International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 02 | Feb 2020 www.irjet.net p-ISSN: 2395-0072

VIRTUAL DATA AUDITING AT OVERCAST ENVIRONMENT

Aruna.K.B[1], Aarthi.S[2], Aishwarya.K.S[3], Arunadevi.P[4]

[1]Assistant Professor, Dept. of Computer Science Engineering, S.A. Engineering College, TamilNadu, India
[2,3,4]Student, Dept. of Computer Science Engineering,S.A. Engineering College, TamilNadu, India

---

ABSTRACT:- *A cloud storage is used to store large amount of data. It provides a storage platform for businesses and individuals. It can store and access data remotely using the cloud storage system. A remote data integrity check is suggested to ensure the integrity of the data stored in the cloud. In electronic health care system - records are stored and managed remotely, the cloud file may contain sensitive information. The confidential information should not be shared with anyone when the cloud file is released. By encrypting the entire shared file, the confidential information can be hidden, but this shared file cannot be used by others. A remote data integrity checking scheme is proposed that implements data exchange with confidential information that is hidden in the document. A disinfection program is used to clean up the data blocks that correspond to the confidential information of the file and to convert the signatures of these data blocks into valid signatures for the cleaned file. As a result, the scheme allows the file stored in the cloud to be shared and used by others, provided the confidential information is hidden, while the remote data integrity check can continue to run efficiently.*

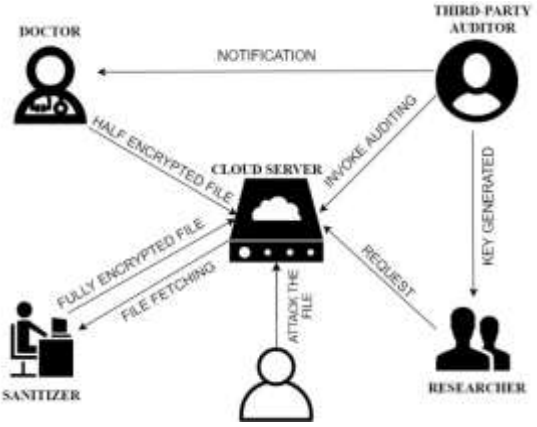
Keywords: *Cloud Storage, Data Integrity Auditing, Data Sharing, Sensitive Information Hiding*

1. INTRODUCTION

Due to the explosive growth in data, it is a major burden for users to store the sheer volume of data locally. Therefore, more and more organizations and individuals want to store their data in the cloud. However, data stored in the cloud can be damaged or lost due to inevitable software, hardware, and human errors in the cloud. To verify that the data is stored correctly in the cloud, many remote data integrity verification schemes have been proposed. With remote data integrity checking schemes, the data owner must first generate signatures for blocks of data before uploading them to the cloud. These signatures are used to prove that the cloud actually owns these data blocks in the integrity check phase. The data owner then uploads these data blocks together with the corresponding signatures to the cloud. The data stored in the cloud is often shared by several users in many cloud storage applications such as Google Drive, Dropbox and iCloud. Sharing data as one of the most common features in cloud storage allows a number of users to share their data with others. However, this shared data stored in the cloud may contain confidential information. The integrity of the EHRs must be ensured due to human error and software / hardware errors in the cloud. Therefore, it is important to perform a remote data integrity check if the confidential information of the shared data is protected. One possible way to solve this problem is to encrypt the entire shared file before sending it to the cloud, and then generate the signatures that verify the integrity of this encrypted file. This encrypted file and the corresponding signatures are then uploaded to the cloud. This method can detect hiding sensitive information because only the data owner can decrypt this file. However, this means that the entire shared file cannot be used by others. For example, encrypting the EHRs of patients with infectious diseases can protect patient and hospital privacy, but these encrypted EHRs can no longer be used effectively by researchers. Distributing the decryption key to the researchers seems to be a possible solution to the above problem. However, it is not possible to use this method in real scenarios for the following reasons. First, the distribution of the decryption key requires secure channels, which in some cases are difficult to fulfill. In addition, it appears to be very difficult for a user to know which researchers will be using their EHRs in the near future when uploading the EHRs to the cloud. Therefore, it is impractical to hide confidential information by encrypting the entire shared file. It is therefore very important and valuable how the data exchange can be realized with confidential information that is hidden in the remote data integrity check. Unfortunately, this problem has not been researched in previous studies.

