EasyChair Preprint

№ 4802

A Review on Assured Deletion of Cloud Data Based on Cryptography

Guan Wang and Yehong Luo

December 25, 2020

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2019) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

2020 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2020

A Review on Assured Deletion of Cloud Data Based on Cryptography

Guan Wang[a,b], Yehong Luo[a,b]

[a]Faculty of Information Technology, Beijing University of Technology, Beijing 100124
[b]Beijing Key Laboratory of Trusted Computing, Beijing 100124

Abstract

In the cloud computing environment, storage is used for users to provide data outsourcing services. After users outsource data to the cloud, privacy protection has become a key concern. As far as we know, there is no systematic analysis of the assured deletion of cloud storage data based on cryptography from the perspective of having a third-party key management center (TKMC) and no TKMC. Our work focuses on analyzing and evaluating the fine-grained, safety, and performance of these schemes, and we also point out that researches can use trusted computing to securely strengthen TKMC in the future, making assured deletion more secure and reliable.

Peer-review under responsibility of the scientific committee of the 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.

Keywords: Cloud storage, assured deletion, cryptography, third-party key management center, trusted computing

1. Introduction

Cloud storage system (CSS) provides users with large-capacity storage space, and access is fast and efficient. While CSS brings opportunities to users, it also brings challenges to privacy protection. Part of it is that after users delete the data, it remains in the hardware storage device of CSS, resulting in the robbery of the data. The data which have been deleted cannot be accessed, restored, or constructed into any other useful information by any role, including data owner (DO), CSS, and other users, which is assured deletion. Delete data in the CSS for certainly, and some researchers applied the cryptography into the assured deletion

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.
E-mail address: wangguan@bjut.edu.cn

Peer-review under responsibility of the scientific committee of the 2018 International Conference on Identification, Information and Knowledge in the Internet of Things.

of cloud storage (ADCD). This concept uses encryption keys to encrypt data and then stores the encrypted data in CSS. The corresponding key is deleted to achieve assured deletion, that is, the data management problem is converted into key management problem. This paper evaluates the performance of the ADCD schemes from the aspects of granularity and security. The proposed solution is required to achieve the function of fine-grained access and deletion. And if a malicious attacker steals the encryption key, the data stored in CSS will also be leaked. Therefore, it is necessary to evaluate the security of ADCD.

In the next section, we introduce the various types of ADCD methods distinguished by the with or without TKMC, and their advantages and disadvantages. In section 3, we are providing future researchers with the research direction of using trusted computing (TC) to ensure the credibility of TKMC to ensure the security of keys, that is, to ensure the security of ADCD. Finally, we provide concluding remarks in section 4.

2. Assured deletion of cloud data based on cryptography

This section is divided into two categories according to the ADCD based on cryptography. One is the introduction of a TKMC, that is, the key to encrypt the data is stored in the key manager; the other is a without TKMC solution, the idea does not need to store a large number of keys in a third-party key store.

2.1. With TKMC

There are four situations for the TKMC in the cryptography-based ADCD. The first is based on hardware. This is the first proposed solution to use a TKMC. The second is based on policy. The third is based on Distributed Hash Table (DHT) [1, 2] network. The fourth is based on TC, which only uses TC to protect the security of internal data, and does not use TC to achieve platform trust and session security.

2.1.1. Based on Hardware Protection(BHP)

Perlman [3] proposed the concept of Ephemerizer in 2005. The key storage and data encryption process are all carried out on a tamper-resistant card called Ephemerizer. The DO uses the public key negotiated with Ephemerizer to encrypt the data. There are two schemes for deletion. One is to delete data on demand. Each data file is encrypted with a public key. Deleting the corresponding key in Ephemerizer at the same time, one for encrypting the key table and one for encrypting the data. The other is the Predetermined Expiration Time Scheme. The DO has a long-term key. Define the time to delete the data before generating the encrypted data. In general, the solution proposed by Perlman can achieve assured deletion well, but users need to keep a long-term key, if the key is stolen, all data of DO will be threatened. Tang [4] based on the concept of Ephemerizer, proposed Time-Ephemerizer. This scheme ensures that data is only available during the pre-defined life cycle. After the time interval comes from the time server timestamp, no attacker can recover the data. This solution effectively prevents the initial disclosure of sensitive data, but there is the possibility of a man-in-the-middle (MITM) attack in the session protocol between the DO and TKMC.

