Fine-grained outsourced data deletion in cloud storage

Zhanpeng Yang\textsuperscript{1,4}, Qiaobin Feng\textsuperscript{1,2,4*}, Jinglan Jiang\textsuperscript{1} and Qiyu Chen\textsuperscript{3}

\textsuperscript{1} School of Information Technology, Liuzhou Railway Vocational Technical College, Liuzhou, Guangxi, Liuzhou, 545616, China
\textsuperscript{2} Department of Mobile Security, Guangdong Achieve Information Technology Development, Guangzhou, Guangdong, 510600, China
\textsuperscript{3} College of Information and Computer Engineering, Northeast Forestry University, Harbin, Heilongjiang, 150000, China
\textsuperscript{4} School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, China

\textsuperscript{*}Corresponding author’s e-mail: csyang@guet.edu.cn

Abstract. Due to the separation of outsourced data management and ownership, user loses the direct control over outsourced data. Therefore, the cloud executes all of the acts (e.g., data erasure) over outsourced data. However, the dishonest cloud may keep the data maliciously. To solve this problem, a new solution is proposed in this paper, which is able to reach fine-grained data erasure. In our protocol, if some outsourced data blocks are no longer needed, the user is able to remove them flexibly, while the useful data blocks remain in the cloud. After that, the user could check the erasure outcome efficiently. If the cloud remains the data blocks dishonestly, the user can fine the data reservation. Meanwhile, we provide the scheme analysis, which is able to show the security and practicality of our scheme.

1. Introduction

Cloud storage, a newly-developing and promising data storage service offered by cloud service providers\cite{1}. Through embracing cloud storage, the resource-constraint local users are able to upload their large-scale personal data to the remote cloud, which is able to greatly reduce local storage cost and computation overhead for the users\cite{2}. Hence, more and more tenants, including companies and individuals employ cloud storage service. As a report shown that 82\% of the companies are able to save both software and hardware investments through employing cloud storage service\cite{3}.

Although there are many attractive advantages, cloud storage inevitably suffers from many austere security challenges. Firstly, in cloud storage, the cloud service provider maintains large-scale data for the user. Note that the outsourced file may contain some user’s private information, which cannot be exposed in the user’s opinion. However, both the cloud and the hackers might try to obtain the outsourced file and dig the private information. As a result, data confidentiality is a security problem\cite{4}. Secondly, the outsourced file may be polluted during the storage process for the following factors. That is, the hackers may destroy the outsourced file maliciously. Moreover, the cloud may remove some data blocks that are rarely accessed arbitrarily\cite{5}. Finally, the cloud may maliciously keep the data blocks. That is when some data blocks are no longer needed, the user is willing to remove them. However, the user loses direct control over the outsourced data due to the separation of
outsourced data management and ownership. Thus, the data erasure operation is transferred to the cloud. However, the cloud may maliciously keep the data blocks for economic interests[6]. In short, despite lots of preponderances, cloud storage service is subjected to plenty of severe security problems, for example, data confidentiality, data integrity and data erasure. The above problems, if not solidly solved, they will greatly impede the application of cloud storage service[7].

1.1. Our contributions
In our paper, we present a fine-grained outsourced data erasure protocol, and the contributions are as follows.

- We adopt RMHT to design a new fine-grained and provable outsourced data erasure protocol, which is able to simultaneously achieve provable data storage and erasure. If the data blocks are not maintained/deleted honestly by the cloud, the user is able to discover the malicious acts with a non-negligible probability.
- The proposed protocol is able to achieve (both the private and public) verifiability without any TTP, which is much better than plenty of the previous protocols. Additionally, we offer the scheme analysis, including the security analysis and performance comparison, which can indicate the security and practicality of our protocol.

1.2. Related works
Outsourced data erasure has been well investigated and emerged many solutions in very recent years[8]-[14]. Bonehand and Lipton[15] presented the first cryptography-based data erasure protocol. In 2016, Hao et al.[16] proposed a trusted-but-verify paradigm, which is called a trusted platform module (TPM). After that, they designed a provable data erasure protocol based on TPM. Dulin et al.[17] proposed an assured data erasure scheme named ADM, which utilized MHT and re-deleting sequences to achieve multi-copy outsourced data erasure. Yang et al.[18] designed a publicly provable outsourced data erasure scheme based on invertible Bloom filter (IBF). Xue et al.[19] designed a key-policy attribute-based encryption protocol. Their protocol is able to reach fine-grained access control and assured data erasure. Specifically, they utilized attribute revocation and MHT to achieve verifiability of erasure outcome. Recently, to solve the problem of data erasure in fog computing, Yu et al.[20] put forward a ciphertext-policy-based secure erasure algorithm. However, all the above solutions require to introduce a trusted third party (TTP), which could be the bottleneck to impede the application of the outsourced data erasure system.