2. SYSTEM MODEL

The user hides the data blocks that correspond to the personal confidential information of the original file, generates the corresponding signatures and then sends them to a sanitizer.The sanitizer cleans up these blended blocks of data in a uniform format and also purges the blocks of data that correspond to the organization's confidential information.In addition, the corresponding signatures are converted into valid signatures for the cleaned file.This method not only implements remote data integrity checking, but also supports data sharing on the condition that confidential information in the cloud storage is protected. Attackers are detected and blocked by the cloud server.



The system model comprises six types of different entities: the cloud, the user, the Sanitizer, the Private Key Generator (PKG) and the Third Party Auditor (TPA), the Attacker.

(1) Cloud: The cloud offers the user enormous data storage space. Through the cloud. The storage service allows users to upload their data to the cloud and share their data with others.

(2) User: The user is a member of an organization in which a large number of files must be saved in the cloud.

(3) Sanitizer: The sanitizeris responsible for sanitizing the data blocks that the sensitive information (personal sensitive information and sensitive information of the organization Information) in the file, the signatures of these data blocks being valid signatures for the cleaned file and uploading the cleaned file and the corresponding signatures to the cloud.

(4) Private Key Generator(PKG): The PKG is classified as trustworthy by other entities. It is responsible for the creation of the public system parameters and the private key for the user according to their identity ID.

(5) Third Party Auditor(TPA): The TPA is a public examiner. It is responsible for checking the integrity of the stored data in the cloud on behalf of users.

(6) Attacker: The attacker is an unauthorized person who changes the content of the data stored in it the cloud without the user's knowledge.

4. DEFINITION

An identity-based shared data integrity auditing scheme with sensitive information hiding for secure cloud storage consists of the following six algorithms: Initiate, Extricate, DSGen, Sanitize, ProofCreate and ProofAuth. Specifically, these algorithms are described as follows:

[4.1]Initiate(1^k) is an initiation algorithm directed by the Private-Key Generator. A security parameter *k* is used as input. It publishes public parameters of the system *pps* and a secret master key *smk*.

[4.2]Extricate(pps, smk, ID) is an algorithm that is executed by the Private-Key Generator for extraction. The public parameters of the system *pps*, the secret master key *smk* and an identity of the user *ID* are used as input. It prints out the user's private key *pvk*. The user can only check the correctness of *pvk* and accept it as his private key if he passes the check.

[4.3]DSGen(F, pvk, spk, name) is a signature generation algorithm executed by the user ID. The input is the original file *F*, the private key *pvk* of the user, the signing private key *spk* of the user and the name of the file identifier. It outputs a blinded file *bF*, the corresponding signature set *s* and a file tag *t*.

[4.4]Sanitize(bF,s) is a disinfection program algorithm for cleaning up confidential information. The blinded file *bF* and its signature sets *s* are used as input. It outputs the cleaned file *bF* and the corresponding signature set *s*.

[4.5]ProofCreate(bF′,s′, ch) is a proof generation algorithm that is executed by the cloud. The cleaned file *bF′*, the corresponding signature set s′ and the test request *ch* are used as input. It issues a verification certificate *P*, which is used to demonstrate that the cloud really owns this cleaned file *bF′*.

[4.6]ProofAuth(ch, pps, P) is a proof verification algorithm executed by the Third Party Auditor. The test request *ch*, the public parameters of the system *pps* and the test certificate *P* are used as input. The Third Party Auditor can check the accuracy of the proof *P*.

5. ALGORITHM

[5.1] RANDOMIZED - ADVANCED ENCRYPTION STANDARD (R-AES)

The main problem with the key extension of the AES algorithm is that the words generated from the original key are related. If a word is understandable, the overall key is derived by the differential method or the liner methods of cryptanalysis.

We modified the key expansion module from AES with the Symmetric Random Function Generator (SRFG). SRFG generates the symmetric symmetric output in terms of the number of ones and zeros in the output string regardless of the input string.

