symmetry

MDPI

Article

Detection Method of Data Integrity in Network Storage Based on Symmetrical Difference

Xiaona Ding

School of Electronics and Information Engineering, Sias University of Zhengzhou, Xinzheng 451150, China; dingxiaona9852@163.com

Received: 15 November 2019; Accepted: 26 December 2019; Published: 3 February 2020

Abstract: In order to enhance the recall and the precision performance of data integrity detection, a method to detect the network storage data integrity based on symmetric difference was proposed. Through the complete automatic image annotation system, the crawler technology was used to capture the image and related text information. According to the automatic word segmentation, pos tagging and Chinese word segmentation, the feature analysis of text data was achieved. Based on the symmetrical difference algorithm and the background subtraction, the feature extraction of image data was realized. On the basis of data collection and feature extraction, the sentry data segment was introduced, and then the sentry data segment was randomly selected to detect the data integrity. Combined with the accountability scheme of data security of the trusted third party, the trusted third party was taken as the core. The online state judgment was made for each user operation. Meanwhile, credentials that cannot be denied by both parties were generated, and thus to prevent the verifier from providing false validation results. Experimental results prove that the proposed method has high precision rate, high recall rate, and strong reliability.

Keywords: symmetric difference; network; data integrity; detection

1. Introduction

In recent years, the cloud computing becomes a new shared infrastructure based on the network. Based on Internet, virtualization, and other technologies, a large number of system pools and other resources are combined to provide users with a series of convenient services [1]. The cloud computing has the advantages: providing convenient computing resource sharing pool, flexible resources, safe and controllable data, cost saving, unified management, and low-cost fault tolerance, so the related products of cloud service are more and more popular with users. With the rapid development of information technology, the traditional data storage mode cannot meet the new needs and challenges [2]. The cloud storage has attracted great attention due to its low cost and high efficiency. As a new storage mode, the cloud storage has attracted more and more attention with the rapid popularization. How to ensure the correctness and integrity of data files stored in cloud servers is one of the key issues in the development of cloud storage technology [3]. Based on the importance of data integrity detection, many excellent research results have been obtained in this field.

One study [4] proposed a method for security data integrity detection. The algorithm performs cross-validation by establishing a dual-evidence mode of integrity verification evidence and untrusted detection evidence. The integrity verification evidence is used to detect data integrity and use untrusted detection. Evidence determines the correctness of data verification results. In addition, the reliability of the verification results is ensured by constructing a detection tree. However, this method has poor recall. Literature [5] proposed a method for data integrity detection in big data storage. Cross-checking the integrity of data in the storage with a two-factor verification method to verify that the data integrity of the big data store is complete. A check tree is constructed to ensure the reliability of the verification results. The data storage integrity of this method is good, but the encoding time of this method is too long and the efficiency is low. Study [6] proposed a method for detecting the integrity of multi-copy data in a cloud storage environment. By introducing a classic B + tree authentication structure and designing a lightweight system model, users can timely grasp the integrity and consistency of multi-copy data on the cloud server side. Adding the rank value to the B + tree enables users to quickly and accurately locate the location of the wrong data, which is convenient for the next processing. However, after this method is used, the data availability rate is low, which cannot satisfy the effective storage of network data.

In order to improve the precision rate and recall rate during the data integrity detection, a method of detecting data integrity of network storage based on symmetric difference was put forward.

2. Research Status

Data has always been an important carrier of information. Traditional data storage methods mainly rely on semiconductors, optical, magnetic, etc. to achieve data storage, data is stored locally. With the rise of the Internet, storage technologies based on network communication emerge in endlessly, such as disk array, network access, storage area network and other storage technologies. However, with the advent of the era of cloud computing, cloud storage has emerged as the times require, and gradually developed into the mainstream storage technology [7]. The traditional method is likely to be replaced by cloud storage. In the cloud platform, the cloud service provider (CSP) is mainly responsible for storing and managing data, and maintaining the availability and reliability of data. However, the data stored in CSP by enterprises or individual users may involve customer related information, personal hobbies, and habits, which often has potential value and is easy to be coveted by people with ulterior motives. In addition to data security, the research on cloud storage also includes data integrity. Because of the important theoretical research and practical application value of data in storage security, it has been concerned by many experts in related fields at home and abroad. Ateniese et al. Took the lead in publishing the paper of PDP (proof of data possession) scheme, which can still ensure the data can be maintained on unreliable storage nodes. In this paper, T represents a combination of tags corresponding to the set of file blocks to be queried. ρ is a hash value under the set of file blocks to be queried. When data integrity is detected. First remove the index information of the tag in t to get the pre calculated tag information, and then compare it with ρ through some established calculation. As long as they are equal, it means that the data stored on the server side keeps the original state without any tampering, which proves the integrity of the file. However, this method ignores the user's demand for dynamic data update, and there are many difficulties in the design level.

3. Method

3.1. Network Storage Data Acquisition

The network crawler is applied to the network storage data crawling, and thus to provide support for the data integrity detection. As a complete automatic image annotation system, it is necessary to use the crawler technology to automatically obtain the network image and build an image annotation database. On this basis, the web crawler technology is used to capture the image and related text information.

Web crawler design:

The main function of web crawler is to obtain the network data. It mainly uses the hypertext link in web page to roam, find, and collect information in the Internet, so as to provide the data source for the next stage of information extraction, organization, and management. Generally, the crawler starts from an initial URL set and uses some search strategy to traverse the web page and download various data resources along the URL of hypertext link, such as breadth-first strategy, depth-first strategy.

The web crawler system will maintain a URL table, including some original URLs. Based on these URLs, Robot downloads the corresponding pages and extracts new URLs from them and then adds them to the URL table. After that, Robot repeats the above process until the URL queue is empty.

On this basis, the basic framework of web crawler is shown in Figure 1 (source: author own conception, adapted from Wu Libing):

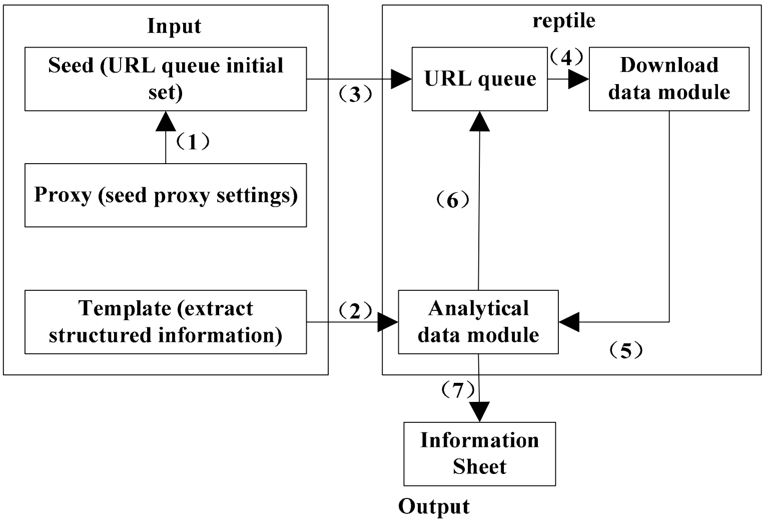

Figure 1. Basic framework of web crawler.

