J Inf Process Syst, Vol.15, No.6, pp.1265~1276, December 2019
https://doi.org/10.3745/JIPS.04.0143

ISSN 1976-913X (Print)
ISSN 2092-805X (Electronic)

JOURNAL OF INFORMATION PROCESSING SYSTEMS JIPS

Cloud Storage Security Deduplication Scheme Based on Dynamic Bloom Filter

Xi-ai Yan*, Wei-qi Shi*, and Hua Tian*

Abstract

Data deduplication is a common method to improve cloud storage efficiency and save network communication bandwidth, but it also brings a series of problems such as privacy disclosure and dictionary attacks. This paper proposes a secure deduplication scheme for cloud storage based on Bloom filter, and dynamically extends the standard Bloom filter. A public dynamic Bloom filter array (PDBFA) is constructed, which improves the efficiency of ownership proof, realizes the fast detection of duplicate data blocks and reduces the false positive rate of the system. In addition, in the process of file encryption and upload, the convergent key is encrypted twice, which can effectively prevent violent dictionary attacks. The experimental results show that the PDBFA scheme has the characteristics of low computational overhead and low false positive rate.

Keywords

Bloom Filter, Cloud Storage, Data Deduplication, Privacy Protection

1. Introduction

With the rapid development of cloud computing technology, more and more users choose to outsource data to cloud service providers. According to IDC analysis, cloud storage space is expected to reach 44 ZB [1] by 2020. How to effectively reduce storage space has become a hot research topic. Data deduplication technology provides an effective solution for minimizing storage capacity. It can automatically search for duplicate data, keep only one copy of all the same data, and make the actual data stored in the system decline geometrically.

In traditional storage systems, data are stored in plaintext. Users have full ownership and management rights. Deduplication technology of data is relatively mature, and more research results have been achieved, such as duplication detection of the same data and duplication detection of similar data. However, in cloud storage systems, the following challenges exist in data deduplication: (1) data is often stored in the ciphertext form, and different keys encrypt the same plaintext to generate different ciphertext; (2) data management and ownership are separated, users lose absolute control of data, and always worry about data security; (3) bandwidth resources and computing capital of cloud data are very valuable. How to detect duplicate data blocks efficiently and cost less is very important.

It uses virtualization technology to achieve on-demand allocation of multi-users storage resources in

Manuscript received March 8, 2019; first revision July 26, 2019; accepted September 11, 2019.
Corresponding Author: Xi-ai Yan (yanxiai222@163.com)
* Dept. of Information Technology, Hunan Police Academy, Changsha, China ({yanxiai222, zhangf783, liuj197807}@163.com)

cloud computing. When cloud servers detect data duplication among multi-users, it is easy to cause user privacy disclosure. How to realize data deduplication and protect user privacy is an urgent problem, in this paper, it proposed a security deduplication scheme based on Bloom filter. A concise global Bloom filter array is constructed to realize data ownership proof and duplication comparison. The Bloom filter is dynamically extended to effectively alleviate the problem that the false positive rate of Bloom filter increases too fast.

2. Related Research

Research on data security deduplication in cloud environment has achieved certain results. Douceur et al. [2] first proposed convergent encryption (CE) technology, which is an encryption technology using hash value of data as the key. CE can effectively protect data confidentiality while deleting duplicate data. Li et al. [3] combined secret sharing AONT-RS with CE, adopted convergent diffusion mechanism and proposed a CDStore scheme, which replaced random information in diffusion algorithm with hash fingerprint of data, and ensured the certainty of the algorithm to achieve data security and deduplication. Stanek et al. [4] combined with CE, proposed a scheme to provide different levels of security encryption for data according to different levels of privacy. In view of the security and privacy problems faced by cloud data security deduplication schemes, Puzio et al. [5] proposed a secure and effective storage system, which provides data block level deduplication and confidentiality. On the basis of CE, additional semantic security encryption schemes and access control mechanisms are added to resist the existing CE attacks.

