Data Leakage Detection Using Encrypted Fake Objects

Anusha.Koneru1
M.Tech(CSE) Student
Department of CSE, K L University
Guntur, Andhra Pradesh, India Guntur,

G. Siva Nageswara Rao2
Associate Professor
Department of CSE, K L University
Andhra Pradesh, India

J.Venkata Rao3
Assistant Professor
Department of CSE
K L University, Guntur

Abstract
Data leakage is a budding security threat to organizations, particularly when data leakage is carried out by trusted agents. In this paper, we present unobtrusive techniques for detecting data leakage and assessing the "guilt" of agents. Water marking is the long-established technique used for data leakage detection which involves some modification to the original data. To overcome the disadvantages of using watermark, data allocation strategies are used to improve the feasibility of detecting guilty agent. Distributor "intelligently" allocates data based on sample request and explicit request using allocation strategies in order to better the effectiveness in detecting guilty agent. Fake objects are designed to look like real objects, and are distributed to agents together with requested data. Fake objects encrypted with a private key are designed to look like real objects, and are distributed to agents together with requested data. By this way we can identify, the guilty agent who leaked the data by decrypting his fake object.

Keywords
Allocation strategies, data leakage, encryption, decryption, fake objects, guilty agent, optimization.

1. INTRODUCTION

Data Leakage can occur through a variety of methods - some are simple, some complex. As such, there is no single "silver bullet" to control Data Leakage. Data leakage detection [2], is an increasingly important part of any organization's ability to manage and protect critical and confidential information. Examples of critical and confidential data that applications can access include: Intellectual Property, Corporate Data, and Customer Data. Watermarks are very useful in a relational database [10], which involves some modification of data.The goal of our paper is to detect when the distributor's sensitive data has been leaked by agents, and show the probability for identifying the agent that leaked the data using encrypted fake objects.

Encryption is the process of encoding messages (or information) in such a way that eavesdroppers or hackers cannot read it, but that authorized parties can. Encrypting data enables confidentiality; this means that if the data falls into unauthorized hands the data is unreadable.

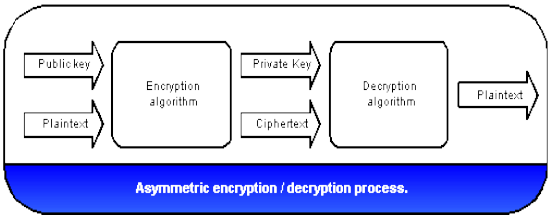

Figure1: Encryption Process

The type and length of the keys utilized depend upon the encryption algorithm and the amount of security needed. In conventional symmetric encryption a single key is used. With this key, the sender can encrypt a message and a recipient can decrypt the message but the security of the key becomes problematic. In asymmetric encryption, the encryption key and the decryption key are different. One is a public key by which the sender can encrypt the message and the other is a private key by which a recipient can decrypt the message.

We study unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, we study the following scenario [14]: After giving a set of objects to agents, the distributor discovers some of those same objects in an illegitimate place. (For example, the data may be found on a web site, or may be obtained through a legal discovery process.) At this point the distributor can assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means.

Manuscript received March 5, 2014
Manuscript revised March 20, 2014

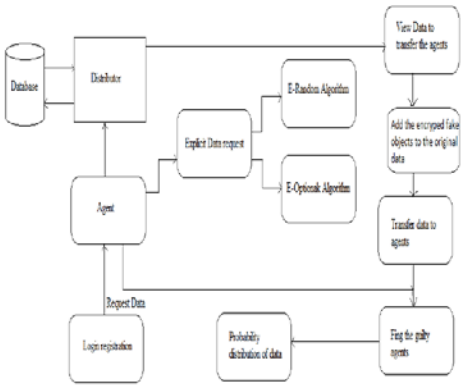

Figure.2: Data Leakage architecture

We develop a model as shown in Figure.2 for assessing the "guilt" of agents by considering the option of adding "encrypted fake" objects to the distributed set. Fake objects are encrypted using RSA algorithm. We also present algorithms for distributing objects to agents, in a way that improves our chances of identifying a leaker.