The working process of crawler is as follows:

Import the seed (the initial URL list) and import the agent. Import templates (regular expressions constructed by different web features). The agent is corresponded to the seed URL, and the seed is put into the crawling queue as the initial set of URL queue. Take the waiting URL from the queue, and then enter the crawling state. After that, the web source file corresponding to the URL is downloaded. Transfer the regular expression in the template object array and the downloaded web page source file into the web page analysis module for matching, so that the structured information can be obtained. The important links contained in the structured information, such as URL of the next page of post list and URL of the post, which continues to be put into the URL queue and waits for crawling. Information required by other users in structured information, such as post name, post sender, reply number, and reply content are stored in the information table.

Extraction of image and related text:

Through the research on the page, the text information related to the image mainly includes:

(1) The texts around the image in page, most of them are too long, including a lot of semantic information. During the page analysis, they are mostly related to the page structure, such as the adjacent texts in the same row or column of the table. When an image exists as an illustration, the surrounding words have limited contribution to the annotation on the image.

(2) File name, title, or description information of image are usually concise phrases or words, which have strong generalization ability.

(3) The title of the image link page. The content of image web page is highly relative to image, and some titles which are used to generalize the content of hyperlink web page. Meanwhile, they also have something to do with image semantics.

The image semantics mainly comes from the analysis of related text. Firstly, Chinese characters and English existing in related text are translated into Chinese, and then the automatic word segmentation and the part of speech tagging are carried out.

When the system receives a large amount of text information through the crawler program, the following problem is how to extract the keywords from documents. One of the obvious differences between Chinese and English is that in Chinese text, there is no obvious natural separator between

Chinese characters or vocabularies. Meanwhile, the number of Chinese words is uncertain, and the collocation is flexible, and the semantics is diverse. Most of them are composed of two or more Chinese characters, and the writing is continuous, which brings more difficulties for Chinese understanding and keyword extraction.

Generally, users may retrieve in the form of words or single character when querying, so the system should label images with shorter words or single character as much as possible. When the annotation platform receives various long texts, it is necessary to divide the whole sentence into smaller vocabulary units at first, and then process them through the keyword selection module [8].

In Chinese word segmentation, it is necessary to consider the complexity of time on the premise of accuracy [9]. A simple way is the maximum matching method of positive word subtraction. The basic thought is to build a dictionary in advance, and then extract a preset length word string from the long sentences of natural language, and compare it with the dictionary. If the string belongs to the dictionary, it will be regarded as a meaningful word string. Then, the separator is used to split it and output it. Otherwise, it is necessary to shorten the word string and search again in the dictionary. Finally, we should move backward and repeat the above steps. The basic description of this algorithm is shown in Figure 2 (source: author own conception, adapted from Wang Ruilei).

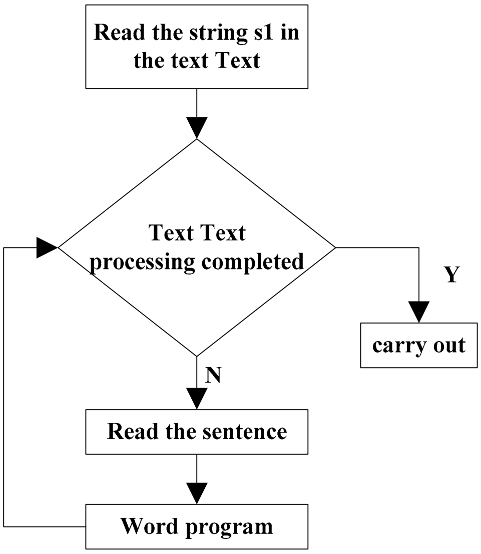
Read the string s1 in the text Text
Text Text processing completed
Y
carry out
N
Read the sentence
Word program

Figure 2. Operational scheme of positive word reduction maximal matching method.

With the distributed crawler as the core, an automatic image download program is implemented, and the main flow chart is shown in Figure 3. It mainly includes the web source file downloading module, regular matching module, next page URL construction and image URL construction module, and image download module whose entry parameter is image URL.

3.2. Data Feature Extraction Based on Symmetrical Difference

In the above process of data acquisition, the automatic word segmentation and pos tagging, Chinese word segmentation are used to achieve the feature analysis of the text data. Then, the symmetrical difference algorithm is taken as the main method and background subtraction is taken as the auxiliary method, so that the image data feature extraction and preliminary recognition of data integrity are achieved. Thus, the precision rate of data integrity detection is improved.

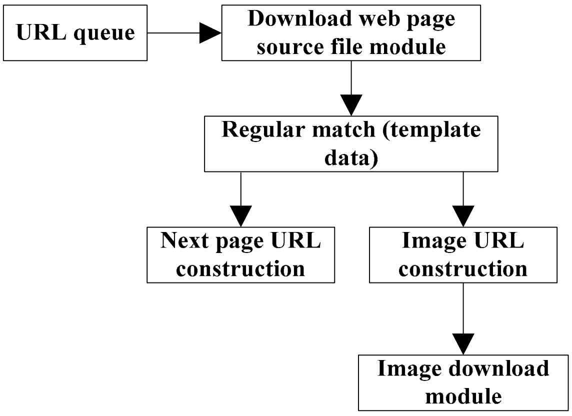
URL queue
Download web page source file module
Regular match (template data)
Next page URL construction
Image URL construction
Image download module

Figure 3. Automatic download process of image.

The symmetrical difference algorithm can remove the influence of the background revealed by the motion and draw the contour of the moving object accurately [10,11]. The basic algorithm is as follows: the source images of three consecutive frames in video sequence are $f_{(k-1)}(x,y)$,$f_{(k)}(x,y)$ and $f_{(k+1)}(x,y)$. The absolute difference gray-scale images of two adjacent source images are calculated respectively, namely $d_{(k,k+1)}(x,y)$ and $d_{(k,k+1)}(x,y)$.

$$\begin{cases} d_{(k-1,k)}(x,y) = \left|W \times f_{(k)}(x,y) - W \times f_{(k-1)}(x,y)\right| \\ d_{(k,k+1)}(x,y) = \left|W \times f_{(k+1)}(x,y) - W \times f_{(k)}(x,y)\right| \end{cases} \tag{1}$$

where, W is a window function to suppress noise. Because the mean filtering will blur the image, leading to the loss of edge information, the median filtering function with 3×3 window is chosen to suppress the noise.