In view of the attack that the opponent obtains the fingerprint information of the user's files and obtains the complete files from the server by using the client's deduplication mechanism, scholars put forward the concept of ownership certification. Halevi et al. [6] first introduced the concept of "proof of ownership" (PoW), which can be verified only when the user has complete data, but can't be validated only if he has partial data. Xu et al. [7] improved Halevi's scheme, first encrypted the original file, then constructed MHT for PoW verification, and proved the scheme to be secure under the random oracle model. In order to reduce the computational overhead and the number of I/O reads and writes, Di Pietro and Sorniotti [8] proposed a PoW optimization scheme (s-PoW) to achieve efficient and information-theoretically secure document ownership certification. In order to further improve the efficiency of server-side in existing PoW schemes, Blasco et al. [9] proposed a flexible, scalable and provably secure cloud data deduplication scheme based on Bloom filter. By comparing the validation information of file blocks with Bloom filter, we can judge whether the user has data or not, which can greatly reduce the overhead of client and server-side. However, data security issue is not discussed in [9]. [10] combines CE with Bloom filter to ensure the security of experimental data, but adopts standard Bloom filter, without discussing the false positive rate of Bloom filter. In Bloom filter research, [11] proposes a high-precision multi-dimensional Bloom filter, which uses bijective function to transform multi-dimensional attributes of elements into one-dimensional values to represent the overall information of elements, which can effectively reduce the misjudgment rate. [12] propose a privacy-preserving reversible Bloom filter. With the help of homomorphic encryption functions, message aggregation can be accomplished by simple multiplication and addition of ciphertext. [13] proposes an efficient grouping classification algorithm based on counting Bloom filter, which applies counting Bloom filters to predict the failures of flow table lookups without traversing flow tables for most mask probing. [14] proposes a ciphertext retrieval

ranking method based on counting Bloom filter (CRRM-CBF) constructs secure retrieval index through Bloom filter to achieve efficient retrieval of ciphertext data and dynamic update of index. [15] proposes a homomorphic encryption protocol based on Bloom filter, which improves the execution efficiency of the protocol by virtue of the advantages of low computational complexity, high spatial utilization and high query efficiency of Bloom filter.

3. Preparatory Knowledge

3.1 Standard Bloom filter

Bloom filter is a spatially efficient randomized data structure for the simplified representation of element sets and membership queries. Bloom filter is widely used in network packet classification, IP routing lookup, deep packet detection, and so on [16].

Standard Bloom filters are m-bit vectors V representing the aggregate information of queries. Assuming that the set of elements is $S=\{s_1,s_2...s_n\}$, K hash functions $h_1(x),h_2(x)...h_k(x)$ corresponding to each element, the range of values of the hash functions is $[0,m-1]$. The initial state of the vector V is all 0. When the element is stored, the hash function $h_1(x),h_2(x)...h_k(x)$ of the V is set to be all 1. When the element is queried, whether the search bits of $h_1(x),h_2(x)...h_k(x)$ are all 1. If all the bits are 1, the element is in the set. Otherwise, the element is not in the set [17].

Example: Let's assume that the length of the vector is $m=6bit$, the number of hash functions is $k=2$, $h_1(x)=x \bmod 6$, $h_2(x)=(2x+1) \bmod 6$, respectively, $S=\{13,14\}$,whether the query element $\{12,19\}$ is in the set or not. The results of insertion and query are shown in Fig. 1.

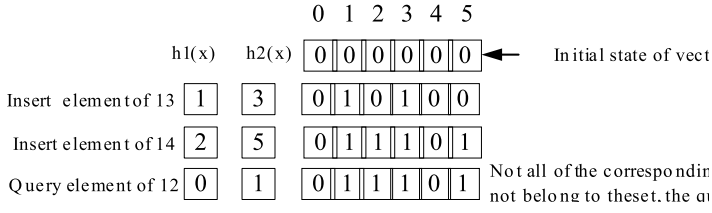

Fig. 1. Insertion and query of standard Bloom filter.

Bloom filters may judge elements that do not belong to this set as elements belonging to this set. Let single Bloom filter false positive rate $FPR(n,m,k)$, where n denotes the number of elements in set S, m denotes the number of filters, K denotes the number of hash functions.

