IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: http://ijdacr.com (Volume 4, Issue 2, September 2015)

Leaker Detection In Open Network [LDION]

Remya .G .Nair
Department of Computer Science and Engineering
Cochin University of Science and Technology
Kottayam , India
nair0869@gmail.com

Mrs. Jyothis Joseph
Department of Computer Science and Engineering
Cochin University of Science and Technology
Kottayam , India
jyothisjoseph@rediffmail.com

Abstract—nowadays leakage of data is common to Industries, academic and Government Offices. Data must be shared for social purpose, research purposes and for business purposes. Data is shared among different enterprises or agents. Once the private data is shared it is not guaranteed that the data will not leak. If leakage happens it will be loss to firms. So we can detect the leaker for avoiding the loss thus occurred and thus avoid business with that agent. Leakage of data happening nowadays also but some firms will not tell their loss because of fear of loss of respect and other matters. Some companies distribute their data to trusted third parties. When Data distributers (Companies) found their some of the data in the web or somebody's laptop that is in unauthorized place. The distributor understands that the leaked data came from one or more agents. Our goal is to detect which agent leaks that data and provide the security to that data. When the distributor's sensitive data have been leaked by agents, and to identify the agent that leaked the data. We propose data allocation strategies (across the agents) that improve the probability of identifying leakages. These methods do not rely on alterations of the released data (e.g., watermarks).The Main Aim of the system can be given as follows:- Identify data leakages from distributed data using some data allocation strategies and find out the fake agent who leak that data.

In this work, we present a generic data lineage framework for data flow across multiple entities that take two characteristic, principal roles (i.e., owner and consumer). We define the exact security guarantees required by such a data lineage mechanism toward identification of a guilty entity, and identify the simplifying non-repudiation and honesty assumptions. With this model we assign a clearly defined role to each involved party and define the inter-relationships between these roles. There are three different roles in LDION: data owner, data consumer and auditor. The data owner is responsible for the management of documents and the consumer receives documents and can carry out some task using them. The auditor is not involved in the transfer of documents, he is only invoked when a leakage occurs and then performs all steps that are necessary to identify the leaker thus provide confidentiality. Whenever a document is transferred to a consumer, the sender embeds information that uniquely identifies the recipient. We call this is as fingerprinting which is cryptographically stored in the document without altering any property of the document. If the consumer or agent leaks this document, it is possible to identify him with the help of the embedded information. A key position in LDION is taken by the auditor. He is invoked by an owner and provided with the leaked data. If the leaked data was transferred using our model, there is identifying information embedded for each consumer who received it. Using this information the auditor can create an ordered chain of consumers who received the document. We call this chain the lineage of the leaked document. The last consumer in the lineage is the leaker. In the process of creating the lineage each consumer can reveal new embedded information to the auditor to point to the next consumer – and to prove his own innocence.

I. Introduction

Nowadays also security of documents or data is a problem. So there is a need of a good method to tackle this security problem. Although it is not guaranteed that leaking of data cannot be avoid today's also but we can find the guilty of leaking. There are forensic methods to find the leaker. But these are very costly methods. In this method (LDION)we can find the guilty of leakage of data using watermarking method. Here in this method we do not find a method to fully avoid leaking of data. Here we define a method to find the guilty of leaking of data. Here watermarking is done by using symmetric key of owner and consumer. So the leaker cannot get the original document without using the key. Watermark can be remove if one malicious agent will come. But here if the leaker will remove watermark and try to

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: http://ijdacr.com (Volume 4, Issue 2, September 2015)

retrieve document then the original document will destroy. Every consumer will have different watermark. That is here multiple rewatermarking method is used. As the owner does not trust the consumer he uses fingerprinting or watermarking every time he passes a document to a consumer. Here for each document owner embedding. Watermark when give to consumer. In the digital era, information leakage by disgruntled employees and malicious external entities, present one of the most serious threats to organization.We can use http://www.privacyrights.org/data-breach" web site for finding breach datas information.

