ISSN No:-2456-2165

Ensuring Cloud Storage Security in Multi-Keyword Search for Multiple Data Owners and Users

Keerthana S
M. Tech Cyber Security, Department of CSE
Sree Narayana Gurukulam College of Engineering
Kadayirippu, India

Dr. Suvanam Sasidhar Babu
Professor, Department of CSE
Sree Narayana Gurukulam College of Engineering
Kadayirippu, India

Abstract:- Outsourcing sensitive information to the cloud has become an imminent need these days. Privacy concerns related with cloud computing make the users more ardent on their data. This is because the service providers can access the data, re-distribute it to other third parties and even delete or alter their personal details. In this paper, we propose a method to enhance the security in multi-keyword search for multi-owner and multi-user domain based on the digital signature scheme. Keywords from a particular file are extracted using the RAKE algorithm. Our scheme is a simplified version of how to impart security on existing PRMSM scheme. Security reasoning have shown that the suggested method can resist various cryptographic attacks.

Keywords:- *Cloud computing;Information Retrieval; RAKE;Public key based Digital Signature.*

I. INTRODUCTION

Cloud computing is an on-demand computing framework that provides access to shared resources and data upon request via the Internet. It provides access to a shared pool of computing resources, such as networks, servers, storage, applications and services, which typically resides in third-party data centers. For a user, the network elements representing the provider-rendered services are invisible, as if concealed by a cloud. The essence is that from the users are unaware of what resources are running in which servers.

With sensitive data migrating on to the cloud, the cloud service providers (CSPs) should consider ways to double down on data security. System and process malfunction resulting in data exposure is also a potential problem. Despite relying on CSP, user's lack of knowledge in security measures that should be taken inside cloud environment also alleviate the problem. Key management in cloud should be done in such a way as to reduce storage costs, communication overhead and increase the computational efficiency. Security risks associated with cloud computing includes physical destruction, user authentication, compliance and legal risks, data and application protection, vendor lock-in, authorization issues, unavailability of service, problems related to virtual machines etc.

An excellent work regarding challenges in cloud computing can be found in [1]. HereTakabi*et.al.* illustrates the unique issues of cloud computing that aggravate security and privacy challenges in clouds. They also discuss various approaches to address these challenges and explains some future work needed to provide a dependable cloud computing environment.

Advanced encryption algorithms such as Attribute-based encryption[2], Fully Homomorphic Encryption(FHE)[3], Searchable Encryption (SE) [4] can be used to increase the protection of privacy.

Our model makes use of digital signature scheme based on public key cryptography. Each person has a public-private key pair. Private key is used for signing and thus it is known as signature key whereas public key is used for verifying and is known as the verification key. Fig. 1 depicts the general scheme in detail.

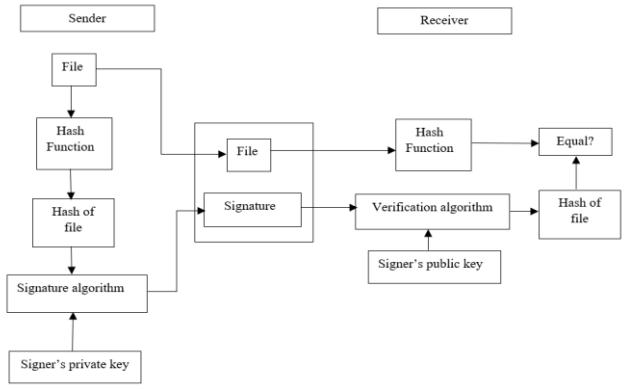

Fig 1:- Model of digital signature scheme

II. RELATED WORK

Wei Zhang *et al.*[5] explores the problem of secure multi-keyword search for multiple data owners and multiple data users in the cloud computing environment and propose schemes to deal with privacy preserving ranked multi-keyword search in a multi-owner model (PRMSM). To efficiently authenticate data users, they propose a novel dynamic secret key generation protocol and a new data user authentication protocol. To enable the cloud server to perform secure search, authors construct a search protocol. An additive order and privacy preserving function family (AOPPF) is proposed to rank the search results and to preserve privacy of relevance scores.

