ISSN : 2394-2975 (Online)
ISSN : 2394-6814 (Print)

Innovative method of Detecting Sensitive Data Leakage and Providing Security using Asymmetric Crypto-Graphic Algorithm

[I]Hamid Iqbal Khan, [II]Akram Khan, [III]M.A.Qadeer

Abstract

The data sets are been distributed to the data distributors which are so called third party agents or third party users. In some circumstances, the distributed data seems to be leaked by some unknown users. This study will propose some data distribution strategies in order to increase and develop the prospects of reducing the data leakages which are not based on the changes done on the data that is released such as watermarks. Alternately, a realistic data records can be created which are not genuine in order to identify and detect the data leakages from third party user.

Keywords

Data leakage, Data distributor, watermarking, sample data request, explicit request, Fake object, Data allocation, RSA, Pseudo code

I. Introduction

Data leakage is the unauthorized transmission of data or information from within an organization to an external destination or recipient. Data leakage is defined as the accidental or intentional distribution of private or sensitive data to an unauthorized entity. Sensitive data of companies and organization includes intellectual property, financial information, patient information, personal credit card data and other information depending upon the business and the industry. A data distributor has given this sensitive data to a set of supposedly trusted agents (third parties). Some of the data are leaked and found in an unauthorized place. 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. We call the owner of the data the distributor and the supposedly trusted third parties the agents. Our goal is to detect when the distributor's sensitive data have been leaked by agents, and if possible to identify the agent that leaked the data. We propose data allocation strategies hat improve the probability of identifying leakages. These methods do not rely on alterations of the released data (e.g.watermarks). In some cases, we can also inject-realistic but fake

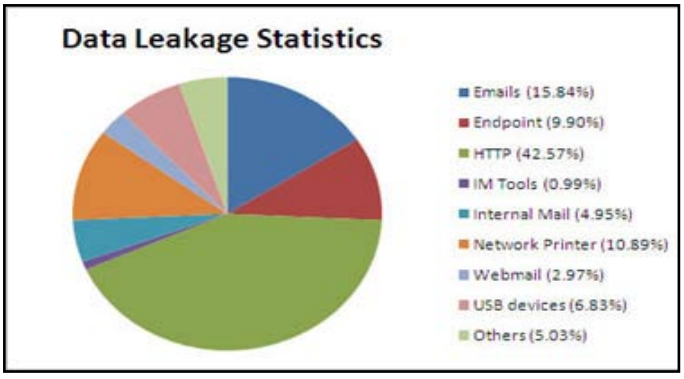


Fig.1 : Detailed survey of Data Leakage Statistics

II. Existing System

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 being handed to agent. In some cases it is important not to alter the original distributor's data. Traditionally, leakage detection is handled by watermarking, 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. Watermarks can be very useful in some cases, but again, involve some modification of the original data. Furthermore, watermarks can sometimes be destroyed if the data recipient is malicious. There is no proper existing system to detect the information leaking in secured manner.E.g. A hospital may give patient records to researchers who will devise new treatments. Similarly, a company may have partnerships with other companies that require sharing customer data. Another enterprise 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.

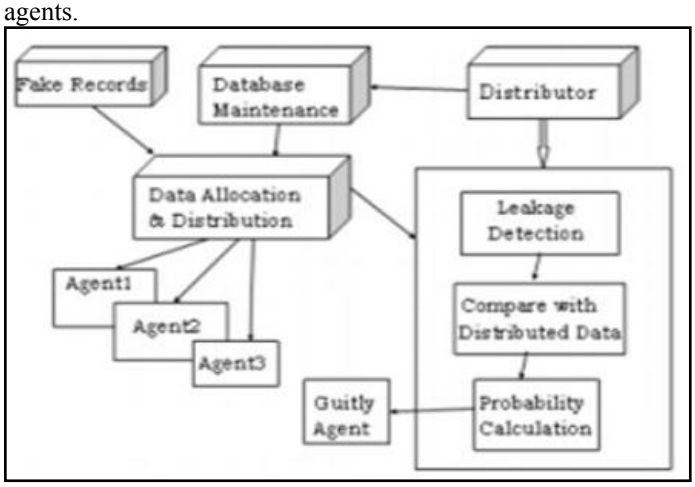
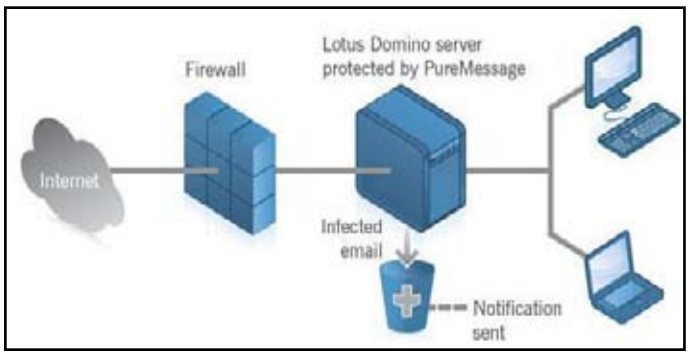


