MODEL FOR FINDING UNRELIABLE AGENTS

[1] Karthick Myilvahanan [2] Mohemmed Yousuf [3] Gothandaraman
[1] Associate Professor [2] Assistant Professor [3] Assistant Professor
[1]Department of Computer Science and Engineering
MVJ College of Engineering, Bangalore, India

Abstract- When a data distributor has given sensitive data to a set of supposedly trusted agents (third parties), some of the data are leaked and found in an unauthorized place (e.g., on the web or somebody's lap) so, the distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means. In this paper, we proposed detection mechanism (across the agents) that improve the probability of identifying leakages and also guilt agent

Index Terms: Allocation strategies, data leakage, data privacy, fake records, leakage model, and guilt agent

INTRODUCTION

In the course of doing business, sometimes sensitive data must be handed over supposedly trusted third parties. For example, a hospital may give patient records to researchers who will devise new treatments. Similarly a university may give students results to some websites to publish the result at some point of time. In case of enterprise, it may outsource its data processing, so data must be given to various other companies. We call the owner of the data the *distributor* and the supposedly trusted third parties the *agents.* Our assumption is that data given to agent is leaked so our goal is to *detect* which agent leaks the data.

We consider applications where the original sensitive data cannot be perturbed. Perturbation is a very useful technique where the data are modified and made "less sensitive" before handed to agents. For example, in case one can replace exact values by ranges . In case of medical research, accurate data is needed.

Traditionally, leakage detection is handled by water-*marking*, e,g., a unique code is embedded in each distributed copy. If that copy is later discovered in the hands of an unauthorized party, the leaker can be identified. But it has some disadvantages, it involves modification of the original data and furthermore, watermarks can sometimes be destroyed if the recipient is *malicious.*

In this paper, we develop the model for assessing the "guilt" of agents. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker. Finally, we also consider the option of adding *"fake"* objects to the distributed set. Such objects do not correspond to real entities but appear realistic to the agents. In such a sense, the fake objects act as a type of watermark for the entire set, without modifying any individual members. If it turns out that an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty.

PROBLEM SETUP AND NOTATION

Entities and Agents

A *distributor* owns a set $T = \{t_1,\ldots,t_m\}$ of valuable data objects. The distributor wants to share some of the objects with a set of agents $U_1,U_2,\ldots,U_n$, but does not wish the objects be leaked to other third parties. The objects in T could be any type and size, e.g., they could be tuples in a relation, or relations in a database.

An agent U_i receives a subset of objects R_i T, determined either by a sample request or an explicit request:

- Sample request R_i = SAMPLE(T, m_i): Any subset of m_i, records from T can be given to U_i.

- Explicit request R_i = EXPLICIT(T, $condi_i$): Agent U_i receives all T objects that satisfy $condi_i$.

Example. Say that T contains customers records for a given company A. Company A hires a marketing agency U_i to do a online suvery, U_1 requests a sample of 1,000 customers records. At the same time, company A subcontracts with agent U_2 to handle billing for all California customers. Thus, U_2 receives all T records that satisfy the condition "state is California"

Guilty Agents

Suppose that after giving objects to agents, the distributor discovers that a set S T has leaked. This means that object leaks to third party.

Since the agents U_1,U_2….,U_N have some of the data, it is reasonable to suspect them leaking the data. However, the agents can argue that they are innocent, and that the S data were obtained by the third party through some other means.

Our goal is to estimate the likelihood that the leaked data came from the agents as opposed to other sources. Intuitively, the more data in S, the harder it is for the agents to argue they did not leak anything. Similarly, the "rarer" the objects, the harder it is to argue that the third party obtained the object through some other means. We also like to
find out if one of them, in particular, was more likely to be the leaker.

We say an agent U_i is guilty and if it contributes one or more objects to the third party. We denote the event that agent U_i is guilty for a given leaked set S by $G_i|S$. Our next step is to estimate $Pr\{G_i|S\}$, i.e., the probability that agent U_i is guilty given evidence S.

RELATED WORK

The guilt detection approach we present is related to the data provenance problem: tracing the lineage of S objects implies essentially the detection of the guilty agents. Our problem formulation simplifies lineage tracing, since we do not consider any data transformation.