2. PROBLEM DEFINITION

Suppose a distributor owns a set T = {t1, tm} of valuable data objects. The distributor wants to share some of the objects with a set of agents U1, U2 … Un but does wish the objects be leaked to other third parties. An agent Ui receives a subset of Ri objects which belongs to T, determined either by a sample request or an explicit request,

- Sample Request R_i = SAMPLE (T, m_i): Any subset of *mi* records from *T* can be given to U_i.
- Explicit Request R_i = EXPLICIT (T, $cond_i$): Agent U_i receives all the *T* objects that satisfy $cond_i$.

The objects in T could be of any type and size, e.g., they could be tuples in a relation, or relations in a database. After giving objects to agents, the distributor discovers that a set S of T has leaked. This means that some third party called the target has been caught in possession of S. For example, this target may be displaying S on its web site, or perhaps as part of a legal discovery process, the target turned over S to the distributor. Since the agents U1, U2 ,…, Un 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 was obtained by the target through other means.

2.1. Agent Guilt 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 the relationship 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 U_i leaked *t* from its own R_i 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∈Ri} that have t in their data sets. Then using Assumption 2 and known probability p, we have

P_r{some agent leaked t to s}=1-p ……..1.1

Assuming that all agents that belong to V_t can leak *t* to *S* with equal probability and using Assumption 2 obtain,

$$p_r\{u_i \text{ leaked } t \text{ to } s\} = \begin{cases} \frac{1-p}{|v_t|}, & if\ u_i \in v_t \\ 0, & otherwise \end{cases} \quad \text{……..…1.2}$$

Given that agent U_i is guilty if he leaks at least one value to *S*, with Assumption 1 and Equation 1.2 compute the probability P_r {G_i | *S*}, agent U_i is guilty,

$$p_r\{G_i/s\} = 1 - \prod t \in s \cap R_i \left(1 - \frac{1-p}{|v_t|}\right) \quad \text{……1.3}$$

2.2. Data Allocation Problem

The distributor "intelligently" gives data to agents in order to improve the chances of detecting a guilty agent. There are four instances of this problem, depending on the type of data requests made by agents and whether "fake objects" are allowed. Agent makes two types of requests,

called sample and explicit. Based on the requests the fakes objects are added to data list. Fake objects are objects generated by the distributor that are not in set T using encryption algorithm. The objects are designed to look like real objects, and are distributed to agents together with the T objects, in order to increase the chances of detecting agents that leak data.

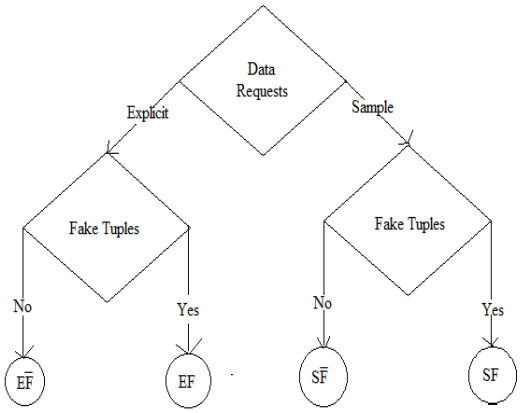

Figure. 3: Leakage Problem Instances

The Figure. 3 represents four problem instances with the names *EF*, $E\bar{F}$, *SF* and $S\bar{F}$, where *E* stands for explicit requests, *S* for sample requests, *F* for the use of fake objects, and *F* for the case where fake objects are not allowed .

The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. Since, fake objects may impact the correctness of what agents do, so they may not always be allowable. Use of fake objects is inspired by the use of "trace" records in mailing lists. The distributor creates and adds fake objects to the data that he distributes to agents. In many cases, the distributor may be limited in how many fake objects he can create.

In EF problems, objective values are initialized by agents' data requests. Say, for example, that T = {t1, t2} and there are two agents with explicit data requests such that R1= { t1, t2 } and R2= { t1 }. The distributor cannot remove or alter the R1 or R2 data to decrease the overlap R1\ R2 . However, say the distributor can create one fake object (B = 1) and both agents can receive one fake object (b1 =b2 = 1). If the distributor is able to create more fake objects, he could further improve the objective.

