A Cross-Domain Secure Deduplication Scheme Based on Threshold Blind Signature

Jinlei Du[1,2], Jide Deng[1,2], Hui Qi[1,2(✉)], Xiaoqiang Di[1,2], and Zhengang Jiang[2]

[1] Jilin Key Laboratory of Network and Information Security, Changchun University of Science and Technology, Changchun, China
[2] School of Computer Science and Technology, Changchun University of Science and Technology, Changchun, China
qihui@cust.edu.cn

Abstract. In the cloud storage environment, client-side deduplication can perform file repetitive detection locally. However, client-side deduplication still faces many security challenges. First, if the file hash value is used as evidence for repetitive detection, the attacker is likely to obtain the entire file information through the hash value of the file. Secondly, in order to protect data privacy, convergence encryption is widely used in the data deduplication scheme. Since the data itself is predictable, convergence encryption is still vulnerable to brute force attacks. In order to solve the above problems, this paper proposes to construct a secure deduplication scheme by using the threshold blind signature method. The generation of the convergence key is coordinated by multiple key servers, ensuring the confidentiality of the convergence key and effectively solving the violent dictionary attack problem. At the same time, since the key center is introduced to centrally manage the keys, the interaction between the key servers is reduced, and the key generation efficiency is improved. In addition, since the key server in this paper can be distributed in multiple independent network domains and interact with the key center through the Internet, the problem of cross-domain deduplication is solved. The experimental results show that the performance of this scheme is greatly improved in terms of system initialization and key generation.

Keywords: Secure deduplication · Encryption · Threshold blind signature · Cross-domain secure deduplication

1 Introduction

Deduplication technology is a special data reduction technology [20,29], which is used to eliminate redundant data and save network bandwidth and storage space in cloud storage systems [12,21]. This technique identifies common data blocks or files, storing only a single instance [26], and duplicate data is replaced with logical pointers [12,24]. Deduplication technology meets the growing demand

W. Li and D. Tang (Eds.): MOBILWARE 2020, LNICST 331, pp. 78–91, 2020.
https://doi.org/10.1007/978-3-030-62205-3_7

for storage capacity. Many cloud storage providers, such as Amazon S3, Bitcasa and Microsoft Azure, are using deduplication technology [30] to improve storage efficiency. There are four deduplication strategies, depending on whether deduplication occurs on the client (before upload) or on the server, and whether deduplication occurs at the block or file level. Client-side deduplication is more advantageous than server-side deduplication because it can ensure that multiple uploads of the same content consume only one upload of network bandwidth and storage space.

Data deduplication technology is widely adopted by cloud service providers. The cloud server verifies whether the data uploaded by the user has been stored by random sampling and extraction of hash values. After verification, if the newly uploaded data of the user is the same as the original stored data, the data deduplication is performed [31]. Experimental research shows that data deduplication will save more than half of the storage space, and the deduplication rate will reach 90%–95% [19,33].

Convergent encryption, in which the encryption key is derived from the plaintext, may be a simple and secure solution allowing deduplication. Unfortunately, it was proved unsafe [11]. In addition, a general impossibility result holds stating that classic semantic security is not achievable for schemes implementing plain convergence encryption [13].

Miao et al. proposed a security deduplication scheme based on threshold blind signature [23], which uses n key management nodes to blindly sign H(M) at the same time. The client obtains the encryption key by combining t ($t \leq n$) signatures, thereby solving the single point failure problem and resisting the collusion attack. However, in the initialization process, a large amount of information needs to be exchanged between the key management nodes, and each signature needs to be verified in the process of generating a key, so communication overhead and computational overhead are large. In addition, the solution does not support application scenarios across network domains.

In this paper, we propose a data security deduplication scheme that supports cross-domains. Based on the idea of threshold blind signature, a multi-server-assisted architecture is constructed. The key management nodes are divided into a master key nodes and the multiple sub-key nodes. The digital signature is completed by multiple parties, which improves the difficulty of signature cracking and ensures the security of the key. Communication and computational overhead is reduced by reducing the interaction between key nodes and reducing the number of signatures. In addition, cross-domain scenarios are also supported, and the practical application prospects are greater.

1.1 Contribution of This Article

Our main contributions are summarized as follows:

(1) Support cross-domain application environments. Key nodes can be distributed in separate network domains, and users can still perform deduplication of encrypted data when roaming between network domains. When a user in one domain moves to another domain, the user can recalculate the new key

through the key nodes in the new domain, and perform secure deduplication of the encrypted data.