2.1.2. Based on Policy Protection(BPP)

Yang Tang et al. [5] proposed a method called FADE. This scheme associates each file with a single atomic file ACP. Each ACP is associated with a control key (CK), and TKMC manages all CKs. The TKMC used in this method is guaranteed using a quorum scheme [6]. Before accessing the TKMC, the DO first needs to provide the TKMC with authentication credentials. The data is encrypted with a data key (DK), and the DK is encrypted with a CK corresponding to the ACP. When the ACP is revoked, the corresponding CK will be deleted from the TKMC. The communication between the DO and the TKMC is encrypted by blinded RSA, which can effectively prevent sniffing attacks during the session. This method is policy-based, and each data file corresponds to a policy so that the data can be deleted in fine granularity. However, the TKMC may still collude with attackers, the TKMC is still unreliable.

Xue.L et al. [7] proposed AD-KP-ABE in 2019, that is, a key-policy attribute-based encryption scheme (KP-ABE) [8]. The TKMC Trusted Authority (TA) completes the generation of DKs and re-encryption keys. AD-KP-ABE associates data with attributes and ACPs with users' private keys (PKs).The data is

deleted by revoking the attributes necessary for the user to access the data. The method also proposed to use Merkle Hash Tree (MHT) [9] to generate deletion evidence form cloud servers. If the evidence is valid, the DO is confident that the data has been deleted. This method has the characteristics of fine-grained access and verifiability, but TA cannot guarantee its credibility and security.

2.1.3. Based on DHT Network Protection(B-DHT-P)

Geambasu et al. [10] proposed a scheme called Vanish. In the data encryption stage, first, a ciphertext C is obtained by generating a random data key K. Then using the Shamir Secret Sharing algorithm [11] to divide the K into N shares, randomly select an access key L, with a secure pseudo-random number to generate the index L_i, and then storing N key shares in the DHT. Over time, the N key shares will disappear from the DHT. The improvement of the scheme proposed by F. Yue et al. [12] relative to Vanish is that they split the decryption key and part of the ciphertext through threshold secret sharing and store it in the DHT. C. Li et al. [14] used two-level encryption to encrypt data. The first level uses the DK to encrypt each database. The second level uses the Minimal Tree Key Set(MTKS) encrypted with the All-Or-Nothing algorithm [15] to generate the CK and then stores the encrypted data block in the DHT. In this method, different data keys encrypt different data blocks to achieve fine-grained access. Moreover, MTKS that distributes access rights through user groups can complete accuracy access control for authorized users. The method increases the overhead of hopping attacks and effectively avoids sniffing attacks. The method proposed by J. Xiong et al. [16] uses a symmetric encryption key to encrypt data. This scheme increases the length of the key by extracting the ciphertext and encrypting the decryption key with the algorithm based on identity-based timed-release encryption (ID-TRE) and combines the two parts to store in the DHT. The key in the DHT will self-destruct according to pre-defined time. Extracting part of the ciphertext which effectively prevents brute-force attacks of the ciphertext, and adding the extracted ciphertext to the key increases the overhead of hopping attacks. The ID-TRE encryption algorithm effectively prevents sniffing attacks [13].

2.1.4. Based on Trusted Computing Protection(BTCP)

Tian et al. [17] proposed an efficient scheme of cloud data assured deletion (ESAD). The TKMC of the scheme is named Attribute Key Management System (AKMS), which is composed of a key generator and an attribute authorizer. The key generator mainly manages the system master key and the system public key. The attribute authorizer mainly in charge of allocating an only identifier ID to each authorized user and preserving the attribute list of authorized users. The two parts independently correspond with other components and use Trusted Platform Module (TPM) security chips to guard core data such as the system master key and the system public key, and the attribute list of authorized users. When deleting data, the DO only needs to create a random number to replace the attribute value that should be deleted in the original attribute list. This scheme effectively prevents collusion attacks, VLAN hopping, sniffing attacks, and single-point failures, but cannot improve fine-grained access while improving performance.

The TKMC introduced above are similar to CSS; they are both not directly controlled by DOs, so the credibility of the TKMC cannot be assured. At present, TC technology is mature enough, and it is a good idea to combine TC with ADCD. The detail of how to apply TC to ADCD has discussed in the section Future Direction. Concerning the ADCD based on cryptography with a TKMC, the analysis of fine-grained access control, safety, and existing limitation was made. As shown in Tables 1 and 2.

