International Journal on Applications in Information and Communication Engineering
Volume 5 : Issue 1: May 2019, pp 11 – 17 www.aetsjournal.com ISSN (Online) : 2394 - 6237

PUBLIC INTEGRITY VERIFICATION SCHEME AND USER LEVEL INTEGRITY IN DYNAMIC CLOUD ENVIRONMENT

M.MATHUMITHA , A.UMAMAHESWARI

Abstract— Cloud Computing has been envisioned as the next generation architecture of IT Enterprise. In contrast to traditional solutions, where the IT services are under proper physical, logical and personnel controls, Cloud Computing moves the application software and databases to the large data centers, where the management of the data and services may not be fully trustworthy. With cloud computing and storage, users are able to access and to share resources offered by cloud service providers at a lower marginal cost. With cloud computing and storage services, data is not only stored in the cloud, but routinely shared among a large number of users in a group. In this project, propose OPoR, a privacy-preserving auditing scheme for shared data with large groups in the cloud and utilize with signature to compute verification information on shared data, so that the TPA is able to audit the correctness of shared data, but cannot reveal the identity of the signer on each block. Finally proposed auditing scheme is to perform efficient public auditing to protect both identity and data privacy in cloud environments.

I. INTRODUCTION

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. It is a practical approach to experience direct cost benefits and it has the potential to transform a data center from a capital-intensive set up to a variable priced environment. The idea of cloud computing is based on a very fundamental principles of reusability of IT capabilities. The difference that cloud computing brings compared to traditional concepts of "grid computing", "distributed computing", "utility computing", or "autonomic computing" is to broaden horizons across organizational boundaries. Forrester [1] defines cloud computing as: "A pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end customer applications and billed by consumption". It is a technology that uses the internet and central remote servers to maintain data and applications and allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows for much more efficient computing by centralizing data storage, processing and bandwidth. Cloud computing examples are Yahoo email, Gmail, or Hotmail.

Cloud computing is known as distributed computing, 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. 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 are being 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-demand manner brings appealing benefits: relief of the burden for storage management, While cloud computing makes these advantages more appealing than ever, it also brings new and challenging security threats toward 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 is 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. Second, there do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. In short, although outsourcing data to the cloud is economically attractive for long-term large-scale storage, it does not

M.MATHUMITHA , Department of Computer Science and Engineering , Mahendra Engineering College , Salem.
(Email ID : mathukriya@gmail.com)
A.UMAMAHESWARI , M.E., Assistant Professor , Department of Computer Science and Engineering , Mahendra Engineering College , Salem. (Email ID : umamaheswari.a@mahendra.info)

immediately offer any guarantee on data integrity and availability.

To address these problems, our work utilizes the technique of public auditing, which enables TPA (Third party Auditor) to perform the auditing without demanding the local copy of data and thus drastically reduces the communication and computation overhead as compared to the straightforward data auditing approaches. By integrating the Merkle Hash Tree with signature, protocol guarantees that the TPA could not learn any knowledge about the data content stored in the cloud server (CS) during the efficient auditing process. The aggregation and algebraic properties of the authenticator further benefit our design for the batch auditing.

OBJECTIVES

Objective of study is to propose a software architecture to enable the use of cloud computing in applications with strict security and privacy requirements. This architecture considers how the components of an application can be integrated so that the privacy and security of user data are guaranteed. The sensitive part of the processing is therefore isolated and the architecture considers different strategies for aggregating sensitive data in environments where there are no guarantees of full confidentiality.

II. REVIEW OF LITERATURE SURVEY

1)" Lightweight and Privacy-Preserving Delegatable Proofs of Storage with Data Dynamics in Cloud Storage "