The binary images $b_{(k-1,k)}(x,y)$ and $b_{(k,k+1)}(x,y)$ are obtained by calculating threshold values of $d_{(k-1,k)}(x,y)$ and $d_{(k,k+1)}(x,y)$ respectively. The binary image $d_S^{(k)}(x,y)$ of symmetrical difference result is obtained by logic operation of $b_{(k-1,k)}(x,y)$ and $b_{(k,k+1)}(x,y)$ at each pixel position. The formula is

$$d_S^{(k)}(x,y) = \begin{cases} 1, & b_{(k-1,k)}(x,y) \cap b_{(k,k+1)}(x,y) = 1 \\ 0, & other \end{cases} \tag{2}$$

The basic idea of background subtraction is to subtract the current image from the background image stored in advance or real-time background image. The pixel point whose difference is greater than a certain threshold value is regarded as the point on the moving target. Otherwise, it is considered as the background point, which is very suitable for detecting the moving target when the background image changes less with time. By comparing the difference of gray values between the current source image $f_{(k)}(x,y)$ and background image $B_k(x,y)$, the foreground image $d_b^{(k)}(x,y)$ can be obtained. The formula is

$$d_b^{(k)}(x,y) = \begin{cases} 1, & \left|W \times f_{(k)}(x,y) - W \times B_k(x,y)\right| \geq T \\ 0, & other \end{cases} \tag{3}$$

where, T denotes the threshold value.

The ideal background $B(x,y) = \cup_{s=0}^{n-1} B_{s,s+1}$ is obtained on the basis of the given N frame image, in which $\cup$ denotes the image splice operator and $B_{s,s+1}$ denotes the common background of frame S and frame $S+1$.

Formula (4) is used to judge the attribution of sub block $B_K(s,j)$,

$$B_K(s,j) = \begin{cases} s, s+1 \\ \text{Frame common background} \quad C[D_{bk}(s,j)] \leq TB \\ \text{target area} \quad other \end{cases} \tag{4}$$

The moving object detection algorithm based on background subtraction and symmetrical difference can accurately extract and update the background model when the image has several moving targets.

3.3. Data Integrity Detection and Accountability Mechanism

Based on the data collection and feature extraction above, the random sentry data segment is introduced to achieve the final data integrity detection. Combined with the accountability scheme of data security of the trusted third party, the trusted third party was taken as the core. The online state judgment was made for each user operation. Meanwhile, credentials that cannot be denied by both parties were generated, so as to ensure the reliability of audit and accountability of trusted third party when the cloud is not trusted. In addition, it is able to prevent the verifier from providing the false verification results.

This scheme randomly selects the sentry data segment to detect the data integrity. Because the sentry data segment contains the selected sentry information, the effective data and other sentry data, this scheme can support the detector to carry out infinite detection, and thus to improve the data integrity detection and recall rate. It is not necessary to worry about the sentry leakage caused by multiple tests.

Next, the scheme flow is introduced according to the data preprocessing stage, the challenge initiation stage and the detection and verification stage.

3.3.1. Data Preprocessing Stage

The original data blocking: firstly, the network storage user runs the key generation formula: $KeyGen(p_u) \rightarrow (sKey_u, pKey_u)$. Secondly, the public key $sKey_u$ and private key $pKey_u$ held by the user are generated, and the private key $pKey_u$ is saved as a secret pair. After that, the user uses the public key $sKey_u$ to encrypt the original data file: $FileEncrypt(F, sKey_u) \rightarrow (F')$. Then, the file block algorithm is carried out, $FileBlock(F', m) \rightarrow \{F_1, \cdots, F_m\}$. F' is divided into m blocks. The sizes of blocks are the same. Finally, each data block in $\{F_1, \cdots, F_m\}$ set is divided into n blocks: $FileBlock(F_i, n) \rightarrow \{F'_{1i}, \cdots, F'_{ni}\}$. Finally, the data block matrix consisting of m vectors is obtained: $\mathbf{F}=\begin{bmatrix} \mathbf{F'_{11}} & \mathbf{L} & \mathbf{F'_{1m}} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{F'_{n1}} & \mathbf{L} & \mathbf{F'_{nm}} \end{bmatrix}$.

Erasure code: the network storage user performs the erasure code on the obtained data matrix, $ECoding\left(B, \left\{F'_{ij}\right\}_{1 \leq i < n, 1 \leq j < m}\right) \rightarrow \mathbf{A}=\begin{bmatrix} \mathbf{a_{11}} & \mathbf{L} & \mathbf{a_{1m}} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{a_{k1}} & \mathbf{L} & \mathbf{a_{km}} \end{bmatrix}$. The vandermonde matrix B' of erasure correction code is a matrix with k rows and n columns, $(k > n)$. Its model is shown in Formula (5)

$$\mathbf{B'}=\begin{bmatrix} 1 & \mathbf{a_1^1} & \mathbf{a_1^2} & \mathbf{L} & \mathbf{a_1^{n\text{-}1}} \\ 1 & \mathbf{a_2^1} & \mathbf{a_2^2} & \mathbf{L} & \mathbf{a_2^{n\text{-}1}} \\ 1 & \mathbf{a_3^1} & \mathbf{a_3^2} & \mathbf{L} & \mathbf{a_3^{n\text{-}1}} \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ 1 & \mathbf{a_k^1} & \mathbf{a_k^2} & \mathbf{L} & \mathbf{a_k^{n\text{-}1}} \end{bmatrix} \tag{5}$$

In fact, data matrix $\mathrm{A} = \begin{bmatrix} \mathbf{a_{11}} & \mathbf{L} & \mathbf{a_{1m}} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{a_{k1}} & \mathbf{L} & \mathbf{a_{km}} \end{bmatrix}$ is obtained by m matrix multiplications, that is to say, each data block set $\{F'_{1i}, \cdots, F'_{ni}\}$ is multiplied by the vandermonde matrix B' of erasure correction code in the form of vector, and then these products constitute the data matrix.

$$\begin{bmatrix} 1 & \mathbf{a}_1^1 & \mathbf{a}_1^2 & \mathbf{L} & \mathbf{a}_1^{\mathbf{n}\text{-}1} \\ 1 & \mathbf{a}_2^1 & \mathbf{a}_2^2 & \mathbf{L} & \mathbf{a}_2^{\mathbf{n}\text{-}1} \\ 1 & \mathbf{a}_3^1 & \mathbf{a}_3^2 & \mathbf{L} & \mathbf{a}_3^{\mathbf{n}\text{-}1} \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{O} & \mathbf{M} \\ 1 & \mathbf{a}_\mathbf{k}^1 & \mathbf{a}_\mathbf{k}^2 & \mathbf{L} & \mathbf{a}_\mathbf{k}^{\mathbf{n}\text{-}1} \end{bmatrix} \times \begin{bmatrix} \mathbf{F}'_{1\mathbf{i}} \\ \mathbf{F}'_{2\mathbf{i}} \\ \mathbf{M} \\ \mathbf{F}'_{\mathbf{n}\mathbf{i}} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{1\mathbf{i}} \\ \mathbf{a}_{2\mathbf{i}} \\ \mathbf{M} \\ \mathbf{a}_{\mathbf{k}\mathbf{i}} \end{bmatrix} \tag{6}$$

According to the vandermonde matrix B' with k rows and n columns, we can see that there are $(k-n)\times m$ redundant data blocks in this error correction code. In other words, for data vector $(a_{1i}, a_{2i}, \cdots, a_{ki})^T$, when the maximum loss is $k-n$ data blocks, the entire data vector can be recovered completely.