(2) Reduce the initialization overhead of the key nodes. Miao et al.'s threshold blind signature scheme requires each key node to communicate with other nodes during initialization, and the number of communications is $n(n-1)$. In the scheme of this paper, the master key node interacts with each sub-key node once, the number of communications is n, and each sub-key node does not need to interact with other sub-key nodes.

(3) Reduce the computational overhead of signature verification. In the scheme of Miao et al., the blind signature generated by each key node needs to be verified, and the number of verifications is large, which results in large computational overhead. However, this solution only needs to verify the signature once, which greatly reduces the number of verifications and reduces the computational overhead.

1.2 Organization

The rest of the paper is organized as follows. Section 2 introduces the related work. Section 3 describes the cross-domain problem of threshold blind signature. In the Sect. 4, the system model of the scheme and the goals it achieves are introduced. A detailed description of the scheme can be found in the Sect. 5. The performance evaluation of the scheme is given in the Sect. 6. Finally, we give the conclusion in Sect. 7.

2 Related Work

In order to solve the problem of data security deduplication in cloud storage, researchers have proposed many deduplication schemes [1,3,13,15,16], showing how deduplication is very attractive to reduce the use of storage resources [10]. Most work didn't consider security to be a consideration for deduplication systems. However, recently Harnik et al. [11] have proposed a number of attacks that may lead to data leakage in storage systems where client deduplication is deployed. In order to defend against this attack, Halevi [9] introduced the concept of proof of ownership.

According to the granularity of data, data deduplication is divided into file-level deduplication [5,13,28] and chunk-level (block-level) deduplication [3,6,14,18,35,36]. File-level deduplication was proposed earlier [4], but that technique was subsequently overshadowed by chunk-level deduplication due to the latter's better compression performance [22,37]. The cloud server is honest and curious [32], and it may try to steal users' data information. Therefore, before users upload data to the cloud server, they usually need to encrypt the data to achieve data privacy protection. However, when different users encrypt the same file with their private keys, different ciphertexts will be generated, which is not conducive to the cloud server to de-duplicate the file. Data deduplication based on convergence encryption can well achieve key sharing and is beneficial

to deduplication of ciphertext data across users. Therefore, some cloud storage data deduplication methods based on convergence encryption are proposed [3,6,8,15,16,34].

Douceur et al. first proposed the Convergent encryption (CE) [7] algorithm, using the hash value of the data as the encryption key, and the same plaintext is encrypted into the same ciphertext. However, the encrypted ciphertext of this scheme relies too much on plaintext information and is vulnerable to offline brute-force attack. Bellare et al. analyzed the security problem of convergence encryption, proposed an abstract Message lock encryption (MLE) scheme [3], and proved the security of the scheme under the random oracle model. Subsequently, they proposed a secure interactive message lock encryption scheme under the standard model [2], and developed a DupLESS system [13] which is mainly consists of clients, cloud storage nodes and a key management node. Since there is only one key management node, there is a single point of failure problem, and once the key management node is compromised, the entire system is broken. Since the DupLESS system was put forward, many scholars have conducted more in-depth research on it and proposed improvements based on it. Miao et al. [23] proposed a multi-server-aided deduplication scheme based on threshold blind signature, which can effectively resist the collusion attack between the cloud server and multiple key servers, and achieve the expected security, but its computational overhead and communication overhead are large during the initialization process. Jan Stanek et al. [27] proposed an enhanced cloud storage security threshold data deduplication scheme, which is based on the original scheme [28] and moves the sensitive data decryption shares and processing of popular state information out of cloud storage. A simpler proof of safety and an easier implementation are given, increasing practicality and increasing efficiency.

3 Cross-Domain Problem of Threshold Blind Signature Scheme

Literature [23] proposed a distributed encryption deduplication scheme based on the idea of threshold blind signature. This scheme is a multi-key server-assisted deduplication architecture, which is mainly composed of clients (users), cloud storage service providers (S-CSP), and multiple key management nodes (K-CSPs), as shown in Fig. 1.