Cloud storage has been in widespread use nowadays, which alleviates users' burden of local data storage. Meanwhile, how to ensure the security and integrity of the outsourced data stored in a cloud storage server has also attracted enormous attention from researchers. Proofs of storage (POS) is the main technique introduced to address this problem. Publicly verifiable POS allowing a third party to verify the data integrity on behalf of the data owner significantly improves the scalability of cloud service. However, most of existing publicly verifiable POS schemes are extremely slow to compute authentication tags for all data blocks due to many expensive group exponentiation operations, even much slower than typical network uploading speed, and thus it becomes the bottleneck of the setup phase of the POS scheme. In this article, we propose a new variant formulation called "Delegatable Proofs of Storage (DPOS)". Then, we construct a lightweight privacy-preserving DPOS scheme, which on one side is as efficient as private POS schemes, and on the other side can support third party auditor and can switch auditors at any time, close to the functionalities of publicly verifiable POS schemes. Compared to traditional publicly verifiable POS schemes, we speed up the tag generation process by at least several hundred times, without sacrificing efficiency in any other aspect. In addition, we extend our scheme to support fully dynamic operations with high efficiency, reducing the computation of any data update to$O(\log n)$and simultaneously only requiring constant communication costs. We prove that our scheme is sound and privacy preserving against auditor in the standard model. Experimental results verify the efficient performance of our scheme.

2)" Secure and Efficient Attribute-Based Access Control for Multiauthority Cloud Storage "

Cloud storage facilitates both individuals and enterprises to cost effectively share their data over the Internet. However, this also brings difficult challenges to the access control of shared data since few cloud servers can be fully trusted. Ciphertext-policy attribute-based encryption (CP-ABE) is a promising approach that enables the data owners themselves to place fine-grained and cryptographically-enforced access control over outsourced data. In this paper, we present secure and cost-effective attribute-based data access control for cloud storage systems. Specifically, we construct a multiauthority CP-ABE scheme that features: 1) the system does not need a fully trusted central authority, and all attribute authorities independently issue secret keys for users; 2) each attribute authority can dynamically remove any user from its domain such that those revoked users cannot access subsequently outsourced data; 3) cloud servers can update the encrypted data from the current time period to the next one such that the revoked users cannot access those previously available data; and 4) the update of secret keys and ciphertext is performed in a public way. We show the merits of our scheme by comparing it with the related works, and further implement it to demonstrate its practicality. In addition, the proposed scheme is proven secure in the random oracle model.

3)" SeSPHR: A Methodology for Secure Sharing of Personal Health Records in the Cloud"

The widespread acceptance of cloud based services in healthcare sector has resulted in cost effective and convenient exchange of Personal Health Records (PHRs) among several participating entities of the e-Health systems. Nevertheless, storing confidential health

information to cloud servers is susceptible to revelation or theft and calls for the development of methodologies that ensure the privacy of the PHRs. Therefore, we propose a methodology called SeSPHR for secure sharing of the PHRs in the cloud. The SeSPHR scheme ensures patient-centric control on the PHRs and preserves the confidentiality of the PHRs. The patients store the encrypted PHRs on the un-trusted cloud servers and selectively grant access to different types of users on different portions of the PHRs. A semi-trusted proxy called Setup and Re-encryption Server (SRS) is introduced to set up the public/private key pairs and to produce the re-encryption keys. Moreover, the methodology is secure against insider threats and also enforces a forward and backward access control. Furthermore, we formally analyze and verify the working of SeSPHR methodology through High Level Petri Nets (HLPN). Performance evaluation with regard to time consumption indicates that the SeSPHR methodology has potential to be employed for securely sharing the PHRs in the cloud.

4)" Privacy-Preserving Smart Semantic Search Based on Conceptual Graphs Over Encrypted Outsourced Data "

The Searchable encryption is an important research area in cloud computing. However, most existing efficient and reliable ciphertext search schemes are based on keywords or shallow semantic parsing, which are not smart enough to meet with users' search intention. Therefore, in this paper, we propose a content-aware search scheme, which can make semantic search more smart. First, we introduce conceptual graphs (CGs) as a knowledge representation tool. Then, we present our two schemes (PRSCG and PRSCG-TF) based on CGs according to different scenarios. In order to conduct numerical calculation, we transfer original CGs into their linear form with some modification and map them to numerical vectors. Second, we employ the technology of multi-keyword ranked search over encrypted cloud data as the basis against two threat models and raise PRSCG and PRSCG-TF to resolve the problem of privacy-preserving smart semantic search based on CGs. Finally, we choose a real-world data set: CNN data set to test our scheme. We also analyze the privacy and efficiency of proposed schemes in detail. The experiment results show that our proposed schemes are efficient.