Watermarks were initially used in images, video considerable redundancy. Recently, and other works have also studied marks insertion to relational data. Our approach and watermarking are similar in the sense of providing agents with some kind of receiver identifying information. However, by its nature, a watermark modifies the item being watermarked. If the object to be watermarked cannot be modified, then a watermark cannot be inserted. In such a cases, methods that attach watermarks to the distributed data are not applicable.

Finally, there are also lot of other works on mechanisms that allow only authorized users to access sensitive data through access control policies. Such approaches prevent in some sense data leakage by sharing information only with trusted parties.

In shadowed watermark generation algorithm is used. It is efficient enough for real world application. However Trusted Watermark Server (TWS) is a stateless manner, it doesn't have permanent storage. In watermark insertion and detection algorithm is used. Here the watermark is blind watermark, it does not provide any information.

BLACKBOX function to create fake object.

4 PROPOSED SYSTEM

In the proposed system, we develop a model for assessing "guilt" of agents. First, original and fake records are created. Depending on the agents' request (sample/explicit) fake objects is added to original records. Second, it is transfer to appropriate agents. Then if data is leaked, we identify who (agents) leaks the data.

4.1 System Architecture

The input of the proposed system is set of records (original and fake record) given to agents based on their request. The output of the system is to identify which agent leaks the data to third party.

System architecture of the proposed system is shown in Figure1. The entities are distributor, agent and third party.

Distributor owns the data set, based on the agent's request distributor allocates data to the appropriate agent. If the agent leaks the data to third party, he is guilt. We need to identify the guilt agent and its proability.

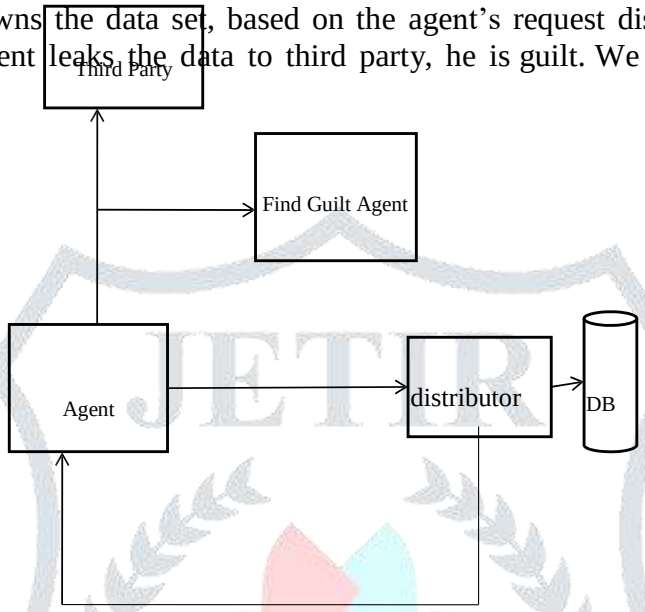

Figure 1: System Architecture

4.2 Functional Architecture

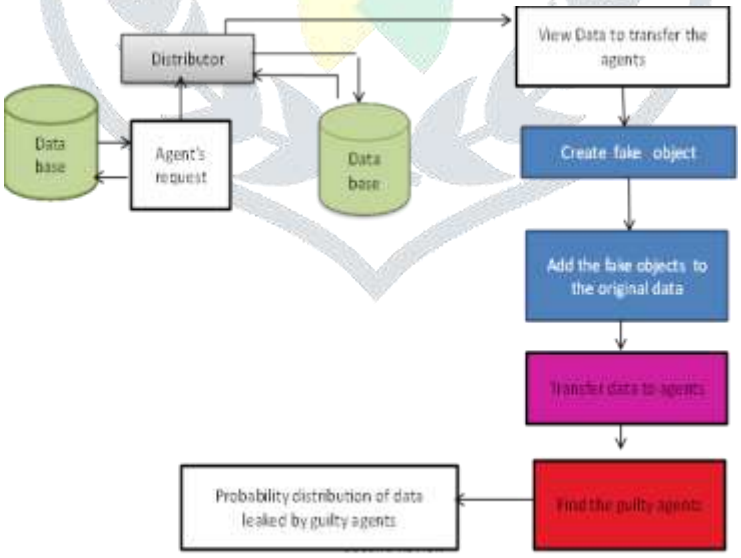

Figure 2 Functional Architecture

It has 3 Paths. They are 1.Data and fake object creation and allocation,2. System optimization and 3.Identifying the guily agent\

4.2.1 Data and Fake Object Creation and allocation

This part mainly focus on how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent.

