RESEARCH ARTICLE OPEN ACCESS

Light Weight Cloud Storage Auditing with Deduplication Supporting Strong Privacy Protection

Mr D Purushothaman MCA., M.E.,[1], M Naveen [2]

[1] Asst. Professor, Department of Computer Applications
[2] Student, Department of Computer Applications
[1], [2] Chadalawada Ramanamma Engineering College (Autonomous)

ABSTRACT

A cloud platform provides users with shared data storage services. To ensure shared data integrity, it is necessary to validate the data effectively. An audit scheme that enables group members to modify data conducts the integrity verification of the shared data, but this approach results in complex calculations for the group members. The audit scheme of the designated agent implements a lightweight calculation for the group members, but it ignores the security risks between the group members and the agents. By introducing Hashgraph technology and designing a Third Party Medium (TPM) management strategy, a lightweight secure auditing scheme for shared data in cloud storage (LSSA) is proposed, which achieves security management of the groups and a lightweight calculation for the group members. Meanwhile, a virtual TPM pool is constructed by combining the TCP sliding window technology and interconnected functions to improve agent security. We evaluate our scheme in numerical analysis and in experiments, the results of which demonstrate that our scheme achieves lightweight computing for the group members and ensures the data verification process for security.

Keywords: Shared data, virtual TPM pool, lightweight calculation, agent security

I. INTRODUCTION

The most recent utility oriented distributed computing model that has envisioned an immense transformation of Information Technology (IT), to increase capacities of the client access to a common pool of platforms, applications and infrastructures without having to really claim them in distributed computing. In the context of deployment, the cloud computing is grouped into four approaches: (i) public, (ii) private, (iii) hybrid and (iv) community clouds that are described below:

Public Cloud: In public cloud, the service suppliers transfer various applications as service and encourage the customers by offering access to the resources by means of concentrated distributed servers over the Internet for example, Amazon Web Services, Google App Engine.

Private Cloud: The services and framework are utilized and supervised absolutely by a performance institution.

Community Cloud: The services and framework are distributed by an arrangement of institutions that are overseen either privately or by a dependable outsider.

Hybrid Cloud: Hybrid cloud adopts a blend of on-premises, private cloud and third-party public cloud services with arrangement among the two platforms.

Liu and his colleagues [1] discusses about the cloud computing reference architecture and taxonomy of three service models i.e., PaaS, SaaS, IaaS. Fox and his colleagues [2] examine the impediments to and opportunities for selection and development of distributed computing and classes of utility computing. Buyya and his colleagues [3] proposed framework for market-oriented distribution of assets inside the clouds. It provides the attributes of cluster, grid and clouds and awareness on market-based assets administration procedures.

With the rapid development of cloud computing, cloud storage has been widely accepted by individuals and enterprises for its advantages of universal access, low costs and on-demand service. Users can outsource complex computations to the cloud to reduce their computational burden. In addition, users also can outsource their large-scale data to the cloud to release their local storage burden.

Under such a trend, it becomes urgent to guarantee the quality of data storage services for the users and the cloud. On one hand, the outsourced data might be corrupted or lost due to the inevitable operation errors or soft- ware/hardware failures in the cloud . Thus, it is critical to develop cloud storage auditing, by which users can verify the integrity of cloud data without downloading the whole data from the cloud. On the other hand, lots of data stored in the cloud are duplicated. Based on the survey by EMC, 75% of cloud data are duplicated copies. In order to improve the storage efficiency of the cloud, it is necessary to perform data de duplication, where the cloud keeps only a single copy of the duplicated le and makes a link to the le for the users. Users usually encrypt their data before outsourcing them to the cloud since they would not like to disclose their sensitive data to the cloud and other parties . In order to realize de duplication over encrypted data, the convergent encryption (CE) [6] was proposed to encrypt data. A convergent encryption algorithm encrypts data with a key deterministically derived from the data (e.g., the le's hash value). Thus, the same le will produce the same cipher text. It means that the de duplication over cipher texts is feasible. However, directly using CE is not secure in some situations. For example, when the le is predictable or from a small space, CE cannot resist brute-force dictionary attacks, in which the malicious cloud can recover the entire le with a number of guesses. In order to deal with this problem, Li proposed a secure auditing and data de duplication scheme by introducing a key server to help user generate the convergent key. In this scheme, the cloud cannot deduce or derive the convergent key from the content of le since a secret ``seed'' is embedded in the convergent key.