5) " Strong Key-Exposure Resilient Auditing for Secure Cloud Storage "

The Key exposure is one serious security problem for cloud storage auditing. In order to deal with this problem, cloud storage auditing scheme with key-exposure resilience has been proposed. However, in such a scheme, the malicious cloud might still forge valid authenticators later than the key-exposure time period if it obtains the current secret key of data owner. In this paper, we innovatively propose a paradigm named strong key-exposure resilient auditing for secure cloud storage, in which the security of cloud storage auditing not only earlier than but also later than the key exposure can be preserved. We formalize the definition and the security model of this new kind of cloud storage auditing and design a concrete scheme. In our proposed scheme, the key exposure in one time period doesn't affect the security of cloud storage auditing in other time periods. The rigorous security proof and the experimental results demonstrate that our proposed scheme achieves desirable security and efficiency.

III. PROPOSED SYSTEM

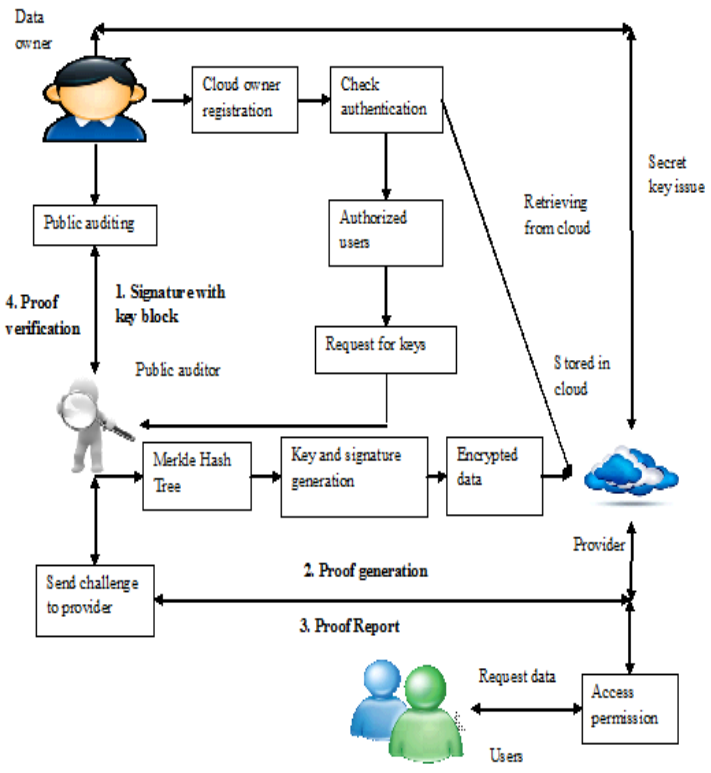
The system model in this project involves three parties: the cloud server, a group of users and a public verifier. There are two types of users in a group: the original user and a number of group users. The original user initially creates shared data in the cloud, and shares it with group users. Both the original user and group users are members of the group. Every member of the group is allowed to access and modify shared data. Shared data and its verification metadata (i.e. signatures) are both stored in the cloud server. A public verifier, such as a third-party auditor (TPA) providing expert data auditing services or a data user outside the group intending to utilize shared data, is able to publicly verify the integrity of shared data stored in the cloud server. When a public verifier wishes to check the integrity of shared data, it first sends an auditing challenge to the cloud server. After receiving the auditing challenge, the cloud server responds to the public verifier with an auditing proof of the possession of shared data. Then, this public verifier checks the correctness of the entire data by verifying the correctness of the auditing proof. Essentially, the process of public auditing is a challenge and-response protocol between a public verifier and the cloud server.

ADVANTAGES

- Introduced the first approach to sharing patient data while providing computational privacy guarantees.
- The proposed scheme is proved secure against reset attacks.