Sentry position generation: after the data matrix is generated, the network storage user sets the number of sentries which need to be placed in each data block, $l'(l'\geq 2)$. Meanwhile, the user uses the sentry insertion position generation algorithm $GuardPosition(i,j,l'pKey_u)\rightarrow\left(P_{i,j,1},\cdots,P_{i,j,w},\cdots,P_{i,j,l'}\right)$ to calculate the position array. $P_{i,j,w}$ denotes the position of the wth sentry which needs to be inserted into the data block. a'_{ij} denotes the data block after inserting the sentry data. $len\left(a'_{ij}\right)=len\left(a_{ij}\right)+q\times l'$ represents the bit length of a'_{ij} after inserting the sentry data. q represents the bit length of each sentry. Thus, the proportion of the actual effective data in data block a'_{ij} to the total storage data is $len\left(a_{ij}\right)/\left(len\left(a_{ij}\right)+q\times l'\right)$.

The generation algorithm $GuardPosition(i,j,l'pKey_u)\rightarrow\left(P_{i,j,1},\cdots,P_{i,j,w},\cdots,P_{i,j,l'}\right)$ of sentry insertion position is to select the random number to hash the bit length of a'_{ij}, namely $P_{i,j,w}=H^w(ran(i,j)\|pKey_u\|User\_id)\mathrm{mod}len\left(a'_{ij}\right)$. The function $ran(i,j)$ denotes the random number generated by i,j. $User\_id$ denotes the unique ID of network storage user. $H(\cdot)$ is the hash function, and $H^1(\cdot)=H(\cdot), H^2(\cdot)=H\left(H^1(\cdot)\right),\cdots,H^{l'}(\cdot)=H\left(H^{l'-1}(\cdot)\right)$. Because $P_{i,j,w}$ is generated by random number, it is necessary to reorder the set of sentry positions from big to small. Finally, the orderly sequence of sentry insertion positions can be obtained [12,13].

Sentry data generation: network storage user uses the array $\left(P_{i,j,1},\cdots,P_{i,j,w},\cdots,P_{i,j,l'}\right)$ of orderly sentry insertion position and preset length q of sentry to calculate sentry at position $P_{i,j,w}$:$G_{i,j,w}$: $GuardGen\left(i,j,P_{i,j,w},q,pKey_u\right)\rightarrow\left(G_{i,j,w}\right)$.

The sentry generation algorithm $GuardGen\left(i,j,P_{i,j,w},q,pKey_u\right)\rightarrow\left(G_{i,j,w}\right)$ is defined as the binary 0/1 string q-bit data with the result $H^w(ran(i,j)\|pKey_u\|User\_id)$. At this stage, network storage user needs to take $pKey_u$ and $ran(i,j)$ as secrets and then save them locally. In addition, they are not disclosed to any other party in the data preprocessing stage.

Sentry insertion: after the network storage user generates the sentry, the sentry insertion algorithm is used to insert the sentry set $G_{i,j,w}$ into the data block a_{ij}:

$$Guardinsert\left(a_{ij},P_{i,j,w},G_{i,j,w}\right)\rightarrow\mathbf{A'}\text{=}\begin{bmatrix} \mathbf{a}'_{11} & \mathbf{L} & \mathbf{a}'_{1\mathbf{m}} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{a}'_{\mathbf{k}1} & \mathbf{L} & \mathbf{a}'_{\mathbf{km}} \end{bmatrix} \tag{7}$$

When calculating the position of sentry $\left(P_{i,j,1},\cdots,P_{i,j,w},\cdots,P_{i,j,l'}\right)$, we do not consider that the insertion of previous $P_{i,j,w}$ will affect the change of sentry position P_{i,j,w^+} behind. When *Guardinsert* is implemented, it is necessary to move the original position $P_{i,j,w}$ back $w-1$ positions of q. After $P'_{i,j,w}=P_{i,j,w}+(w-1)\times q$ transformation, we will insert $G_{i,j,w}$ into the data block matrix $\mathbf{A}\text{=}\begin{bmatrix} \mathbf{a}_{11} & \mathbf{L} & \mathbf{a}_{1\mathbf{m}} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{a}_{\mathbf{k}1} & \mathbf{L} & \mathbf{a}_{\mathbf{km}} \end{bmatrix}$ to generate the final confusion data matrix $\mathbf{A'}\text{=}\begin{bmatrix} \mathbf{a}'_{11} & \mathbf{L} & \mathbf{a}'_{1\mathbf{m}} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{a}'_{\mathbf{k}1} & \mathbf{L} & \mathbf{a}'_{\mathbf{km}} \end{bmatrix}$, and then upload the data matrix to the network server.

Upload parameter to trusted third party: the network cloud storage user uses the public key $sKey_t$ of trusted third party to encrypt the parameters $\left(n,m,k,l',q,ran(i,j)_{1\le i<k,1\le j<m}, len\left(a'_{ij}\right), pKey_u\right)$ which are used in the data pre-processing stage, sec $= ParaEncrypt\left(\left(n,m,k,l',q,ran(i,j)_{1\le i<k,1\le j<m}\right), len\left(a'_{ij}\right), pKey_u\right)$. After that, we can store them in the trusted third party. In the subsequent data integrity detection, the network storage user authorization can directly use the private key $pKey_u$ to decrypt sec. According to the calculated parameters, the challenge can be sent to the cloud.

3.3.2. Start Challenge

Cloud storage users put forward detection request. When cloud storage users need to detect the integrity of data files stored in cloud server, they will send a detection request to the trusted third party, and then the trusted third party will perform the data integrity detection instead of users.

Detection parameter analysis: after receiving the detection request from the user, the trusted third party checks the user rights at first, so as to judge whether the user has the read permission for the data file. If the applicant has the read permission, the trusted third party uses $pKey_u$ to decrypt the pre-processing parameter set sec to get $\left(n,m,k,l',q,ran(i,j)_{1\le i<k,1\le j<m}, len\left(a'_{ij}\right), pKey_u\right)$. The random generation algorithm $RanPos(k,m,r) \to \{(i_1,j_1),\cdots,(i_r,j_r)\}$ is performed. Where, k is the number of rows and m is the number of columns of matrix $\mathbf{A}'=\begin{bmatrix} \mathbf{a}'_{11} & \mathbf{L} & \mathbf{a}'_{1m} \\ \mathbf{M} & \mathbf{O} & \mathbf{M} \\ \mathbf{a}'_{k1} & \mathbf{L} & \mathbf{a}'_{km} \end{bmatrix}$. r denotes the number of data blocks a'_{ij} that the trusted third-party plans to detect. According to the detection intensity and detection environment, r can be determined by the trusted third party. If the trusted third party wants to conduct comprehensive data integrity detection, r can be bigger at this time. If the trusted third party wants to conduct periodic data integrity detection, the value of r can be moderate at this time. If the current network fluctuates greatly or the soft and hard environment is lack, the value of r can be reduced [14,15].

The output $I = [(i_1,j_1),\cdots,(i_r,j_r)]$ of algorithm $RanPos$ denotes the subscript of the selected detection data block. Then, the trusted third party uses the array I and algorithm $GuardPosition(i,j,l'pKey_u) \to \left(P_{i,j,1},\cdots,P_{i,j,w},\cdots,P_{i,j,l'}\right)$ to generate $r \times l'$ sentry positions of r data blocks $a'_{ij}(i,j) \in I$. Finally, the algorithm $P'_{i,j,w} = P_{i,j,w} + (w-1)\times q$ is used to calculate the real positions of l' sentries in data block $a'_{ij}(i,j) \in I$.

