Available online at www.internationaleJournals.com

International eJournals

International eJournal of Mathematics and Engineering 107 (2010) 981 - 984

ISSN 0976 – 1411

INTERNATIONAL eJOURNAL OF MATHEMATICS AND ENGINEERING

www.internationaleJournals.com

ALLOCATION STRATEGIC FOR DETECTION OF DATA MISUSE USING DATA MINING TECHNIQUES

[1]G.Ravikumar.
Department of Computer Science
Rayalaseema University,
Kurnool., A.P, India

[2]G.A.Ramachandra
Head & Associate Professor
Department of Computer Science
&Technology, S. K. University,
Anantapur, A.P., India

[3]K.Nagamani
Department of Computer Science
Rayalaseema University, Kurnool.
Kanchar.lamni@gmail.com

Abstract- this research paper Nagmani deals with data distributor has given sensitive data to a set of supposedly trusted agents. Some of the data are misused and found in an unauthorized place i.e. some one is using that data. 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 propose data allocation strategies across the agents that improve the probability of identifying data misuse or 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" data records to further improve our chances of detecting leakage and identifying the guilty party.

Key words- Allocation strategies, data leakage, data privacy, fake records, leakage model.

1 INTRODUCTION

When we do business, sometimes sensitive data must be handed over to supposedly trusted third parties. For example, a Cricket club may give player records to researchers who will devise new. 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. 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 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 agents. For example, one can add random noise to certain attributes, or one can replace exact values by ranges [18]. However, in some cases, it is important not to alter the original distributor's data. For example, if outsourcers are doing our payroll, he must have the exact salary and customer bank account numbers. If medical researchers will be treating patients (as opposed to simply computing statistics), they may need accurate data for the patients. 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

misuser 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. In this research paper, we study unobtrusive techniques for detecting leakage of a set of objects or records. Specifically, we study the following scenario: After giving a set of objects to agents, the distributor discovers some of those same objects in an unauthorized place. (For example, the data may be found on a website, 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. Using an analogy with cookies stolen from a cookie jar, if we catch Freddie with a single cookie, he can argue that a friend gave him the cookie. But if we catch Freddie with five cookies, it will be much harder for him to argue that his hands were not in the cookie jar. If the distributor sees "enough evidence" that an agent leaked data, he may stop doing business with him, or may initiate legal proceedings.

2. DATA ALLOCATION STRATEGIC PROBLEM

The main focus of this paper is the data allocation problem: how can the distributor "intelligently" give data to agents in order to improve the chances of detecting a guilty agent? As illustrated in Fig. 2, there are four instances of this problem we address, depending on the type of data requests made by agents and whether "fake objects" are allowed.

The two types of requests we handle were defined in Section 2: sample and explicit. Fake objects are objects generated by the distributor that are not in set T. The objects are designed to look like real objects, and are distributed to agents together with T objects, in order to increase the chances of detecting agents that leak data. We discuss fake objects in more detail in Section 2.1. As shown in Fig. 1, we represent our four problem instances with the names EF, EF, SF, and SF, where E stands for explicit requests, S for sample requests, F for the use of fake objects, and F or the case where fake objects are not allowed. Note that, for simplicity, we are assuming that in the E problem instances, all agents make explicit requests, while in the S instances, all agents make sample requests. Our results can be extended to handle mixed cases, with some explicit and some sample requests. We provide here a small example to illustrate how mixed requests can be handled, but then do not elaborate further. Assume that we have two agents with requests R_1= EXPLICIT(T', $cond_1$) and R_2 =SAMPLE(T' 1), where T'= EXPLICIT(T', $cond_2$.) Further, say that $cond_1$ is "state=CA" (objects have a state field). If agent U2 has the same condition $cond_2$ =$cond_1$, we can create an equivalent problem with sample data requests on set T'. That is, our problem will be how to distribute the CA objects to two agents, with R_1 =SAMPLE(T', |T'|) and R_2 = SAMPLE(T', 1). If instead U_2 uses condition "state = NY," we can solve two different problems for sets T' and T- T'. In each problem, we will have only one agent. Finally, if the conditions partially overlap, R1 $\cap$T'$\neq$ Ø but $R_1 \neq$T', we can solve three different problems for sets R_1 –T', $R_1 \cap$T', and T'- R_1.