Under this architecture, no key server can obtain the distributed key knowledge among all key servers, which can effectively resist collusion attacks between the cloud server and multiple key servers and solve the problem of single point of failure. However, this solution does not support cross-domain scenarios. As shown in Fig. 2, there are two network domains A and B. Domain A and domain B both have multiple key nodes and users, and generate virtual keys K_a, K_b, respectively. It is assumed that the networks of domain A and domain B cannot communicate with each other, but both domain A and domain B can access the Internet. For example, domain A is a campus network, and domain B is another campus network. When the user of domain A moves to domain B, because the

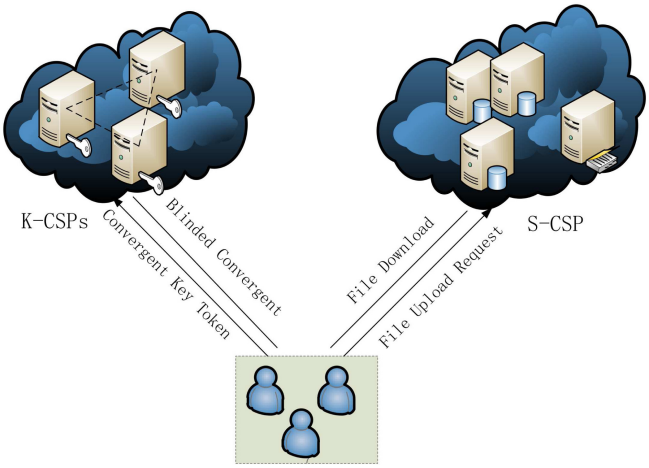

Fig. 1. Architecture of data secure deduplication.

key nodes of the two domains cannot be negotiated, resulting in K_a and K_b being unequal, the data encrypted by K_a and K_b cannot be safely deduplicated. In order to solve the problem of security de-duplication in cross-domain environment, this paper adds a master key node based on the architecture of Fig. 2. As shown in Fig. 3, the master key node is located on the Internet, and the keys of domain A and domain B are allocated by the master key node, so that it is ensured that K_a of domain A and K_b of domain B are equal. In this case, when the user of domain A moves to domain B, the data encrypted by K_a and K_b can be safely deduplicated.

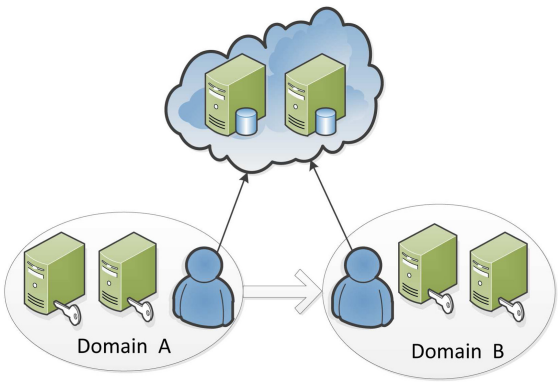

Fig. 2. Cross-domain network topology.

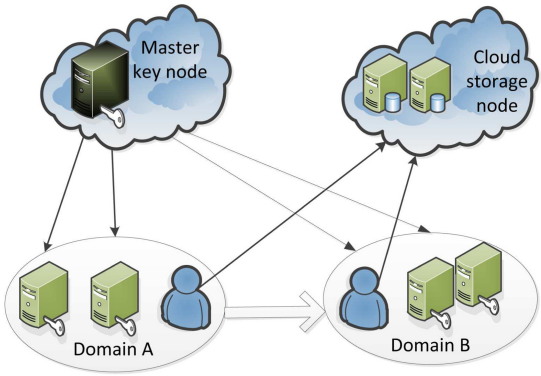

Fig. 3. Supporting cross-domain network topology.

The scheme of this paper introduces the master key node based on the Miao et al scheme [23]. If the master key node is compromised, the entire system will no longer be secure. It appears that there is a single point of failure risk, but since the master key node only communicates with the sub-key node and only distributes the partial key to the sub-key node. After the key distribution is completed, the master key node can be offline and does not receive network requests. Therefore, its security protection is easier than that of the sub-key node, and the possibility of being compromised is also lower. In addition, the scheme of this paper can distribute the sub-key nodes to each network domain, making the key nodes closer to the end users, which can effectively reduce the network communication overhead and shorten the key generation time. It is very suitable for large-scale network environments such as the Internet of Things.

4 System Model

This paper proposes a cross-domain deduplication scheme. Its architecture is shown in Fig. 4. It consists mainly of a master key node, a sub-key node, a client (users) and a cloud storage node. The introduction of each part is as follows.