N. Deepa *et.al.* [6] have identified several issues in the protocol proposed in [5] and shows that the proposed method is vulnerable to impersonation attack. Therefore, when the attacker submits a request, the cloud server cannot find the difference between the request submitted by the legitimate user and the attacker. Authors suggests that it is essential to develop a new efficient secure ranking based multi-keyword searching mechanism which enables the data users to perform a secure

data access from the untrusted public cloud and have also done a detailed review of [5] in this work.

In this paper, we try to address the issues found out by [6]. We propose a scheme based on digital signature for search request as well as file upload and create the index using RAKE[7]. Since there is only single cloud server involved we are not going to partition the files as this method is more suited in distributed environments.

The rest of the paper is organized as follows: Section 3 describes the proposed model. Section 4 presents the security analysis. Section 5 demonstrates the results of work. In section 6, paper is concluded.

III. PROPOSED METHODOLOGY

In this section, we describe the architecture of the proposed model for ensuring cloud storage security as shown in Fig.2 and the corresponding threat model.

A. Overview

The proposed multi-owner and multi-user system comprises of four entities: data owner, data user, administration server and cloud server. Data owners have a collection of files F. Data owners perform an initial authentication and connection setup with the cloud like a 3-way handshake. Through this, owners will propagate their keyword w, key key_i for decrypting the index created corresponding to each document and finally the signed document. They also extract a keyword set W from the file F using RAKE, encrypt them to obtain index I and send to the administration server. Administration server outsources I to cloud server. Once a data user wants to search t keywords, he/she performs authentication with the cloud and submit the request to the cloud server. Upon receiving the request, the cloud server searches the encrypted keywords of each data owner and try to compare the user request and keywords. If there is a matching, cloud returns the corresponding file set based on the score produced by RAKE. Once the data user receives the files from the cloud server, he/she will decrypt it using the key exchanged beforehand.

B. Adversary Model

Assume administration server to be a trusted entity. Also, the file owners and file users who passed the authentication process with administration server are presumed to be trusted entities. Cloud server is treated as partially reliable. It is imagined as an entity which is ardent to retrieve contents of the files uploaded by the file owners. Time after which the file users' session will be deactivated is out of the scope of the proposed model since this will create an extra load on the administration server. Also, how the decrypted keys are exchanged between users and owners are out of the scope.

Here we are trying to increase the confidentiality of users and owners with the cloud server as well as integrity of files uploaded by the file owners. Communication between cloud server and administration server is assumed to be secure since administration server can be any trusted third party (TTP). File maintenance and operations are not considered in this work as there are some excellent papers in the literature regarding this.

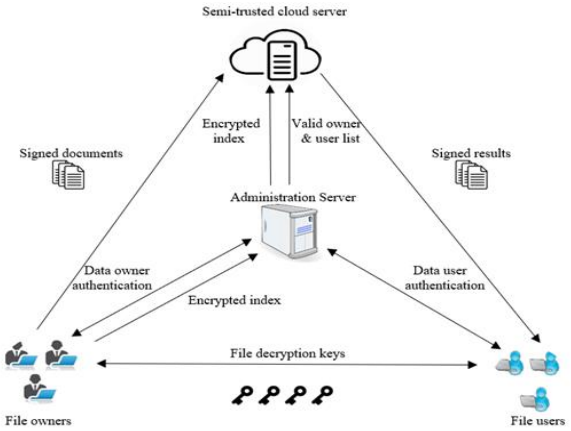

Fig 2:- Architecture of proposed model

C. Notations

Symbols	Inference
$h_k(.)$	one-way hash function with key k
$E_k(m)$	encrypting message m with key k
$Sig_k(m)$	signing the message m with key k
(p_c, s_c)	public and private key pair of cloud
(p_k, s_k)	public and private key pair of owner
(p_u, s_u)	public and private key pair of user

Table 1. Notations

D. File owner and user authentication

Both file owners and file users are enabled by the administration server. Administration server will send a list of valid users and owners V_i and their last login time to the cloud after authentication. Once a user submits a request, cloud will search this list to find whether it is an authentic user. Also, cloud will search this list to ensure that the owner trying to upload a file is a valid one. Users and owners can start their operations only after the activation by the administration server.