2.3. Optimization Problem

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 {Gj|S =Ri} or simply Pr {Gj|Ri } is the probability that agent Uj is guilty if the distributor discovers a leaked table S that contains all Ri objects.

The difference functions Δ (i, j) is defined as:

$$(i, j) = \Pr \{G_i | R_i \} - \Pr \{G_j | R_i \} \text{ ……. } 1.5$$

2.3.1 Problem definition: Let the distributor have data requests from n agents. The distributor wants to give tables R1,R2……,Rn to agents U1, . . .,Un, respectively, so that

- 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 Ri sets satisfy the agents' requests, we can express the problem as a multi-criterion

2.3.2 *Optimization problem:*

$$\text{maximize}_{(\text{over } R_1,\ldots,R_n)}(\ldots,\Delta(i,j),\ldots) \quad i \neq j \ldots\ldots 1.6$$

The approximation of objective of the above equation does not depend on agent's probabilities and therefore minimize the relative overlap among the agents as

$$\text{maximize}_{(\text{over } R_1,\ldots,R_n)}(\ldots,\frac{|R_i \cap R_j|}{|R_i|}\ldots..) \qquad i \neq j \ldots\ldots 1.7$$

This approximation is valid if minimizing these relative

Overlap $\frac{|R_i \cap R_j|}{|R_i|}$ maximizes Δ (*i, j*).

2.4. Objective Approximation

In case of sample request, all requests are of fixed size. Therefore, maximizing the chance of detecting a guilty agent that leaks all his data by minimizing is $\frac{|R_i \cap R_j|}{|R_i|}$ equivalent to minimizing($|R_i \cap R_j|$).The minimum value of $|R_i \cap R_j|$. maximizes $\prod(|R_i \cap R_j|)$ and Δ(i,j), since $\prod(|Ri|)$

is fixed. If agents have explicit data requests, then overlaps ($|R_i \cap R_j|$). are defined by their own requests and $|R_i \cap R_j|$ are fixed. Therefore, minimizing |Ri| j is equivalent to maximizing |Ri | (with the addition of fake objects).The maximum value of |Ri | minimizes Π(Ri) and maximizes Δ(i, j),since $\prod(R_i \cap R_j)$ is fixed.

3. ALLOCATION STRATEGIES

In this section the allocation strategies [4], solve exactly or approximately the scalar versions of Equation 1.7 for the different instances presented in Fig. 1. In Section A deals with problems with explicit data requests and in Section B with problems with sample data requests.

A. Explicit Data Request

In case of explicit data request with fake not allowed, the distributor is not allowed to add fake objects to the distributed data. So Data allocation is fully defined by the agent's data request. In case of explicit data request with fake allowed, the distributor cannot remove or alter the requests R from the agent. However distributor can add the fake object. In algorithm for data allocation for explicit request, the input to this is a set of request R1,R2,……,Rn from n agents and different conditions for requests. The e-optimal algorithm finds the agents that are eligible to receiving fake objects. Then create one fake object in iteration and allocate it to the agent selected. The e-optimal algorithm minimizes every term of the objective summation by adding maximum number bi of fake objects to every set Ri yielding optimal solution.

Step 1: Calculate total fake records as sum of fake records allowed.
Step 2: While total fake objects > 0
Step 3: Select agent that will yield the greatest improvement in the sum objective

i.e. i = $argmax(\frac{1}{|R_i|} - \frac{1}{|R_i|+1})\sum_j R_i \cap R_j$

Step 4: Create fake record
Step 5: Add this fake record to the agent and also to fake record set.
Step 6: Decrement fake record from total fake record set.

Algorithm makes a greedy choice by selecting the agent that will yield the greatest improvement in the sum-objective.

B. Sample Data Request:

With sample data requests, each agent U_i mayreceive any T from a subset out of $\binom{|T|}{m_i}$different ones. Hence, there are $\Pi_{i=1}^{n}\binom{|T|}{m_i}$ different allocations. In every allocation, the distributor can permute T objects and keep the same chances of guilty agent detection. The reason is that the guilt probability depends only on which agents have received the leaked objects and not on the identity of the leaked objects. Therefore, from the distributor's perspective there are different allocations. An object allocation that satisfies requests and ignores the distributor's objective is to give each agent a unique subset of T of size m. The s-max algorithm allocates to an agent the data record that yields the minimum increase of the maximum relative overlap among any pair of agents.

