Research on the Detection Algorithm of Data Integrity Verification Results in Big Data Storage

Shuai Yin*
Jinan University, Shandong, China, 262200
*E-mail: swjdybz@163.com

Abstract. As one of the most widely used big data computing technologies, data cloud storage technology brings great convenience to users. However, while sharing data, it also causes data corruption and missing data integrity problems. Today's verification of the integrity of remote data is publicly performed by trusted third parties. This makes the verifier have the potential threat of providing false information. The resulting validation data is untrustworthy. In order to ensure the security and integrity of data, technicians need to establish a data authentication results detection algorithm. Moreover, this algorithm ensures the reliability and utilization of data by improving the limited verification results.

Keywords: Data Storage, Verification, Detection Algorithm

1. Introduction

Cloud computing is a dynamically scalable virtualization resource for Internet-based related services. It can use the network virtual technology to get the terminal network and application resources at a small cost. As the most potential application of cloud computing, cloud storage technology is a dynamically scalable and virtualized storage method through the Internet. In order to avoid the loss of the data stored in the cloud and guarantee the integrity of the user's data storage, people have developed an algorithm to verify the integrity of the remote data.

These algorithms include data holding technology and data recovery technology. The actual cloud storage environment is completely open to the outside world. There is no absolutely reliable data verifier. A detection algorithm of data verification results should generate integrity verification evidence to detect data integrity\(^1\). It should generate untrusted test evidence to determine the correctness of data validation results. It should be able to resist the non-tamper ability of verification results by building a detection tree of verification results. It can use double evidence to perform cross validation to ensure data integrity.
2. Verification model and problem proposal

2.1. Establishment of validation model

We can select the publicly verifiable model in the existing data validation algorithm to describe the data integrity. The validation model should include three entities. These entities include data owners, cloud storage providers, and data validators. The process of data validation should be described in the following five phases.

\[
\begin{align*}
\text{KeyGen}() & \rightarrow (sk, pk) \quad (1) \\
\text{TagGen}(sk, M) & \rightarrow \emptyset. Do \quad (2) \\
\text{ChalGen}(M) & \rightarrow \Omega. Do \quad (3) \\
\text{ProofGen}(M, \emptyset, \Omega) & \rightarrow P. CSP \quad (4) \\
\text{Verify}(\Omega, P, pk) & \rightarrow \text{true}\backslash\text{false}. DV \quad (5)
\end{align*}
\]

2.2. Establishment of adversary model

Today’s data validation is mainly aimed at the possible data corruption and deception of cloud storage providers. DV performing data validation tasks may also provide spurious validation results. Because there is no corresponding detection mechanism in the data validation model, DO can only blindly obey the validation results. This will cause DO to lose its main role. In addition, the collusion of CSP and DV may cause greater loss of DO.

2.3. Questions raised

We can see from the data validation model and adversary model that it is necessary to detect the untrusted validation results of the model. We need to solve three problems in the algorithm. Identify issues with untrusted validation data for CSP. The problem of identifying untrusted verification results of DV. Identify the problem of conspiracy attack of CSP and DV.

3. Detection algorithm of untrusted verification results

For the convenience described below, we can define some symbols. \( Q \) is the challenge set. \( t_i \) is the label of data block \( m_i \). \( P \) is proof of completeness. \( P' \) is untrustworthy test evidence. \( sk \) is a private key. \( pk_1, pk_2 \) are two public keys. \( G_1, G_2 \) is a bilinear map. \( TP \) is evidence of labeling. \( DP \) is the actual data evidence.

3.1. Generation of test evidence for untrusted verification results

In the traditional data verification scheme, after CSP accepts the challenge of DV, the computer calculates the integrity evidence set \( P \) according to the challenge set, data set and label set. The value of set \( P \) will be returned to DV by the system as the basis of decision amount. It is very important to calculate the evidence of integrity verification and the evidence of untrusted detection. The calculation formula of \( DP_2 \) is:
\[ DP_2 = e(u,R)\sum_{i \in Q} v_i m_i \times e(\prod_{i \in Q} p_{k_2}^{m_i} g_2^{v_i}) \]  

\( v_i, r \) is the random number of the corresponding data \( m_i \) selected in \( Z_p \). \( u \) is an element of \( G_1 \). \( R \) is the challenge mark. Therefore, the calculation formula of \( TP_2 \) is:

\[ TP_2 = \prod_{i \in Q} (t_i, pk_2)^{m_i} \]  

### 3.2. Detection of untrusted verification results

In the process of data verification in the cloud storage environment, there are many data to be verified. The difference of verifier's equipment will also restrict the implementation of data verification tasks. Therefore, the verification task can be performed by multiple DVS. We take a DV data verification result as an example to study the detection method of untrusted verification result. We can use the technology of binary tree to build detection tree.

