Information Leak Identification and Secure Search over Encrypted Data in Cloud Storage Utilities

A Anusha, J.Dayanika, Kranthi Kiran G

[1]Department of CSE, CMR TECHNICAL CAMPUS, Hyderabad, India.

Abstract— *As we know that cloud computing is used in big IT organization as well as IT enterprises which provideconvenient, direct, indirect remote access to data storage and application services that allow users to store their private and sensitive data. In cloud computing many organization notice as well as report that important and sensitive data have been leak repeatedly in recent years in which human mistake is one of the most important reason of data loss. Hence in this paper we represent data leak identification solution to solve this problem and we identify the system requirements and challengestowards achieving privacy assured searchable outsourced cloud data services. This paper present a general methodology for this, using searchable encryption techniques, which allows encrypted data to be searched by users without leaking information about the data itself and users queries.The resulting design is able to facilitate efficient server side rankingwithout losing keyword privacy.*

Keywords —Cloud Storage, Data Owner, Data User, Data leak, Network Security, Privacy, Collection Intersection.

I. INTRODUCTION

We have seen the report from Risk Based Security(RBS)[1],there are number of leak sensitive data records that can be gain during the last few years, i.e., from 412 million in 2012 to 822 million in 2013. Hacker attacks, inadvertent leaks and human mistakes lead to data leak incidents [2]. The leaks of data can be detected and avoided by the various solution which may include data-leak detection [3][4], data confinement [5]–[6], stealthy malware detection and policy enforcement. In this paper, we use the solution for data leaks by design, implement and evaluate *fuzzy fingerprint* technique that gains the privacy of data during operation. The process of this technique is that the owner of data computes a set of fingerprints from the sensitive data and then make visible only small amount of them to the DLD provider. The DLD provider then calculates fingerprints from the network traffic and recognizes leaks in them.On the other hand to increase privacy in cloud computing we have to protect data and sensitive data need to be encrypted before the data going to store on cloud storage. Most of the searchable encryption technique allow user to search encrypted data via keywords. But this technique supports only Boolean Search which is insufficient to meet the effective data utilization which is demanded by huge amount of users. Hence in this paper, wesolve this problem of secure ranked keyword search over encrypted cloud data. In this paper, we use the statistical measure approach, i.e. relevance score, from information retrieval to build a secure searchable index, and also protect those sensitive score information. The resulting design is able to facilitate efficient server side ranking without losing keyword privacy. A privacy approach for owners to take back control of their data is to adopt end-to-enddata encryption. Therefore, to build a full-fledged cloud data service, it is highly desirable to enable privacy assured search over encrypted data, which ideally does not leak any sensitive user information to the cloud, such as business secrets or private personal activities. Without being able to effectively utilize the outsourced data, the cloud will merely be a remote storage with limited values.

II. DETAILS EXPERIMENTAL

In this section we review some related works concerned with security and privacy issues in the cloud. Also, we discuss the work which adopts similar techniques as our approach.

Model, Related Technique and Overview

In our paper, we represent key role of data owner and DLD provider in which the data owner share the sensitive data and allow DLD provider to check network traffic for anomalies, namely inadvertent data leak. Similarly, on the other hand DLD provider detects network traffic and it can be done offline without causing any real-time delay in routing the packets.

A. Security Goal and Threat Model:

We categorize three causes for sensitive data to appear on the outbound traffic of an organization

- Case I Inadvertent data leak: Inadvertent data leak may be due to human errors such as forgetting to use encryption, carelessly forwarding an internal email and attachments to outsiders, or due to application flaws.[7]

- Case II Malicious data leak: A rogue insider or a piece of stealthy software may steal sensitive personal or organizational data from a host. Host based defenses (such as detecting the infection onset) need to be deployed instead.
- Case III Legitimate and intended data transfer: In this paper, we assume that the data owner is aware of legitimate data transfers and permits such transfers. So the data owner can tell whether a piece of sensitive data in the network traffic is a leak using legitimate data transfer policies.

B. Privacy Goal and Threat Model:

First we describe the two most important players in our abstract model: the organization (i.e., data owner) and the data-leak detection (DLD) provider as shown in figure 1.