2.2. Without TKMC

There is no TKMC for ADCD. First is the Master Key Method, that is, the DO only needs to maintain a master key locally. The second is the Processing Ciphertext Method, that is, the ciphertext encrypted by the DK for further processing, such as extracting ciphertext using an algorithm. The third is the Like FADE [5] Method, which is, based on the FADE method but without TKMC.

2.2.1. Master Key Method

The scheme proposed by Cachin.C et al. [18] initially generates protection class based on attributes, and then generates a directed acyclic strategy graph. The master key is generated by security parameter and the

policy graph, and an auxiliary state is also generated. Both the DK and decryption key are derived from the data attributes and the master key. Deleting the attributes which are the file to be deleted in the protection class, and generate a new master key that is used to replace the one in the non-erasabe memory according to the new strategy graph. Data with different attributes are encrypted with different keys to achieving a certain degree of fine-grained access. If an attacker steals the master key, all data files may be leaked.

Mo et al. [19] proposed a ADCD scheme based on a multi-level key structure of a Recursively Encrypted Red-black key tree (RERK). The DO randomly generates n data keys based on the symmetric encryption algorithm to construct the RERK. Each data file is encrypted by one of the DKs. After the encryption is completed, the recursive encryption key generates the key sequence k_1 to k_n, and finally, store the key sequence and ciphertext c_1 to c_n to the cloud. The DO only saves the metakey k and the tag of the root of RERK. When you need to delete data, first finding the i^{th} key, the CSS returns the key node and its parent node and sibling nodes. And the corresponding key node needs to be deleted, the tree is updated in order to balance the tree node again, and then a new key node sequence is generated. Therefore, the DO's metakey is also changed. In this way, the key is deleted successfully and assuredly. This method prevents the leakage of key, but it is only suitable for encryption and assured deletion of a small amount of sensitive data.

2.2.2. Processing Ciphertext Method

Chen et al. [20] proposed a scheme called AD-IHSE. This scheme uses the method of logistic chaotic mapping [21] to generate a random sequence of data extraction locations. After successfully extracting the ciphertext, the partial ciphertext is hidden in the vector image using the Least Significant Bits (LSB). Therefore, the ciphertext will be divided into two parts, one part is the ciphertext encrypted once by the data encryption key, and the other is the ciphertext hidden in the vector image after the random sequence extraction. This method can ensure data security in the case of key leakage. However, the position sequence of the extracted ciphertext may also be leaked. In addition, when the DO sends the chaotic sequence of files and encryption keys to the data sharing user, there will be MITM attack in the communication process.

2.2.3. Like FADE Method

Habib et al. [22] proposed a scheme named SFADE which uses the concept of FADE [5] to remove its TKMC. The DO uses a secret word P_1 to generate K_1 and then generates a random key K_2, where K_1 is similar to the CK in FADE, and K_2 is similar to the DK. The encrypted files and K_2 are stored in the cloud. SFADE has many limitations, such as the failure to achieve fine-grained data deletion and data sharing. Nusrat et al. [23] proposed the SFADE+ scheme based on SFADE [22], which added the sharing of data with other users but still cannot achieve fine-grained deletion. Zakaria, I. et al. [24] also made improvements on the basis of FADE [5], the scheme is called FADETPM. The TPM is used to protect the user's RSA private key. This private key cannot be directly obtained through software means under the protection of the DO's hardware, ensuring the security of the DK. However, the performance of encryption and decryption and access is relatively low, which can also be a field for further research in the future.

Table 1. Fine-grained access control and safety analysis of ADCD based on cryptography with TKMC

Types	Schemes	Fine-grained Access Control	Safety
BHP	Ephemerizer	can not achieve	No security certify
	Time-Ephemerizer	can not achieve	Prevent initial disclosure of data
BPP	FADE	Achievable, ACP is associated with CK	Secure the session with blind functions
	AD-KP-ABE	Achievable, ACP is associated with PK	Use MHT to generate deletion certify
B-DHT-P	Vanish	can not achieve	Low degree of resistance to DK's leakage
	SSDD	can not achieve	Increasing the overhead of hopping attacks
	MTKS-AON	Achievable, different data have different DK	Preventing sniffing and hopping attacks
	FullPP	Achievable, data is accessible with ID-TRE	Increasing the overhead of hopping attacks Preventing brute-force and sniffing attacks
BTCP	ESAD	Achievable, ACP based on user attributes	Preventing collusion attacks, SPOF, VLAN hopping and Sniffing attacks