It will result in the data privacy leakage because the malicious cloud or other parties might guess or derive the content of le by performing the brute-force dictionary attacks. Thus, how to realize de duplication supporting strong privacy protection in cloud storage auditing is very important and valuable. Unfortunately, previous schemes are weak in privacy protection because they cannot fully defend against the brute-force dictionary attacks. Our main contributions can be summarized as below: In this paper, we investigate how to fully resist the brute-force dictionary attacks and realize de duplication with strong privacy protection in cloud storage auditing, and propose a concrete scheme satisfying this property. In order to realize de duplication with strong privacy protection, we design a novel method to generate the le index, and employ a new strategy to generate the key for le encryption. In the detailed design, the le index is generated with the help of an Agency Server (AS) instead of directly being produced by the hash value of le. The key for le encryption is generated with the le and the le label. The le label is kept by the user secretly. In this way, the privacy of the user's le is protected against the cloud and the AS. In order to improve the storage efficiency, the users, who own the same le, are able to generate the same cipher text and the same authenticators. The proposed scheme effectively achieves data deduplication and authenticator de duplication. Furthermore, to reduce the computation burden on the user side, the user only needs to perform lightweight computation to generate data authenticators, verify the integrity of the cloud data, and retrieve his le from the cloud. We give the security analysis of the proposed scheme, showing that the proposed scheme satisfies correctness, soundness and strong privacy protection. We also justify the performance by concrete implementations. The result shows that the proposed scheme is efficient.

II. RELATED WORK

Provable Data Possession at Untrusted Stores.
Introduce a model for provable data possession (PDP) that allows a client that has stored data at an untrusted server to verify that the server possesses the original data without retrieving it. The model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. We present two provably-secure PDP schemes that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In

particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. Experiments using our implementation verify the practicality of PDP and reveal that the performance of PDP is bounded by disk I/O and not by cryptographic computation.

Message-Locked Encryption and Secure Deduplication

We formalize a new cryptographic primitive, Message-Locked Encryption (MLE), where the key under which encryption and decryption are performed is itself derived from the message. MLE provides a way to achieve secure deduplication (space-efficient secure outsourced storage), a goal currently targeted by numerous cloud-storage providers. We provide definitions both for privacy and for a form of integrity that we call tag consistency. Based on this foundation, we make both practical and theoretical contributions. On the practical side, we provide ROM security analyses of a natural family of MLE schemes that includes deployed schemes. On the theoretical side the challenge is standard model solutions, and we make connections with deterministic encryption, hash functions secure on correlated inputs and the sample-then-extract paradigm to deliver schemes under different assumptions and for different classes of message sources. Our work shows that MLE is a primitive of both practical and theoretical interest.

Attribute-Based Storage Supporting Secure Deduplication of Encrypted Data in Cloud

Attribute-based encryption (ABE) has been widely used in cloud computing where a data provider outsources his/her encrypted data to a cloud service provider, and can share the data with users possessing specific credentials (or attributes). However, the standard ABE system does not support secure deduplication, which is crucial for eliminating duplicate copies of identical data in order to save storage space and network bandwidth. In this paper, we present an attribute-based storage system with secure deduplication in a hybrid cloud setting, where a private cloud is responsible for duplicate detection and a public cloud manages the storage. Compared with the prior data deduplication systems, our system has two advantages. First, it can be used to confidentially share data with users by specifying access policies rather than sharing decryption keys. Second, it achieves the standard notion of semantic security for data confidentiality while existing systems only achieve it by defining a weaker security notion. In addition, we put forth a methodology to modify a ciphertext over one access policy into ciphertexts of the same plaintext but under other access policies without revealing the underlying plaintext.