- Data Owner: the sensitive data and authorizes the DLD provider to inspect the network traffic from the organizational networks for anomalies, namely inadvertent data leak. However, the organization does not want to directly reveal the sensitive data to the provider.
- DLD provider: inspects the network traffic for potential data leaks. The inspection can be performed offline without causing any real-time delay in routing the packets. However, the DLD provider may attempt to gain knowledge about the sensitive data.

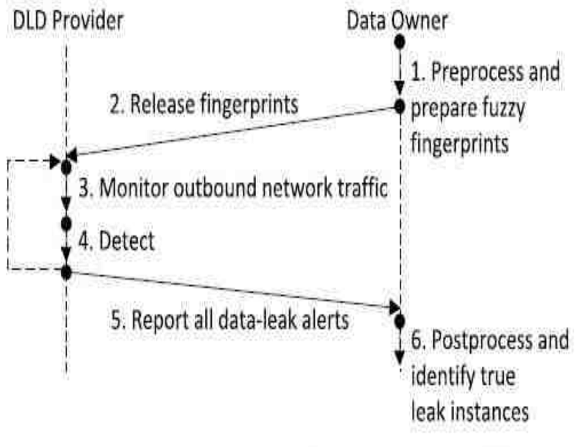
DLD Provider
Data Owner
1. Preprocess and prepare fuzzy fingerprints
2. Release fingerprints
3. Monitor outbound network traffic
4. Detect
5. Report all data-leak alerts
6. Postprocess and identify true leak instances

Fig. 1. Our Privacy-preserving Data-Leak Detection Model.

Privacy-preserving keyword search [8] or fuzzy keyword search [9] provide string matching approaches in semi-honest environments, but keywords usually do not cover enough sensitive data segments for data-leak detection.

Related Technique For Safe Search:

First, we briefly discuss and compare several existing techniques, and their relevance to the privacy-assured cloud-based search problem.

- *Secure multiparty computation (SMC):* InSMC, each party *Pi* possess some private input *xi*, and every party computes some (public)function *f*(*x*1, , *xn*) without revealing *xi* to others, except what can be derived from the input and output.
- *Private information retrieval (PIR):*PIRinvolves two parties: a client and a server. In asymmetric PIR, the server hosts a public database D, while the client retrieves a record *i* from D without revealing *i* to the server. In symmetric PIR (a.k.a. oblivious transfer), the non-retrieved records should also be withheld from the client, which can be regarded as a special case of SMC.
- *Searchable encryption (SE):* SE also involvesa client and a server, where the latter stores an encrypted database ~ D, and the former possesses a private query *Q* that wants to obtain the query result *Q*(D) without revealing both *Q* and plaintext D to the server.
- *Order-Preserving Symmetric Encryption (OPSE)***:**In OPSE [10], the numerical orderingof plaintext is preserved after encryption that provide the first cryptographic construction of OPSE that is provably secure under the security framework of pseudorandom function or pseudorandom permutation. It can be regarded as a function *g* (◊) from a domain D = {1…*M*} to a range R = {1…*N*}.

Methodology:

We describe a top-down approach in which the search functionality in the plaintext domain, one can decompose it into a certain data index structure and primitive data operations using relevant information retrieval (IR) principles.

Model and Overview:

We begin by describing a general cloud data storage service architecture involving three (types of) entities (Fig. 2). The *data owner* (or data contributor) is one or multiple entities who generate and encrypt data, and upload them to the cloud server. The owner can be either an organization or an individual. The *cloudserver* belonging to a CSP possesses significantstorage and computation resources, and provides them to end users in a pay per user manner. There are one or more *data users* in the system, who may need to perform queries over the outsourced data in order to extract useful information. In addition, in order to enable public auditing, a third party auditor can be employed, which is discussed in [11] and is outside the scope of this article. The owner's data are encrypted end-to-end using secret keys created by him/herself, and a searchable index is usually created and encrypted along with the outsourced data. To allow data access and search by users, the data owner usually generates and distributes search tokens (or trapdoors), which are encrypted queries to users, either actively or upon users' requests. When a user wants to gain file access or initiate a query, he/she submits a corresponding token to the server, who then returns a matching set of documents in an encrypted format. In some situations, the data user and data owner can be the same physical entity.