The generation of detection interval: the trusted third party uses the form of selecting data interval to determine the data range. That is to say, the data interval $[x_c,y_c], 1 \le c < r$ is selected from data block a'_{ij} for final detection of data integrity. The data block a'_{ij} corresponds to an interval, and the lengths of r intervals $[x_c,y_c], 1 \le c < r$ are the same, but their positions are different in corresponding a'_{ij} (see Figure 4).

The selection of interval length $y_c - x_c$ is similar to the selection condition of the number r of detection data block a'_{ij}, which can be determined by the trusted third party through the detection intensity and detection environment. Meanwhile, the minimum length of interval $[x_c,y_c]$ should be bigger than the length of $len\left(a'_{ij}\right)/l'$. Because $len\left(a'_{ij}\right) = len\left(a_{ij}\right) + q \times l'$, $len\left(a'_{ij}\right)/l'$ must be bigger than the position length q of sentry. That is to say, the detection interval range must be bigger than the length of sentry, so that the reliability of data integrity detection can be guaranteed, and thus avoiding the sentry leakage caused by multiple detection [16–18].

In order to ensure that the data interval $[x_c,y_c], 1 \le c < r$ contains the sentry information, the trusted third party randomly selects r random numbers $\{u_e\}, 1 \le e < r$ from the integer interval $[1,l']$, and then extracts the sentry position $P'_{i,j,e}$ of corresponding random number $\{u_e\}, 1 \le e < r$ from the sentry position set $\left(P_{i,j,1},\cdots,P_{i,j,w},\cdots,P_{i,j,l'}\right)$ of data block a'_{ij} corresponding to $I = [(i_1,j_1),\cdots,(i_r,j_r)]$. That is the aggregation $\left(a'_{i_1,j_1} : P'_{i_1,j_1,u_1},\cdots,a'_{i_w,j_w} : P'_{i_w,j_w,u_w},\cdots,a'_{i_r,j_r} : P'_{i_r,j_r,u_r}\right)$. See Figure 5:

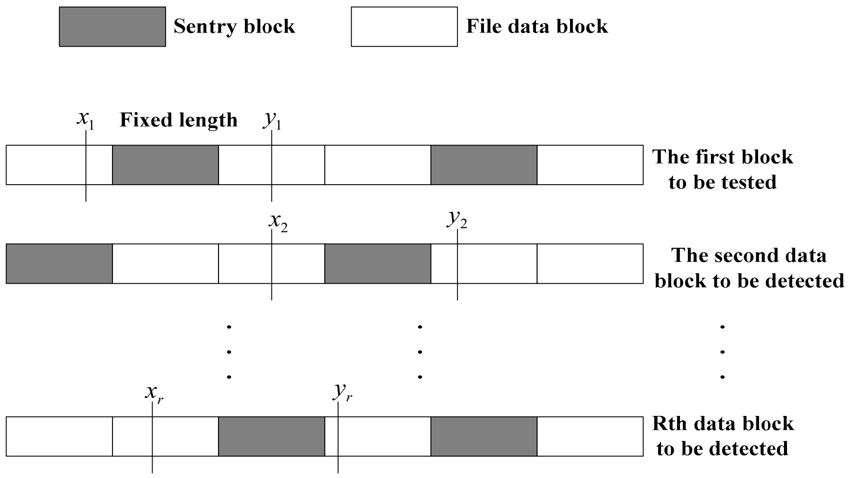
Sentry block
File data block
x_1
Fixed length
y_1
The first block to be tested
x_2
y_2
The second data block to be detected
x_r
y_r
Rth data block to be detected

Figure 4. *r* detection intervals.

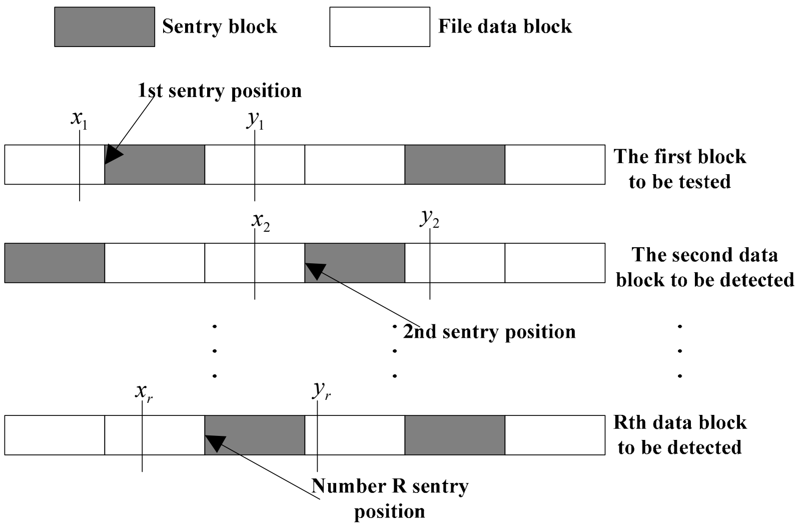

Figure 5. Sentry position.

The length of detection interval the trusted third party is set as γ. In order to put the sentry $a'_{i_w,j_w} : G_{i_w,j_w,u_w}$ corresponding to the set $\left(a'_{i_1,j_1} : P'_{i_1,j_1,u_1}, \cdots, a'_{i_w,j_w} : P'_{i_w,j_w,u_w}, \cdots, a'_{i_r,j_r} : P'_{i_r,j_r,u_r}\right)$ in the detection interval $[x_c, y_c], 1 \leq c < r$, the distance $dis = random(0, \delta) \times \gamma, 0 < \delta \leq 1$ between $a'_{i_w,j_w} : P'_{i_w,j_w,u_w}$ and the left vertex of interval is defined. δ denotes the basic threshold which is randomly generated at *dis*. If δ is close to 1, the probability that the length of *dis* is close to γ. $a'_{i_w,j_w} : P'_{i_w,j_w,u_w}$ is far away from the left vertex of interval with the increase of *dis*, and the probability that sentry in the interval is incomplete.

In the detection interval $[x_w, y_w]$ of data block a'_{i_w,j_w}, the sentry G_{i_w,j_w,u_w} is uncertain, so the number of sentries in an interval may be uncertain. A small number of sentries may only be included in some information [19,20]. The situation of sentry in interval is shown in Figure 6. (1) shows that sentry G_{i_w,j_w,u_w} is located in the interval $[x_w, y_w]$ completely, namely $x_w \leq P'_{i_w,j_w,u_w} < P'_{i_w,j_w,u_w} + q - 1 < y_w$. (2) shows that only partial data of sentry G_{i_w,j_w,u_w} is located in the interval $[x_w, y_w]$, namely $x_w < P'_{i_w,j_w,u_w} \leq y_w < P'_{i_w,j_w,u_w} + q - 1$. Due to the uncertainty of the length γ of interval, this interval may include several sentries of position w in addition to sentry G_{i_w,j_w,u_w}, which is shown in (3) and (4) of Figure 6.