- Supporting efficient public verifiability and dynamic data operations simultaneously.
- The approach uses sequence alignment and clustering-based heuristics to anonymize longitudinal patient records.
- The investigations suggest that it can generate longitudinal data with a low level of information loss and remain useful for biomedical analysis.
- The approach is not guided by specific utility (e.g., satisfaction of GWAS validation), but it can be extended to support with confident.
- The new system checks whether the database inserted with the tuple is still k-anonymous, without letting two different users know the contents of the tuple and the database, respectively.
- The two protocols solve this problem on suppression-based and generalization-based k-anonymous and confidential databases.
- The protocols rely on well-known cryptographic assumptions.
- The new system provides theoretical analyses to proof their soundness and experimental results to illustrate their efficiency.

IV. BLOCK DIAGRAM



1) SYSTEM ARCHITECTURE

The system architecture of the proposed system, in which we aim to solve the security and privacy issue in data integration and aggregation. Also resource allocation problem is studied for efficient integration of dynamic cloud server.

2) ALGORITHM AND TECHNIQUES:

Existing public auditing mechanisms can actually be extended to verify shared data integrity and data freshness. However, a new significant privacy issue introduced in the case of shared data with the use of existing mechanisms is the leakage of identity privacy to public verifiers.. To solve the above privacy issue on shared data, propose OPOR, a novel privacy preserving public auditing mechanism. More specifically, utilize hash signatures to construct homomorphic authenticators in OPoR, 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, extend this mechanism to support batch auditing, which can perform multiple auditing tasks simultaneously and improve the efficiency of verification for multiple auditing tasks.

Meanwhile, OPoR is compatible with random masking; In this project, to prove the data freshness (prove the cloud possesses the latest version of shared data) while still preserving identity privacy. And ensures that retrieved data always reflects the most recent updates and prevents rollback attacks. In this project, the TPA will be fully automated and will be able to properly monitor confidentiality and integrity of the data and uniquely integrate it with random mask technique to achieve a privacy-preserving public auditing system for cloud data storage security while keeping all above requirements in mind. Extensive security and performance analysis shows the proposed schemes are provably secure and highly efficient. We also show how to extent our main scheme to support batch auditing for TPA upon delegations from multi-users.

3) Merkle Hash Tree (MHT)

To achieve privacy-preserving public auditing, propose to uniquely integrate the linear authenticator with binary tree technique. In our protocol, the linear combination of sampled blocks in the server's response is masked with randomness 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 correctness 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 MHT, to equip the auditing protocol with public audit ability. A MHT Encryption scheme is comprised of a tuple of algorithms (Gen, E,D, Eval), and is defined with respect to a circuit C with t inputs. Though a MHT scheme can be either a public-key or symmetric-key system, we will define it as a public-key system here. The key generation algorithm Gen takes the security parameter 1^k as input, and outputs the public key and private key for the system (Notation: (pk, sk) ← Gen(1^k)).

- Assume that messages M ∈ $\{0, 1\}^{l(k)}$.
- The encryption algorithm E takes a public key and a message as input, and outputs a ciphertext C, (Notation: C ← E(pk,M) for M ∈ $\{0, 1\}^{l(k)}$).
- The decryption algorithm D takes a secret key and a ciphertext, and returns a message, (Notation: M ← D(sk,C) and M ∈ $\{0, 1\}^{l}$).
- Finally, the evaluation algorithm Eval takes as input a public key, a description of a t-input circuit C, and t ciphertexts $C_1, \ldots, C_t$ such that Ci← E(pk,Mi), and produces as output C*, (Notation: C* ← Eval(pk,C, $C_1, \ldots, C_t$)).
- We add a new correctness property to the standard correctness requirement for an encryption scheme as follows. We say that an encryption scheme is homomorphic with respect to a t-input circuit C if ∀k, ∀M1, . . . ,Mt, Pr[(pk, sk) ← Gen(1k); C1, . . . Ct ← E(pk,M1), . . . , E(pk,Mt); C* ← Eval(pk,C, $C_1, \ldots, C_t$) : D(sk,C*) = C($M_1, \ldots, M_t$)] = 1.

Similarly, a scheme with respect to a family of circuits $\{C_i\}$ if the correctness property holds for any circuit C ∈ $\{C_i\}$. Note that so far, our definition makes no requirement that the output C* of Eval should look like a standard ciphertext. Indeed, without some additional restriction on C*, every standard encryption scheme (Gen, E,D) can be trivially modified to yield a homomorphic encryption scheme (Gen', E',D', Eval') with respect to all circuits as follows.