To remove the TTP, Paul and Saxena[21] proposed an ensuring comprehensive data erasure scheme. Their proposed scheme achieves data erasure by overwriting. In their scheme, they have to overwrite all the backups in the distributed storage system. As a result, their scheme is not attractive in efficiency. Yang et al.[22] adopted blockchain to construct a publicly verifiable outsourced data erasure protocol, which does not require any TTP. However, their scheme cannot achieve fine-grained data erasure. Luo et al.[23] utilized overwriting model to design an assured erasure scheme without TTP. However, their scheme relies on an ideal assumption. Meanwhile, the verification may be failed due to network instability. Very recently, Yang et al.[24] adopted vector commitment (VC) to construct a novel scheme, which can realize provable data migration and erasure. However, their scheme needs to execute some complex calculations, such as modular exponential computation and bilinear pairings calculation. As a result, it is very meaningful to design more efficient fine-grained and provable data erasure schemes without requiring any TTP.

1.3. Organization
This paper is organized as follows. We describe the preliminary of RMHT in Section 2, which is the building block of the proposed protocol. The system model, security problems and design goals are presented in Section 3. The concrete scheme and detailed scheme analysis are respectively presented in Section 4 and Section 5. Finally, we conclude this paper simply in Section 6.
2. Preliminaries
Rank-based Merkle Hash Tree (RMHT) could be viewed as an extension of the traditional Merkle hash tree (MHT)[25]. In the RMHT, in order to obtain a hash value in the corresponding node, the inputs are the concatenation of its two children and its rank, where rank is the number of leaf nodes this node contains, as demonstrated in Figure 1. In every leaf node, the hash value $h_i = H(1||d_i)$, where $i = 1, 2, 3, 4$. In the internal node, $h_a = H(2||h_1||h_2)$, $h_b = H(2||h_3||h_4)$ and the root node $h_r = H(4||h_a||h_b)$. Similarly, a leaf node’s auxiliary verification object $\phi$ is a set of the hash values on the sibling path from the specific leaf to the root. For example, for verifying data block $d_2$, the auxiliary verification object is $\phi = (h_3, h_4)$.

![Figure 1. An Example of RMHT](image)

3. Problem statement

3.1. System model
We investigate the scenario that the user outsources his data blocks to the cloud, and deletes them when they are no longer needed. Therefore, in our system model, there is a user and a cloud, as demonstrated in Figure 2.

![Figure 2. The system model](image)

The user has large-scale personal data. However, the user is a resource-constraint entity, thus, he cannot maintain so many data blocks locally. Therefore, the user outsources his data blocks to the cloud. Meanwhile, when the outsourced data blocks are no longer needed, the user may require the cloud to remove them and check the returned data erasure outcome. The cloud has large-scale storage capacity. Thus, the cloud provides convenient and high-quality data storage service for the user and benefits from it. Moreover, when the user will not need the data anymore, the cloud deletes the related outsourced data and returns a corresponding data erasure evidence, which is utilized to check the data erasure outcome by the user.
3.2. Security challenges and goals
In the proposed protocol, we have to consider the following problems and achieve corresponding design goals. Firstly, data pollution is a primary concern of the user. On the one hand, the attackers might maliciously access and destroy the outsourced file. On the other hand, the dishonest cloud may remove some data blocks that are rarely accessed for saving storage overhead. However, the user cannot discover the malicious destroy easily. Secondly, the outsourced data might contain some sensitive information. Therefore, the cloud might not perform data erasure operation honestly. Further, the cloud could dig much private information from dishonest data backups.

