ISSN: 2321-2152

IJMECE

International Journal of modern
electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

AUTHORIZE AND RELIABILITY PHRASE CHECKING WITH AWARE OF DATA SECURITY FOR ENCRYPTED CLOUD DATA

Spurthi K[1] , Dr.Kishore Kumar K[2] , Mr. G Naga Kumar Kakarla[3]

[1]Assistant Professor, Dept of CSE, Siddhartha Institute of Technology & Sciences, Narapally, Hyderabad, 500088, India.

[2]Associate Professor Dept of CSE, Siddhartha Institute of Technology & Sciences, Narapally, Hyderabad, 500088, India.

[3]Associate Professor, Computer Science and Engineering Gokaraju Lailavathi Womens Engineering College Hyderabad, 500049

ABSTRAT

Public cloud data integrity auditing technology is used to verify the integrity of cloud data through a third-party auditor (TPA). To make it more practical, we propose a new model called integrity auditing based on the keyword privacy of sensitive information for encrypted data in the cloud. This model is designed for one of the most common scenarios, which is a user's concern about the integrity of a portion of encrypted files in the cloud that contain keywords of interest. In our proposed scheme, only a TPA provided with the encrypted keyword can audit the integrity of all encrypted files in the cloud containing the user's interested keyword. At the same time, TPA cannot infer sensitive information about which files contain the keyword and how many files contain this keyword. These prominent features are achieved by taking advantage of the recently proposed Relationship Authentication Tag (RAL). Not only can RAL authenticate the relationship containing files containing the queried keyword, but it can also be used to create audit proof without revealing sensitive information. We provide a concrete security analysis that shows that the proposed scheme satisfies the authenticity, robustness, and privacy of sensitive information. We also conduct detailed experiments to demonstrate the efficiency of our scheme.

Keywords: Cloud storage, sensitive information privacy, keyword search, data auditing, privacy

I INTRODUCTION

Cloud storage allows people to easily outsource large amounts of their data to centralized cloud servers. Taking the electronic medical record (EMR) as an example, doctors can upload patients' electronic medical records to cloud servers, which will then be accessed by other doctors from different departments. The integrity of electronic medical records is of great importance, since tampered with electronic medical records may cause an incorrect diagnosis, or even the death of the patient. Cloud data integrity auditing techniques can verify whether users' files are properly stored in the cloud. The safety audit task is usually performed by a third-party auditor (TPA) with strong computational capabilities that the user does not possess. In general, TPA typically adopts a “pay-as-you-go” model of charging users according to the workload of the audit services it provides. The more cloud files are audited, the more money the user needs to pay. The Internet Data Center estimates that the data held by each user will reach 5,200 GB in 2020 [1]. When such large-scale files are transferred to the cloud, periodically auditing the integrity of all cloud files would impose a heavy economic burden on the user. Moreover, it will cause unavoidable waste of resources. In most cases, the user may only care about the safety of certain files that will be used soon. For example, when a patient comes to a hospital for treatment, the doctor only cares about the safety of that patient's electronic medical records. The doctor may search and extract these electronic medical records from the cloud according to the identity of this patient. When medical scientists conduct research on diabetes, they may only care about the safety of electronic medical records that contain the keyword “diabetes” or “GLU” in the cloud. In these scenarios, it may be reasonable and cost-effective to audit the integrity of only files that contain the keyword of interest. Since keywords in files often contain user-sensitive information, the user needs to encrypt the files before uploading them to the cloud. When a user wants to verify the integrity of all encrypted cloud files containing the keyword in question, he or she only provides the TPA with the encrypted keyword (the search trapdoor). This makes achieving integrity auditing based on keyword encrypted cloud data more difficult. In short, it faces two critical challenges. The first challenge is how to audit the integrity of all encrypted cloud files containing the queried keyword provided that the TPA is only provided through the search trapdoor. When TPA does not know which files contain this queried keyword, the malicious cloud may provide a valid directory computed from files that do not contain this keyword or a portion of files that do contain this keyword to pass the verification process. The second challenge is that, by performing the integrity audit, TPA does not have to know which files contain this queried keyword, or how many files contain this queried keyword. Since the TPA's mission is only to conduct the integrity audit, this sensitive information should not be disclosed to the TPA. Sensitive information may reveal the most important encrypted keyword and even reveal the internal relationship between files. In order to address the above challenges, we explore how to achieve integrity auditing based on the sensitive information privacy keyword of encrypted cloud data. The contributions of this paper can be summarized as follows: (1) We propose a new model called integrity auditing based on the sensitive information privacy keyword of encrypted cloud data. Unlike previous schemes, TPA can verify the integrity of all encrypted cloud files containing only one specific keyword through the search door of such a scheme. Evidence from the cloud can only pass TPA verification if the cloud correctly stores all encrypted files containing this keyword. In addition, TPA cannot obtain any sensitive information, for example, which files contain the queried keyword and how many files contain the queried keyword. Current businesses cannot achieve such security. Therefore, this new model is different from traditional cloud data integrity auditing. In integrity auditing procedures, our proposal is that only with Oð1Þ computation complexity and communication complexity in terms of N total number of files containing the queried keyword, outperforms OðNÞ complexity in data auditing based on verifiable and searchable encryption. (2) We propose the first integrity auditing scheme based on the sensitive information privacy keyword for encrypted cloud data. To build this scheme, we designed a new label model called Relation Authentication Label (RAL). This mark plays an important role in achieving our design goals. On the one hand, RAL can authenticate the relationship that keyword files contain. On the other hand, it can be used to create a pro-audit