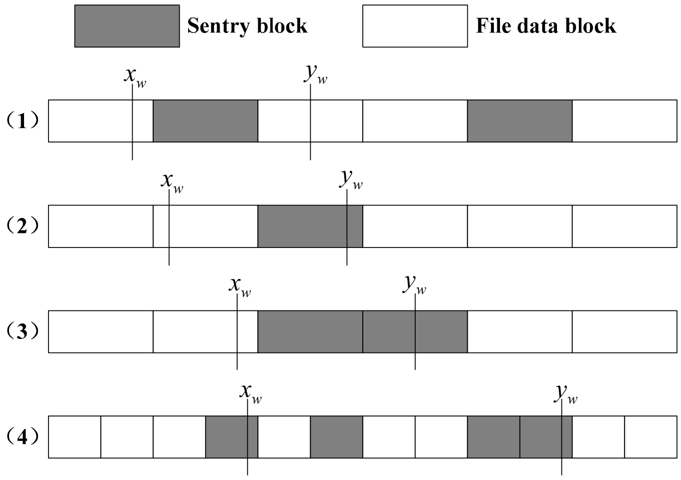

Figure 6. Detection of sentries within the interval.

Start challenge request: the trusted third party encrypts the array $I = [(i_1, j_1), \cdots, (i_r, j_r)]$ of the serial number of detected data block and the array $Q' = [(x_1, y_1), \cdots, (x_r, y_r)]$ which is composed of detection interval $[x_w, y_w]$ corresponding to serial number (i_w, j_w) through the cloud public key $sKey_c$ and then sends them to the remote server as the challenge request *chal*.

3.3.3. Detection and Verification Stage

Evidence generation: after receiving the challenge request *chal*, the server uses the private key $pKey_c$ to decrypt *chal*, then uses the decrypted I and Q' to perform the evidence generation algorithm $Gen\mathrm{Pr}oof(I, Q') \rightarrow \{i, j, \mathrm{Pr}o\}_{(i,j)\in I}$, so as to output the credentials Pr*o* which is used to submit the trusted third party for data integrity detection. Finally, the public key $sKey_t$ of trusted third party is used to encrypt Pr*o* and return it to the trusted third party.

Evidence verification: after receiving the detection credentials Pr*o* sent from the cloud, the trusted third party uses the evidence validation algorithm $Check\mathrm{Pr}oof\left(I, Q', P_{i,j,w}, q, pKey_u, \mathrm{Pr}o\right) \rightarrow \{''Su'', ''Fa''\}$ to judge whether the data file is complete. The detailed process is as follows: the trusted third party locates the θ sentry positions included in the detection interval $[x_w, y_w]$ according to the sentry position array $\left(P_{i_w,j_w,1}, \cdots, P_{i_w,j_w,z}, \cdots, P_{i_w,j_w,l'}\right)$ of data block a'_{i_w,j_w}, namely $x_w + 1 - q \leq \left\{P'_{i,j,m}\right\}_{1\leq m<\theta} < y_w$. Then, the trusted third party calculates the detailed information of sentry by the sentry generation algorithm $GuardGen\left(i, j, P_{i,j,w}, q, pKey_u\right) \rightarrow \left(G_{i,j,w}\right)$. By comparing with the corresponding position in detection evidence Pr*o*, we can see that only partial data appears in the sentry of detection area. If R' data segments in Pr*o* are compared, it means that the data has not been modified or damaged in this time. Otherwise, it means that the data stored in remote server is incomplete. Finally, the trusted third party submits the data integrity detection result to cloud storage user.

(4) Data security accountability based on trusted third party

With the popularization and development of cloud storage, people pay more and more attention to the security of cloud data. When the data stored in the cloud is illegally modified, neither the user nor the cloud can provide cogent credentials to divide the responsibility. Therefore, a data security accountability scheme based on trusted third party. This scheme takes the trusted third party as the core and carries out the online status judgment in each user operation, and then the credentials that cannot be denied by both parties are generated, so as to ensure the reliability of the audit and accountability of trusted third party when the cloud is not trusted.

A credible third-party accountability system needs to provide the following functions:

Any operation on cloud data is recorded in trusted third party and cloud;

When disputes occur, the trusted third party can accurately find the responsible party and determine the responsibility;

The certificate that is used to judge the responsible party has non repudiation.

Figure 7 is the framework of data security accountability based on trusted third party

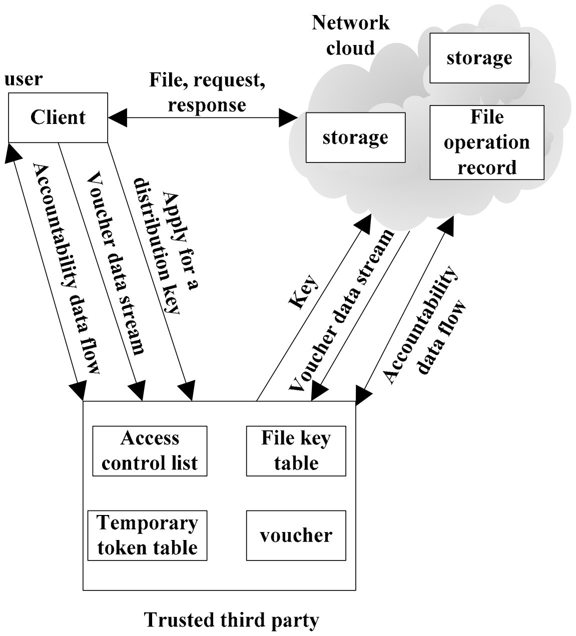

Figure 7. Framework of data security accountability scheme based on trusted third party.

In Figure 7, the user operates the cloud files through the browser or other clients. When the users log in to the cloud through the browser or other clients in each time, they can get a temporary token from the trusted third party. The file key version is formed after each user operation. When the user and the cloud operate the data, the accountability voucher will be generated and saved in the table of trusted third-party voucher. When the accountability is proposed or some disputes between the two parties occur, the trusted third party can judge the responsibility based on the clouding file operation records and local vouchers.

4. Results

In the process of verifying the performance of the network storage data integrity detection method based on symmetric difference, the experimental environment was 1 PC and the operating system was Windows 10. The installed memory (RAM) is 16 GB and the processor is AMD Ryzen Threadripper 2990WX@3.5GHz. The disk size is 250 GB and the disk read/write speed is 5400 RPM. Java language is adopted. In this experiment, the text data and image data are randomly crawled in the network storage database by the above method, so that the feasibility of the proposed method is verified by the precision rate of data integrity detection.

Figure 8 shows the precision rate of data integrity detection.

In Figure 8, the overall precision rate of the proposed method is high, showing good performance. This method achieves the feature analysis of text data by automatic word segmentation, part of speech tagging and Chinese word segmentation. Based on the symmetrical difference algorithm and the background subtraction, the feature extraction of image data is realized. Meanwhile, the preliminary recognition of data integrity is achieved, and thus to improve the precision rate of data integrity detection.

Three different methods are used to further verify the data integrity. In this paper, the availability of network storage data is verified, and the results are as follows.