Based on the above security problems, we should achieve the following two security goals in the proposed protocol. On the one hand, the proposed protocol should guarantee the outsourced data integrity. This is, the user can verify the outsourced data integrity efficiently. If the outsourced data is polluted, the user is able to discover the data pollution through data integrity checking. On the other hand, the dishonest cloud may remove some data blocks that are rarely accessed for saving storage overhead. However, the user maliciously access and destroy the outsourced file. On the other hand, the dishonest cloud may

4. The concrete scheme
We minutely present our new protocol in this part, which contains six polynomial time algorithms. Initialize. The user first chooses a secure hash function \( H(\cdot) \) (e.g., SHA-1). Meanwhile, the user chooses a file name \( n_f \) for the outsourced file \( F \), where \( n_f \) is unique and \( F \) is the file that is needed to outsource to the cloud. Moreover, we could assume that the user and the cloud respectively has ECDSA public/private key pairs \( (PK_u, SK_u) \) and \( (PK_s, SK_s) \).

Outsource. For saving storage overhead and computation cost, the user outsources his file to the cloud. Specifically, the user firstly executes data encryption operation \( C = Enc_K(F) \), where \( Enc \) is a secure symmetrical encryption scheme and \( K = H(n_f \ || SK_u) \). Then, the user may divide the ciphertext \( C \) into \( n \) data blocks \( (C_1, \ldots, C_n) \). Subsequently, the user uploads the data set \( D \) to the cloud, where \( D = (n_f, C_1, \ldots, C_n) \).

Store. Upon receiving \( D \), the cloud constructs a RMHT to maintain the outsourced data set \( D \). Specifically, the cloud maintains the data block \( C_i \) in the leaf node \( i \) of the RMHT. Meanwhile, the cloud can obtain the root node \( R \). Subsequently, the cloud generates an ECDSA signature on the root node \( R \). Specifically, the cloud computes \( sig_r = Sign_{SK_s}(H_R) \). Finally, the cloud returns \( sig_r \) and \( R \) to the user.

StoCheck. The user can check the storage result. Specifically, the user randomly chooses a data block \( C_x \), where \( 1 \leq x \leq n \). Then, the user obtains the corresponding auxiliary verification object \( \phi_x \). Further, the user can utilize \( C_x \) and \( \phi_x \) to re-compute a new root node \( R_s \), and then compares it with \( R \). Meanwhile, the user verifies the signature \( sig_r \). If and only if both the checkings succeed, the user believes which the cloud honestly maintains the data blocks.

Delete. When data block \( C_y \) (\( 1 \leq y \leq n \)) is no longer needed, the user may send an erasure command \( DR \) to the cloud to remove \( C_y \). Subsequently, the cloud deletes data block \( C_y \) from the outsourced data set. Meanwhile, the cloud deletes the leaf node \( y \) of the RMHT, as demonstrated in Figure 3. After that, the cloud uses the remained data blocks to reconstruct a new rank-based Merkle hash tree RMHT'. At last, the cloud returns the new root node \( R' \), signature \( sig_r' \) and auxiliary verification object \( \phi_y \) to the user.

DelCheck. On receipt of \( R', sig_r' \) and \( \phi_y \), the user is able to check the data erasure outcome. Firstly, the user utilizes \( \phi_y \) to re-compute a new root node \( R'_s \), and compares it with \( R' \). Meanwhile, the user checks the signature \( sig_r' \). Only the equation \( R' = R'_s \) holds and signature \( sig_r' \) is valid, the user trusts that the cloud honestly removes the data blocks.
5. Scheme analysis

5.1. Security analysis

Theorem 1. The proposed protocol can achieve provable data erasure.

Proof. The user can utilize the auxiliary verification object to re-compute a new root value $R'$ and then compares it with $R$ that received from the cloud. The user might have removed some blocks before, which may change the path from one block to the root node, even during this data erasure process the path is also changing. That is, there will be plenty of paths which are able to reach the root node from the remaining leaf nodes. We assume that there are $l$ blocks left in the RMHT', there are $\frac{(2L-2)!}{k!(k-1)!}$ ways to reach the root node of RMHT'. Therefore, the cloud could not guess the right path with a non-negligible probability if it does not honestly remove the data block. Further, the cloud almost cannot generate the correct root $R'$. Therefore, if the root node passes the verification, the cloud had indeed removed the block. Moreover, the verification phase does not require any private information. Hence, the proposed protocol can achieve publicly verifiable data erasure.