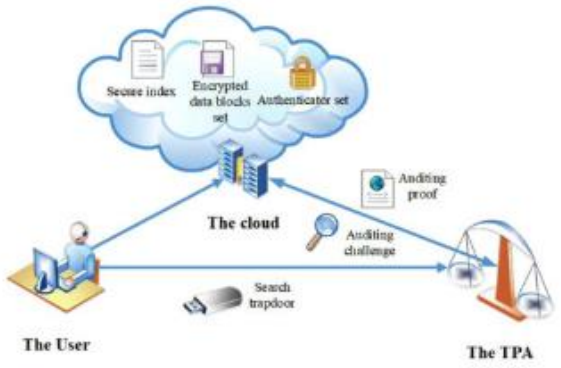

Fig. 1. System model

II EXISTING SYSTEM

As shown in Figure 1, the system model in the proposed scheme consists of three entities: user, cloud and TPA. the user. It is the person who wants to store a large number of encrypted files in the cloud. It creates the secure index and authentications, and uploads them along with the encrypted file blocks to the cloud. To allow TPA to perform the audit task on files that contain the query keyword, send the search gate to TPA. Cloud. It is an entity with enormous storage capacity and computing power. When an audit trail receives a certain keyword, it first searches the secure index for the corresponding encrypted files. It then calculates the audit evidence according to the audit challenge and sends it back to the TPA. TPA. He is the one who performs the audit task on behalf of the user. It interacts with the cloud in the audit phase, verifying the integrity of all files containing the queried keyword. Cloud. Data stored in the cloud can be modified or deleted without the user's consent. To make matters worse, the cloud will hide cases of data corruption. The cloud tries to trick the TPA into accepting its audit proof when it doesn't have all the data. Furthermore, the cloud is curious about the plain text of the queried file/keyword and the relationship between the file and the queried keyword. Determined. TPA is honest in verifying the integrity of users' files. Furthermore, it does not actively carry out leak and misuse attacks. Therefore, in this article we do not consider forward and backward specificity in TPA. However, the sensitive information in the file is interesting. It tries to infer the plain text of the keyword and the file. It is also interesting to know the identities and the number of files that contain the queried keyword.