Fig. 2 : Detailed System Architecture

III. Proposed System

Our goal is to detect, when the distributor's sensitive data has been leaked by agents, and if possible to identify the agent that leaked the data. Perturbation is a very useful technique where the data is modified and made "less sensitive" before being handed to agents. We propose to develop unobtrusive techniques for detecting leakage of a set of objects or records. In this section, we propose to develop a 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 a sense, the fake objects acts as a type of watermark for the entire set, without modifying any individual members [2]. If it turns out an agent was given one or more fake objects that were leaked, then the distributor can be more confident that agent was guilty.



A. Modules

1. Login / Registration
2. Data Distributor
3. Data Allocation Module
4. Fake Object Module
5. Data Leakage protection Module
6. Finding Guilty Agents Module

B. Modules Description

1. Login / Registration:

This is a module mainly designed to provide the authority to a user/agent in order to access the other modules of the project. Here a user/agent can have the accessibility authority after the registration

2. Data Distributor

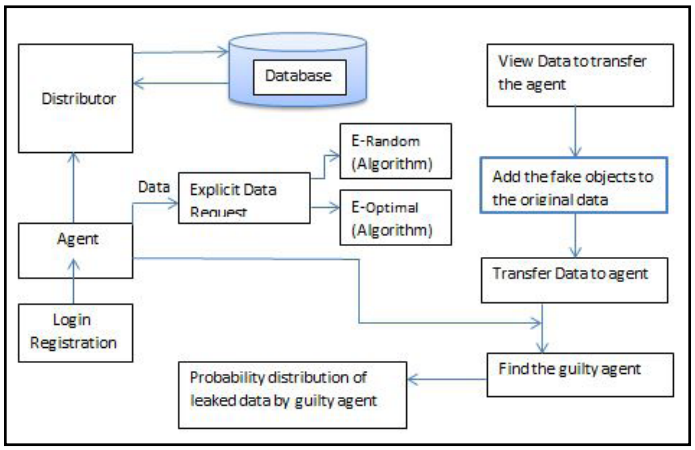
A data Distributor part is developed in this module. A data distributor has given sensitive data to a set of supposedly trusted agents (third parties). Some of the data is leaked and found in an unauthorized place (e.g., on the web or somebody's laptop). The distributor must assess the likelihood that the leaked data came from one or more agents, as opposed to having been independently gathered by other means.

3. Fake Object module

Fake objects are objects generated by the distributor in order to increase the chances of detecting agents that leak data. The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. Our use of fake objects is inspired by the use of "trace" records in mailing lists.

4. Data Allocation Module:

The main focus of our project is the data allocation problem as how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent.



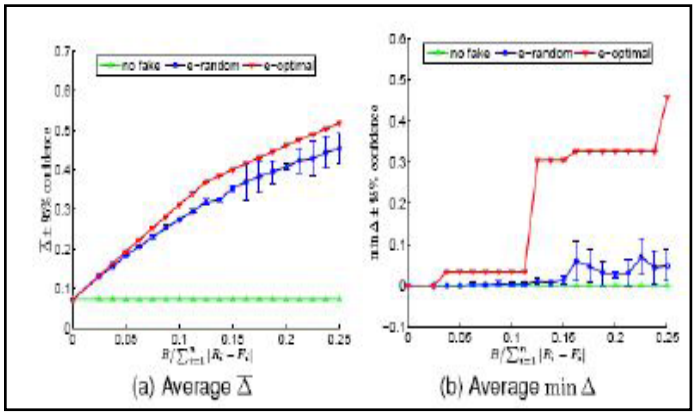
Explicit Data Reques

This is general algorithm utilized by other alorithm

Algorithm 1 Allocation for Explicit Data Requests (EF)

Input: $R_1,\ldots,R_n$, $cond_1,\ldots,cond_n$, $b_1,\ldots,b_n$, B
Output: $R_1,\ldots,R_n$, $F_1,\ldots,F_n$

1: $R \leftarrow \phi$ ▷ Agents that can receive fake objects
2: for i=1, ……,n do
3: if $b_i > 0$ then
4: $R \leftarrow R \cup \{i\}$
5: $F_i \leftarrow \phi$
6: while B> 0 do
7: $i \leftarrow SELECTAGENT(R, R_1,\ldots,R_n)$
8: $f \leftarrow CREATEFAKEOBJECT(R_i, F_i, cond_i)$
9: $R_i \leftarrow R_i \cup \{f\}$
10: $F_i \leftarrow F_i \cup \{f\}$
11: $b_i \leftarrow b_i - 1$
12: if $b_i = 0$ then
13: $R \leftarrow R \setminus \{R_i\}$
14: $B \leftarrow B-1$



