J. Parallel Distrib. Comput. 111 (2018) 115–125

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

A rapid detection algorithm of corrupted data in cloud storage

Guangwei Xu *, Zhifeng Sun, Cairong Yan, Yanglan Gan

School of Computer Science and Technology, Donghua University, Shanghai 201620, China

HIGHLIGHTS

- A rapidly detecting algorithm based on three-dimensional data locating is proposed.
- The consistent hash is utilized to balance the computation cost.
- The cube splitting is applied to locate the corrupted blocks.
- Data concatenated and data blind technology are utilized in detection process.

ARTICLE INFO

Article history:
Received 5 May 2016
Received in revised form 22 October 2016
Accepted 13 August 2017
Available online 18 August 2017

MSC 2010:
00-01
99-00

Keywords:
Cloud storage
Data integrity verification
Corrupted data detecting
Cube-based hierarchical verification

ABSTRACT

The cloud computing provides dynamically scalable and virtualized resource service for users to access the storage data. Although having been bringing enormous convenience, it also incurs the threat of users' data loss or corruption, such as data intentionally deleted or corrupted, the service providers' hardware error and careless operation. Most of the data verification schemes based POR or PDP are proposed to verify the integrity of a data block or even a batch of data blocks. However, once the batch verification fails, it results in that all the blocks in the batch of data cannot be judged to be intact or corrupted since the corrupted blocks are not accurately identified. To improve the efficiency of corrupted data identified, we propose a rapid detection algorithm of corrupted data based on three-dimensional data locating, which is called cube-based detection. Furthermore, the consistent hash is utilized to balance the computation cost, and the cube splitting is applied to locate the corrupted blocks by narrowing the range of suspicious blocks step by step. Finally, both data hierarchically concatenated and data blind technology are utilized to improve the verification efficiency and preserve users' data privacy in the detection process. Theoretic analysis and simulation results demonstrate that our algorithm has strong detection capability and identification capability for all the corrupted blocks, and greatly decreases the cost of verification data transmission and computation.

1. Introduction

Cloud computing offers dynamic and scalable virtualized resources and services via the Internet, and provides a new model to big data storage and management. A growing number of data are stored in cloud service providers (CSP), which offer great convenience for data access, but data owners (DO) also lose effective control over their data. Especially when the data are stored in untrusted CSP, DO often suffers the threat of data loss or corruption (collectively referred to as data corruption). Data integrity verification has become an important technology to solve this problem [15]. The technology detects whether data are corrupted by service providers's hardware error or careless operation, and an adversary malicious attacks on the data integrity [9,23], and prevents CSP from concealing the data corruption [11].

Traditional data verification techniques such as hash (e.g., MD5) and digital signature (e.g., RSA) need to download remote data to the local and then verify them. However, these techniques are not applied into cloud storage environment, since the amount of data in cloud is usually massive and the process of downloading data will bring bulk network transmission consumption, especially for energy, storage, and connectivity-constrained devices [11].

In recent years, many data verification algorithms [1,7,10,18,19,22,23] based on provable data possession (PDP) [2] or proofs of retrievability (POR) [14] have been proposed. These algorithms focus on public verification, dynamic data operations, blockless verification, stateless verification, and user data privacy preservation. Moreover, to improve the efficiency of data verification, many researchers utilize data sampling to ensure the verification at high detection rate [2,10,18,19,23]. They all use block-oriented verification and the verification of a batch of data

* Corresponding author.
E-mail addresses: gwxu@dhu.edu.cn (G. Xu), china-firstszf1989@163.com (Z. Sun), cryan@dhu.edu.cn (C. Yan), yanglan.gan@gmail.com (Y. Gan).

http://dx.doi.org/10.1016/j.jpdc.2017.08.004
0743-7315/

to reduce computation and transmission overhead. Even though the batch verification improves the verification efficiency since it accumulates the verification tags and merges the verified data, it also causes a serious issue that the verifier cannot rapidly and accurately identify the corrupted blocks in these merged data once the batch verification fails. To identify the corrupted data, the verification must re-execute one block by one block verification. Thus, even though the batch verification improves the verification efficiency, we do not still ignore the issue of verification interference. For example, only a corrupted block results in the batch verification failure.

To rapidly and efficiently identify the corrupted data and reduce data computation and transmission consumption in detection process, we propose a cube-based corrupted data detection algorithm (CDD) based on three-dimensional location technology. In the algorithm, we construct a challenge cube which ingeniously merges the extracted data blocks' verification proofs and accumulates the extracted blocks' verification tags to rapidly detect the integrity of all the extracted data. Then we split the cube step by step to accurately identify the corrupted blocks in all the extracted data. Furthermore, algorithm analysis and simulation results show that our algorithm achieves a lightweight and efficient detection under the condition of ensuring the verification security. Our contributions in this paper can be summarized as follows.