III. PROBLEM DEFINITION

A malicious cloud server is able to discard all the shared data and generate a valid proof of data possession by reserving some intermediate results or a previous valid proof, which we refer to as a replace attack and a replay attack, respectively. A malicious group member is able to modify other member's data in that group without being discovered. A malicious agent is able to collude with illegal group members to steal user data and identity information. As far as we know, the three points mentioned above are still open challenges to design a secure integrity auditing scheme for shared data with lightweight computing on the client side.

IV. IMPLEMENTATION SCENARIO

In this project, we investigate how to fully resist the bruteforce dictionary attacks and realize deduplication with strong privacy protection in cloud storage auditing, and propose a concrete scheme satisfying this property. In order to realize deduplication with strong privacy protection, we design a novel method to generate the file index, and employ a new strategy to generate the key for file encryption. In the detailed design, the file index is generated with the help of an Agency Server (AS) instead of directly being produced by the hash value of file. The key for file encryption is generated with the file and the file label. The file label is kept by the user secretly. In this way, the privacy of the user's file is protected against the cloud and the AS. In order to improve the storage efficiency, the users, who own the same file, are able to generate the same ciphertext and the same authenticators. The proposed scheme effectively achieves data deduplication and authenticator deduplication. Furthermore, to reduce the computation burden on the user side, the user

only needs to perform lightweight computation to generate data authenticators, verify the integrity of the cloud data, and retrieve his file from the cloud. .

Advantages

- Satisfies correctness
- Strong privacy protection
- Efficient

V. IMPLEMENTATION MODEL

The system model of this scheme consists of four different entities: the Group members (M), the Cloud, the Group Manager (GM), and the TPM. As shown bellow Figure 1, there are multiple group members in a group. After the data owner (the individual or organization that owns the original data) creates the data file and uploads it to the cloud, any group member can access and modify the corresponding shared data. Note that the original data owner can play the role of GM and there is only one GM in each group. TheMplay two important roles: 1) blind data, and 2) record blind data and broadcast within the group through a Hashgraph. The cloud (e.g., iCloud, OneDrive, and Baidu Cloud) provides data storage services for group members and provides a platform for group members to share data. The GM plays three important roles: 1) generate the TPM's public-private key pair, 2) for- mulate the TPM management strategy, and 3) generate the secret seed that is used to blind the data for group members and to recover the real data for the cloud. The TPM plays two important roles: 1) generate data authentication label for group members, and 2) verify the integrity of the cloud data on behalf of the group members.