The probability of one bit being zero in the filter is $(1-\frac{1}{m})^{kn}$, the probability of one bit being 1 is $1-(1-\frac{1}{m})^{kn}$, if the hash function is uniformly distributed, and then $FPR(n,m,k)$ can be expressed as:

$$FPR(n,m,k)=(1-(1-\frac{1}{m})^{kn})^k \approx (1-e^{-kn/m})^k=\exp(k\ln(1-e^{-kn/m})) \quad (1)$$

Let the number of Bloom filters is *r*, and use $GFPR(n,m,k,r)$ to express the false positive rate of global Bloom filters:

$$GFPR(n,m,k,r) = 1-(1-FPR(n,m,k))^r \tag{2}$$

Formula (1) can be changed into:

$$n = -\frac{\ln(1-e^{\ln FPR/k})m}{k} \tag{3}$$

Formula (1) shows that the false positive rate of the filter increases with the increase of the number of elements in the set. If the upper limit of the false positive rate of a single Bloom filter is f and the size of the elements in the set is n_1, it can be seen from formula (2) that if the false positive rate is within the controllable range, it must:

$$n_1 \le -\frac{\ln(1-e^{\ln f/k})m}{k} \tag{4}$$

3.2 Convergent Encryption

To solve the problem that the traditional random encryption algorithm can't detect repeatability and store multiple copies of the same file in the cloud at the same time, which seriously wastes storage space, Douceur et al. [2] proposed CE algorithm. The key generation algorithm is a deterministic algorithm, which is usually obtained by hashing the original data to ensure that the same data gets the same key.

The basic cryptographic primitives based on CE algorithm are as follows:

(1) Key generation： $K = H(F)$, H is Hash function;

(2) Encryption： $C = E_{H(F)}(F)$, the client encrypts the file;

(3) Decryption： $F = D_{H(F)}(C)$, users get ciphertext from the server and decrypt it;

(4) Label generation： $id = GEN(F)$, generating file or data block labels.

The ciphertext verifiability of CE algorithm makes it widely used in the field of cloud data deduplication. Many research results combine CE with various mechanisms to achieve deduplication of ciphertext data. CE is widely used to construct secure data deduplication system, but it also faces many dangers such as data leakage, copy forgery attack, choice plaintext attack, and content distribution network attack, and so on [18,19].

4. System Model

The cloud storage security deduplication model based on dynamic Bloom filter (DBF) is shown in Fig. 2. The main entities include users, cloud service providers (CSP) and third-party servers (TPS).

The users divides file F into blocks of the same size, performs CE, and generates ciphertext blocks and block labels. A public dynamic Bloom filter is deployed on the TPS to realize user privilege detection and data block repeatability detection. According to the test results, it decides whether to issue the privilege label, upload data or return a storage pointer to the user. Another function of TPS is to achieve secondary

encryption of convergent keys and effectively prevent violent dictionary attacks. CSP clothing is responsible for storing data blocks and block labels, and accepting user uploads and downloads of data blocks according to the privilege labels.

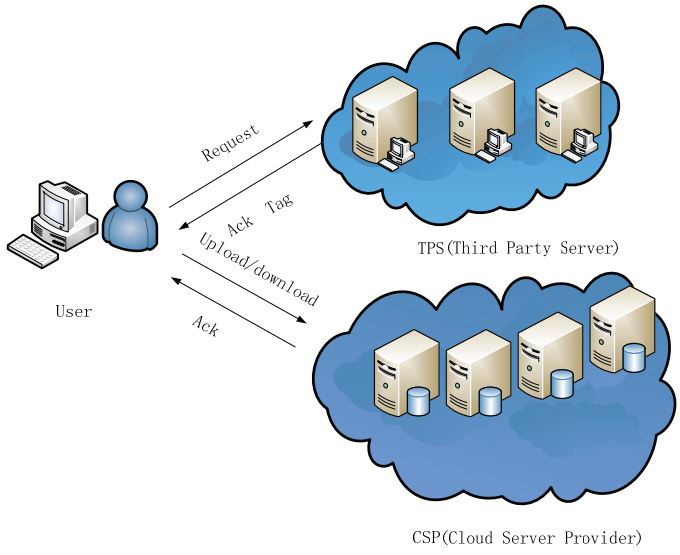

Fig. 2. Cloud storage security deduplication model based on dynamic Bloom filter.