The expression for the proposed combined function generator is given as**:**

$f_c = f_i^L$

where, four universal gates: AND, OR, NOT and XOR; L represents the length of expression (number of terms in the combined function); and represents the random combination. In the experiments we used L = 5. In order to emphasize the randomness in such a combined function generator, the above equation can be further

expressed in the form of the randomness of the input variables during the selection

$f_c(V_1,V_2,....,V_N)=f_i^L[rand(V_1,V_2,..,V_N)]$

For experimentation, the previous equation can be rewritten as,

$f_c(V_1,V_2)= f_i^5[rand(V_1,V_2)]$

The main goal of adding SRFG in AES is to activate the key expansion module with a random function. This helps prevent the words from being derived from keys even though a subkey is in hand.

[5.2] DIGITAL SIGNATURE

A digital signature is a mathematical strategy to verify the authenticity of digital messages or documents. A valid digital signature that meets the requirements gives a recipient strong reason to believe that the message was created by a known sender (authentication) and that the message was not changed during transmission (integrity).

A digital signature scheme typically made up of 3 algorithms:

• A key generation algorithm that randomly selects a private key from a set of possible private keys. The algorithm provides the private key and a corresponding public key.

• A signature algorithm that generates a signature based on a message and a private key.

• An algorithm for checking the signature, which, given the message, the public key and the signature, either accepts or rejects the authenticity claim of the message.

[5.3] RANDOM KEY CRYPTOGRAPHY

For creating a random key matrix from size (16x16), take every key. The size of the key must be less than or 16 characters long. This 16characters can be any 256 Characters (ASCII code 0 to 255). The relative position and the character itself is very important in the method to calculate the randomization number, the encryption number and the relative shifting of the characters in the start keymatrix. Here we demonstrate a method:

Suppose key = AB

Select the following table to calculate the place value and character strength of the incoming key:

Length Of Key(n)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Base Value(b)	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2

Equation: Sum = Σ ASCII Code * bm ----(1)

m=1

Example-1:

Now calculate the sum for key="AB" using equation(1)

Sum=65*161 + 66 * 162=17936

Now compute 3 parameters from the above sum: (i) randomization number (n1), (ii) encryption number (n2) and (iii) relative shift (n3) with the following procedure:

(i) Randomization number(n1):

num1=1*1+7*2+9*3+3*4+6*5=84

n1=sum mod num1=17936 mod

84=44

Note: if n1=0 then n1=num1 and n1<=128

(ii) Encryption number(n2):

num2=6*1+3*2+9*3+7*4+1*5=72

n2=sum mod num2 =17936

mod 72 =8

Note: if n2=0 then n2=num2 and n2<=64

(iii) Relative shift(n3):

n3= Σall digits in

sum=1+7+9+3+6=26

6. RELATED WORK

[6.1] Cross-Domain Data Sharing in Distributed Electronic Health Record Systems

A secure EHR system based on cryptographic constructions is proposed to enable the secure sharing of sensitive patient data during the collaboration and to ensure the protection of the patient data.

Standard identity-based cryptography (IBC) enables an entity's public key to be derived from public identity information such as name, email address, etc.

With HIBPKI (Hierarchical Identity Based Public Key Infrastructure), entities from different domains can authenticate themselves directly with one another without having to rely on certificates from a certification authority.

Different possible attacks can be applied to the EHR system and should be recognized to take appropriate countermeasures.

[6.2] Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud

A novel public verification mechanism for the integrity of shared data with a view to efficient revocation of users is proposed.

Since collusion-resistant proxy re-signature schemes generally have two levels of signature (that is, the first level is re-signed by a user and the second level is re-signed by the proxy), the two levels signature are different forms and must be verified differently.

The unblockable verifiability at both signature levels and their joint verification in a public verification mechanism is a challenge.

[6.3] Identity-Based Remote Data Integrity Checking With Perfect Data Privacy Preserving for Cloud Storage

A new structure of the identity-based (ID-based) RDIC protocol is proposed using a key-homomorphic cryptographic basic element in order to reduce the system complexity and the costs for setting up and managing the public-key authentication framework in PKI-based RDIC schemes.

ID-based RDIC and its security model is formalized, including security against a malicious cloud server and zero-knowledge data protection against a third-party auditor.

Short signature algorithm: Because of the short signature, hackers can easily find the key.

7. PROPOSED SCHEME

In the proposed scheme we design a new efficient signature algorithm in the signature generation phase. The designed signature scheme supports non-blocking verifiability, allowing the verifier to verify the integrity of data without having to download all of the data from the cloud. In addition, it is based on identity-based cryptography, which simplifies complicated certificate management.