(1) Master Key Node: The master key node must be secure and trusted. It generates the master key and the public key, exposes the public key, generates part of the private key of each sub-key node, and securely shares it to the sub-key node.

(2) Sub-key Node: There are multiple sub-key nodes, which receive part of the private key generated by the master key node, use it to sign the blind messages of the client, and return the result to the client.

(3) Client (users): The client blinds the message, sends it to the sub-key node, then combines t ($t \leq n$) blind signatures, obtains the encryption key after unblinding, and then encrypts the data with the encryption key and sends it to the cloud storage node.

(4) Cloud storage node: Stores ciphertext data and safely deduplicates it.

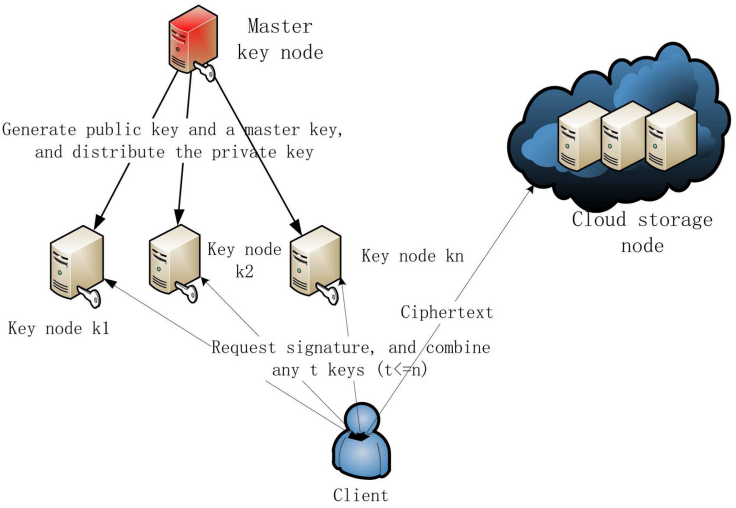

Fig. 4. Data security deduplication architecture.

5 Detailed Plan

In this section, we will introduce this scheme in detail. This scheme adopts threshold blind signature algorithm, which is based on bilinear pairings. Bilinear pairing will be constructed in the following way:

5.1 Bilinear Pairing

Let G_1 be a cyclic addition group generated by P, whose order is prime q; and G_2 be a cyclic multiplication group of the same order q. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_2$ with the following properties:

(1) Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in G_1$, and $a, b \in \mathrm{Z}_q^*$.

(2) Non-degenerate: There exists $P, Q \in G_1$ such that $e(P, Q) \neq 1$.

(3) Computable: There is an efficient algorithms to computed $e(P, Q)$ for all $P, Q \in G_1$.

Based on bilinear mapping, we can get the following definitions.

Define 1: Discrete Logarithms Problem (DLP): Given two group elements P and Q, find an integer n such that Q=nP whenever such an integer exists.

Define 2: Computational Diffe-Hellman problem (CDHP): For $a, b \in \mathrm{Z}_q^*$, give P, aP, bP, compute abP.

Define 3: Dicision Diffe-Hellman problem (DDHP): $a, b, c \in \mathrm{Z}_q^*$, give P, aP, bP, cP, decide whether c $\equiv$ ab mod q.

5.2 Detailed Steps

The main steps are as follows: system initialization, key generation, blinding, signature, verification, and encrypted storage.

First, system initialization is performed. The master key node generates the public key and the master key, and publicizes the public key. The n sub-key nodes are numbered k_1,k_2,...,k_n, and the $t-1$ degree polynomial is constructed: f(x)=a_0+a_1x+...+$a_{t-1}x^{t-1}$; calculate the partial private key corresponding to each sub-key node: f(k_1),f(k_2),...,f(k_n) and transmit it securely to the sub-key node.The client then blinds the message and sends the blinded message to the sub-key node. The sub-key node signs the message and returns it to the client. The client uploads the ciphertext data to the cloud server. Specific steps are as follows.