III PROPOSED SYSTEM

We first present two straightforward methods to achieve keyword-based integrity auditing of encrypted data in the cloud. The first is a naive approach that requires the cloud to return all files containing the queried keyword to the TPA in the audit phase. It will bear the heavy burden of communication. Furthermore, this approach cannot achieve the privacy of sensitive information. The second method is a little better, it has higher communication efficiency, but is still capable of revealing sensitive information to the TPA. We then present our basic scheme to achieve keyword-based integrity auditing with sensitive information privacy on encrypted data in the cloud. This approach is designed in part based on verifiable and searchable encryption (VSE) technology[5]. We call it VSE-based data audit. The user and the TPA share a secret key for a MAC algorithm. The cloud stores encrypted files and a secure index with MAC array. To generate a MAC value on this MAC set, the user runs the MAC algorithm by entering each encrypted keyword and all encrypted files containing that keyword. When TPA wants to verify the integrity of files containing the specified keyword, it sends the encrypted keyword as audit proof to the cloud. Based on the secure index, the cloud finds all encrypted files that contain this keyword. It then returns it along with the MAC value corresponding to the TPA. Since TPA holds the secret key of the MAC algorithm, it can verify whether this MAC value is valid based on the received encrypted files. If valid, it means that all files containing this keyword are intact. However, in this approach, the cloud needs to return to the TPA all MACs and files containing the queried keyword. Suppose there are N files containing the query keyword in total. OðNÞ will bear the overhead costs of communication in integrity audit procedures. Additionally, the TPA must independently verify the authenticity of these MACs based on the OðNÞ files received. OðNÞ will incur indirect accounting costs in integrity audit procedures. Obviously, VSE-based data auditing is not efficient, especially when the number or size of files containing this keyword is large. Furthermore, it is inevitable that sensitive information, such as files containing this query keyword, will be exposed to the TPA. Therefore, this approach is not practical. Similar to the first method, the user and TPA also share a secret key for a MAC algorithm in this method. Additionally, the cloud stores encrypted files and a secure index with MAC array. Unlike the first method, the user sets the encrypted keyword and corresponding file identities as input to the MAC.

SYSTEM ARCHITECTURE:

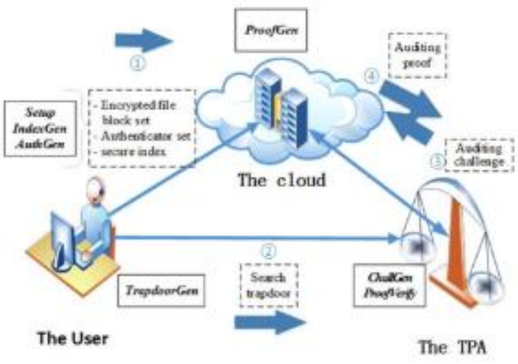

FIGURE 2. The process of the proposed scheme

A. Our Core Scheme

- **SysIniðÞ!ðpp; x; y:** Choose the system parameters pp as follows: two q order multiplicative cyclic groups G1; G2, a bilinear map e : G1 G1 ! G2, two generators u; g 2 G1, three secure hash functions H1 : f0; 1g ! G1; H2 : f0; 1g ! G1; H3 : f0; 1g ! G1, a symmetric encryption algorithm Encð ; k0Þ with key k0, a pseudo random permutation(PRP) pk1 ð Þ with key k1, and a pseudo random function(PRF) fk2 ðÞ with key k2. For simplification, we will use pðÞ to denote pk1 ðÞ and fðÞ to denote fk2 ðÞ in the detailed scheme. b) Randomly choose the secret key for the user x 2 Z——————————
q , and the corresponding public key y ¼ gx

Algorithm 1. IndexGen

Input: The secret key x, the keyword set W, the index vector set V . Output: The secure index I.

```
1: for each wk 2 Wð1 k mÞ do
2: Extract vwk from V ;
3: Compute pðwkÞ;
4: Compute evpðwkÞ ¼ vwk fðpðwkÞÞ;
5: Create an empty set Swk ¼ ;;
6: for each i 2 ½1; n do
7: if vwk ½i ¼¼ 1 then
8: Add i to set Swk ;
9: end if
10: end for
11: for each j 2 ½1; sdo
12: Compute:
```