✓Gen' runs as Gen.

✓E' runs as E.

✓The Eval' is constructed to take a public key, a circuit description, and up to t ciphertexts, and then output the circuit description concatenated with each of the ciphertexts, as C* ← Eval"(pk,C, C1, . . . ,Ct) = C|C1| . . . |Ct, with | used to denote concatenation.

✓On special cipher texts C* containing a circuit description, D' parses its input into C, $C_1, \ldots, C_t$, runs the original decryption algorithm D on the ciphertexts to obtain messages Mi ← D(sk,Ci), and runs the circuit C on these messages, to obtain D'(sk,C*) = C($M_1, \ldots, M_t$), satisfying the homomorphic correctness property. On ciphertexts without circuit descriptions, D'(sk,C) simply returns D(sk,C).

4) OPOR (Public Auditing)

With the establishment of privacy-preserving public auditing, the TPA may concurrently handle multiple auditing upon different users' delegation. The individual auditing of these tasks for the TPA can be tedious and very inefficient. Given K auditing delegations on K distinct data files from K different users, it is more advantageous for the TPA to batch these multiple tasks together and audit at one time. Keeping this natural demand in mind, we slightly modify the protocol in a single user case, and achieve the aggregation of K verification equations (for K auditing tasks) into a single one. As a result, a secure batch auditing protocol for simultaneous auditing of multiple tasks is obtained.

1. Verify file tag tk for each user k, and quit if fail
   For each user k ($1 \leq k \leq K$)
2. Generate a random challenge
3. Compute $\mu k, \sigma k, Rk$ as single user case;
   Chal = {(I, Vi)} i$\in$ I
4. Compute R=R1, R2,….Rk
   L = vk1||vk2||…..||vk_k
5. Compute $\mu k = rk + \gamma k \mu' k \bmod p$
6. Compute $\gamma k = h(R||Vk||L)$ for each user k and do batch auditing

5) Cloud Framework

In this module, cloud data storage service three different entities such as the cloud user, who has large amount of data files to be stored in the cloud; the cloud server, which is managed by the cloud service provider to provide data storage service and has significant storage space and computation resources; the third-party auditor, 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.

6) Key management

Merkle hash tree contains three algorithms: KeyGen, Sign and Verify. In KeyGen, each user in the group generates his/her public key and private key. In Sign, a

user in the group is able to generate a signature on a block and its block identifier with his/her private key and all the group members' public keys. A block identifier is a string that can distinguish the corresponding block from others. A verifier is able to check whether a given block is signed by a group member in Ring Verify.

MHT Encryption systems are used to perform operations on encrypted data without knowing the private key (without decryption), the client is the only holder of the secret key. When decrypt the result of any operation, it is the same as if had carried out the calculation on the raw data. The protocol provides public verifiability without the help of a third party auditor. It doesn't leak any privacy information to third party, which provides good performance without the support of the trusted third party and provides a method for independent arbitration of data retention contracts. But it gives unnecessary computation and communication costIn this protocol, the linear combination of sampled blocks in the server's response is masked with randomness generated by the server.

7) Data integrity analysis

In this module, TPA checks the correctness of data storage to ensure that there exists no cheating cloud server that can pass the TPA's audit without indeed storing users' data intact to ensure that the TPA cannot derive users' data content from the information collected during the auditing process. Then implement the batch auditing scheme in OPOR to enable TPA with secure and efficient auditing capability to cope with multiple auditing delegations from possibly large number of different users simultaneously. Auditor monitors the manipulations between the data owner and cloud service provider, receives the meta information of the data component, tag generation key and random challenge from the data owner, now by making a request to the cloud server auditor gets the meta information of the data component, before processing the request checks for authentication the and checks with the meta information which is received from the data owner. Data owner hosts the data over cloud servers.

8) Dynamic auditing

To fully ensure the data integrity and save the cloud users' computation resources as well as online burden, it is of critical importance to enable public auditing service for cloud data storage, so that users may resort to an independent third-party auditor (TPA) to audit the outsourced data when needed**.** In this module, 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. Through the organization of privacy-preserving public auditing in Cloud Computing, TPA may concurrently handle multiple auditing delegations upon different user requests. The individual auditing of these tasks for TPA can be and very difficult and inefficient. Batch auditing not only allows TPA to perform the multiple auditing tasks at the same time, but also greatly reduces the computation cost on the TPA side.