System initialization: The master key node selects a cyclic addition group G_1, the generator is p, and the order is prime q; Then select a cyclic multiplication group G_2 of the same order; to generate and a bilinear map e: $G_1 \times G_1 \rightarrow G_2$. The master key node selects an integer $K_M \in \mathrm{Z}_q^*$, calculates the system public key

$$Q = K_M P \tag{1}$$

and selects the following strong collision-free hash function H_1:{0,1}$^* \rightarrow G_1$, H_2:{0,1}$^* \rightarrow \mathrm{Z}_q^*$; The master key node then saves K_M as the system private key and exposes the parameters {G_1,G_2,e,q,P,Q,H_1,H_2 }.

Key generation: The n sub-key nodes are numbered k_1,k_2,...,k_n,and a random $t-1$ degree polynomial,

$$f(x) = a_0 + a_1 x + ... + a_{t-1}x^{t-1} \tag{2}$$

is constructed, where $t \leq n$. Let $f(0)$=a_0=K_M, calculate partial private keys $f(k_1)$,$f(k_2)$,...,$f(k_n)$ and Lagrange polynomials $l(k_1)$,$l(k_2)$,...,$l(k_n)$ corresponding to each sub-key node, where

$$l(k_i) = \prod_{k_i \neq k_j} \frac{-k_j}{k_i - k_j} \tag{3}$$

Finally, $f(k_i)$ and $l(k_i)$ are securely transmitted to the sub-key node.

Blinding: The client selects the random number $\alpha \in \mathrm{Z}_q^*$ to blind the message H_1(m),obtains the blinded message

$$W = \alpha H_1(m) \tag{4}$$

and then sends W to the sub-key node.

Signature: After receiving the blind message w of the client,the sub-key node k_i performs signature using the partial key $f(k_i)$ calculates

$$\sigma_i = w \times f(k_i) \times l(k_i) \tag{5}$$

and return σ_i to the client. The client combines the t signatures to obtain $\sum_{i=1}^{t} \sigma_i$ and unblinds it by α^{-1} to obtain:

$$\begin{aligned}\sigma &= \alpha^{-1}\sum_{i=1}^{t}\sigma_i \\ &= \alpha^{-1}W\sum_{i=1}^{t}f(k_i)\times l(k_i) \\ &= H_1(m)\sum_{i=1}^{t}f(k_i)\times l(k_i)\end{aligned} \tag{6}$$

where

$$\sum_{i=1}^{t}f(k_i)\times l(k_i) = a_0 = f(0) = K_M \tag{7}$$

so that $\sigma{=}K_M\ H_1(m)$ is obtained.

Verification: The client calculates $e(\sigma, P)$ and $e(H_1(m), Q)$. If they are equal, it proves that σ is obtained by K_M encryption. The left side of the equation is:

$$\begin{aligned}e(\sigma, P) &= e(\alpha^{-1}\sum_{i=1}^{t}\sigma_i, P) \\ &= e(\alpha^{-1}W\sum_{i=1}^{t}f(k_i)\times l(k_i), P) \\ &= e(H_1(m)\sum_{i=1}^{t}f(k_i)\times l(k_i), P) \\ &= e(H_1(m)K_M, P) \\ &= e(H_1(m), K_M P) \\ &= e(H_1(m), Q)\end{aligned} \tag{8}$$

6 Performance Evaluation

The data deduplication scheme in this paper is essentially a threshold blind signature scheme, and its security is similar to that of [23]. This section mainly evaluates the computational performance of the proposed scheme, and compares it with Miao et al.'s scheme in terms of system initialization time and key generation time.

6.1 Experimental Method

Since the network transmission performance will vary greatly in practical applications, this difference should be ignored during the experiment. For this reason, this paper carries out simulation experiments on a physical computer. The master key node, each sub-key node, the client, and the cloud storage are simulated by separate threads, and the communication between the nodes is simulated

by the Socket communication interface. Since the local communication jitter between multiple threads on the same physical computer is extremely small, this method can effectively shield the impact of network transmission on experimental results.

6.2 Experimental Environment

The experimental hard ware consists of a 2.5 GHz Intel Core i5-4200 CPU processor and 4G memory. The software is based on the Windows operating system and the Eclipse integrated development environment. The experimental program is implemented by the Java programming language, and the cryptographic operations mentioned in Sect. 5.2 are performed using the JPBC (Java Pairing-Based Cryptography) library.

6.3 Experimental Results and Analysis

As with the experiment by Miao et al., experiments were performed with 5 to 10 key node. In order to accurately calculate the time and reduce the error, the average of ten experimental data was taken. The time comparison of system initialization consumption is given in Fig. 5.