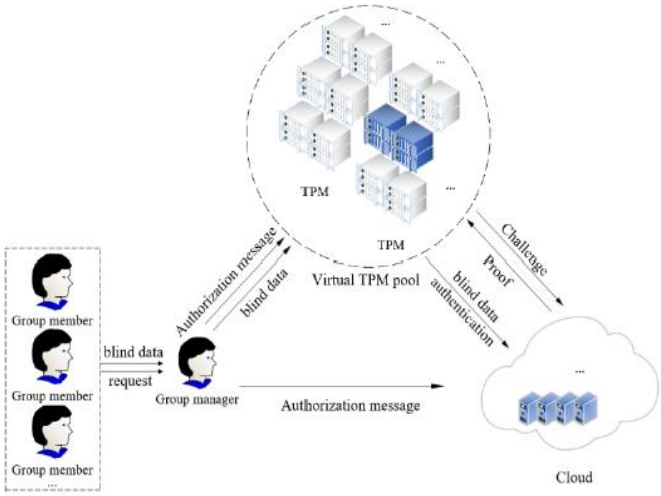

Fig1: implementation architecture model

The execution procedure is divided into the data upload stage and the audit stage. Before the group members make a request to upload the modified data to the cloud, the data are first blinded by the secret seed and recorded by the Hashgraph, and then sent to the group manager. According to the TPM management strategy, the group manager selects a TPM from the virtual TPM pool for authorization, and the authorized TPM calculates the corresponding authentication labels for these blinded data within the authorization time. Then, the blind data and authentication label are sent to the cloud. Before receiving these messages, the cloud will check whether or not the authorization from the TPM is valid at the current time. If it is, he verifies whether or not the authentication labels are correct. If they are correct, he will recover the real data and compute their authentication labels. Finally, the cloud stores these real data and authentication labels. Before executing the auditing procedure, the group manager selects a TPM and creates the authorization according to the TPM management strategy. Then, the authorized TPM sends the challenge messages to the cloud. Before receiving these messages, the cloud will check whether or not the authorization from the TPM is valid. If it is, the cloud generates a proof of possession of the shared data. Finally, the TPM can verify the integrity of shared data in the cloud by checking the correctness of the proof.

VI. IMPLEMENTATION DESIGN

Lightweight computing: This approach ensures that group members do not need to perform time-consuming calculations during the generation of authentication labels or during the audit of the shared data. Multiple TPMs take part in the calculation, thereby ensuring a lightweight calculation of a single TPM.

Identity traceability: The modification of data by illegal group members may lead to disputes among the group members using the same shared data. This goal ensures that the GM can find and remove any illegal group members, thereby achieving the security management of groups.

TPM management security: Each TPM works independently to ensure legal participation of the TPM. This goal ensures that the cloud only accepts and stores the data of TPMs that are authorized by the GM, and it only responds to the challenge of the TPMs that are authorized by the GM.

Data privacy and identity privacy: When the TPM generates authentication labels instead of group members, it is impossible to know the actual information of the data block. The TPM cannot acquire the identity information of group members at the stages of uploading data and auditing data.

Audit correctness and security: The TPM can verify the integrity of the shared data through the audit process. Malicious cloud service providers cannot complete the audit process through replace or replay attacks.