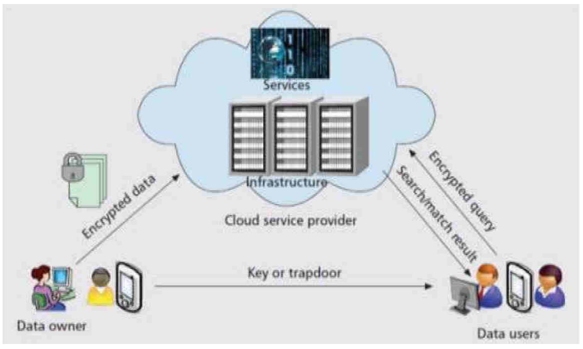

Fig. 2: System architecture for searchable cloud data storage services.

System Requirement:

Functional Properties For data search, perhaps the most important property is *usability*, which is the basis for attracting customers. The current Google search is a great example of what is necessary in plaintext domain search. The following is an (incomplete, but typical) list of them:

- *Multi-keyword search*: The search conditionshould support Boolean expressions consisting of combinations of multiple keywords, including conjunctive normal form (CNF) and disjunctive normal form (DNF).
- *Result ranking*: The ranked search functiongreatly enhances the relevance of returned search results and reduces communication overhead, which is highly desirable for building usable cloud data services.
- *Error tolerance*: To accommodate varioustypos, representation inconsistencies, and so forth, search schemes should have a fuzzy nature. This means a search needs to also return relevant results for keywords within a certain edit distance from the input query.
- *Handle structured data*: A large portion oftoday's online data is represented using rich structures beyond simple text form, such as social network graphs. Without being able to utilize those structured data, the economic potential of cloud services will not be fully realized. We note that in the encrypted domain, it is very difficult for the above properties to be simultaneously achieved. We describe how the state-of-the-art schemes achieve some combination of them.
- *Privacy Assurance*: In a searchable cloudstorage service, both the owner's outsourced data and users' queries over those data may contain sensitive information and need protection against an adversary. More specifically, the system should meet the following privacy requirements:
- *Data and index confidentiality*: Without thesecret key *K*, no one, including the cloud server, should be able to learn sensitive information from the owner's private data. Similarly, they should not be able to deduce sensitive information underlying the data index, because the index is often closely related to the data itself.
- *Query confidentiality*: Users' most importantconcern is to hide the search criteria on which they are evaluating the data (e.g., their query keywords). These should not be derivable from the search trapdoor and data/index sent to the cloud server, even when the server possesses some additional background information such as keyword distribution. A higher-level requirement is *query unlinkabilit*y, that is, the cloud server shall not learn whether two queries have the same criteria. Note that this intrinsically requires the trapdoor to be nondeterministic.
- *Efficiency*: A privacy assured data searchscheme should have low computation, communication, and storage overheads. For such a scheme to be deployed in a large scale cloud storage system with economic practicality, we argue that the search process should be completed within both constant communication round and computation time (independent of the database size). In general, the privacy guarantee conflicts with efficiency and functionality goals.

III. EXPERIMENTAL EVALUATION

We implement our fuzzy fingerprint framework in Hadoop, including packet collection, shingling, Rabin fingerprinting, as well as partial disclosure and fingerprint filter extensions. Our implementation of Rabin fingerprint is based on cyclic redundancy code (CRC). We use the padding scheme mentioned in [12] to handle small inputs. In all experiments, the shingles are in 8-byte, and the fingerprints are in 32-bit (33-bit irreducible polynomials in Rabin fingerprint). We set up a networking environment in Virtual Box, and make a scenario where the sensitive data is leaked from a local network to the Internet. Multiple users' hosts are put in the local network, which connect to the Internet via a gateway (Fedora). Multiple servers (HTTP, FTP, etc.) and an attacker-controlled host are put on the Internet side. The gateway dumps the network traffic and sends it to a DLD server/provider (Linux). Using the sensitive-data fingerprints defined by the users in the local network, the DLD server performs off-line data-leak detection. The speed aspect of privacy preserving data-leak detection is another topic and we study it in [13].

