Journal of Network and Computer Applications 77 (2017) 146–151

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

A data integrity verification scheme in mobile cloud computing ☆

Chen Lin [a], Zhidong Shen [a,*], Qian Chen [b], Frederick T. Sheldon [c]

[a] International School of Software, Wuhan University, Wuhan, Hubei 430079, China
[b] Engineering Technology Department, Computer Science Technology Program, Savannah State University, GA 31404, USA
[c] Department of Computer Science, University of Idaho, ID, USA

ARTICLE INFO

Article history:
Received 7 March 2016
Received in revised form
7 June 2016
Accepted 15 August 2016
Available online 16 August 2016

Keywords:
Mobile cloud computing
Provable data possession
Hash tree
BLS signature scheme

ABSTRACT

The provable data possession (PDP) scheme is vital to data-oriented mobile cloud computing security architecture. Without an efficient PDP scheme, users cannot verify whether the server retrieves original data before the data is being processed. One big issue of recent PDP verification schemes is that computation complexity and space overheads are very high. In this paper, we develop 2 comprehensive mobile provable data possession schemes (MPDP) using a hash tree data structure and a Boneh–Lynn–Shacham short signature scheme. Our MPDP schemes support data dynamics via verification outsourcing, blockless verification, stateless verification, and dynamic data operations. Experimental results show that these 2 MPDP schemes are highly accurate in the data verification process, and have a low data transmission cost.

1. Introduction

The number of smart phone users is reaching 2 billion in 2016, and will increase to 6.1 billion by 2020 (Consumers, 2016; Smartphone, 2016). Smart devices use wireless LAN technology and 3G/4G LTE connections for Internet access. Similar to the Software as a Service (SaaS) model of cloud computing, mobile SaaS is a mobile app distribution model that allows customers access to applications hosted by vendors or service providers in real time over the Internet. Mobile SaaS is the fundamental component of mobile cloud computing (MCC), and has unique benefits to mobile environments such as enhancing battery life and overcoming storage and memory limitations. However, data security in the mobile SaaS environment is a challenge. Service providers who suffer from Byzantine failure may hide data errors or delete infrequently used data (Wang et al., 2009). In this case, data corruption is only noticed when users attempt to retrieve their data from service providers.

To ensure that a storage server faithfully stores users' outsourced data, extra security mechanisms must be provided to provable data possession (PDP). However, low computing and storage capacity of mobile devices prevents them from running high complexity algorithms. One mechanism to solve this problem is sharing the computing burden to a third party, whose security level cannot be guaranteed.

Another issue is the inefficiency of re-uploading a large data file. Normally, if a few data blocks are modified or damaged, users are required to upload the original data again. To enhance data integrity and efficiency, we introduce 2 mobile provable data possession (MPDP) schemes using the Merkle hash tree (MHT) data structure and the Boneh–Lynn–Shacham (BLS) short signature scheme. Our scheme supports blockless, stateless, and data outsourcing verification, and dynamic data operations. MHT is a flexible data structure for arranging data block hash. The MHT increases the damaged block detection rate to 99%, and has a low transmission cost. The BLS scheme and bilinear mapping provide easy verification of data corruption. Experimental results are shown in Section 4.

2. Related works

Ateniese et al. (2007) first proposed a model for provable data possession (PDP) in 2007. This technique uses homomorphic verifiable tags to periodically and remotely audit the integrity of data stored on untrusted cloud servers. The utilization of homomorphic verifiable tags enhances the efficiency of data integrity proof. This is because the PDP scheme allows servers to prove data integrity without accessing entire files, and both space and bandwidth complexity is O(1) (constant). The proposed PDP scheme provides probabilistic proofs of data with low overheads, guarantees data possession, and practically verifies the possession of large datasets.

However, computing workloads and storage burden of clients utilizing the PDP scheme provided by Ateniese et al. (2007) is still

☆This work was supported by National Natural Science Foundation of China (Nos. 61003185 and 61402339).

* Corresponding author.
E-mail addresses: lcholmes@163.com (C. Lin), shenzd@whu.edu.cn (Z. Shen), chenq@savannahstate.edu (Q. Chen), sheldon@uidaho.edu (F.T. Sheldon).

http://dx.doi.org/10.1016/j.jnca.2016.08.017
1084-8045/

very high. To solve this problem, Yang et al. (2011) developed a new framework using bilinear signature and the Merkle Hash Tree (MHT) to aggregate verification tokens into one signature. Mobile devices need to generate secret keys and random numbers with the help of trusted platform model chips (Yang et al., 2011). Xu et al. (2016) proposed another PDP scheme called EPOS, which is 400 times faster than Ateniese et al.'s in proving data integrity on untrusted servers. EPOS enhances time complexity by reducing group exponentiation operations. Juels et al. (2007) proposed POR (Proof Of Retrievability) using a sentinel-based scheme, but it does not support public databases. Wang et al. (2009) improved POR with public verifiability and dynamic data operations using MHT construction for block tag authentication.