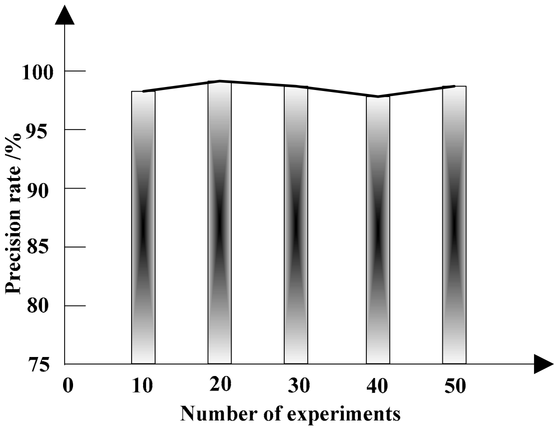

Figure 8. Precision rate of data integrity detection.

Analysis of the above Figure 9 shows that the availability of different methods is different under different data volumes. When the data volume is 5 GB, the data availability rate of literature [4] method is 72%, the data availability rate of literature [5] method is 82%, the data availability rate of this method is 94%, while the data availability rate of this method is significantly higher than the other two methods, and the data integrity is better.

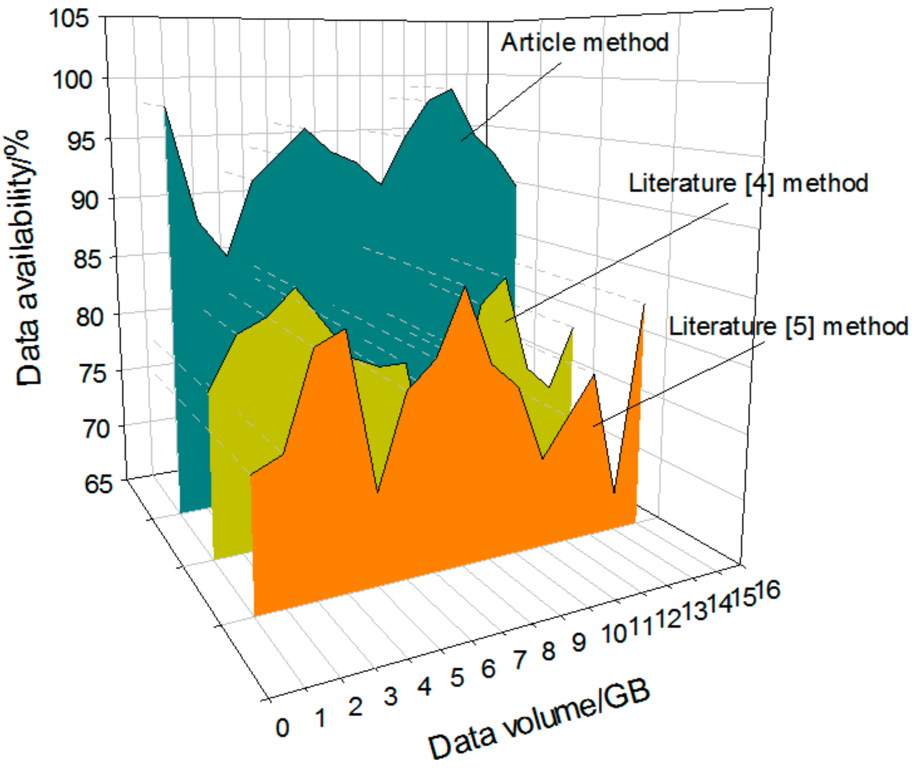

Figure 9. Storage data availability of different methods.

5. Discussion

The recall rate of data integrity detection is used as the discussion index to analyze the recall performance of the proposed method. The results are shown in Figure 10.

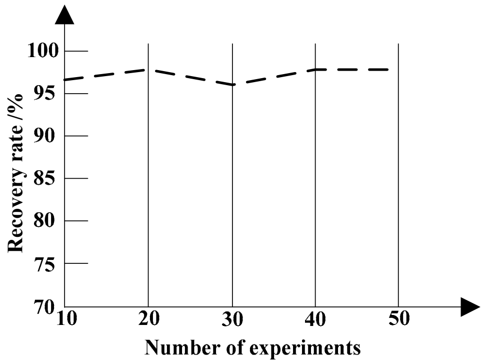

Figure 10. Recall rate of data integrity detection.

Figure 10 show that the proposed method uses the random sentry data segments to detect the data integrity. Because the sentry data segments not only contain the selected sentry information, but also contain the effective data and other sentry data, the scheme can support the infinite detection of detection party, and thus to improve the recall rate of data integrity detection.

In order to further verify the performance of this method, three different methods of coding time are used for detection, and the results are as follows.

Analysis of Figure 11 shows that different methods have different encoding times, that is to say, the shorter the encoding time, the higher the efficiency. When the amount of data stored is 14 MB, the coding time of document [4] method is 8 s, that of document [5] method is 12 s, and that of document [4] method is 5 s. The coding time of this method is the shortest and the efficiency is higher.

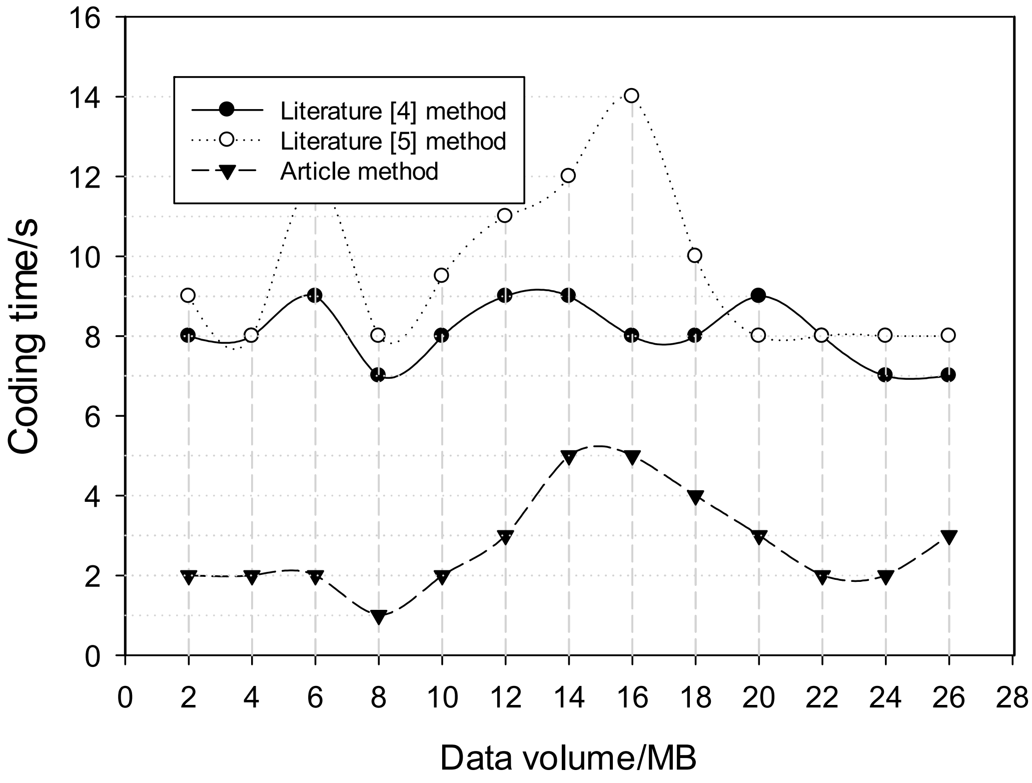

Figure 11. Coding time of different methods.