5. System Design and Implementation

5.1 Design of Dynamic Bloom Filter

In the design of Bloom filter, firstly a Bloom filter is constructed for permission query to achieve identity authentication, and then a Bloom filter is constructed for each storage node to achieve data deduplication. All Bloom filters are isomorphic and form a public Bloom filter array (PBFA) globally. Bloom filter array can achieve public weight removal, not limited to a single storage node. The PBFA is shown in Fig. 3.

	Bit 1										Bit m		
BF_1	0	1	0	0	0	1	0	0	0		1	0	0
BF_2	1	0	0	1	0	0	1	0	1		0	1	0
BF_3	0	0	1	0	1	0	0	0	1	...	1	0	0
	...	...	...	...	...	...	...	...	...		...	...	...
BF_{r-1}	1	0	1	0	0	1	0	1	0		0	1	1
BF_r	0	0	0	0	1	0	1	0	0		1	0	0

Fig. 3. A public Bloom filter array.

In order to slow down the growth of false alarm rate, a DBF is designed for each storage node. DBF is dynamically composed of several standard sub-filters according to the size of the set elements, $DBF = \{DBF_1, DBF_2, DBF_3, \ldots DBF_{t-1}, DBF_t\}$, a PDBFA is constructed globally.

The DBF creation algorithm is as follows:

Step 1: Initialize the standard Bloom filter, set the upper limit f and bit m of the false positive rate of the filter, and calculate $n_1$according to formula (3).

Step 2: Create the first Bloom filter vector DBF_1.

Step 3: if $\leq n_1$, map elements to DBF_1.

Step 4: if $n > n_1$, create DBF_2 (isomorphic toDBF_1), map elements to DBF_2 .

Step 5: if $n > 2n_1$, create DBF_3 (isomorphic toDBF_1), map elements to DBF_3.

Step 6: if $n > (t-1)n_1$, where$t = \lfloor n/n_1 \rfloor + 1$, create DBF_t (isomorphic toDBF_1), map elements to DBF_t.

The maximum number of elements that DBF can represent in a controllable range is *n*:

$$n = -\frac{ln(1-e^{ln\, f/k})tm}{k} = tn_1 \tag{5}$$

$DFPR(n, n_1 m, k)$denotes false positive rate of DBF , the number of last filter elements is $i = n - n_1 \times \lfloor n/n_1 \rfloor$, According to formula (1), $DFPR(n, n_1 m, k)$ can be expressed as:

$$\begin{aligned} DFPR(n, n_1 m, k) &= 1 - \left(1 - FPR(n_1, m, k)\right)^{\lfloor n/n_1 \rfloor} \left(1 - FPR(i, m, k)\right) \\ &= 1 - \left(1 - \left(1 - e^{-kn_1/m}\right)^k\right)^{\lfloor n/n_1 \rfloor} \left(1 - \left(1 - e^{-ki/m}\right)^k\right) \end{aligned} \tag{6}$$

$GDFPR(n, n_1 m, k, r)$ denotes false positive rate of DBFA:

$$GDFPR(n, n_1 m, k, r) = 1 - (1 - DFPR(n, n_1 m, k))^r \tag{7}$$

Assuming k = 4, m = 6400, the upper limit of false positive rate of single standard sub-filter is f = 0.01, it can be obtained by formula (3):

$$n1 = -\frac{\ln(1-e^{\ln(0.01)/4}) \times 6400}{4} = 405.4724$$

The number of elements that a single sub-filter can tolerate most is 405.

According to formulas (2) and (7), the false positive rate of PBFA and PDBFA increases with the increase of elements. Assuming r = 10, the change of false positive rate is shown in Fig. 4. From Fig. 4, PDBFA can dynamically build new sub-filters, which can effectively control the growth of false positive rate at the expense of a certain storage space.