This scheme uses n number verification steps to help the auditor to saves a considerable amount of auditing time. Data dynamics supports to replace information index in computational block with best data structure and supporting data dynamics for privacy-preserving public risk auditing is also of supreme importance.

9) Secure Data sharing

Each user is assigned to data owner from the Provider. Each user can freely get the cipher texts from the server. To decrypt a cipher text, each user may submit their secret keys issued by data owner together with its global public key to the server and ask it to generate decryption token for some cipher text. Upon receiving the decryption token, the user can decrypt the cipher text by using its global secret key. The users those who are having matching keys as in the access policy defined in the cipher text can retrieve the entire data content. It aims to allow the users with eligible attributes to decrypt the entire data stored in the cloud server. However it cannot limit the users from accessing the data's which are not accessible to them. That is it cannot limit the data access control to the authorized users.

V.CONCLUSION

Cloud computing securities are discussed and analyzed in previous study. In this project, 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. Data freshness is essential to protect against misconfiguration errors or rollbacks caused intentionally and can develop an authenticated file system that supports the migration of an enterprise-class distributed file system into the cloud efficiently,

transparently and in a scalable manner. It's authenticated in the sense that enables an enterprise tenant to verify the freshness of retrieved data while performing the file system operations. The user must be given complete access control over the published data. Also, powerful security mechanisms must always supplement every cloud application. Attaining all these would end up in achieving the long dreamt vision of secured Cloud Computing in the nearest future.

FUTURE WORK

In future, this proposed model could be used to get the secure cloud computing environment which would be a great enhancement in the privacy preservation. Then overcome user revocation problem using public key updation algorithm with dynamic group management.

References

[1] Chang E.C and Xu, Forrester,. J, "Remote integrity check with dishonest storage server," in Proceedings of ESORICS 2008, volume 5283 of LNCS. Springer-Verlag, 2008, pp. 223–237.

[2] Jiawei Yuan and Shucheng Yu "Public Integrity Auditing for Dynamic Data Sharing With Multiuser Modification" IEEE Trans. Information Forensic and security. Syst., vol. 10, no.8, 2015

[3] Jin Li, Xiao Tan, Xiaofeng Chen, Duncan S. Wong, Fatos Xhafa, "OPoR: Enabling Proof of Retrievability in Cloud Computing with Resource-Constrained Devices" IEEE TRANSACTIONS ON CLOUD COMPUTING, VOL. 3, NO. 2, APRIL-JUNE 2015

[4] Li.J and Kim.K, "Hidden attribute-based signatures without anonymity revocation" Information Sciences, vol. 180, no. 9, pp. 1681–1689, 2010

[5] Shah.M.A, Swaminathan.R, and Baker.M, "Privacy-preserving audit and extraction of digital contents," Cryptology ePrint Archive, Report 2008/186, 2008, http://eprint.iacr.org/.

[6] Tao Jiang, Xiaofeng Chen, and Jianfeng Ma, "Public Integrity Auditing for Shared Dynamic Cloud Data with Group User Revocation**"** IEEE Trans. Parallel Distrib. Syst., vol. PP, no. 99, 2015

[7] Wang. C, Wang. Q, Ren. K, and Lou. W, "Privacy-preserving public auditing for data storage security in cloud computing," in INFOCOM, 2010, pp. 525–533

[8] Zheng .Q and Xu. S, "Fair and dynamic proofs of retrievability," in CODASPY, 2011, pp. 237–248.

[9] Zhu.Y, Wang.H, Hu .Z, Ahn. G.J., Hu.H, and Yau. S. S., "Dynamic audit services for integrity verification of outsourced storages in clouds," in SAC, 2011, pp. 1550–1557.

[10] Zhu. Y, Hu.H, Ahn G.J, and Yu. M., "Cooperative provable data possession for integrity verification in multicloud storage," IEEE Trans. Parallel Distrib. Syst., vol. 23, no. 12, pp. 2231–2244, 2012.