6. Conclusions

In order to solve the problem about network data security and integrity, a method to detect the network storage data integrity based on symmetric difference is put forward. According to data acquisition, feature analysis and extraction, the sentry data segment is introduced to detect the data integrity. The research results and work of this paper can be summarized as follows:

(1) Through in-depth study of data feature extraction and data integrity technology, this paper proposes a method of network storage data integrity detection based on symmetric difference. In this method, the integrity of the network storage data is detected according to the automatic image annotation system. For data recovery, this paper provides a strong anti-corruption ability by two rounds of coding for the original file. In the face of large area and high frequency file damage, it can still recover the original file with a high recovery rate, providing a high data availability. Every time the data recovery algorithm is called, this method is better.

(2) This paper presents an efficient data recovery algorithm. After locating the faulty storage nodes, it is necessary to recover the faulty data on these nodes. The algorithm ensures the recovery of data with low communication overhead and high recovery rate, and provides better security for network storage data.

(3) The integrity detection method proposed in this paper supports not only static data detection, but also dynamic data detection.

In this paper, the integrity detection method of network storage data based on symmetric difference is studied, and the phased research results are obtained. However, due to the limited time and energy and the criss-cross problems involved, the work done in this paper will inevitably be insufficient, which needs further improvement and improvement, and further research in the future. With the continuous growth of data volume, the development and expansion of trusted third party has become the next problem to be solved, and fault tolerance is also a very important research focus.

Author Contributions: This paper studies the integrity of network storage data based on symmetric difference. Based on a complete automatic image annotation system, use web crawler technology to capture images and related text information. Feature analysis of text data based on automatic word segmentation and part-of-speech tagging and Chinese word segmentation. It mainly uses symmetric difference algorithm, supplemented by background subtraction, to achieve image data feature extraction. With a trusted third party as the core, it conducts online status judgment for each user operation and generates credentials that both parties cannot deny, effectively preventing the verifier from providing a potential threat of false and falsified verification results. The experimental results show that the proposed method has higher precision and recall, and is more reliable. The methodology, concepts, experimental analysis, and so on of this article were completed by the author of this article, X.D., who read and approved the final manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Research Funds of Hyundai Heavy Industries. It was also supported by the Korea Institute of Energy Technology Evaluation and Planning (KETEP) and the Ministry of Trade, Industry, & Energy (MOTIE) of the Republic of Korea (No. 20172510102130).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lee, K.M.; Lee, K.M.; Lee, S.H. Remote data integrity check for remotely acquired and stored stream data. *J. Supercomput.* **2017**, *74*, 1182–1201. [CrossRef]
2. Han, W.J.; Yu, C.S. Distributed data storage architecture for data storage management of power transmission and transformation engineering. *J. Shenyang Univ. Technol.* **2019**, *41*, 366–371.
3. Fazzinga, B.; Flesca, S.; Furfaro, F.; Parisi, F. Exploiting Integrity Constraints for Cleaning Trajectories of RFID-Monitored Objects. *ACM Trans. Database Syst.* **2016**, *41*, 24. [CrossRef]
4. Liu, H.J.; Chen, Y.H.; Tian, H.; Wang, T.; Cai, Y.Q. Integrity-checking Security Data Aggregation Protocol. *Comput. Sci.* **2016**, *43*, 353–356.
5. Xu, G.W.; Bai, Y.K.; Yan, C.R.; Yang, Y.B.; Huang, Y.F. Check Algorithm of Data Integrity Verification Resultsin Big Data Storage. *J. Comput. Res. Dev.* **2017**, *54*, 2487–2496.
6. Li, H.Y.; Zhang, L.J.; Li, Q.P. Multiple-replica data integrity verification scheme in cloud storage environment. *China Sci. Pap.* **2017**, *12*, 98–102.
7. Tian, J.F.; Li, T.L. Data integrity verification based on model cloud federation of TPA. *J. Commun.* **2018**, *39*, 113–124.

8. Ma, L. Performance Analysis of Batch Processing for Big Data on Spark and Flink. *J. China Acad. Electron. Inf. Technol.* **2018**, *13*, 81–85, 103.
9. Rong, D.S.; Hu, J.S.; Zhao, J.J.; Yang, X.P. Low-ordercoal-to-methane production forecasting model based on data fusion IGA-RGRNN. *J. Power Supply* **2018**, *75*, 182–188.
10. Klymchuk, T. Regularizing algorithm for mixed matrix pencils. *Appl. Math. Nonlinear Sci.* **2017**, *2*, 123–130. [CrossRef]
11. Chen, H.; Jiang, J.; Cao, D.; Fan, X. Numerical investigation on global dynamics for nonlinear stochastic heat conduction via global random attractors theory. *Appl. Math. Nonlinear Sci.* **2018**, *3*, 175–186. [CrossRef]
12. Tan, X.F.; Xuan, T.T.; Zhang, P.C. Load identification and classification in Non-intrusive load monitoring system based on data stream. *Chin. J. Power Sources* **2016**, *40*, 1110–1112, 1141.
13. Sun, L. Research on efficient data mining algorithms for large-scale data sets. *Autom. Instrum.* **2016**, *3*, 192–193.
14. Fei, X.J.; Li, X.F. Wireless Sensor Network Data Fusion Algorithm Based on Compressed Sensing Theory. *J. Jilin Univ. (Sci. Ed.)* **2016**, *54*, 575–579.
15. Radhappa, H.; Pan, L.; Xi Zheng, J.; Wen, S. Practical overview of security issues in wireless sensor network applications. *Int. J. Comput. Appl.* **2018**, *40*, 202–213. [CrossRef]
16. Yang, Y.; Zhong, M.; Yao, H.; Yu, F.; Fu, X.; Postolache, O. Internet of Things for Smart Ports: Technologies and Challenges. *IEEE Instrum. Meas. Mag.* **2018**, *21*, 34–43. [CrossRef]
17. Wu, Z.G. Steady-State Network Massively Open Data Integrity and Efficient Detection Simulation. *Comput. Simul.* **2019**, *36*, 456–459.
18. Shen, W.T.; Yu, J.; Yang, G.Y.; Yang, G.Y.; Cheng, X.G.; Hao, R. Cloud Storage Integrity Checking Scheme with Private Key Recovery Capability. *J. Softw.* **2016**, *27*, 1451–1462.
19. Xu, X.F.; Chen, L. LZ77 power data compression algorithm based on integer wavelet transform. *J. Xi'an Polytech. Univ.* **2018**, *32*, 337–342.
20. Bardsiri, A.K. A new combinatorial framework for software services development effort estimation. *Int. J. Comput. Appl.* **2018**, *40*, 14–24.