VII. RESULTS:

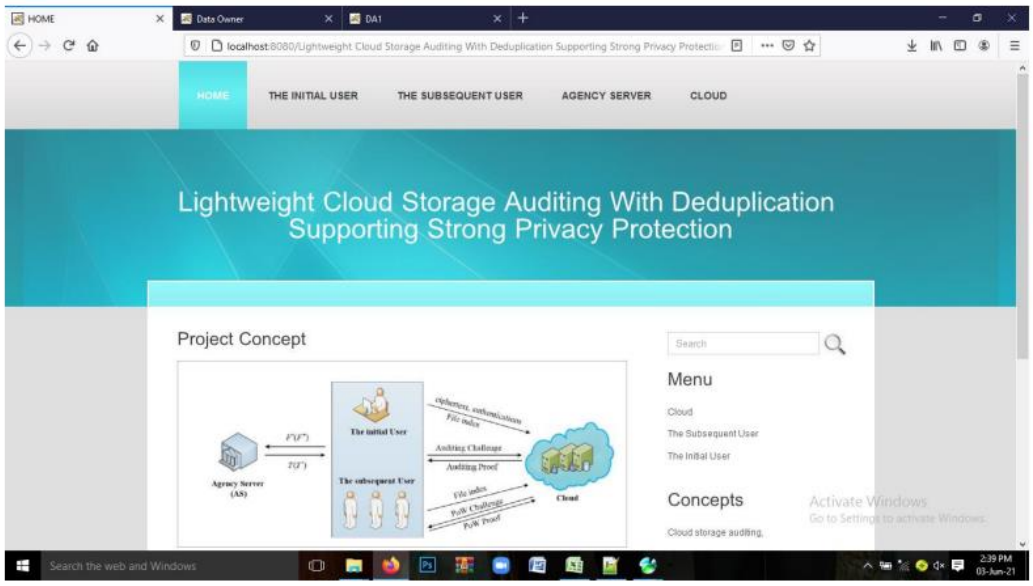

Fig 1 Home Page

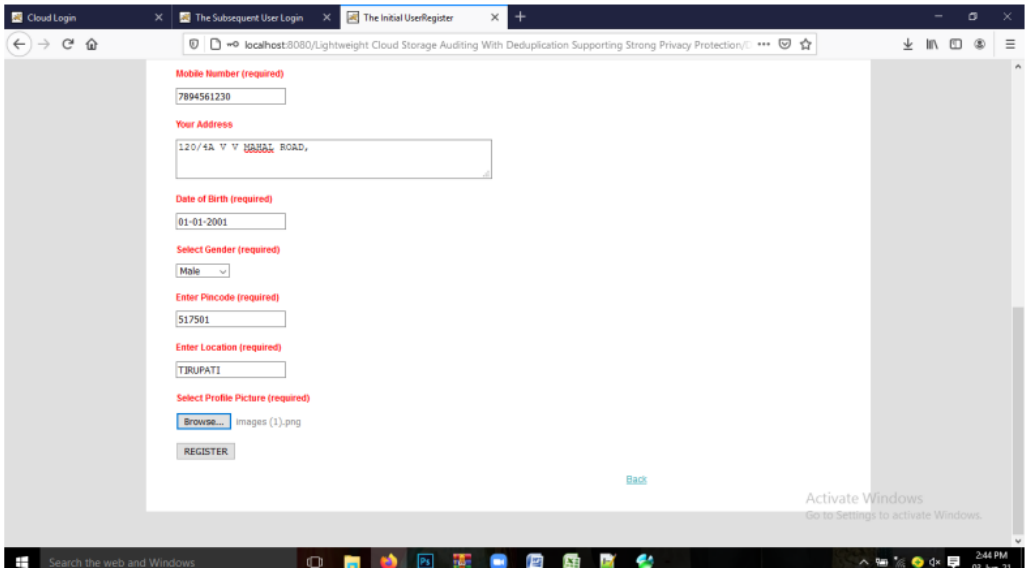

Fig 2 User Registration

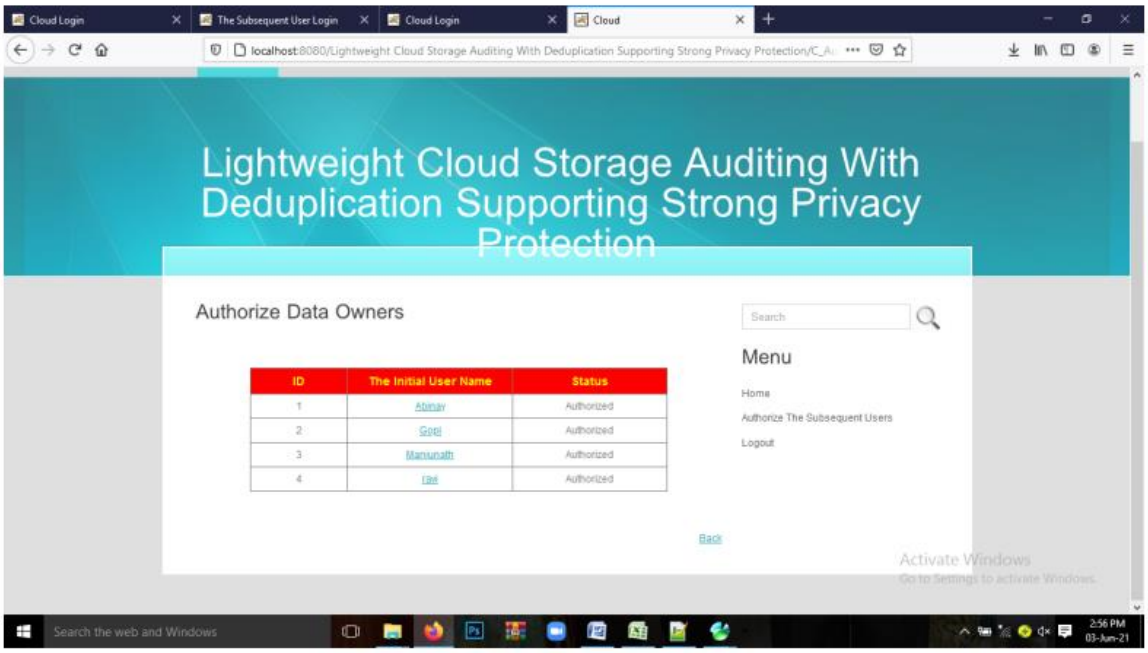

Fig 3 Authorize Data Owner

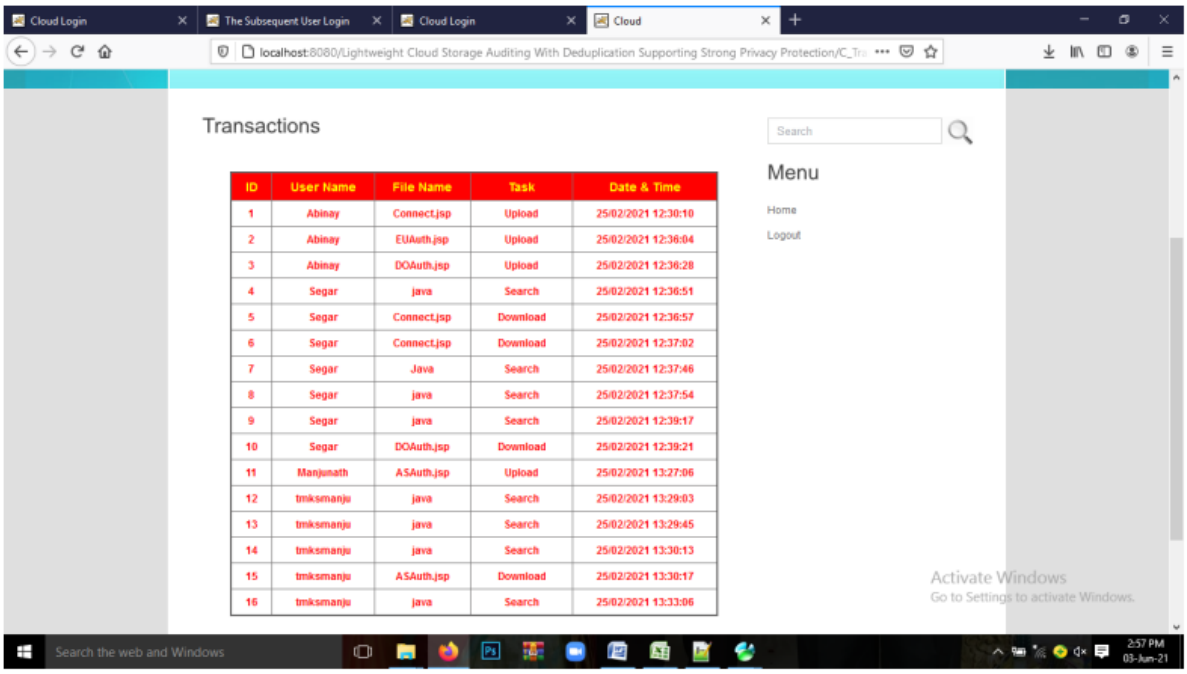

Fig 4 Transactions

VIII. CONCLUSION

In this project study on how to solve the problem of user's privacy leakage in cloud storage auditing with de duplication when brute-force dictionary attacks are launched and design a lightweight cloud storage auditing scheme with de duplication supporting strong privacy protection. In the proposed scheme, the privacy of user can be well preserved against the cloud and other parties. The user relieves the heavy computation burden for generating data authenticators and verifying data integrity. The security proof shows that the proposed scheme is secure and also provide detailed comparisons among our proposed scheme and other existing schemes by experiments. Experimental results show the proposed scheme achieves higher storage efficiency and is more efficient in authenticator generation phase and auditing phase. Provide a multi-level access policy construction with secret sharing scheme, in which each media layer is assigned a random secret that is shared by the access tree in this layer, and also the lower media layers. It ensures that the users who view the higher media layer must satisfy the access sub-trees at a lower access level.