5.2. Efficiency evaluation

In this following part, we describe the efficiency comparison of the proposed protocol and the existing scheme[24]. In Table 1, we respectively utilize symbols $n$, $E$ and $H$ to represent the number of outsourced data blocks, an AES encryption performance and a hash calculation. Meanwhile, we denote by $S$ a ECDSA signature operation, $V$ a signature checking operation, $Exp$ a modular exponentiation computation. Moreover, symbols $P$ and $M$ represent a computation of bilinear pairings and a multiplication. Then, the theoretical comparison outcomes are presented in Table 1.

From Table 1, we are able to obtain the following discoverings. Firstly, in data encryption process, scheme[24] needs more hash calculations and modular exponentiations. Secondly, in data storage process, scheme[24] needs $n(n-1)$ modular exponentiations and $n(n-2)$ multiplication calculations. However, the proposed protocol only needs a signature and $(2n-1)$ hash computations. Thirdly, note that bilinear pairings are much more inefficient than hash computation and signature verification operation. That is, scheme[24] costs more overhead to verify data storage and erasure results. Finally, to delete a data block, scheme[24] requires two signature generations, a signature verification and a modular exponentiation. The proposed protocol only costs two signature generations.
and \( \log_2 n \) hash calculations. From the above analysis, we can say that the proposed protocol is more efficient than the existing scheme[24].

| Table 1. Comparison Between Two Schemes |
|------------------------------------------|
| Computation(Encrypt)                     | Scheme[24]                      | Our Scheme                      |
|                                          | \( 1E + (2n + 1)H + n\text{Exp} \) | \( 1E + 1H \)                   |
| Computation(Store)                       | \( n(n - 1)\text{Exp} + n(n - 2)M \) | \( (2n - 1)H + 1S \)           |
| Computation(StoCheck)                    | \( 2nP + qM \)                  | \( 1V + \log_2 nH \)           |
| Computation(Delete)                      | \( 2S + 1V + 1\text{Exp} \)     | \( 2S + \log_2 nH \)           |
| Computation(DelCheck)                    | \( 2P + 1\text{Exp} \)         | \( 1V + \log_2 nH \)           |

6. Conclusions
In this paper, an RMHT-based novel scheme is proposed for cloud storage, which can achieve publicly provable and fine-grained data erasure. Anybody who owns the corresponding data erasure evidence (i.e., verifier) can conveniently and efficiently verify the data erasure outcome. Meanwhile, the proposed protocol can achieve outsourced data confidentiality that can ensure only the user is able to decrypt the outsourced ciphertext correctly. Moreover, the performance evaluation is able to show the efficiency and practicability of the proposed protocol.

Acknowledgment
This work was supported by the National Natural Science Foundation of China under Grant 61962015 and the Natural Science Foundation of Guangxi under Grant 215 2016GXNSFAA380098.

References
[1] C. Yang, J. Ye. (2015) Secure and efficient fine-grained data access control scheme in cloud computing. Journal of High Speed Networks, 21(4): 259-271.
[2] J. Wei, W. Liu, X. Hu. (2018) Secure data sharing in cloud computing using revocable-storage identity-based encryption. IEEE Transactions on Cloud Computing, 6(4): 1136-1148.
[3] C. Yang, X. Tao, F. Zhao, Y. Wang. (2020) Secure data transfer and deletion from counting bloom filter in cloud computing. Chinese Journal of Electronics, 29(2): 273-280.
[4] J. Hong, K. Xue, Y. Xue, W. Chen, D. S. L. Wei, N. Yu, P. Hong. (2020) TAFC: Time and attribute factors combined access control for time-sensitive data in public cloud. IEEE Transactions on Services Computing, 13(1): 158-171.
[5] W. Guo, S. Qin, F. Gao, H. Zhang, W. Li, Z. Jin, Q. Wen. (2020) Comments on provable multicopy dynamic data possession in cloud computing systems. IEEE Transactions on Information Forensics and Security, 15 2584-2586.
[6] Y. Tang, P. P. C. Lee, J. C. S. Lui, R. Perlman. (2012) Secure overlay cloud storage with access control and assured deletion. IEEE Transactions on Dependable and Secure Computing, 9(6): 903-916.
[7] C. Yang, Y. Liu, X. Tao, F. Zhao. (2020) Publicly verifiable and efficient fine-grained data deletion scheme in cloud computing. IEEE Access, 8 99393-99403.
[8] J. Reardon, D. A. Basin, S. Capkun. (2013) Sok: Secure data deletion. in: Proceedings of the 2013 IEEE Symposium on Security and Privacy. Berkeley. pp. 301-315.
[9] J. Xiong, X. Liu, Z. Yao, J. Ma, Q. Li, K. Geng, P. S. Chen. (2014) A secure data self-destroying scheme in cloud computing. IEEE Transactions on Cloud Computing, 2(4): 448-458.
[10] C. Yang, X. Tao, F. Zhao. (2019) Publicly verifiable data transfer and deletion scheme for cloud storage. International Journal of Distributed Sensor Networks, 15(10): 1-12.
[11] Y. Tang, P. P. C. Lee, J. C. S. Liu, R. Perlman. (2010) Fade: Secure overlay cloud storage with file assured deletion. in: Proceedings of the SecureComm 2010 - 6th International ICST Conference on Security and Privacy in Communication Networks. Shanghai. pp. 380-397.