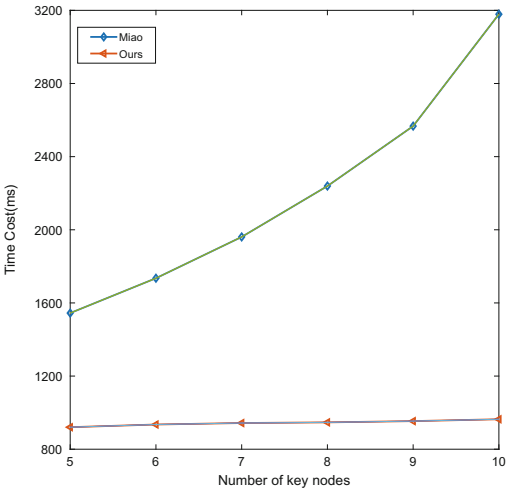

Fig. 5. Time comparison of system initialization consumption.

It can be seen from Fig. 5 that the initialization time of the scheme of Miao [23] increases approximately as a power function with the increase of the number of key nodes, and the scheme proposed in this paper increases linearly. Because Miao et al.'s scheme needs to communicate with other nodes during the system initialization, the total number of communications is $n(n-1)$. However, in the

scheme proposed in this paper, the master key node only needs to communicate with n sub-key nodes for n times to complete initialization, which reduces communication and computational overhead. The experimental results show that the average performance is increased by 54.83%. Figure 6 gives a comparison of the time to generate the key.

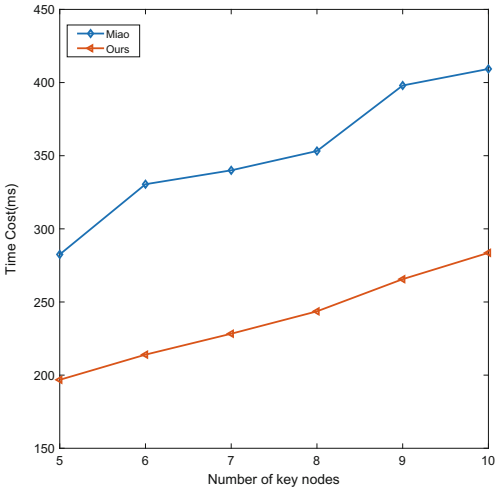

Fig. 6. Generate key time comparison.

As can be seen from Fig. 6, as the number of key nodes increases, the time to generate the key increases linearly. In the scheme of Miao et al. [23], the user needs to verify the signatures of all key nodes. If the verification passes, the blind signature is calculated and verified again. However, in this scheme, the client does not need to verify the signature of each sub-key node, and the client only needs to verify once, reducing the verification and computational overhead. The experimental results show that the average performance is improved by 32.24%.

7 Conclusion and Future Work

In order to solve the problem of cloud storage data security, the predecessors have done a lot of research work. An existing security deduplication scheme based on threshold blind signature requires a large amount of information to be exchanged between key management nodes, and the computational cost of verifying the signature is large. In addition, cross-domain scenarios are not supported. In this paper, a new threshold blind signature scheme is proposed. The key management node is divided into a master key node and multiple sub-key nodes. The key of the sub-key node is generated and allocated by the master

key node, so that the number of communications is reduced from $n(n-1)$ times to n times, and the verification is reduced from multiple times to once. The scheme of this paper supports secure deduplication in a cross-domain environment in addition to ensuring data encryption security, preventing brute force attacks, and satisfying convergence key security. The experimental results show that the performance of the scheme is improved in terms of system initialization and key generation, and it is more suitable for computing resource-constrained application scenarios, such as the Internet of Things.

Acknowledgment. This work is supported in part by the National key research and development plan of China under Grant No. 2018YFB1800303 and the 13th Five-Year Science and Technology Research Project of the Education Department of Jilin Province under Grant No. JJKH20190598KJ.