First, we construct a challenge cube and analyze its relevant parameters in detecting corrupted blocks.

Second, the cube-based hierarchical verification proofs and tags are generated. The algorithm utilizes the bilinear mapping and BLS short signature [4] to generate homomorphic verification tags, and achieve public verification. Also, the algorithm combines the accumulative verification tags [20] with Diffie–Hellman key exchange technology [6] to preserve the data privacy and resist the attack of data tampering [12].

Finally, the cube-based verification algorithm is designed to rapidly identify corrupted data and improve the verification efficiency. Moreover, we comprehensively analyze the security of algorithm, and design an experiment system to evaluate the performance of our algorithm.

We organize the paper as follows. In Section 2, we briefly summarize the current research on the verification of data integrity. Section 3 describes the verification model, threat model and design goals. Section 4 advances a cube-based corrupted data detection algorithm. Section 5 analyzes the security and performance of algorithm. Section 6 evaluates the performance of algorithm by simulations. Finally, Section 7 concludes the contributions of this paper and presents future extension to the work.

2. Related work

In recent years, many data verification schemes based on PDP or POR have been proposed in [1,2,7,10,14,16–19,22,23]. These schemes usually consider the following issues:

- Public verification. Both data owners and third-party verifiers can verify the integrity of data stored in cloud servers [2,16–19]. The principle is to store the non-forgery verification tags in cloud servers.
- Dynamic data operation. Users can efficiently execute block-level or more fine-grained updates to outsourced data without recomputing the verification tags of unrenewed data blocks. Erway [7] utilized a rank-based authentication skip list to verify the updated data. Zhu [23] achieved an integrity verification scheme based on the index hash table (IHT) to support dynamic data operations. Applying merkle hash tree (MHT), Wang [19] presented a method for public auditability and data dynamics. On the basis of MHT, Liu [10] expanded it to rank-based merkle hash tree (R-MHT) and achieved the authorized public verification with efficient verifiable fine-grained updates. Zhang [22] improved the unbalanced disadvantage of MHT updating to achieve a balanced update tree.
- Stateless verification. Verifier does not need to maintain or update any state information in the verification process, since the status information is difficult to be maintained in such a complex cloud environment [19].
- Blockless verification. To ensure the safety and efficiency of verification, the verifier checks the integrity of outsourced data without downloading or accessing any actual data block from the remote storage [16,17,19].
- Batch verification. To improve the efficiency of verification, the verifier can handle a batch of data or multiple verification tasks from different users simultaneously [17,19];
- Privacy preservation. To avoid information leakage in the public verification, Wang [18] protected data privacy with blind information technology and prevented the third party auditor from disclosing data privacy. Thereafter, Wang [17] exploited a homomorphic authenticable ring signature (HARS) technology to preserve user's identity privacy in verified data for the cloud storage.
- Data confidentiality. Before data are uploaded, data owners encrypt their data to prevent CSP from using their data without permission [1,14].

These schemes pay more attention to the accuracy of data verification. Also, they utilize the batch verification to improve the verification efficiency. Some methods are proposed to detect the corrupted data in the batch verification.

- One by one. It is a simple approach and the abbreviation for one block by one block checked. Each verification tag is checked to judge the integrity of each corresponding data block. This method can accurately identify the corrupted data blocks from all the checked data. However, it will lead to much cost of verification proofs' computation and transmission with the increase of the number of checked data blocks. Also, it decreases the efficiency of corrupted data detecting.
- Binary search. Wang [17] utilized a binary search approach to find the incorrect proofs corresponding to the data blocks. The binary approach divides all data blocks into a binary tree where each parent at most includes 2 children. Hwang [8] exploited the detecting illegal signature technology and public verification scheme based on BLS short signature technology [4], and proposed a data integrity verification scheme to report locations of corrupted blocks. However, the scheme cannot identify all corrupted data blocks in most cases, or requires the verification to execute many times to find them all. Also, it costs a large overhead of data computation and transmission.
- Matrix crossover. In Li's verification scheme [9], a signature matrix was constructed by generating a random number for each message signature. Hwang [8] utilized a random matrix to detect the corrupted blocks. The method is to detect whether the signature is illegal through each row and column crossover element of the matrix and eventually locate the actual corrupted blocks. The matrix is constructed by data block indices according to the order from left to right and top to bottom. CSP generates a data integrity proof with a row or column element in the matrix for each data block, and then the verifier identifies the corrupted blocks via individually checking each proof of each row or each column element. Also, it cannot accurately identify the corrupted blocks while the verification of many row and column elements fails.