3. Future Direction

Over the past 15 years, many scholars have developed a lot of ADCD based on cryptography, and we can apply other technologies to ADCD. The main security issues of the TKMC are the security of the conversation between the DO and TKMC and the credibility of the TKMC. Researchers can use TC for ADCD. Generally speaking, TC has the functions of protection execution, sealing storage, remote attestation, and I/O protection [25]. The protection execution mechanism refers to providing a safe area in the CPU chip to run some sensitive applications so that users can still trust the security of this area even when they are in a terrible environment. TKMC can use the mechanism to protect data, such as DKs and CKs. The sealing storage mechanism is the use of encryption to protect data so that users can trust the confidentiality and integrity of data. The remote attestation mechanism is to enable users to believe that a remote platform is trustworthy. The I/O protection mechanism is to make the interaction path between the user and the application a trusted path, and DO can establish a secure session with TKMC for the secure transmission of information. Nowadays, TC technology has matured, such as Intel Trusted Execution Technology (Intel TXT) [26] based on TPM, the security extension of ARM TrustZone for embedded platform and Intel Software Guard Extensions (Intel SGX) [27] instruction set extensions provide users with a trusted execution environment and verify the credibility of the platform. The no TKMC solution can avoid the threatens brought by TKMC, but the problem that follows is that researchers need to study the performance consumption of DO key management and data encryption and decryption. Researchers can develop effective key derivation algorithms, and use TC's protection execution and sealing storage mechanism to manage the key for its root key to ensuring the security of the key.

TC can ensure the integrity and confidentiality of system data, as well as the trustworthiness of remote platforms, applying the mechanism of TC to ADCD, which has great research value. The overall security architecture and active immune security system led by TC have become an indispensable part of cyberspace security, and also provide the cornerstone of security for ADCD.

Table 2. Limitation analysis of ADCD based on cryptography with TKMC

Types	Schemes	Limitation
BHP	Ephemerizer	MITM, SPOF, brute-force attacks
	Time-Ephemerizer	MITM, SPOF, brute-force attacks
BPP	FADE	SPOF, brute-force attacks, Collusion attack
	AD-KP-ABE	Collusion attack, user's private key leakage issues
	Vanish	Hopping attack, sniffing attack, brute-force attacks
B-DHT-P	SSDD	Sniffing attack
	MTKS-AON	MTKS root key and AON leakage issues
	FullPP	Only for assured deletion of small amounts of data
BTCP	ESAD	Increasing fine-grained access will reduce performance

4. Conclusion

In the CSS, users can only passively trust the reliability of the service after outsourcing their data to the CSP and believe that other malicious users and CSP managers will not steal their private data. The main idea of the ADCD method based on cryptography is to convert data security issues into key security issues, which solves the problem of difficult control of outsourced data. We divide the schemes into with TKMC and without TKMC to summarize the implementation methods of each scheme and its fine-grained access, security, and propose future directions. Researchers can use the active immune of TC to ensure the integrity and confidentiality of the platform so that TKMC and DO themselves can safely manage keys to achieve ADCD.

References

[1] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," IEEE Communications Surveys and Tutorials, vol. 7, no. 2, pp. 72–93, 2005.

[2] G. Urdaneta, G. Pierre, and M. Van Steen, "A survey of dht security techniques," ACM Computing Surveys, vol. 43, no. 2, p. 8, 2011.

[3] R. Perlman, "File system design with assured delete," Third IEEE International Security in Storage Workshop (SISW'05), pp. 83–88, 2005.

[4] Q. Tang, "Timed-ephemerizer: Make assured data appear and disappear," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6391 LNCS, pp. 195–208, 2010.

[5] Y. Tang, P. P. C. Lee, J. C. S. Lui, and R. Perlman, "FADE: Secure Overlay Cloud Storage with File Assured Deletion," Security and Privacy in Communication Networks, pp. 380–397, 2010.