IV. RESULTS AND DISCUSSION

We begin by describing a general cloud data storage service architecture involving three entities. The data owner (or data contributor) is one or multiple entities who generate and encrypt data, and upload them to the cloud server. The owner can be either anorganization or an individual. The owner's data are encrypted end-to-end using secret keys created by him/her, and a searchable index is usually created and encrypted along with the outsourced data. To allow data access and search by users, the data owner usually generates and distributes search tokens (or trapdoors), which are encrypted queries to users, either actively or upon users' requests. When a user wants to gain file access or initiate a query, he/she submits a corresponding token to the server, who then returns a matching set of documents in an encrypted format. In some situations, the data user and data owner can be the same physical entity. In our detection procedure, the data owner computes a special set of digests or fingerprints from the sensitive data and then discloses only a small amount of them to the DLD provider. The DLD provider computes fingerprints from network traffic and identifies potential leaks in them. To prevent the DLD provider from gathering exact knowledge about the sensitive data, the collection of potential leaks is composed of real leaks and noises. It is the data owner, who post-processes the potential leaks sent back by the DLD provider and determines whether there is any real data leak.

V. CONCLUSION

We identify the problem and challenges of enabling privacy assured searchable cloud data storage services, which suggest that achieving functionally rich, usable, and efficient search on encrypted data is possible without sacrificing privacy guarantee too much as well as we proposed fuzzy fingerprint, a privacy-preserving data-leak detection model and present its realization. Using special digests, the exposure of the sensitive data is kept to a minimum during the detection.

REFERENCES

[1] Risk Based Security. (Feb. 2014). Data Breach Quick-View: An Executive's Guide to 2013 Data Breach
[2] Trends.[Online].Available:https://www.riskbasedsecurity.com/reports/2013-DataBreachQuickView.pdf, accessed Oct. 2014.
[3] Ponemon Institute. (May 2013). 2013 Cost of Data Breach Study: Global Analysis. [Online]. Available: https://www4.symantec.com/mktginfo/ whitepaper/053013_GL_NA_WP_Ponemon-2013-Cost-of-a-Data-Breach-Report_daiNA_cta72382.pdf, accessed Oct. 2014.
[4] Identity Finder. Discover Sensitive Data Prevent Breaches DLP Data Loss Prevention. [Online]. Available: http://www.identityfinder.com/,accessed Oct. 2014.
[5] K. Borders and A. Prakash, "Quantifying information leaks in outbound web traffic," in Proc. 30th IEEE Symp. Secur. Privacy, May 2009,pp. 129–140.
[6] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama: Capturing system-wide information flow for malware detection and analysis," in Proc. 14th ACM Conf. Comput. Commun. Secur., 2007,pp. 116–127.
[7] A. Nadkarni and W. Enck, "Preventing accidental data disclosure in modern operating systems," in Proc. 20th ACM Conf. Compute. Common. Secure., 2013, pp. 1029– 1042.
[8] J. Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno, "Privacy oracle: A system for finding application leaks with black box differential testing," in Proc. 15th ACM Conf. Comput. Commun. Secur., 2008, pp. 279–288.
[9] S. Ananthi, M. SadishSendil, and S. Karthik, "Privacy preservingkeyword search over encrypted cloud data," in Advances in Computing and Communications (Communications in Computer and Information Science), vol. 190. Berlin, Germany: Springer-Verlag, 2011,pp. 480–487.
[10] J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proc. 29th IEEEConf. Comput. Commun., Mar. 2010, pp. 1–5.
[11] A. Boldyreva et al., ―Order-Preserving Symmetric Encryption,‖ Proc. Eurocrypt '09, LNCS, vol. 5479, Springer, 2009
[12] C. Wang et al., ―Toward Publicly Auditable Secure Cloud Data Storage Services,‖ IEEE Network, vol. 24, no. 4, July– Aug. 2010, pp. 19–24.
[13] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. MIT/LCS/TR-212, 1979.
[14] F. Liu, X. Shu, D. Yao, and A. R. Butt, "Privacy-preserving scanning of big content for sensitive data exposure with MapReduce," in Proc. ACM CODASPY, 2015