Graph 1: Evaluation of Explicit Data Request

Algorithm 2. Agent Selection for e-random
1 function SELECTAGENT (R, R1; . . . Rn)
2: i_ select at random an agent from R
3: return i

In lines 1-5, Algorithm 1 finds agents that are eligible to receiving fake objects in O (n) time. Then, in the main loop in lines 6-14, the algorithm creates one fake object in every iteration and allocates it to random agent. The main loop takes O(B) time. Hence, the running time of the algorithm is O

(n+B).If B ≥Σn i-1 bi, the algorithm minimizes every term of the objective summation by adding the maximum number bi of fake objects to every set Ri, yielding the optimal solution. Otherwise, if B ≤ Σn i-1 bi, the algorithm just selects at random the agents that are provided with fake objects. Algorithm 3 It denote the combination of Algorithms 1 and 3 by e-optimal

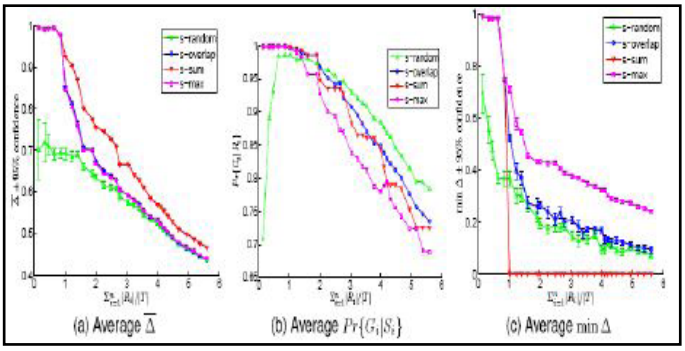
Algorithm 3. Agent Selection for e-optimal
1: function SELECTAGENT (R;R1; . . .;Rn)
2: i←argama(1÷|Ri|-1÷|Ri|+1∑ |Ri intersect Rj|)
3: return i
Algorithm 3 makes a greedy choice by selecting the agent that will yield the greatest improvement in the sum objective. The cost of this greedy choice is Odn2T in every iteration [1]. The overall running time of e-optimal is O(n+ n2B) =O(n2B).

Sample Data Request

This algorithm is meant for making a greedy choice of choosing an agent that causes improvement in the sum-objective

Algorithm 4: Allocation for Sample Data Requests(SF)

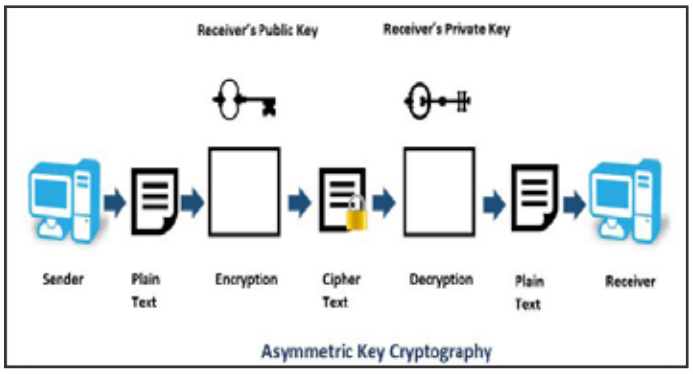
Input: $m_1,.....,m_n$, $|T|$ ▷ Assuming $m_i \le |T|$
Output: $R_1,.....,R_n$
1: a ← $0_{|T|}$ ▷ a[k]: number of agents who have received object t_k
2: $R_1 \leftarrow \phi,.....,R_n \leftarrow \phi$
3: remaining ← $\Sigma^n_{i=1} m_i$
4: while remaining > 0 do
5: for all i=1,.....,n:$|R_i| < m_i$ do
6: k ← SELECTOBJECT (i, R_i) ▷ May also use additional parameters
7: $R_i \leftarrow R_i \cup \{t_k\}$
8: a[k] ← a[k] + 1
9: remaining ← remaining-1