[6] A. Shamir, "How to Share a Secret," Communications of the ACM, vol. 22, no. 11, pp. 612–613, 1979. [Online]. Available: https://doi.org/10.1145/359168.359176

[7] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, "Efficient attribute-based encryption with attribute revocation for assured data deletion," Information Sciences, vol. 479, pp. 640–650, 2019.

[8] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," Proceedings of the ACM Conference on Computer and Communications Security, pp. 89–98, 2006.

[9] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," Advances in Cryptology — CRYPTO '87, pp. 369–378, 1988.

[10] R. Geambasu, T. Kohno, A. A. Levy, and H. M. Levy, "Vanish: Increasing data privacy with self-destructing data," Proceedings of the 18th USENIX Security Symposium, pp. 299–315, 2009.

[11] B. Poettering, "Shamir's secret sharing scheme," http://point-at-infinity.org/ssss/.

[12] G. Wang, F. Yue, and Q. Liu, "A secure self-destructing scheme for electronic data," Journal of Computer and System Sciences, vol. 79, no. 2, pp. 279–290, 2013.

[13] S. Wolchok, O. S. Hofmann, N. Heninger, E. W. Felten, J. A. Halderman, C. J. Rossbach, B. Waters, and E. Witchel, "Defeating Vanish with Low-Cost Sybil Attacks Against Large DHTs," Ndss, pp. 1–20, 2010.

[14] C. Li, Y. Chen, and Y. Zhou, "A data assured deletion scheme in cloud storage," China Communications, vol. 11, no. 4, pp. 98–110, 2014.

[15] R. L. Rivest, "All-or-nothing encryption and the package transform," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 1267, pp. 210–218, 1997. [Online]. Available: https://doi.org/10.1007/bfb0052348

[16] J. Xiong, F. Li, J. Ma, X. Liu, Z. Yao, and P. S. Chen, "A full lifecycle privacy protection scheme for sensitive data in cloud computing," Peer-to-Peer Networking and Applications, vol. 8, no. 6, pp. 1025–1037, 2015. [Online]. Available: https://doi.org/10.1007/s12083-014-0295-x

[17] Y. Tian, T. Shao, and Z. Li, "An Efficient Scheme of Cloud Data Assured Deletion," Mobile Networks and Applications, pp. 1–12, 2020.

[18] C. Cachin, K. Haralambiev, H. C. Hsiao, and A. Sorniotti, "Policy-based secure deletion," Proceedings of the ACM Conference on Computer and Communications Security, pp. 259–270, 2013. [Online]. Available: https://doi.org/10.1145/2508859.2516690

[19] Z. Mo, Q. Xiao, Y. Zhou, and S. Chen, "On deletion of outsourced data in cloud computing," IEEE International Conference on Cloud Computing, CLOUD, pp. 344–351, 2014.

[20] Y. Chen and W. Yao, "Cloud Data Assured Deletion Based on Information Hiding and Secondary Encryption with Chaos Sequence," Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST, pp. 613–623, 2018.

[21] V. Patidar, N. K. Pareek, G. Purohit, and K. K. Sud, "A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption," Optics Communications, vol. 284, no. 19, pp. 4331–4339, 2011.

[22] A. B. Habib, T. Khanam, and R. Palit, "Simplified File Assured Deletion (SFADE) - A user friendly overlay approach for data security in cloud storage system," Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2013, pp. 1640–1644, 2013.

[23] R. Nusrat and R. Palit, "Simplified FADE with sharing feature (SFADE+): A overlay approach for cloud storage system," 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), pp. 1–6, 2017.

[24] I. Zakaria and H. Mustaha, "FADETPM: Novel approach of file assured deletion based on trusted platform module," 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech), pp. 1–4, 2017.

[25] W. FENG, Y. QIN, J. LIU, and D. FENG, "Trusted computing theory and technology in innovation-driven development," SCIENTIA SINICA Informationis, vol. 50, no. 8, pp. 1127–1147, 2020.

[26] Intel Corporation, "Intel Trusted Execution Technology Software Development Guide," 2013, https://www.intel.com/content/dam/www/public/us/en/documents/white-papers/trusted-execution-technology-security-paper.pdf.

[27] F. McKeen, I. Alexandrovich, I. Anati, D. Caspi, S. Johnson, R. Leslie-Hurd, and C. Rozas, "Intel® Software Guard Extensions (Intel® SGX) support for dynamic memory management inside an enclave," pp. 10–19, 2016.