Aside from having good efficiency and frequency, PDP should allow clients to securely verify data integrity on malicious and unreliable servers. In addition, the PDP scheme must ensure storage servers are not cheating clients by deleting data or saving data in tertiary or offline storages (Ateniese et al., 2008). To realize security features in PDP schemes, Ateniese et al. (2007) used the public-key-based technique in their first PDP design. This technique permits unlimited verifications, but only supports static databases. Furthermore, this PDP is neither efficient in setup or verification phases. Therefore, the public-key-based technique was later replaced by symmetric-key cryptography (Ateniese et al., 2008). This new PDP scheme supports dynamic operations and requires less storage space and bandwidth on the client side. However, it does not support public verifications.

Considering resource constraints and the harsh environment of mobile devices, we designed 2 mobile PDP (MPDP) schemes adopting the Merkle hash tree (MHT) data structure and Boneh–Lynn–Shacham (BLS) short signature scheme. The MHT data structure (Merkle, 1982) is a simple structure that flexibly organizes file hash blocks and can be easily upgraded to support dynamic data. This structure has been applied to check data integrity in cryptographic file systems (Oprea and Reiter, 2007) and provides security in cloud storage service-level security (Popa et al., 2011). The BLS short signature scheme (Juels et al., 2007) based on bilinear mapping is widely used as signatures based on homomorphic cryptography. By adopting these 2 approaches, our PDP schemes successfully support verification outsourcing, blockless verification, stateless verification, and dynamic data operations.

3. Preliminaries

3.1. System structure

Fig. 1 shows the architecture of our MPDP scheme. There are 4 entities in MPDP: (1) data owner (DO), the mobile device user whose data is stored in the cloud; (2) Cloud Service Provider (CSP), the cloud providing services via the Internet; (3) trusted third party (TTP), a neutral organization providing data verification services and acting as an agent between the DO and CSP; (4) Storage Service Provider (SSP), an organization providing actual data storage service. Sometimes an entity can serve as both a CSP and SSP, and therefore, we use this concept for simplicity.

3.2. Design goals

MPDP scheme properties:

- Verification outsourcing: Heavy computational burden is shifted to the TTP to release mobile devices' computing and storage exhaustion. The TTP interacts with the CSP to accomplish verification, and then returns results to the DO via a security link.
- Blockless verification: No origin file blocks are directly involved in this step. Instead, we use pre-processed blocks. MPDP applies a byte-mapping function H to map file block to an elliptic curve group (one-way mapping). We manipulate the mapped value "file-*block*-tag" instead of actual file block during the entire verification process. In this way, the TTP is unable to access original blocks during verification, therefore safely isolating them.
- Stateless verification: Verifiers, even if not data owners, can achieve successful data verification. This is very useful and important when the MPDP scheme is used in file sharing services.
- Dynamic data support: Files stored in the CSP can be arbitrarily block-updated, block-inserted, or block-deleted, with little computational burden.

3.3. Functionalities

There are 3 main functionalities supported by MPDP: (1) data storage; (2) data integrity verification; and (3) dynamic data operations. We elaborate these 3 functionalities as follows:

- Data storage: The DO encrypts data file blocks using symmetric cryptography techniques. Then, the DO uploads these data file

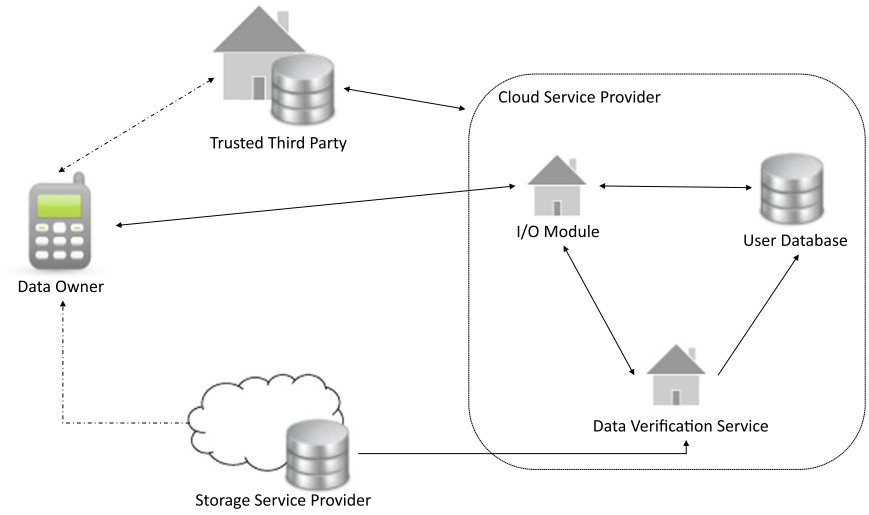

Fig. 1. Architecture of MPDP scheme.