E. Creating search request

Dialog between a data user and cloud server consists of following steps:

Step 1: User calculate

$$E_{p_c}(t, r, ID_u), Sig_{s_u}\big(h(t, r, ID_u)\big) \quad (1)$$

where t is the keyword to be searched, r is a random number and ID_u is the user id of the current user. Now send this to cloud server.

Step 2: Upon receiving (1), cloud decrypt $E_{p_c}(t, r, ID_u)$ using its private key, find the hash value and compares the

hashes. Cloud checks whether this is a valid user by searching in the list V_i. If matching cloud computes

$$E_{p_u}(t, r, r', ID_c), Sig_{s_c}(h(t, r, r', ID_c)) \quad (2)$$

and send this to user. Here t is the keyword sent by the user,$r'$is another random number, r is the random number sent by user and ID_c is the cloud id.

Step 3: User decrypt $E_{p_u}(t, r, r', ID_c)$ using the private key s_u and try to verify the signature. If successful, user calculates

$$E_{p_c}(r', ID_c, ID_u), Sig_{s_u}(h(r', ID_c, ID_u)) \quad (3)$$

and forwards this to cloud. This completes the conversation between user and cloud server.

F. File Upload

File owners will upload their files on to the cloud server for storage. They also construct an encrypted index I and send that to administration server which will simply forward this to cloud. Conversation between a file owner and cloud server is as follows:

Step 1: Owner calculates

$$E_{P_c}(key_i, r_1, ID_o), Sig_{s_k}(h(key_i, r_1, ID_o)) (4)$$

wherekey_i is the key for decrypting the index, r_1 is a random number and ID_o is the owner id.

Step 2: Upon obtaining (4), cloud server verify signature using p_k. Then cloud computes

$$E_{p_k}(r_1, r_2, ID_c, key_i), Sig_{s_c}(h(r_1, r_2, ID_c, key_i)) (5)$$

where key_i is the key send by file owner, r_1 is the random number send by owner,r_2 is the new random number chosen by cloud and ID_c is the cloud id. Send this value to the owner.

Step 3: Owner verifies the signature and computes

$$E_{P_c}(r_2, ID_c, ID_o), Sig_{s_k}(h(r_2, ID_c, ID_o)) (6)$$

wherer_2 is the random number send by cloud server. After this step, both cloud and owner obtain assurance of their identity. Now owner can send the keyword which can aid the cloud while a user start searching for a file in the cloud repository.

Step 4: File owner send the following value to the cloud server:

$$E_{P_c}(w, ID_o, r_2, r_3), Sig_{s_k}(h(w, ID_o, r_2, r_3)) (7)$$

where$w$ is the keyword from file owner part, r_2 is the random number send by cloud and r_3 is the new random number.

Step 5: On receiving (7), cloud server verifies signature and if successful calculates

$$E_{p_k}(w, r_3, r_4, ID_c), Sig_{s_c}\big(h(w, r_3, r_4, ID_c)\big) (8)$$

wherer_4 is a new random number.

Step 6: Actual file upload starts in this step. Once the owner verifies (8), he/she prepares the file for transport as follows:

$$E_{P_c}\big(Sig_{key_d}(file), ID_c, r_4, ID_f, ID_o\big), Sig_{s_k}(h\big(Sig_{key_d}(file), ID_c, r_4,$$
(9)

where$Sig_{key_d}(file)$ is the file signed using key_d which is the key that will be exchanged with user upon request by the owner. This is to prevent the cloud server from reading the file. $ID_f$is the file id that can be used by the cloud to retrieve the file from the index.

Step 7: Cloud verifies signature and stores the owner id, file id and$Sig_{key_d}(file)$. Then computes

$$E_{p_k}\big(ID_c, ID_f, r_4\big), Sig_{s_c}(h(ID_c, ID_f, r_4)) (10)$$

and send to owner. Upon receiving this, owner can verify the signatures.