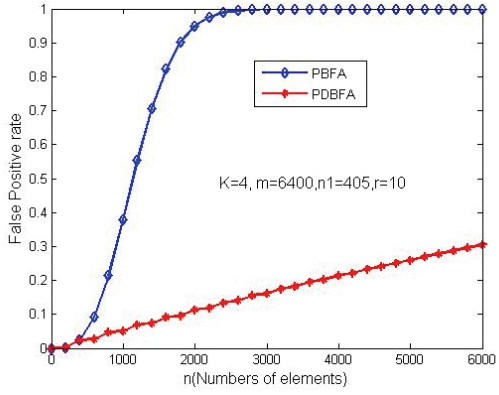

Fig. 4. Theoretical comparison of false positive rates between PBFA and PDBFA.

5.2 Implementation of Safe Deduplication Scheme

The first section is system initialization as follows:

(1) Data preprocessing: dividing the file into blocks: $F=(F_1,F_2,F_3...F_i)$, initializing four algorithms of CE scheme.

(2) Privilege initialization: System privileges are entirely S, suppose the system has N users, each user's privileges is S_U , the access privilege is S_{F_i} to data block F_i .

(3) Bloom filter initialization: Mapping access right of data block F_i to the first Bloom filter through K hash functions, and mapping data blocks of each storage node to corresponding Bloom filters through K hash functions.

The second section is file upload as follows:

Step 1: Authentication request, user asks TSP for query S_U in Bloom filters.

Step 2: Bloom filters queries and updates. Whether the corresponding K bit of S_U is all 1, if all 1, the query passes; otherwise, the user is required to prove the ownership of the data, if prove ok, the corresponding K bit is set to 1 by updating the Bloom filter.

Step 3: Distribution of permission labels. After the permission query is passed, the corresponding permission key K_{S_i} is selected, the permission label $\phi_i=id\oplus K_{S_i}$ is generated and distributed to the user.

Step 4: Repetitive data block detection. User uses ϕ_i for repetitive detection through row manner in PDBFA. When duplication is found, a storage pointer is issued to user, and corresponding permission is allocated, so data is no longer uploaded to the cloud server.

Step 5: Non-duplicate data processing. When the user access all data in PDBFA and no duplicate block are found, it proves to be a new data block.

Step 6: Data block encryption upload. User randomly select $r\in Z_q$ as the blinding factor, and send the convergent key K_i to TPS after blinding process, TPS encrypts the blinded convergent key twice to obtain $K_i^{'}$. User process the blinded key $K_i^{'}$ to obtain $K_i^{''}$, and upload the ciphertext $C_i=E_{K_i^{''}}(F_i)$ to CSP.

The third section is file download as follows:

Step 1: Users request to download data block F_i, send data block ID and corresponding permissions S_U to TPS.

Step 2: TPS queries the DBF for access rights. If not, the access fails. If yes, a privilege label ϕ_i is issued to the user, and then the user will send it to CSP.

Step 3: CSP will send the corresponding ciphertext block C_i and $K_i^{'}$ to the user, the user will obtain $K_i^{''}$ after de-blinding process of $K_i^{'}$, decrypt $F_i=D_{K_i^{''}}(C_i)$.

6. Analysis of Experimental Performance

6.1 Experimental Environment

Datasets are collected from various portals by using web downloaders. The selected websites are Sina, Tencent, and Netease. The file formats are web pages, text, pictures, music, animation and video. The total number of files in the dataset is 562352, with a total size of 725.78 GB. The hardware environment of the experiment is shown in Table 1.

Table 1. Hardware configuration of experimental environment

Identity	CPU	Hard disk	Memory	Network bandwidth
User	Core i5, 3.3 GHz	WD4T	DDR3, 8G	100 M
TPS	Core i7, 4.0 GHz	WD 6T	DDR3, 8G	100 M
CSP	Core i5, 3.3 GHz	WD 4T	DDR3, 8G	100 M

In the experiment, the SHA-1 algorithm is used to calculate the data block label, and the SHA-2 algorithm is used to calculate the convergence key of the data block. The capacity of single Bloom filter is m = 65536 bit, and the number of hash functions is K = 6.

6.2 Upload Time Overhead of Data Block

Baseline scheme proposed in [20] is a classical scheme for cloud storage security deduplication. Baseline scheme first calculates the file tag and blocks, then calculates the block convergence key to encrypt the block, calculates the block ciphertext hash value to prove ownership, and finally uses the private key to encrypt the convergence key and upload it to the storage server. This experiment compares PDBFA scheme with Baseline scheme, and compares the time cost of file upload from three aspects: no duplication, duplication, and partial duplication.