### 3.3. Detection method of untrusted verification

The algorithm used in this paper can not only verify the integrity of data stored in CSP, but also check the reliability of DV verification results. Therefore, we need to improve the traditional data validation model. The algorithm is described in detail as follows:

1. **KeyGen(\( \lambda \))→(sk,pk_1,pk_2)**
2. **TagGen(sk,M)→Ø.DO**
3. **ChalGen(M)→Ø.DV**
4. **ProofGen(M,Ø,Ø)→(P_1,P_2)**
5. **VerifyData(Ø,P_1,pk_1)→(i,true\{false\)**
6. **ResultCheck(R_0,DP_2,Ø)→Y\N.DO**

### 3.4. Segment detection support

In order to improve the adaptability of detection evidence, we also need to provide a generation method. We can assume that data \( m_i \) is divided into \( s \) data segments \( m_{ij} \). The label of block \( m_i \) is:

\[ t_i = (h(W_i) \times \prod_{j=1}^{s} u_i^{m_{ij}})^{sk} \]  

The calculation formula of the data evidence of the test evidence is:

\[ DP_2 = \prod_{j=1}^{s} e(u_j,R)^{MP_j} \times e(pk_2,R) \]

### 4. Analysis of algorithm security

The analysis of the verification results provided by the detection algorithm for \( V \) of data integrity depends on the correctness of the algorithm. The verification formula for the correctness of the algorithm is:
\[ left = e(u,R) \sum_{i \in Q} \nu_i m_i \times e(\prod_{i \in Q} p_{Q_i}^{k_i}, q_{Q_i}^{k_i}) \times e(\prod_{i \in Q} h(W_i)^{r_{Q_i}} pk_i) \] 

(16)

5. Experimental analysis

5.1. Verifying ability

Verification capability refers to the calculation time spent in verifying the same number of data blocks (see Figure 1).

![Figure 1. Verifying ability](image)

5.2. Storage overhead

Storage overhead refers to the storage calculation of validation data generated in the process of data validation (see Figure 2).

![Figure 2. Storage overhead](image)
5.3. Reliability verification

The reliability of verification refers to the proportion of reliability data in the verification results provided by the verifier (see Figure 3).

![Figure 3. Reliability of verification results](image)

6. Conclusion

In order to avoid the problem of unreliable verification results in data integrity verification, the verification results are forged and cheated. We use a detection algorithm of data verification results to resist the forgery attack of untrusted verification results. Experimental results show that the algorithm can guarantee the reliability of data verification.

References

[1] Guangwei X, Yanke B, Cairong Y, et al. Check Algorithm of Data Integrity Verification Results in Big Data Storage[J]. Journal of computer research and development, 2017.

[2] Guang-Wei X U, Yan-Bin Y, Ke L, et al. Research on Verification Algorithm of Data Integrity Inspecting in Remote Untrusted Storage[J]. Journal of Chinese Computer Systems, 2016.

[3] Xia Y H, Hong H S, Lin G F, et al. Efficient Data Integrity Verification Using CRC Based on HDFS in Cloud Storage[J]. Itm Web of Conferences, 2017, 11.

[4] Liu C, Ranjan R, Zhang X, et al. Public Auditing for Big Data Storage in Cloud Computing -- A Survey[C] Proceedings of the 2013 IEEE 16th International Conference on Computational Science and Engineering. IEEE, 2013.

[5] Xu G, Sun Z, Yan C, et al. A rapid detection algorithm of corrupted data in cloud storage[J]. Journal of Parallel and Distributed Computing, 2018, 111(jan.):115-125.

[6] Xu G, Bai Y, Pan Q, et al. Data verification tasks scheduling based on dynamic resource
allocation in mobile big data storage[J]. Computer networks, 2017, 126(oct.24):246-255.