References

1. Anderson, P., Zhang, L.: Fast and secure laptop backups with encrypted deduplication. In: van Drunen, R. (ed.) Uncovering the Secrets of System Administration: Proceedings of the 24th Large Installation System Administration Conference, LISA 2010, San Jose, CA, USA, 7–12 November 2010. USENIX Association (2010)
2. Bellare, M., Keelveedhi, S.: Interactive message-locked encryption and secure deduplication. In: Katz, J. (ed.) PKC 2015. LNCS, vol. 9020, pp. 516–538. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46447-2_23
3. Bellare, M., Keelveedhi, S., Ristenpart, T.: Message-locked encryption and secure deduplication. In: Johansson, T., Nguyen, P.Q. (eds.) EUROCRYPT 2013. LNCS, vol. 7881, pp. 296–312. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38348-9_18
4. Bolosky, W.J., Douceur, J.R., Ely, D., Theimer, M.: Feasibility of a serverless distributed file system deployed on an existing set of desktop pcs. In: Brandwajn, A., Kurose, J., Nain, P. (eds.) Proceedings of the 2000 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, Santa Clara, CA, USA, 18–21 June 2000, pp. 34–43. ACM (2000)
5. Cao, Z., Wen, H., Ge, X., Ma, J., Diehl, J., Du, D.H.C.: TDDFS: a tier-aware data deduplication-based file system. TOS **15**(1), 4:1–4:26 (2019)
6. Chen, R., Mu, Y., Yang, G., Guo, F.: BL-MLE: block-level message-locked encryption for secure large file deduplication. IEEE Trans. Inf. Forens. Secur. **10**(12), 2643–2652 (2015)
7. Douceur, J.R., Adya, A., Bolosky, W.J., Simon, D., Theimer, M.: Reclaiming space from duplicate files in a serverless distributed file system. In: Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS'02), Vienna, Austria, 2–5 July 2002, pp. 617–624. IEEE Computer Society (2002)
8. González-Manzano, L., Orfila, A.: An efficient confidentiality-preserving proof of ownership for deduplication. J. Netw. Comput. Appl. **50**, 49–59 (2015)
9. Halevi, S., Harnik, D., Pinkas, B., Shulman-Peleg, A.: Proofs of ownership in remote storage systems. In: Chen, Y., Danezis, G., Shmatikov, V. (eds.) Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, 17–21 October 2011, pp. 491–500. ACM (2011)

10. Harnik, D., Margalit, O., Naor, D., Sotnikov, D., Vernik, G.: Estimation of deduplication ratios in large data sets. In: IEEE 28th Symposium on Mass Storage Systems and Technologies, MSST 2012, 16–20 April 2012, Asilomar Conference Grounds, Pacific Grove, CA, USA, pp. 1–11. IEEE Computer Society (2012)
11. Harnikand, D., Pinkasand, B., Shulman-Peleg, A.: Side channels in cloud services: deduplication in cloud storage. IEEE Secur. Privacy **8**(6), 40–47 (2010)
12. Hovhannisyan, H., Qi, W., Lu, K., Yang, R., Wang, J.: Whispers in the cloud storage: a novel cross-user deduplication-based covert channel design. Peer-to-Peer Netw. Appl. **11**(2), 277–286 (2016). https://doi.org/10.1007/s12083-016-0483-y
13. Keelveedhi, S., Bellare, M., Ristenpart, T.: Dupless: server-aided encryption for deduplicated storage. In: King, S.T. (ed.) Proceedings of the 22nd USENIX Security Symposium, Washington, DC, USA, 14–16 August 2013, pp. 179–194. USENIX Association (2013)
14. Kumar, N., Antwal, S., Jain, S.C.: Differential evolution based bucket indexed data deduplication for big data storage. J. Intell. Fuzzy Syst. **34**(1), 491–505 (2018)
15. Li, J., Chen, X., Li, M., Li, J., Lee, P.P.C., Lou, W.: Secure deduplication with efficient and reliable convergent key management. IEEE Trans. Parallel Distrib. Syst. **25**(6), 1615–1625 (2014)
16. Li, J., Li, Y.K., Chen, X., Lee, P.P.C., Lou, W.: A hybrid cloud approach for secure authorized deduplication. IEEE Trans. Parallel Distrib. Syst. **26**(5), 1206–1216 (2015)
17. Li, J., Qin, C., Lee, P.P.C., Li, J.: Rekeying for encrypted deduplication storage. In: 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2016, Toulouse, France, June 28 - July 1, 2016, pp. 618–629. IEEE Computer Society (2016)
18. Li, M., Qin, C., Li, J., Lee, P.P.C.: CDStore: toward reliable, secure, and cost-efficient cloud storage via convergent dispersal. IEEE Internet Comput. **20**(3), 45–53 (2016)
19. Liu, J., Asokan, N., Pinkas, B.: Secure deduplication of encrypted data without additional independent servers. In: Ray, I., Li, N., Kruegel, C. (eds.) Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, 12–16 October 2015, pp. 874–885. ACM (2015)
20. Ma, J., et al.: Lazy exact deduplication. TOS **13**(2), 11:1–11:26 (2017)
21. Mandagere, N., Zhou, P., Smith, M.A., Uttamchandani, S.: Demystifying data deduplication. In: Douglis, F. (ed.) Middleware 2008, ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, Belgium, 1–5 December 2008, Companion Proceedings, pp. 12–17. ACM (2008)
22. Meyer, D.T., Bolosky, W.J.: A study of practical deduplication. In: Ganger, G.R., Wilkes, J. (eds.) 9th USENIX Conference on File and Storage Technologies, San Jose, CA, USA, 15–17 February 2011, pp. 1–13. USENIX (2011)
23. Miao, M., Wang, J., Li, H., Chen, X.: Secure multi-server-aided data deduplication in cloud computing. Pervasive Mobile Comput. **24**, 129–137 (2015)
24. Paulo, J., Pereira, J.: A survey and classification of storage deduplication systems. ACM Comput. Surv. **47**(1), 11:1–11:30 (2014)
25. Puzio, P., Molva, R., Önen, M., Loureiro, S.: Cloudedup: secure deduplication with encrypted data for cloud storage. In: IEEE 5th International Conference on Cloud Computing Technology and Science, CloudCom 2013, Bristol, United Kingdom, 2–5 December 2013, vol. 1, pp. 363–370. IEEE Computer Society (2013)
26. Shin, Y., Koo, D., Hur, J.: A survey of secure data deduplication schemes for cloud storage systems. ACM Comput. Surv. **49**(4), 74:1–74:38 (2017)

