trans. Inform. Syst.Security, vol. 2, no. 2, pp. 159 – 176, May 1999.
[44] D. Ma and G. Tsudik, ―A new approach to securelogging,‖ ACM Trans .Storage, vol. 5, no. 1, pp. 2:1 – 2:21,Mar. 2009
[45] D. Dolev and A. Yao, ―On the security of public key protocols,‖ IEEE Trans. Inform. Theory, vol. 29, no. 2, pp.198 –208, Mar. 1983.