Fake objects are objects created by distributor that are not in T. It looks like real object, and are distributed to agents together with T, in order to increase the chances of identifying agent who leaks data.

Fake objects *cannot be added in the following cases:*

- Affecting the privacy of agents
- Small modifications to some medical records is undesirable

CONDITION FOR CREATING FAKE OBJECT*:*

- Statistics shouldn't change by introducing fake object, if agents use that statistics
- Agents should not identify and distinguish the fake object from the real object

ADVANTAGES OF FAKE OBJECT:

- Provides "enough evidence" that agent leaked data
- Leakers cannot argue
- Fake object seems like realistic object

LIMIATATIONS OF FAKE OBJECT:

- Creation of fake object in mailing list-creating and monitoring email accounts consumes resources
- Limit the number of fake objects received by agent

PROPOSED PSEUDOCODE FOR CREATING FAKE OBJECT:

RETRIVE records from original database
MODIFY original records and store it in FAKE database
CREATE a circular queue as follows
S[ ] = {A,B,……Z } Do
BEGIN
S_1[ ] = {S[0],S[1],……..,S[9]}
Map 0,1,…..9 to A,B, ……..J respectively
Find the number of strokes in each alphabet and replace it in appropriate digits of card number
END
REPEAT the above steps for next credit card number by taking next 10 alphabets i.e., S_1 = {S[1],S[2],….S[10]}

Example: number alphabet [no of strokes]

0	[6] A	5	[4] F
1	[7] B	6	[6] G
2	[3] C	7	[5] H
3	[6] E	8	[3] I
4	[5]	9	[4] J

Card number: 734xxxxxxxxx124

565xxxxxxxxx735

Here is the simple **example** for data allocation

Distributor Database:

Original record T = $\{t_1,t_2,\ldots,t_{10}\}$ Fake record F = $\{f_1,f_2,\ldots,f_{10}\}$ California record = $\{t_1,t_5,t_6\}$
Ny record = $\{t_2,t_3,t_8\}$ Loan record = $\{t_4,t_7,t_9,t_{10}\}$

REQUEST from agents

Issue 1: Equivalence problem
Agent U_1: R_1 = EXP(T,California) = $\{t_1,t_5,t_6,\mathbf{f_1}\}$
Agent U_2: R_2 = SAM (T',1) = $\{t_1,t_5,t_6,\mathbf{f_5}\}$
Where T'= EXP (T,California)
Issue 2: Difference problem
Agent U_3: R_3 = EXP(T,Ny) = $\{t_2,t_3,t_8,\mathbf{f_2}\}$
Agent U_4: R_4 = EXP(T,California) = $\{t_1,t_5,t_6,\mathbf{f_6}\}$

Issue 3: partial overlap
Agent U_5: R_5 = SAM(T,5) = $\{t_1,t_2,t_4,t_7,t_8,\mathbf{f_7}\}$
Agent U_6: R_6 = EXP(T,Loan) = $\{t_2,t_3,t_8,\mathbf{f_3}\}$

4.2.2 System Optimization

The distributor *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. The distributor *objective* is to be able to detect an agent who leaks data. We

consider fake object distribution is the only possible *constraint relaxation.* Our objective is to *maximize the chances of detecting a guilty agent that leaks his data.*

PSEUDOCODE FOR E-RANDOM :

Input : Record, Agent and Condition

Output: Record with fake object
Step 1: Identify agents who can receive fake objects *Step 2:*
Calculate the total number of fake objects *Step 3*: While total
fake objects > 0
do
SELECTAGENT at random
SELECTFAKEOBJECT at random
Step 4: $R_i \leftarrow R_i U \{f\}$
Step 5: Add fake object to the agent set and also to the fake record set
Step 6: Decrement one from total fake record