REFERENCES

[1] M. Armbrust *et al.*, ``Above the clouds: A Berkeley view of cloud comput- ing," Dept. Elect. Eng. Comput. Sci., Univ. California, Berkeley, Berkeley, CA, USA, Tech. Rep. UCB/EECS-2009-28, 2009.

[2] P. Mell and T. Grance, ``The National Institute of Standards and Technol- ogy (NIST) denition of cloud computing," NIST, Washington, DC, USA, NIST Special Publication 800-145, 2011.

[3] K. Julisch and M. Hall, ``Security and control in the cloud," *Inf. Secur. J. Global Perspective*, vol. 19, no. 6, pp. 299_309, 2010.

[4] D. G. Feng, M. Zhang, Y. Zhang, and Z. Xu, ``Study on cloud computing security," *J. Softw.*, vol. 22, no. 1, pp. 71_83, 2011.

[5] G. Ateniese *et al.*, ``Provable data possession at untrusted stores," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 598_609.

[6] A. Juels and B. S. Kaliski, ``Pors: Proofs of retrievability for large files," in *Proc. 14th ACM Conf. Comput. Commun. Secur. (CCS)*, 2007, pp. 584_597.

[7] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, ``Scalable and efficient provable data

possession," in *Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (ICST)*, Istanbul, Turkey, 2008, pp. 22_25.

[8] H. Shacham and B. Waters, ``Compact proofs of retrievability," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur*. Berlin, Germany: Springer, 2008, pp. 90_107.

[9] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, ``Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," *J. Syst. Softw.*, vol. 113, pp. 130_139, Mar. 2016.

[10] T. Jiang, X. Chen, and J. Ma, ``Public integrity auditing for shared dynamic cloud data with group user revocation," *IEEE Trans. Comput.*, vol. 65, no. 8, pp. 2363_2373, Aug. 2016.

[11] Y. Luo, M. Xu, K. Huang, D. Wang, and S. Fu, ``Efficient auditing for shared data in the cloud with secure user revocation and computations outsourcing," *Comput. Secur.*, vol. 73, pp. 492_506, Mar. 2018.

[12] L. Huang, G. Zhang, and A. Fu, ``Privacy-preserving public audit- ing for dynamic group based on hierarchical tree," *J. Comput. Res. Develop.*, vol. 53, no. 10, pp. 2334_2342, 2016.

[13] L. X. Huang, G. M. Zhang, and A. M. Fu, ``Certi_cateless public veri_ca- tion scheme with privacy-preserving and message recovery for dynamic group," in *Proc. Australas. Comput. Sci. Week Multiconf.*, Melbourne, VIC, Australia, 2017, p. 76.

[14] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang, ``NPP: A new privacy-aware public auditing scheme for cloud data sharing with group users," *IEEE Trans. Big Data*, to be published.

[15] J. Li, X. Tan, X. Chen, D. S. Wong, and F. Xhafa, ``OPoR: Enabling proof of retrievability in cloud computing with resource-constrained devices," *IEEE Trans. Cloud Comput.*, vol. 3, no. 2, pp. 195_205, Apr. 2015.

[16] C.W. Guan, K. Ren, F. Zhang, F. Kerschbaum, and J. Yu, ``Symmetric-key based proofs of retrievability supporting public veri_cation," in *Proc. 20th Eur. Symp. Comput. Secur*. Berlin, Germany: Springer-Verlag, 2015.

[17] H. Wang, D. He, and S. Tang, ``Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1165_1176, Jun. 2016.

[18] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, ``Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," *J. Netw. Comput. Appl.*, vol. 82, pp. 56_64, 2017.