[12] C. Yang, X. Tao. (2018) New publicly verifiable cloud data deletion scheme with efficient tracking. in: Proceedings of the Second International Conference on Security with Intelligent Computing and Big-Data Services. Xi’an. pp. 359-372.

[13] B. Hall, M. Govindarasu. (2018) An assured deletion technique for cloud-based IoT. in: Proceedings of the 27th International Conference on Computer Communication and Networks. Xi’an. pp. 1-9.

[14] J. Lai, J. Xiong, C. Wang, G. Wu, Y. Li. (2017) A secure cloud backup system with deduplication and assured deletion. in: Proceedings of the 11th International Conference on Provable Security. Xi’an. pp. 74-83.

[15] D. Boneh, R. Lipton. (1996) A revocable backup system. in: Proceedings of the Sixth USENIX Security Symposium. San Jose. pp. 91-96.

[16] F. Hao, D. Clarke, A. F. Zorzo. (2016) Deleting secret data with public verifiability. IEEE Transactions on Dependable and Secure Computing, 13(6): 617-629.

[17] Dulin, Z. Zhang, S. Tan, J.Wang, X. Tao. (2018) An associated deletion scheme for multi-copy in cloud storage. in: Proceedings of the 18th International Conference on Algorithms and Architectures for Parallel Processing. Guangzhou. pp. 511-526.

[18] C. Yang, X. Tao, F. Zhao, Y.Wang. (2019) A new outsourced data deletion scheme with public verifiability. in: Proceedings of the 14th International Conference on Wireless Algorithms, Systems, and Applications. Hawaii. pp. 631-638.

[19] L. Xue, Y. Yu, Y. i, M. H. Au, X. Du, B. Yang. (2018) Efficient attribute-based encryption with attribute revocation for assured data deletion. Information Sciences, 479 640-650.

[20] Y. Yu, L. Xue, Y. Li, X. Du, M. Guizani, B. Yang. (2018) Assured data deletion with fine-grained access control for fog-based industrial applications. IEEE Transactions on Industrial Informatics, 14(10) 4538-4547.

[21] M. Paul, A. Saxena. (2010) Proof of erasability for ensuring comprehensive data deletion in cloud computing. in: Proceedings of the Third International Conference on Recent Trends in Network Security and Applications. Peking. pp. 340-348.

[22] C. Yang, X. Chen, Y. Xiang. (2018) Blockchain-based publicly verifiable data deletion scheme for cloud storage. Journal of Network and Computer Applications, 103 185-193.

[23] Y. Luo, M. Xu, S. Fu, D. Wang. (2016) Enabling assured deletion in the cloud storage by overwriting. in: Proceedings of the 4th ACM International Workshop on Security in Cloud Computing. Guangzhou. pp. 17-23.

[24] C. Yang, J. Wang, X. Tao, X. Chen. (2018) Publicly verifiable data transfer and deletion scheme for cloud storage. in: Proceedings of the 20th International Conference on Information and Communications Security. Chongqing. pp. 445-458.

[25] C. Yang, Q. Chen, Y. Liu. (2019) Fine-grained outsourced data deletion scheme in cloud computing. International Journal of Electronics and Information Engineering. 11(2): 81-98.