The s-max algorithm is as follows:

Step 1: Initialize Min_overlap ← 1, the minimum out of the maximum relative overlaps that the allocations of different objects to U_i

Step 2: for k ∈ {k | $t_k \in R_i$ } do

Initialize max_rel_ov ← 0, the maximum relative overlap between R_i and any set R_j that the allocation of t_k to U_i

Step 3: for all j = 1,..., n : j = i and $t_k \in R_j$ do

Calculate absolute overlap as

abs_ov ← $|R_i \cap R_j|$ + 1

Calculate relative overlap as

rel_ov ← abs_ov / min (m_i , m_j)

Step 4: Find maximum relative as

max_rel_ov ← MAX (max_rel_ov, rel_ov)

If max_rel_ov ≤ min_overlap then

min_overlap ← max_rel_ov

ret_k ← k

Return ret_k

It can be shown that algorithm s-max is optimal for the sum-objective and the max-objective in problems where M ≤ |T| and n < |T|. It is also optimal for the max-objective if |T| ≤ M ≤ 2 |T| or all agents request data of the same size. It is observed that the relative performance of algorithm and main conclusion do not change. If p approaches to 0, it becomes easier to find guilty agents and algorithm performance converges. On the other hand, if p approaches 1, the relative differences among algorithms grow since more evidence is need to find an Agent guilty.

The algorithm presented implements a variety of data distribution strategies that can improve the distributor's chances of identifying a leaker. It is shown that distributing objects judiciously can make a significant difference in identifying guilty agents, especially in cases where there is large overlap in the data that agents must receive.

4. SECURING THE FAKE OBJECTS USING RSA ALGORITHM

In this paper encryption is done by using RSA algorithm. RSA is the most widely used asymmetric key cryptography. It uses two different keys:

Public Key: known to every communicating entity in the network.

Private Key: known uniquely to the user

Generally, the receiver's public key is used for encrypting information and is sent to the receiver who decrypts it by his unique private key (known only him). This ensures confidentiality because it is assumed that only the key is known to the receiver. Incase of RSA algorithm, both the plain-text and the cipher-text are integers between 0 to (n-1) for some n.

RSA encrypts messages through the following algorithm, which is divided into 3 steps:

Step 1:Key Generation

I. Choose two distinct prime numbers p and q.

II. Find n such that n = pq.

n will be used as the modulus for both the public and private keys.

III. Find the totient of n, $\phi(n)$

$\phi(n)=(p-1)(q-1)$.

IV. Choose an e such that $1 < e < \phi(n)$, and such that e and $\phi(n)$ share no divisors other than 1 (e and $\phi(n)$ are relatively prime).

e is kept as the public key exponent.

V. Determine d (using modular arithmetic) which satisfies the congruence relation

$de \equiv 1 \pmod{\phi(n)}$.

In other words, pick d such that de - 1 can be evenly divided by (p-1)(q-1), the totient, or $\phi(n)$.

This is often computed using the Extended Euclidean Algorithm, since e and $\phi(n)$ are relatively prime and d is to be the modular multiplicative inverse of e.

d is kept as the private key exponent.

The public key has modulus n and the public (or encryption) exponent e. The private key has modulus n and the private (or decryption) exponent d, which is kept secret.

Step 2: Encryption

I. Person A transmits his/her public key (modulus n and exponent e) to Person B, keeping his/her private key secret.

II. When Person B wishes to send the message "M" to Person A, he first converts M to an integer such that $0 < m < n$ by using agreed upon reversible protocol known as a padding scheme.

III. Person B computes, with Person A's public key information, the ciphertext c corresponding to

$c \equiv me \pmod{n}$.

IV. Person B now sends message "M" in ciphertext, or c, to Person A.

Step 3: Decryption

I. Person A recovers m from c by using his/her private key exponent, d, by the computation

$m \equiv cd \pmod{n}$.