II Model

Suppose an agent Ui is guilty if it contributes one or more objects to the target. The event that agent Ui is guilty for a given leaked set S is denoted by Gi| S. The next step is to estimate Pr { Gi| S }, i.e., the probability that agent Gi is guilty given evidence S. To compute the Pr{ Gi| S}, estimate the probability that values in S can be "guessed" by the target. For instance, say some of the objects in t are emails of individuals. Conduct an experiment and ask a person to find the email of say 100 individuals, the person may only discover say 20, leading to an estimate of 0.2. Call this estimate as pt, the probability that object t can be guessed by the target. The two assumptions regarding theRelationship among the various leakage events.

Assumption 1: For all t, t ∈S such that t ≠ T the provenance of t is independent of the provenance of T. The term provenance in this assumption statement refers to the source of a value t that appears in the leaked set. The source can be any of the agents who have t in their sets or the target itself.

Assumption 2: An object t ∈S can only be obtained by the target in one of two ways. • A single agent Ui leaked t from its own Ri set, or the target guessed (or obtained through other means) t without the help of any of the n agents. To find the probability that an agent Ui is guilty given a set S, consider the target guessed t1 with probability p and that agent leaks t1 to S with the probability 1-p. First compute the probability that he leaks a single object t to S. To compute this, define the set of agents Vt= {Ui | t<-Rt} that have t in their data sets. Then using Assumption 2 and known probability p, We have, Pr {some agent leaked t to S} = 1- p (1.1) Assuming that all agents that belong to Vt can leak t to S with equal probability and using Assumption 2 obtain, Pr {Ui leaked t to S} = (1.2) Given that agentUi is guilty if he leaks at least one value to S, with Assumption 1 and Equation 1.2 compute the probability Pr { Gr| S}, agentUi is guilty, Pr {Gi| S} = (1.3)

III Optimization Problems

The distributor's data allocation to agents has one constraint and one objective. The distributor's constraint is to satisfy agents' requests, by providing them with the number of objects they request or with all available objects that satisfy their conditions. His objective is to be able to detect an agent who leaks any portion of his data. We consider the constraint as strict. The distributor may not deny serving an agent request and may not provide agents with different perturbed versions of the same objects. The fake object distribution as the only possible constraint relaxation. The objective is to maximize the chances of detecting a guilty agent that leaks all his data objects.

The Pr {Gi |S =Ri } or simply Pr {Gi |Ri } is the probability that agent Ui is guilty if the distributor Discovers a leaked table S that contains all Ri objects. The difference functions Δ (i, j) is defined as:

Δ (i, j) = Pr {Gi |Ri } – Pr {G |Ri } (1.4)

A) Problem Definition

Let the distributor have data requests from n agents. The distributor wants to give tables R1, .Rn. to agents, U1 . . . , Un respectively, so that

IJDACR
ISSN: 2319-4863

IJDACR
International Journal Of Digital Application & Contemporary Research

International Journal of Digital Application & Contemporary Research
Website: http://ijdacr.com (Volume 4, Issue 2, September 2015)

• Distribution satisfies agents' requests; and

• Maximizes the guilt probability differences Δ (i, j) for all i, j = 1. . . n and i= j. Assuming that the sets satisfy the agents requests, we can express the problem as a multi criterion

B) Optimization Problem

Maximize (. . . , Δ (i, j), . . .) i! = j (1.5) (Over R1,….., Rn,) The approximation [3] of objective of the above equation does not depend on agent's probabilities and therefore minimize the relative overlap among the agents as Minimize (. . . ,(|Ri∩Rj|) / Ri , . . .) i != j (1.6) (over R1 , . . . ,Rn) This approximation is valid if minimizing the relative overlap, (|Ri∩Rj|) / Ri maximizes Δ (i, j).

IV Provenance

The most general approach to provenance is one in which one record a complete history of the derivation of some data set. This is called workflow or coarse-grain provenance. This may involve not only tracking the interaction of programs, but also the involvement of external devices such as sensors, cameras or other data collecting equipment. It may also involve a record of human interaction with the process. A proper record of workflow provenance is essential in many scientific experiments as it enables experiments to be systematically repeated and validated by others.