If the user wants to upload data to the cloud to get the personal confidential information of the original file from the disinfectant, that user must use a blinding factor to blind the data blocks that correspond to the personal confidential information of the original file.

If necessary, the user can use this blinding factor to restore the original file from the blinded one. This user then uses the designed signature algorithm to generate signatures for the blinded file.

Finally, the disinfectant uploads the cleaned file and the corresponding signatures to the cloud.

The PKG generates the private key for the user based on his identity ID. When researchers access the file, they are given a key to decrypt the contents of the file, which hides the confidential information.

When the data integrity verification task runs, the cloud generates verification evidence according to the TPA challenge. The TPA can verify the integrity of the cleaned file stored in the cloud by verifying that this verification is correct or not. The unauthorized user who changes the content of the file stored in the cloud can be identified and reported to the user.

8. CONCLUSION

An identity-based data integrity verification scheme for secure cloud storage is suggested that supports data exchange with hidden confidential information. The file stored in the cloud can be shared and used by others, provided the confidential information of the file is concealed. However, the remote data integrity check can be performed efficiently. It shows the safety case and the experimental analysis that the proposed scheme achieves the desired safety and efficiency. The further development of this project can be made to ensure enhanced failure recovery.

REFERENCES

[1] J. Sun and Y. Fang, “Cross-domain data sharing in distributed electronic health record systems,” IEEE Transactionson Parallel and Distributed Systems, vol. 21, no. 6,pp. 754–764, June 2010.

[2] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, “Enablingpublic auditability and data dynamics for storage securityin cloud computing,” IEEE Transactions on Parallel andDistributed Systems, vol. 22, no. 5, pp. 847–859, May2011.

[3] K. Ren, C. Wang, and Q. Wang, “Security challengesfor the public cloud,” IEEE Internet Computing, vol. 16,no. 1, pp. 69–73, Jan 2012.

[4] C. Wang, S. S. M. Chow, Q. Wang, K. Ren, and W. Lou,“Privacy-preserving public auditing for secure cloud storage,”

IEEE Transactions on Computers, vol. 62, no. 2,pp. 362–375, 2013.

[5]HarithaNuthi, HemalathaGoli, Ramakrishna Mathe,” Data Integrity Proof for Cloud Storage” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue 9, September 2014

[6] B. Wang, B. Li, and H. Li, “Panda: Public auditing forshared data with efficient user revocation in the cloud,”

IEEE Transactions on Services Computing, vol. 8, no. 1,pp. 92–106, Jan.-Feb. 2015.

[7] H. Wang, "Identity-based distributed provable data possessioninmulticloudstorage," IEEE Transactions onServices Computing, vol. 8, no. 2, pp. 328–340, 2015.

[8] J. Yu, K. Ren, and C. Wang, "Enabling cloud storageauditing with verifiable outsourcing of key updates,"

IEEE Transactions on Information Forensics and Security,vol. 11, no. 6, pp. 1362–1375, June 2016.

[9] H. Wang, D. He, J. Yu, and Z. Wang, "Incentiveand unconditionally anonymous identity-based publicprovable data possession," IEEE Transactions on Services Computing, 2016. Available: DOI:10.1109/TSC.2016.2633260

[10] J. Li, J. Li, D. Xie, and Z. Cai, "Secure auditinganddeduplicating data in cloud," IEEE Transactions onComputers, vol. 65, no. 8, pp. 2386–2396, Aug 2016.

[11] A. Fu, S. Yu, Y. Zhang, H. Wang, and C. Huang,"Npp: A new privacy-aware public auditing schemefor cloud data sharing with group users," IEEETransactions on Big Data, 2017. Available:DOI:10.1109/TBDATA.2017.2701347

[12] Y. Yu, M. H. Au, G. Ateniese, X. Huang, W. Susilo,Y. Dai, and G. Min, "Identity-based remote data integritychecking with perfect data privacy preserving for cloudstorage," IEEE Transactions on Information Forensicsand Security, vol. 12, no. 4, pp. 767–778, April 2017.

[13]NihilaFathima M, Remitha M. S, Rithanya R. K, "Blockchain based Secure Data Sharing andSensitive Information Hiding in Cloud Storage Environment", International Journal for Research in Applied Science & Engineering Technology (IJRASET)ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 6.887Volume 7 Issue III, Mar 2019- Available at www.ijraset.com