PSEUDOCODE FOR S-RANDOM:

Input : Agent, Set of Records
Output: random object is selected
Step 1: Decide on the number of agents who have received objects
Step 2: For remaining agents do
SELECTOBJECT at random
Step 3: Decrement by one from total number of agents

PSEUDOCODE FOR E-OPTIMAL Input :

Agent,records
Output: Optimal agent with records
Step 1: Identify the agents that can receive fake object
Step 2: Select the optimal agent that will yield greatest chance of identifying the leaker
Step 3: Allocate the data set (original with fake) to agents

CONCLUSION AND FUTURE WORKS

In a perfect world, there would be no need to hand over sensitive data to agents That may unknowingly or maliciously leak it. And even if we had to hand over sensitive data, in a perfect world, we could watermark each object so that we could trace its origins with absolute certainty. However, in many cases, we must indeed work with agents that may not be certain if a leaked object came from an agent or from some other source, since certain data cannot admit watermarks.

In spite of these difficulties, we have shown that it is possible to assess the likelihood that an agent is and the data of responsible for a leak, based on the overlap of his data with the leaked data of other agents, and based on the probability that objects can

be "guessed" by other means. The algorithm we presented for creating fake objects, would generate automatically, by reducing the burden of distributor. And data distribution strategies improve the distributor's chances of identifying a leaker.

Proposed system is suitable to many applications too. Since, we are implemented data and fake object allocation and optimization module, our future work includes the implementation of identification of guilt agent.

ACKNOWLEDGEMENT

This work was supported by Department of Computer Science and Engineering, MVJ Engineering College, Chennai. The authors would like to thank the Management for providing infrastructure

REFERENCES

[1] P.Papadimitriou and H.Garcia-Molina, "Data Leakage Detection", In the *International conference on Data Engineering, Stanford University*, vol 23,No.1, 2011.

[2] R.Agrawal and J.Kiernan,"Watermarking Relational Databases," Proc. 28th *Int'l conf. Very Large Data Bases (VLDB '02), VLDB* Endowment,pp.155-166,2002.

[3] P.Bonatti, S.D.C. di vimercati,and P.Samarathi,"An Algebra for Composing Access Control Policies," *ACM Trans. Information and System Security,*vol.5,no.1,pp.1-35,2002

[4] P.Buneman, S.Khanna, and W.C. Tan,"Why and Where:A Charaterization of Data Provenance," Proc.Eighth *Int'l Conf. Database Theory (ICDT'01),* J.V.den Bussche and V.Vianu,eds., pp.316-330, jan 2001.

[5] F.Guo, J.Wang, Z.Zhang, X. Ye, and d.Li, "An Improved Algorithm to Watermark numeric Relational Data," *Information Security Applications*, pp.138-149, springer, 2006.

[6] S.jajodia, P.Samarathi, M.L.Sapino, and V.S. Subrahmanian, "Flexible Support For Multiple access Control Policies," *ACM Trans.Database Systems,*vol.26 ,no.2, pp.214-260,2001.

[7] Y.Li, v.Swapup, and S.Jajodia, "Fingerprinting Relational Databases :Schemes and Specialities," *IEEE Trans.Dependable and Secure Computing,* vol.2, no.1, pp.34-35, Jan-Mar.2005.

[8] J.J.K.O. Ruanaidh, W.J.Dowling, and F.M.Bonald, "Watermarking Digital Images for Copyright Protection," *IEEE Proc.Vision, Signal and Image Processing,* vol.143, no.4, pp.250-256,1996.

[9] R. Sion, M.Atallah, and S.prabhakar, "Rights Protection for Relational Data," *proc. ACM SIGMOD,* pp. 98-109,2003.

[10] L.Sweeney, "Achieving K-Anonymity Privacy Protection Using Generalization and Suppression, " http://en.scientificcommons.org/43196131,2002

[11] Hequn Xian, Dengguo Feng "Leakage Identification for Secret Relational Data Using Shadowed Watermarks", *International Conference* ,2009

[12] YinFan , Yu Rongwei,Wang Lina, ,Ma Xiaoyan "A distribution model for data leakage prevention" Proc 2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer (MEC),2014.