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

ABSTRACT

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.

**EXISTING SYSTEM**

* In 2008, Shacham and Waters [3], [15] proposed a publicly verifiable POR scheme with a comprehensive proof of security under the POR model [1]. Similar to [2], their scheme utilized homomorphic authenticators built from BLS signatures [42]. Subsequently, Zeng et al. [36], Wang et al. [16], [17] gave similar constructions for publicly verifiable remote data integrity check, which adopted the BLS based homomorphic authenticators. With the same reason as [2], these protocols do not support data privacy.
* In [8], [18], Wang et al. designed a privacy-preserving POR scheme. The idea is to mask the linear combination of sampled blocks in the server’s response with some random value. With the similar masking technique, Zhu et al. [19] introduced another privacy-preserving public auditing scheme. Later, Hao et al. [20] and Yang et al. [21] proposed two privacy preserving public auditing schemes without applying the masking technique. Yuan et al. [22] gave a POR scheme with
* public verifiability and constant communication cost.
* Zhang et al. [37] proposed a block update tree structure where each tree node is associated with a range of data blocks rather than one block. In applications that users access/update files in sequentially, their scheme becomes more efficient and requires less complexity and communication overheads compared with other tree based schemes.
* Cash et al. [41] employed oblivious RAM techniques to achieve full dynamic POR. Shi et al. [38] proposed a practical dynamic POR scheme with constant client storage. Very recently, Tian et al. [39] proposed to support full dynamics based on dynamic hash table which is essentially a single linked sequence table, while Shen et al. [40] proposed a new dynamic structure consisting of a doubly linked info table and a location array.
* **Disadvantages**
* A block updates tree structure where each tree node is associated with a range of data blocks rather than one block.
* There is no dynamic data integrity proof instead manual.

**PROPOSED SYSTEM**

* In the proposed system, the system aims to protect data integrity and privacy of data owner’s file. The data owner is fully trusted, and the cloud storage server and ODA are semi-trusted in different sense:
* The cloud storage server is trusted in data privacy (We assume the server has to access plaintext to provide additional services to the data owner), and is not trusted in maintaining data integrity (e.g. the server might delete some rarely accessed data for economic benefits, or hide the data corruption events caused by server failures or attacks to maintain reputation).
* Before he/she is revoked, the ODA is trusted in performing the delegated auditing task and protecting his/her verification secret key securely, but is not trusted in data privacy. A revoked ODA could be potentially malicious and might surrender his/her verification secret key to the cloud storage server.

**Advantages**

* An effective and dynamic data integrity proof.
* Scheme is sound under Bilinear Strong Diffie-Hellman Assumption, and privacy preserving under Discrete Log Assumption, both in the standard model..

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Back End - MySQL