www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 6 Issue 3 March 2017, Page No. 20464-20466
Index Copernicus value (2015): 58.10 DOI: 10.18535/ijecs/v6i3.11

Multi-Key Searchable Encryption for Group Sharing Via Cloud Storage

V.Amudha[1], A.Ashwini[2], G.KaviBharathi3, and P.Ranjitha[4]
UG (CSE), Sri Krishna College of Technology, Anna University, Coimbatore, India
[4]Assistant Professor (CSE), Sri Krishna College of Technology, Anna University, Coimbatore, India
amudhavelu36@gmail.com[1], ashwiniaruchamy18@gmail.com[2] , kavisekar5595@gmail.com[3] and ranjise01@gmail.com[4]

***Abstract*—In case of sharing the group of documents in cloud storage the document owners uses the encrypted key for secured sharing. For a single owner, the document contains one trapdoor key then the user can download it using the key, but for multiple owners this concept does not works. To overcome this practical problem this paper proposes a solution for KASE in case of federated clouds. This is a practice of interconnecting the cloud computing environment of two or more service providers for the purpose of load balancing traffic.**

Keywords—Searchable encryption,trapdoor,data sharing, cloud storage, data privacy

I. INTRODUCTION

Cloud storage has emerged as a promising answer for providing ubiquitous, convenient, and on-demand accesses to large amounts of data shared over the Internet. Today, millions of users are sharing personal data, such as image and videos, with their friends through social network applications based on cloud storage on a daily basis. Business users are also being draw in by cloud storage due to its many benefits, including lower cost, greater agility, and better resource utilization. However, while enjoying the comfort of sharing data via cloud storage, users are also progressively concerned about inadvertent data leaks in the cloud. Such data leaks, caused by a despiteful adversary or a misbehaving cloud operator, can usually track to serious breaches of personal privacy or business secrets. To address users' anxiety over potential data leaks in cloud storage, a public approach is for the data owner to encrypt all the data previous to uploading them to the cloud, such that later the encrypted data may be recover and decrypted by those who have the decryption keys. Such a cloud storage is frequently called the cryptographic cloud storage.

However, the encryption of data makes it challenging for users to search and then selectively retrieve only the data containing given keywords. A usual solution is to employ a searchable encryption (SE) scheme in which the data owner is required to encrypt potential keywords and upload them to the cloud combined with encrypted data, such that, for retrieving data matching a keyword, the user will send the corresponding keyword trapdoor to the cloud for performing search over the encrypted data.

A. Achieving secure, scalable, and fine-grained data access control

Cloud computing is an emerging computing prototype in which resources of the computing infrastructure are provided as services over the Internet. As positive as it is, this paradigm also brings forth many new questioning for data security and access control when users outsource sensitive data for allocating on cloud servers, which are not within the same *trusted* domain *as* data owners. However, in doing so, these solutions necessarily introduce a heavy computation surface on the data owner for key distribution and data management when fine grained data access control is preferred, and thus do not scale well. The problem of at the same time achieving fine grainedness, scalability, and data confidentiality of access control actually still remains unresolved.

B. Secure provenance

Secure provenance that records ownership and process history of data objects is essential to the success of data forensics in cloud computing, yet it is still a challenging issue to tackle this undiscovered area in cloud computing, we proposed a new secure provenance scheme based on the bilinear pairing techniques. As an important bread and butter of data forensics and post investigation in cloud computing, the proposed scheme is defined by providing the information confidentiality on delicate documents stored in cloud, anonymous authentication on user access, and provenance pursuit on disputed documents. With the obvious security techniques, we formally demonstrate the proposed scheme is secure in the standard model.

C. Secure multi-owner data sharing for dynamic groups

With the character of low maintenance, cloud computing provides an economical and proficient solution for sharing group resource among cloud users. Inappropriately, sharing data in a multi-owner mode while preserving data and identity secrecy from an untrusted cloud is still a challenging issue, due to the common change of the membership. By leveraging group signature and dynamical broadcast encryption method, any cloud user can anonymously share data with others. Meanwhile, the storage overhead and encryption calculation cost of our outline are independent with the number of revoked

users

Fig. 1. System architecture of search over encrypted data in cloud computing

II. METHODOLOGY

Java is an platform independent programming language that extends it's features wide over the network.

- It's a light weight package, as they are not implemented by platform-specific code.
- Related classes are limited in javax.swing and its sub packages, such as javax.swing.tree.
- Components explained in the Swing have more ability than those of AWT.
- Compilation happens just once; interpretation occurs each time the program is executed

A. The Java Platform

A platform is the hardware or software environment in which a program tests. The Java platform differs from most other platforms in that it's a software-only platform that tracks on top of other, hardware-based platforms. Most other platforms are represented as a combination of hardware and operating system.