Algorithm 2. ProofGen

Input: The auditing challenge Chal, the secure index I, the encrypted data block set C, the authenticator set F. Output: The auditing proof Proof.

```
1: Extract the auditing challenge Chal ¼ fTw0 ; fj; vjgj2Qg, where Tw0 ¼ fpðw0 Þ; fðpðw0 ÞÞg;
2: Search the corresponding row in the secure index, where pðwkÞ ¼ pðw0 Þ;
3: Compute vwk ¼ evpðwkÞ fðpðwkÞÞ;
4: Initiate an empty set: Swk ¼ ;;
5: for each i 2 ½1; n do
6: if vwk ½i ¼¼ 1 then
```

```
        Add i to Swk ;
    end if
end for
```

IV Experiment Results

We utilize C-programming language, GMP library[20] and Pairing-Based-Cryptography (PBC) library[21] to simulate the proposed scheme. We test our experiment on Ubuntu 16.04 LTS with 1 CPU, 25 GB Storage, and 1 GB RAM. Since the computation overhead mainly comes from keyword searching and cloud data auditing, we test the efficiency of them, respectively. In our experiments, we utilize type-A pairing with 160 bits group order and 512 bits base field order. The length of each element in Zq is 20 bytes, and the length of each element in G1 is 128 bytes

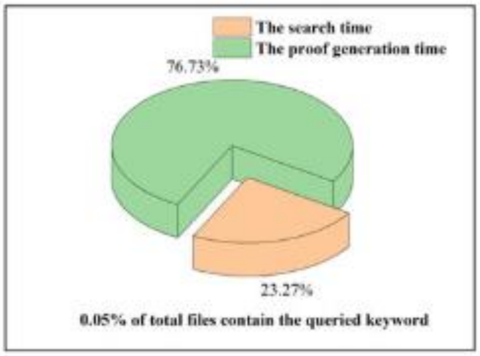

Fig no 3: The challenge generation time

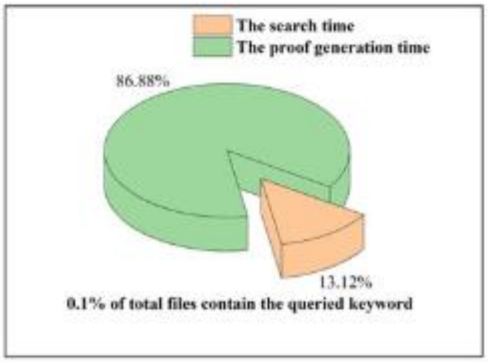

Fig no 4: The proof verification time.

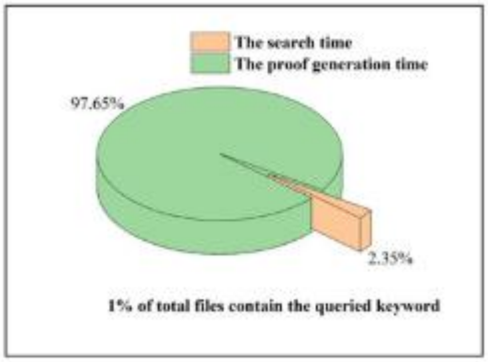

Fig no 5: The computation overhead comparison

CONCLUSION

We address a new problem of how to achieve a data integrity audit in the cloud based on the privacy keyword of sensitive information. We have designed a new tag called RAL, which is used to not only authenticate the relationship containing the files containing the query keyword, but also to create audit evidence without revealing any identity of the file containing the query keyword. We demonstrate the security of the proposed scheme and evaluate its practical effectiveness through comprehensive experiments.

REFERANCES

[1] By 2020, there will be 5,200 GB of data for every person on earth. Accessed: Aug. 2021. [Online]. Available: https://www. computerworld.com/article/2493701/

[2] X. Ge, J. Yu, C. Hu, H. Zhang, and R. Hao, "Enabling efficient verifiable fuzzy keyword search over encrypted data in cloud computing," IEEE Access, vol. 6, pp. 45725–45739, 2018.