II. Given m, Person A can recover the original message "M" by reversing the padding scheme.

This procedure works since

$c \equiv me \pmod{n}$,

$cd \equiv (me)d \pmod{n}$,

$cd \equiv mde \pmod{n}$.

By the symmetry property of mods we have that

$mde \equiv mde \pmod{n}$.

Since $de = 1 + k\phi(n)$, we can write

$mde \equiv m1+k\phi(n) \pmod{n}$,

$mde \equiv m(mk)\phi(n) \pmod{n}$,

$mde \equiv m(mod n)$.

From Euler's Theorem and the Chinese Remainder Theorem, we can show that this is true for all m and the original message

$cd \equiv m \pmod{n}$, is obtained.

5. CONCLUSION

In the business management system, we have a set clients,patners and trusted members. The manager will transmit sum times 100 percent of data for the trusted third parties along with the authorized intended persons. We may not certain incase of a data leakage. Hence sharing of data should proceed by considering assumptions specified and may reduce the leakage through our efficient algorithm and by the process of asymmetric key encryption algorithm for the fake object creation and which includes our chances of detection process even when the intended persons are colluded.

6. FUTURE WORK

Our future work includes the inquiring of agent guilt models that capture leakage scenarios that are not studied in this paper. For instance, what is the appropriate model for cases where agents can collude and identify fake tuples? Another open problem is the extension of our allocation strategies so that they can handle agent requests in an online fashion (the presented strategies assume that there is a fixed set of agents with requests known in advance).

7. ACKNOWLEDGMENT

Students work is incomplete until they thank the almighty & his teachers. We sincerely believe in this and would like to thank Dr.V.Srikanth, Head of the Department, Computer Science & Engineering, Kluniversity, Vaddeswaram for his encouragement and motivation to write this paper. Also we are grateful to G.Siva Nageswara Rao, (CSE), K L University, Vaddeswaram for guiding us in writing this paper.

REFERENCES

[1] Wu, Jiangjiang, "An Active Data Leakage Prevention Model for Insider Threat Intelligence Information Processing and Trusted Computing" (IPTC), 2011 2nd International Symposium on, 22-23 Oct. 2011.

[2] Papadimitriou, Panagiotis, "A Model for Data Leakage Detection", Data Engineering, 2009. ICDE '09. IEEE 25th International Conference, March 29 2009-April 2 2009.

[3] Xiaosong Zhang , "Research and Application of the Transparent Data Encpryption in Intranet Data Leakage Prevention," Computational Intelligence and Security, 2009. CIS '09. International Conference. 11-14 Dec. 2009.

[4] Guo, Hongyu , "Identifying and Preventing Data Leakage in Multi-relational Classification," Data Mining Workshops (ICDMW), 2010 IEEE International Conference, 13 Dec,2010.

[5] Bhaduri, Kanishka , "Privacy-Preserving Outlier Detection Through Random Nonlinear Data Distortion," Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions, **Volume:** 41, Issue: 1 Feb. 2011.

[6] Li, Xiao-Bai , "A Tree-Based Data Perturbation Approach for Privacy-Preserving Data Mining", Knowledge and Data Engineering, IEEE Transactions, **Volume:** 18, Issue: 9, 2006.

[7] Lizhong Zhang, Wei Gao; Nan Jiang; "Relational databases watermarking for textual and numerical data", Mechatronic Science, Electric Engineering and Computer (MnEC), 2011 International Conference, 9-22 Aug. 2011.

[8] Yoshida, Maki , "Watermarking Cryptographic Data", Intelligent Information Hiding and Multimedia Signal Processing, 2009. IIH-MSP '09. Fifth International Conference, 12-14 Sept. 2009.

[9] L. Sweeney. "Achieving k-anonymity privacy protection using generalization and suppression", 2002.

[10] 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.

[11] P. Buneman and W.-C. Tan,"Provenance in Databases", Proc. ACM SIGMOD, pp. 1171-1173, 2007.

[12] Y.Li,V. Swarup, and S. Jajodia, "Fingerprinting Relational Databases: Schemes and Specialties", IEEE Trans. Dependable and Secure Computing, vol. 2, no. 1, pp. 34-45, Jan.-Mar. 2005.