The figure 3 depicts a Java program, such as an application or applet, that's running on the Java platform.

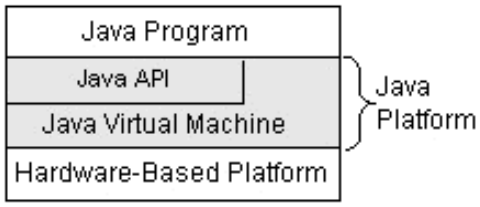

Fig. 2. The java api and virtual machine

B. Core API features

The core API has the following features:

The Essentials: Objects, strings, threads, numbers, input and output, data structures, system properties, date and time, and so on.

Applets: The set of conventions utilised by Java applets.

Networking: URLs, TCP and UDP sockets, and IP addresses.

Internationalization: Help for writing programs that can be localized for users worldwide.

Security: Both low-level and high-level, including electronic signatures, public/private key management, access control, and certificates.

Software components: Known as JavaBeans, can plug into existing section architectures such as Microsoft's OLE/COM/Active-X architecture, OpenDoc, and Netscape's Live Connect.

Object serialization: Allows cipher persistence and communication via Remote Method Invocation (RMI).

Java Database Connectivity (JDBC): Offers uniform access to a wide range of relational databases. Java not only has a core API, but also standardized extensions. The standardized extensions define APIs for 3D, servers, collaboration, telephony, speech, animation, and more.

C. Networking basics

1) Socket overview

Internet protocol (IP) is a low-level routing protocol that breaks data into small collection and sends them to an address across a network. Transmission Control Protocol (TCP) is a higher-level protocol that manages to reliably transfer data. A third protocol, User Datagram Protocol (UDP), sits next to TCP and can be used directly to support fast, connectionless, unreliable transport of packets.

2) Reserved sockets

Once connected, a higher-level protocol ensues, which is dependent on which port you are using. TCP/IP reserves the lower, 1,024 ports for specific protocols. Port number 21 is for FTP, 23 is for Telnet, 25 is for e-mail, 79 is for finger, 80 is for HTTP, 119 is for Net-news and the list goes on. It is up to each protocol to determine how a client should interact with the port.

3) Java and the net

Java supports TCP/IP both by extending the already established stream I/O interface. Java supports both the TCP and UDP protocol families. TCP is in use for reliable stream-based I/O across the network. UDP assist a simpler, hence faster, point-to-point datagram-oriented model.

4) Inet address

The InetAddress class is used to encapsulate both the numerical IP address and the domain name for that address. We move with this class by using the name of an IP host, which is more convenient and understandable than its IP address. The InetAddress class conceal the number inside. As of Java 2, version 1.4, InetAddress can handle both IPv4 and IPv6 addresses.

a) Factory Methods

Three commonly in use InetAddress factory method acting are shown here.

- static InetAddress getLocalHost() throws UnknownHostException
- static InetAddress getByName(String hostName) throws UnknowsHostException
- static InetAddress[ ] getAllByName(String hostName) throws UnknownHostException

III. MODULES

A. Searchable encryption

Generally searchable encryption schemes fall into two categories,

1. Searchable symmetric encryption (SSE) and
2. Public key encryption with keyword search (PEKS).

Both SSE and PEKS can described as the tuple

SE= (Setup, Encrypt, Trapdoor Test): Setup(1): this algorithm is run by the owner set up the scheme. It takes as input a security parameter 1, and outputs the necessary keys.

Encrypt(k; m): this algorithm is run by the ownership to encrypt the data and return its keyword cipher-texts. It takes as input the data m, owner necessary keys including searchable encryption key k and data encryption key, outputs data cipher text and keyword cipher-texts.

C m Trpdr(k; w): this algorithm is run by a user generate a trapdoor Tr for a keyword w using key k.

Test(Tr,C): this algorithm is run by the cloud server to do a keyword search over encrypted data. It takes as input trapdoor Tr and the keyword cipher-texts C m m ,outputs whether C contains the specified keyword. the problem of searching on data that is encrypted using a public key system.

B. Data Group sharing

Server can use this aggregate trapdoor and some public information to perform keyword search and return the result to Bob. Therefore, in MKSE, the delegation of keyword search right can be achieved by sharing the single aggregate key. We note that the delegation of decryption rights can be achieved using the key-aggregate encryption approach, but it remains an open problem to delegate the keyword search rights together with the decryption rights. To summarize, the problem of constructing a MKSE.