[3] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," J. Comput. Secur., vol. 19, no. 5, pp. 895–934, 2011.

[4] S. Kamara and C. Papamanthou, "Parallel and dynamic searchable symmetric encryption," in Proc. Int. Conf. Financial Cryptography Data Secur., 2013, pp. 258–274.

[5] W. Sun, X. Liu, W. Lou, Y. T. Hou, and H. Li, "Catch you if you lie to me: Efficient verifiable conjunctive keyword search over large dynamic encrypted cloud data," in Proc. IEEE Conf. Comput. Commun., 2015, pp. 2110–2118.

[6] G. Ateniese et al."Provable data possession at untrusted stores," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007, pp. 598–609.

[7] G. Yang, J. Yu, W. Shen, Q. Su, Z. Fu, and R. Hao, "Enabling public auditing for shared data in cloud storage supporting identity privacy and traceability," J. Syst. Softw., vol. 113, pp. 130–139, 2016.

[8] Y. Yu, J. Ni, M. H. Au, Y. Mu, B. Wang, and H. Li, "Comments on a public auditing mechanism for shared cloud data service," IEEE Trans. Serv. Comput., vol. 8, no. 6, pp. 998–999, Nov./Dec. 2015.

[9] R. Bost, P.-A. Fouque, and D. Pointcheval, "Verifiable dynamic symmetric searchable encryption: Optimality and forward security," IACR, Lyon, France, Rep. 2016/062, 2016.

[10] X. Zhu, Q. Liu, and G. Wang, "A novel verifiable and dynamic fuzzy keyword search scheme over encrypted data in cloud computing," in Proc. IEEE Trustcom/BigDataSE/ISPA, 2016, pp. 845–851.

[11] X. Ge et al., “Towards achieving keyword search over dynamic encrypted cloud data with symmetric-key based verification,” IEEE Trans. Dependable Secure Comput., vol. 18, no. 1, pp. 490–504, Jan./Feb. 2021.

[12] J. Mao, Y. Zhang, P. Li, T. Li, Q. Wu, and J. Liu, “A position-aware merkle tree for dynamic cloud data integrity verification,” Soft Comput., vol. 21, no. 8, pp. 2151–2164, 2017.

[13] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, “Enabling public verifiability and data dynamics for storage security in cloud computing,” in Proc. Eur. Symp. Res. Comput. Secur., 2009, pp. 355–370.

[14] D. Cash, A. Kupc € ¸u, and D. Wichs, “Dynamic proofs of retrievabil- € ity via oblivious RAM,” J. Cryptol., vol. 30, no. 1, pp. 22–57, 2017.

[15] C. C. Erway, A. Kupc € ¸u, C. Papamanthou, and R. Tamassia, € “Dynamic provable data possession,” ACM Trans. Inf. Syst. Secur., vol. 17, no. 4, pp. 1–29, 2015.

[16] J. Shen, J. Shen, X. Chen, X. Huang, and W. Susilo, “An efficient public auditing protocol with novel dynamic structure for cloud data,” IEEE Trans. Inf. Forensics Secur., vol. 12, no. 10, pp. 2402– 2415, Oct. 2017.

[17] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-preserving public auditing for data storage security in cloud computing,” in Proc. IEEE INFOCOM, 2010, pp. 1–9.

[18] M. S. Islam, M. Kuzu, and M. Kantarcioglu, “Access pattern disclosure on searchable encryption: Ramification, attack and mitigation,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2012, pp. 1–15.

[19] D. Cash, P. Grubbs, J. Perry, and T. Ristenpart, “Leakage-abuse attacks against searchable encryption,” in Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur., 2015, pp. 668–679.

[20] The GNU multiple precision arithmetic library (GMP). Accessed: Aug. 2021. [Online]. Available: http://gmplib.org/