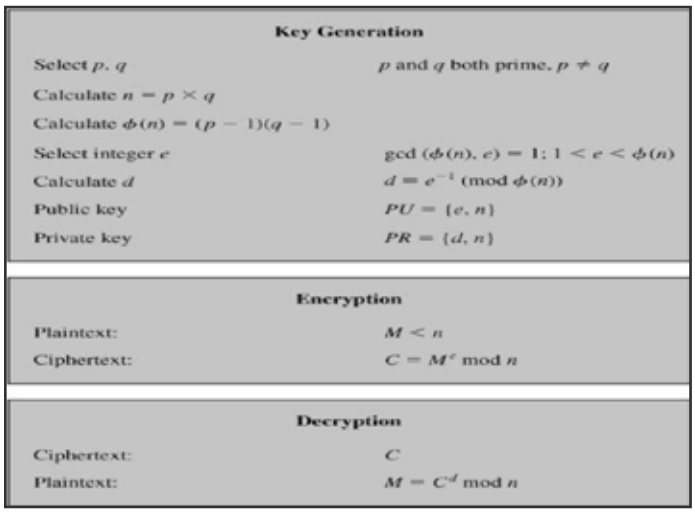


Graph 2: Evaluation of sample data request

5. Data Leakage protection Module

In this module, to protect the data leakage, a secret key(private key) is sent to the agent who requests for the files. The secret key is sent through the email id of the registered agents. Without the secret key the agent cannot access the file sent by the distributor. There are very many encryption algorithms but in this paper we are describing the Rivest, Shamir, Adleman (RSA) Algorithm [8]. The RSA algorithm is the most commonly used public key encryption algorithm. Two keys are used: Public Key and Private Key. So in a public key cryptosystem, the sender encrypts the data using the public key of the receiver and uses an encryption algorithm that is also decided by the receiver and the receiver sends only the encryption algorithm and public key. But by using the public key, data can only be encrypted but not decrypted, and the data is only decrypted by the private key that only the receiver has. So no one can hack our data.





Pseudo code

Key Generation Algorithm

1. Choose two very large random prime integers:
   p and q
2. Compute n and φ(n):
   n = pq and φ(n) = (p-1)(q-1)
3. Choose an integer e, 1 < e < φ(n) such that:
   gcd(e, φ(n)) = 1 (where gcd means greatest common denominator)
4. Compute d, 1 < d < φ(n) such that:
   ed ≡ 1 (mod φ(n))

- the public key is (n, e) and the private key is (n, d)
- the values of p, q and φ(n) are private
- e is the public or encryption exponent
- d is the private or decryption exponent

Encryption

The cyphertext C is found by the equation 'C = M^e mod n' where M is the original message.

Decryption

The message M can be found form the cyphertext C by the equation 'M = C^d mod n'.

6. Finding Guilty Agents Module

The Optimization Module is 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. This module is designed using the agent – guilt model. Here a count value (also called as fake objects) is incremented for any transfer of data occurrence when agent transfers data. Fake objects are stored in database.

7. Optimization Module

The Optimization Module is the distributor's data allocation to agents has one constraint and one objective. The agent's constraint is to satisfy distributor's 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. User can able to lock and unlock the files for secure.

IV. Result

V. Future Scope

One open problem is 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).

VI. Conclusion

From the study of the data leakage, we can detect and prevent the data from the leak by using some algorithms and techniques. 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.

References

[1] *Panagiotis Papadimitriou and Hector Garcia-Molina,"Data Leakage Detection," IEEE Trans, Knowledge and Data Engineering, vol. 23, no. 1, January 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] *Sandip A.Kale, Prof. Kulkarni S.V. (Department Of Computer Sci.&Engg,MIT College of Engg, Dr.B.A.M.University, Aurangabad(M.S),India, Data Leakage Detection: A Survey, (IOSR Journal of Computer Engineering (IOSRJCE)ISSN : 2278-0661 Volume 1, Issue 6 (July-Aug 2012), PP 32-35 www.iosrjournals.org*

[4] *IEEE Transactions On Knowledge And Data Engineering, Vol. 22,No. 3, March 2011 Data Leakage Detection Panagiotis Papadimitriou,Member, IEEE, Hector Garcia-Molina, Member, IEEE P.P (2,4-5)*

[5] *Faith M. Heikkila, Data Leakage: What You Need to Know, Pivot Group Information Security Consultant. P.P (1-3)*

[6] *Rudragouda G Patil Dept Of CSE, The Oxford College Of Engg, Bangalore. International Journal Of Computer Applications In Engineering Sciences [VOL I, ISSUE II, JUNE 2011] [ISSN: 2231- 4946] P.P (1, 4) Development Of Data Leakage Detection Using Data Allocation Strategies*

[7] *International Journal of Computer Applications in Engineering Sciences [ISSN: 2231-4946]197 | P a g e Development of Data leakage Detection Using Data Allocation Strategies Rudragouda G Patil Dept of CSE,The Oxford College of Engg, Bangalore.*

[8] *http://en.wikipedia.org/wiki/RSA_(algorithm)*