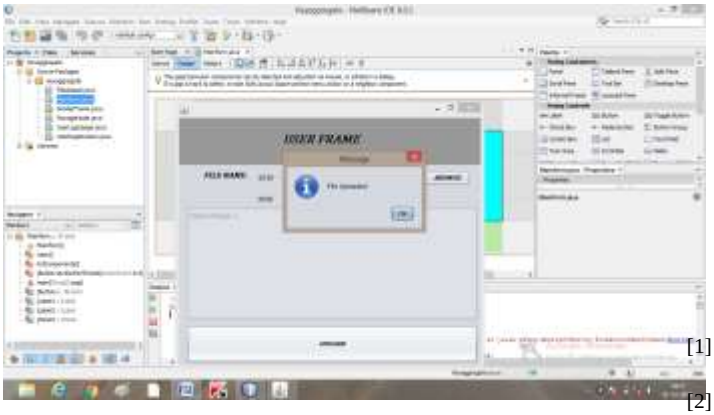

Fig. 3. File upload in user frame

C. Cloud Data privacy

Cloud Data privacy issues are among the key concerns for companies moving to the cloud. In most countries and in most industries, data privacy modulation apply whenever Personally Identifiable Information (PII) is collected and stored. When this information resides in the cloud, it presents a exclusive challenge because cloud computing resources are distributed, making it challenging to know where data is located and who has access at any given time. In addition to the cloud data privacy laws highlighted below, many enterprises need to also obey to series

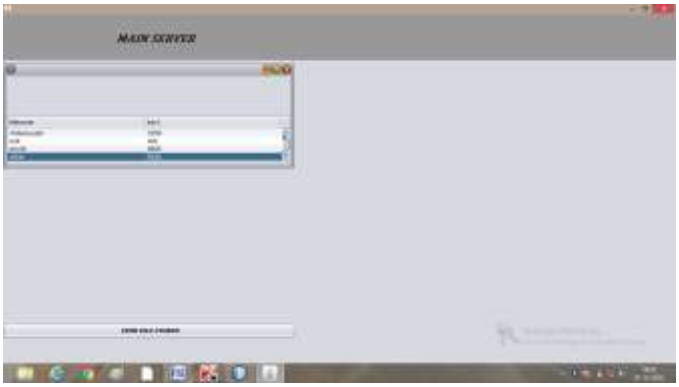

Fig. 4. Storage frame and key generated in data table

IV. CONCLUSION AND FUTURE SCOPE

Considering the functional problem of privacy-preserving data sharing scheme based on public cloud storage which requires a data owner to distribute a large number of keys to users to modify them to access his/her documents, we for the first time suggest the concept of key-aggregate searchable encryption and construct a concrete KASE scheme. Both analysis and evaluation results confirm that our work can provide an effectual solution to building functional data sharing system based on public cloud storage. In a MKSE scheme, the owner only needs to distribute a single key to a user when sharing stacks of documents with the user, and the user only needs to submit a single trapdoor when he queries over all documents shared by the same owner. However, if a user wants to query over documents shared by multiple owners, he must return multiple trapdoors to the cloud. How to reduce the number of trapdoors under multi-owners setting is a future work. Moreover, federated clouds have draw in a lot of attention nowadays, but our MKSE cannot be applied in this case directly. It is also a future work to provide the solution for MKSE in the case of federated clouds.

REFERENCES

[1] Michael A, Armando F, et al., "A View of Cloud Computing," Communications of the ACM, Vol.53, April 2010, pp.50-58.

[2] George Pallis, "Cloud Computing: The New Frontier of Internet Computing," IEEE Internet Computing, September-October 2010, pp.70-73.

[3] Rimal, B., et al., "A Taxonomy, Survey, and Issues of Cloud Computing Ecosystems," Springer,London, 2010, pp.21-46

[4] Qi Zhang, Lu Cheng et al., "Cloud Computing: state-of-the-art and research challenges," Journal of Internet Services and Applications, Volume 1, Springer, 2010, pp.7-18.

[5] Daniele Catteddu, "Cloud Computing: Benefits, Risks and Recommendations for Information

[6] Security," Communications in Computer and Information Science, Vol. 72, Springer 2010.

[7] Sean C and Kevin C, "Cloud Computing Security, International Journal of Ambient Computing and Intelligence," Vol. 3, pp.38-46, April-June 2011, IGI Publishing.

[8] Paresh D Sharma, "A classification of distinct vulnerabilities in cloud computing," World Journal of Science and Technology, Vol. 2, 2012.

[9] Fugini M., "Security and trust in Cloud scenarios," In 1st International Workshop on Securing Services on the Cloud (IWSSC), September 2011, pp.22-29.

[10] R.K.L. Ko, B.S. Lee and S. Pearson, "Towards Achieving Accountability, Auditability and Trust in Cloud Computing," Proc. International workshop on Cloud Computing: Architecture, Algorithms and Applications, Springer, 2011, pp.5-18.

[11] Hassan T, James B.D. Joshi, et al., "Security and Privacy Challenges in Cloud Computing