27. Stanek, J., Kencl, L.: Enhanced secure thresholded data deduplication scheme for cloud storage. IEEE Trans. Dependable Sec. Comput. **15**(4), 694–707 (2018)
28. Stanek, J., Sorniotti, A., Androulaki, E., Kencl, L.: A secure data deduplication scheme for cloud storage. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 99–118. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_8
29. Tian, Y., Khan, S.M., Jiménez, D.A., Loh, G.H.: Last-level cache deduplication. In: Bode, A., Gerndt, M., Stenström, P., Rauchwerger, L., Miller, B.P., Schulz, M. (eds.) 2014 International Conference on Supercomputing, ICS'14, Muenchen, Germany, 10–13 June 2014, pp. 53–62. ACM (2014)
30. Wang, J., Chen, X.: Efficient and secure storage for outsourced data: a survey. Data Sci. Eng. **1**(3), 178–188 (2016)
31. Xia, W., et al.: A comprehensive study of the past, present, and future of data deduplication. Proc. IEEE **104**(9), 1681–1710 (2016)
32. Xiong, J., Li, F., Ma, J., Liu, X., Yao, Z., Chen, P.S.: A full lifecycle privacy protection scheme for sensitive data in cloud computing. Peer-to-Peer Netw. Appl. **8**(6), 1025–1037 (2014). https://doi.org/10.1007/s12083-014-0295-x
33. Xiong, J., Zhang, Y., Li, X., Lin, M., Yao, Z., Liu, G.: Rse-pow: a role symmetric encryption pow scheme with authorized deduplication for multimedia data. MONET **23**(3), 650–663 (2018)
34. Yan, F., Tan, Y., Zhang, Q., Wu, F., Cheng, Z., Zheng, J.: An effective raid data layout for object-based de-duplication backup system. Chin. J. Electron. **25**(5), 832–840 (2016)
35. Zhang, C., et al.: MII: a novel content defined chunking algorithm for finding incremental data in data synchronization. IEEE Access **7**, 86932–86945 (2019)
36. Zhang, Y., et al.: A fast asymmetric extremum content defined chunking algorithm for data deduplication in backup storage systems. IEEE Trans. Comput. **66**(2), 199–211 (2017)
37. Zhu, B., Li, K., Patterson, R.H.: Avoiding the disk bottleneck in the data domain deduplication file system. In: Baker, M., Riedel, E. (eds.) 6th USENIX Conference on File and Storage Technologies, FAST 2008, 26–29 February 2008, San Jose, CA, USA, pp. 269–282. USENIX (2008)