When there is no duplication of data blocks, the complexity of computing hash value increases with the increase of data blocks, and the upload time of data blocks also increases. In PDBFA, it takes a part of time to initialize the Bloom filter, so there is little difference between them. But because PDBFA uses DBF to prove ownership, the overall time overhead is slightly better, and the comparison results are shown in Fig. 5.

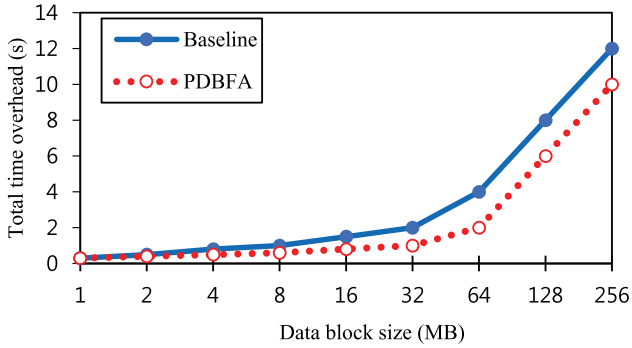

Fig. 5. Comparisons of upload time overhead for data blocks without duplication.

When data blocks are duplicated, baseline scheme needs to calculate block ciphertext hash value for ownership certification in the global scope. In PDBFA scheme, because of the use of DBF, users can only query within the scope of data with access rights efficiently, which reduces the time of authorization verification. With the increase of data blocks, the advantages of Baseline scheme are more obvious, and the results are compared as shown in Fig. 6.

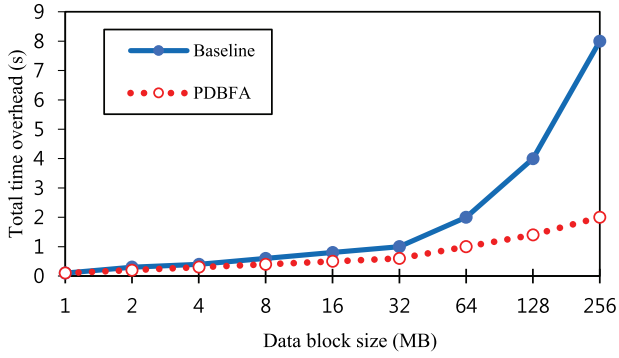

Fig. 6. Comparisons of upload time overhead when data blocks are fully duplicated.

When the repeatability of data blocks is different, the fixed size of data blocks is 100 MB. The data blocks are compared at different repeatability rates of 0%, 25%, 50%, 75%, and 100%, respectively. As the repetition rate increases, the encryption operation decreases, so the overall upload time overhead decreases. The key encryption time of baseline scheme is fixed. But in PDBFA scheme, the number of convergent key encryption decreases greatly due to the increasing of repetition rate. Compared with baseline scheme, the time overhead advantage is obvious. The comparison results are shown in Fig. 7.

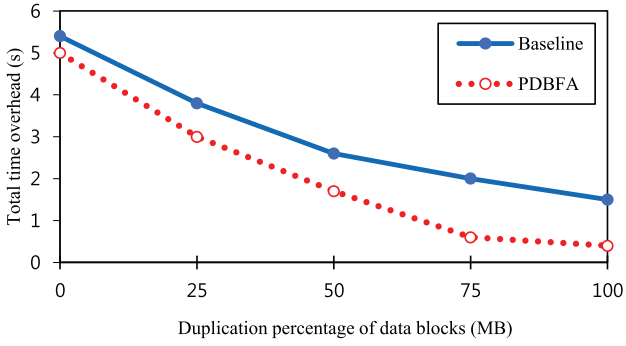

Fig. 7. Comparisons of data block upload time overhead at different repetition rates.

6.3 False Positive Rate Analysis

Set the bit of standard Bloom filter is $m = 65536$, the upper limit of false positive rate of standard filter is $f = 0.001$, according to formula (3), obtained $n_1 = 4152$. The comparison of false positive rate between PBFA and PDBFA is shown in Fig. 8 when the number of fingerprints in coding block is different.