2.1 Fake Objects

The distributor may be able to add fake objects to the distributed data in order to improve his effectiveness in detecting guilty agents. However, fake objects may impact the correctness of what agents do, so they may not always be allowable. The idea of perturbing data to detect leakage is not new, e.g., [1]. However, in most cases, individual objects are perturbed, e.g., by adding random noise to sensitive salaries, or adding a watermark to an image. In our case, we are perturbing the set of distributor objects by adding

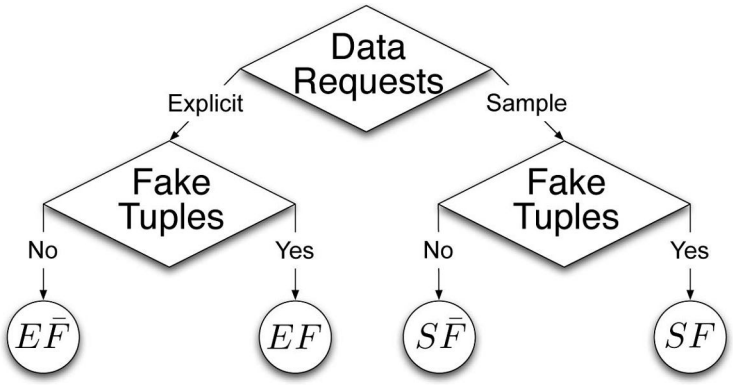

Fig.1. Leakage problem instances.

Algorithm 1. Allocation for Explicit Data Requests (EF)
Input: R_1; . . .,Rn, $cond_1$,. . . ,condn, b_1; . . . , bn, B
Output: R_1, . . .,Rn, F_1, . . . , Fn
1: R← Ø -Agents that can receive fake objects
2: for i= 1, . . . , n do
3: if bi > 0 then
4: R← R U̍ {i}
5: F← Ø
6: while B>0 do
7: i ←SELECTAGENT(R, R_1 . . .,Rn)
8: f← CREATEFAKEOBJECT (Ri, Fi, condi)
9: Ri ←R U̍ {f}
10: Fi← Fi U̍{f}
11: bi← bi -1
12: if bi =0 then
13: R ← Rn\ {R_i}
14: B ← B -1

CONCLUSIONS

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 100 percent trusted, and we 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 responsible for a leak, based on the overlap of his data with the leaked data and the data of other agents, and based on the probability that objects can be “guessed” by other means. Our model is relatively simple, but we believe that it captures the essential trade-offs. The algorithms we have presented implement a variety of data distribution strategies that can improve the distributor’s chances of identifying a leaker. We have 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. Our future work includes the investigation of agent guilt models that capture leakage scenarios that are not studied in this paper. For example, what is the appropriate model for cases where agents

can collude and identify fake tuples? A preliminary discussion of such a model is available in [14]. 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).

REFERENCES

[1] 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.
[2] P. Bonatti, S.D.C. di Vimercati, and P. Samarati, "An Algebra for Composing Access Control Policies," ACM Trans. Information and System Security, vol. 5, no. 1, pp. 1-35, 2002.
[3] P. Buneman, S. Khanna, and W.C. Tan, "Why and Where: A Characterization 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.
[4] P. Buneman and W.-C. Tan, "Provenance in Databases," Proc. ACM SIGMOD, pp. 1171-1173, 2007.
[5] Y. Cui and J. Widom, "Lineage Tracing for General Data Warehouse Transformations," The VLDB J., vol. 12, pp. 41-58, 2003.
[6] S. Czerwinski, R. Fromm, and T. Hodes, "Digital Music Distribution and Audio Watermarking," http://www.scientificcommons. org/43025658, 2007.
[7] 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.
[8] F. Hartung and B. Girod, "Watermarking of Uncompressed and Compressed Video," Signal Processing, vol. 66, no. 3, pp. 283-301, 1998.
[9] S. Jajodia, P. Samarati, 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.
[10] 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.
[11] B. Mungamuru and H. Garcia-Molina, "Privacy, Preservation and Performance: The 3 P's of Distributed Data Management," technical report, Stanford Univ., 2008.
[12] V.N. Murty, "Counting the Integer Solutions of a Linear Equation with Unit Coefficients," Math. Magazine, vol. 54, no. 2, pp. 79-81, 1981.
[13] S.U. Nabar, B. Marthi, K. Kenthapadi, N. Mishra, and R. Motwani, "Towards Robustness in Query Auditing," Proc. 32nd Int'l Conf. Very Large Data Bases (VLDB '06), VLDB Endowment, pp. 151-162, 2006.
[14] P. Papadimitriou and H. Garcia-Molina, "Data Leakage Detection," technical report, Stanford Univ., 2008.
[15] P.M. Pardalos and S.A. Vavasis, "Quadratic Programming with One Negative Eigenvalue Is NP-Hard," J. Global Optimization, vol. 1, no. 1, pp. 15-22, 1991.
[16] J.J.K.O. Ruanaidh, W.J. Dowling, and F.M. Boland, "Watermarking Digital Images for Copyright Protection," IEE Proc. Vision, Signal and Image Processing, vol. 143, no. 4, pp. 250-256, 1996.
[17] R. Sion, M. Atallah, and S. Prabhakar, "Rights Protection for Relational Data," Proc. ACM SIGMOD, pp. 98-109, 2003.
[18] L. Sweeney, "Achieving K-Anonymity Privacy Protection Using Generalization and Suppression," http://en.scientificcommons. org/43196131, 2002.