IV. SECURITY ANALYSIS

This section analyses the security of our proposed scheme.

Since the method calculates signature of messages send this ensures message authentication, non-repudiation and data integrity. Also use of digital signatures and hashes prevents the malicious users from performing Man in-the middle (MitM) attack.

Use of random numbers ensure the freshness property of messages. Also, this protects the scheme from replay attack. If any unauthorized person tries to replay some captured messages from the communication, previously used random numbers would depict that data have been altered.

The scheme is also free of impersonation attack as ID of each entity ensures that communication with the person who he/she claims is correct.

Since the key for decrypting the document is obtained only after request from the file owner, it ensures that cloud can never read the contents of the uploaded documents. Only a user who have the correct decryption key can decrypt it.

Our proposed framework removes many hardships from the administration server side as the search request of user and the keyword upload of ownerare handled by the cloud server. Administration server have only the duty of enabling the owners and users and carrying the indices.

Storage cost on the cloud side is also cut down as the random numbers and hashes are re-computed for each conversation. Thus, cloud only have to store the document, key for decrypting index, file id, keyword and valid user and owner list.

V. RESULTS AND IMPLEMENTATION

The proposed system is implemented using Java programming language with My SQL server as back end. Cloud server is set up in Oracle VM Virtual Box as an Ubuntu 16.04 machine with 1024 MB base memory and a single processor. Documents used for uploading are Notepad files with no graphic content. Hash value is calculated using hash function from SHA family and public-private key pairs are calculated using RSA algorithm.

VI. CONCLUSION

An approach for ensuring cloud storage security in case of multi-keyword search for a multi-user and multi-owner environment is proposed here. Efficiency of RAKE and data security provided by public-key based digital signature are the backbone of the scheme. Extra work on the administration server in the existing system is removed by moving the search request and the keyword upload of owner to the cloud server part. From the security analysis, it is clear that the proposed scheme achieves protection from MitM attack, replay attack, etc. and provides message authentication, non-repudiation and data integrity. Storage costs are also reduced when compared to existing approach. The work can be further extended by securing the communication between the cloud server and the administration server and by designing a secure key distribution protocol.

REFERENCES

[1]. Hassan Takabi and James B.D. Joshi and Gail-Joon Ahn, "Security and Privacy challenges in cloud computing environments,"IEEE Security & Privacy, Vol:8, Issue:6, Nov-Dec 2010, pp.24 – 31.

[2]. Amit Sahai and Brent Waters, "Fuzzy Identity-Based Encryption,"Advances in Cryptology – EUROCRYPT 2005, pp 457-473.

[3]. Craig Gentry, "Fully homomorphic encryption using ideal lattices," Proceedings of the forty-first annual ACM symposium on theory of computing(STOC), 2009.

[4]. Md Iftekhar Salam, Wei-Chuen Yau, Ji-Jian Chin, Swee-Huay Heng, Huo-Chong Ling, Raphael C-W Phan, Geong Sen Poh, Syh-Yuan Tan and Wun-She Yap, "Implementation of searchable symmetric encryption for privacy-preserving keyword search on cloud storage," Human-centric computing and information sciences(HCIS), Springer,July 2015.

[5]. Wei Zhang, Yaping Lin, Sheng Xiao, Jie Wu and Siwang Zhou , "Privacy preserving ranked multi-keyword search for multiple data owners in cloud computing,"IEEE Transactions on Computers, Vol:65, Issue:5, May 2016,pp. 1566 - 1577.

[6]. N. Deepa, P. Vijayakumar, Bharat S. Rawal and B. Balamurugan, "An extensive review and possible attack on the privacy preserving ranked multi-keyword search for multiple data owners in cloud computing,"2017 IEEE International Conference on Smart Cloud (SmartCloud), Nov 2017.

[7]. Automatic keyword extraction from individual documents, Text Mining: Applications and Theory, John Wiley & Sons, 2010,pp.1-20.