The experimental results show that when the number of data blocks $n < 4152$, the two mechanisms are the same, and similar false positive rate can be obtained. When $n = 2n_1$, the false positive rate of PBFA

was about 8 times that of PDBFA. The reason was that PDBFA expanded the capacity of Bloom filter and controlled the actual false positive rate of single Bloom filter within the effective threshold. Subsequently, as the number of data blocks continues to grow, the false positive rate difference between them further enlarges.

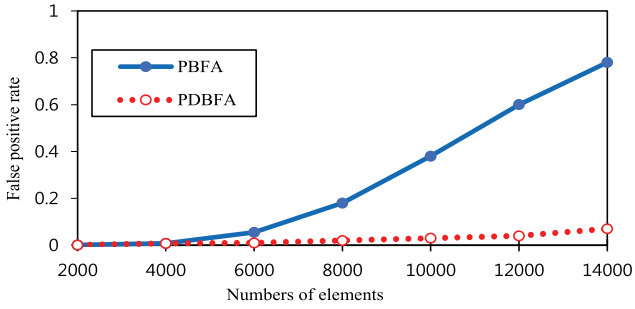

Fig. 8. Experimental comparison of false positive rates between PBFA and PBFA.

7. Summary

Data security deduplication in cloud environment is a very active research direction. At present, it is still in its infancy, and there is still a big gap between key technologies and practical applications. Bloom filter uses a very compact form to represent information. Hash lookup time is constant, storage space is low, and cache is quite low. It is especially suitable for the search and recognition of feature characters in massive data sets.

Aiming at the privacy protection and fast deduplication of cloud storage, this paper proposes a secure deduplication scheme of cloud storage based on DBF, which improves the efficiency of ownership certification and realizes fast detection of duplicate data blocks. In addition, the convergent key is encrypted twice by means of blind processing method, which can effectively resist violent dictionary attacks. In the scheme, a scalable DBF is designed to control the false positive rate within a certain threshold. The experimental results show that the computational overhead of file upload is relatively small, and it can effectively control the false positive rate of the system.

Compared with previous studies, the contribution of this paper is to design dynamic global Bloom filter array to complete de-duplication and proof of ownership, which greatly reduces the cost of query and proof time for duplicate data blocks. The disadvantage is that the sequence structure and time structure of the stream data are not designed in Bloom filter, and the dynamic stream data related to time cannot be processed directly. The future plan is to study the time dimension expansion of Bloom filter, reduce the storage space and computing overhead of stream data processing, and provide efficient security protection for stream data.

Acknowledgement

This paper is supported by Hunan Science and Technology Major Special Subsidy Project (No. 2017SK1040), Hunan Education Science 13th Five-Year Plan Subsidy Project (No. XJK18CXX014), and Hunan Provincial Education Department Innovation Platform Open Fund Project (No. 18K110).

References

[1] Cisco Global cloud index [Online]. Available: https://www.cisco.com/c/dam/assets/sol/sp/gci/global-cloud-index-infographic.html.

[2] J. R. Douceur, A. Adya, W. J. Bolosky, P. Simon, and M. Theimer, "Reclaiming space from duplicate files in a serverless distributed file system," in *Proceedings 22nd International Conference on Distributed Computing Systems*, Vienna, Austria, 2002, pp. 617-624.

[3] M. Li, C. Qin, and P. P. Lee, "CDStore: toward reliable, secure, and cost-efficient cloud storage via convergent dispersal," in *Proceedings of 2015 USENIX Annual Technical Conference*, Santa Clara, CA, 2015, pp. 111-124.

[4] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl, "A secure data deduplication scheme for cloud storage," in *Financial Cryptography and Data Security*. Heidelberg: Springer, 2014, pp. 99-118.

[5] P. Puzio, R. Molva, M. Onen, and S. Loureiro, "PerfectDedup: secure data deduplication," in *Data Privacy Management, and Security Assurance*. Cham: Springer, 2015, pp. 150-166.