V Lineages Tracing for General Data Warehouse Transformations

Transformations

→ Let a data set be any set of data items—tuples, values, complex objects—with no duplicates in the set. (The effect duplicates have on lineage tracing has been addressed in some detail in [CWW00].) A transformation T is any procedure that takes data sets as input and produces data sets as output. Here we will consider only transformations that take a single data set as input and produce a single output set. We extend our results to transformations with multiple input sets and output sets in [CW01]. For any input data set I, we say that the application of T to I resulting in an output set O, denoted T (I) = O, is an instance of T .

→ Given transformations T1 and T2, their composition T = T1 □ T2 is the transformation that first applies T1 to I to obtain I', then applies T2 to I' to obtain O. T1 and T2 are called T's component transformations. The composition operation is associative: (T1 □ T2) □ T3 = T1 □(T2□T3). Thus, given transformations T1, T2…, Tn, we represent the composition ((T1 □ T2)□ ….) □Tn as a transformation sequence T1□….□Tn. A transformation that is not defined as a composition of other transformations is atomic.

For now we will assume that all of our transformations are stable and deterministic. A transformation T is stable if it never produces spurious output items, i.e., T (□) = □. A transformation is deterministic if it always produces the same output set given the same input set. All of the example transformations we have seen are stable and deterministic. An example of an unstable transformation is one that appends a fixed data item or set of items to every output set,

regardless of the input. An example of a non-deterministic transformation is one that transforms a random sample of the input set. In practice we usually require transformations to be stable but often do not require them to be deterministic. See [CW01] for a discussion of when the deterministic assumption can be dropped.

CONCLUSION

LDION is a flexible method. Here watermarking is done by using fingerprinting method. It is a method for increasing accuracy of security of documents in networks. By using digital signatures and watermarking we can achieve this in this method. We differentiate between trusted senders (usually owners) and untrusted senders (usually consumers). In the case of the trusted sender, a very simple protocol with little overhead is possible. The untrusted sender requires a more complicated protocol, but the results are not based on trust assumptions and therefore they should be able to convince a neutral entity (e.g. a judge).Our work motivates further research on data leakage detection techniques for various document types and scenarios. For example, it will be an interesting future research direction to design a verifiable lineage protocol for derived data.

ACKNOWLEDGEMENT

The author wish to acknowledge the support of many respected persons who provided me with enchanting inspirations, enduring motivations and invaluable advices in successfully completing my research work. I have no hesitation whatsoever, to say that I would not have been able to complete the work without the guidance of those whom I could remind only with great reverence and gratitude. I would like to thank **Mrs. Jyothis Joseph** and Mrs.Anitha R for their collaboration and engagement in the projects.

REFERENCES

[1].http://www.privacyrights.org/data-breach. "Data breach cost,".

[2] P. Papadimitriou and H. Garcia-Molina, "Data leakage detection, "Knowledge and Data Engineering, IEEE Transactions on, vol. 23, no. 1,pp. 51–63, 2011.

[3]A.-R. Sadeghi, "The Marriage of Cryptography and Watermarking Beneficial and Challenging for Secure Watermarking and Detection," in Proceedings of the 6th International Workshop on Digital Watermarking, ser. IWDW '07, 2008, pp. 2–18. [4 I. J. Cox and J.-P. M. Linnartz, "Public watermarks and resistance to tampering," in International Conference on Image Processing (ICIP'97), 1997, pp. 26–29..

[5]N. P. Sheppard, R. Safavi-Naini, and P. Ogunbona, "On multiple watermarking," in MM&Sec, 2001, pp. 3–6.

[6] A.-R. Sadeghi, "Secure fingerprinting on sound foundations," Ph.D.dissertation, 2004.

[7] V. M. Potdar, S. Han, and E. Chang, "A survey of digital image watermarking techniques," in International Conference on Industrial Informatics, 2005. INDIN'05. 2005 3rd IEEE. IEEE, 2005, pp.709–716.