[6] S. Halevi, D. Harnik, B. Pinkas, and A. Shulman-Peleg, "Proofs of ownership in remote storage systems," in *Proceedings of the 18th ACM Conference on Computer and Communications Security*, Chicago, IL, 2011, pp. 491-500.

[7] J. Xu, E. C. Chang, and J. Zhou, "Weak leakage-resilient client-side deduplication of encrypted data in cloud storage," in *Proceedings of the 8th ACM SIGSAC Symposium on Information, Computer and Communications Security*, Hangzhou, China, 2013, pp. 195-206.

[8] R. Di Pietro and A. Sorniotti, "Boosting efficiency and security in proof of ownership for deduplication," in *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security*, Seoul, Korea, 2012, pp. 81-82.

[9] J. Blasco, R. Di Pietro, A. Orfila, and A. Sorniotti, "A tunable proof of ownership scheme for deduplication using bloom filters," in *Proceedings of 2014 IEEE Conference on Communications and Network Security*, San Francisco, CA, 2014, pp. 481-489.

[10] Z. Liu and Z. Yang, "Efficient and secure deduplication cloud storage scheme based on proof of ownership by Bloom filter," *Journal of Computer Applications*, vol. 37, no. 3, pp. 766-770, 2017.

[11] W. Li, D. F. Zhang, K. Huang, and K. Xie, "Accurate multi-dimension counting Bloom filter for big data processing," *Chinese Journal of Electronics*, vol. 43, no. 4, pp. 652-657, 2015.

[12] K. Xie and W. Shi, "PPIBF: a privacy preservation invertible Bloom filter," *Computer Engineering & Science*, vol. 39, no. 6, pp. 1104-1111, 2017.

[13] J. Zhao, Z. Hu, B. Xiong, and K. Li, "Accelerating packet classification with counting bloom filters for virtual openflow switching," *China Communications*, vol. 15, no. 10, pp. 117-128, 2018.

[14] Y. Li and Z. Xiang, "Ciphertext retrieval ranking method based on counting Bloom filter," *Journal of Computer Applications*, vol. 38, no. 9, pp. 2554-2559, 2018

[15] E. Zhang and G. Jin, "Cloud outsourcing multiparty private set intersection protocol based on homomorphic encryption and Bloom filter," *Journal of Computer Applications*, vol. 38, no. 8, pp. 2256-2260, 2018.

[16] S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor, "Longest prefix matching using bloom filters," *IEEE/ACM Transactions on Networking*, vol. 14, no. 2, pp. 397-409, 2006.

[17] C. M. Tseng, J. R. Ciou, and T. J. Liu, "A cluster-based data de-duplication technology," in *Proceedings of the 2nd International Symposium on Computing and Networking*, Shizuoka, Japan, 2014, pp. 226-230.

[18] J. Xiong, Y. Zhang, F. Li, S. Li, J. Ren, and Z. Yao, "Research progress on secure data deduplication in cloud," *Journal on Communications*, vol. 37, no. 11, pp. 169-180, 2016.

[19] P. Anderson and L. Zhang, "Fast and secure laptop backups with encrypted de-duplication," in *Proceedings of the 24th Large Installation System Administration (LISA)*, San Jose, CA, 2010.

[20] L. Gonzalez-Manzano and A. Orfila, "An efficient confidentiality-preserving proof of ownership for deduplication," *Journal of Network and Computer Applications*, vol. 50, pp. 49-59, 2015.

Xi-ai Yan https://orcid.org/0000-0003-2338-9234

He received his M.S. and Ph.D. degrees in Computer Science from Hunan University, China, in 2007 and 2016, respectively. Now he is professor of Computer Science in Hunan Police Academy, Changsha, China. His research interests include cloud storage security and information retrieval.

Wei-qi Shi https://orcid.org/0000-0002-8649-5539

He received M.S. degree in Computer Science from Hunan University, China, in 2005. Now he is professor of Computer Science in Hunan Police Academy, Changsha, China. His current research interests include pattern recognition and information security.

Hua Tian https://orcid.org/0000-0003-2095-8120

She received B.Sc. degree from Hunan Normal University, China, in 1998. Now she is associate professor in Hunan Police Academy, Changsha, China. Her current research interests include fault-tolerance computing and information security.