第 41 卷 第 4 期 计 算 机 工 程 2015 年 4 月
Vol. 41 No. 4 Computer Engineering April 2015

·安全技术· 文章编号：1000-3428(2015)04-0140-05 文献标识码：A 中图分类号：TP309

基于数据分配策略的数据泄漏检测研究

唐昌龙，刘吉强

（北京交通大学计算机与信息技术学院，北京 100044）

摘 要：使用数据分配策略（DAS）能有效检测数据泄露。鉴于此，分别介绍基于第 1 代和第 2 代 DAS 的检测方法。利用过失模型在解决数据泄漏检测问题上的优势，研究基于过失代理模型的数据分配算法，对其中的明确数据请求分配算法、随机假对象分配算法和优化 Agent 选择算法进行优化，并对代理模型进行实验仿真，结果表明，优化算法在添加伪数据后能明显提高对过失代理的检测率。此外，还给出针对过失模型的研究方向。

关键词：数据分配策略；过失模型；数据泄漏保护；数据隐私；伪数据

中文引用格式：唐昌龙，刘吉强. 基于数据分配策略的数据泄漏检测研究[J]. 计算机工程，2015，41(4)：140-144.

英文引用格式：Tang Changlong，Liu Jiqiang. Research on Data Leakage Detection Based on Data Allocation Strategy[J]. Computer Engineering，2015，41(4)：140-144.

Research on Data Leakage Detection Based on Data Allocation Strategy

TANG Changlong，LIU Jiqiang

（School of Computer and Information Technology，Beijing Jiaotong University，Beijing 100044，China）

【Abstract】 Data Allocation Strategy（DAS） system shows its advantages to address data leakage problems. This paper further illustrates the recent advances in DAS-based data leakage detection techniques including the first and second generation detection methods. Guilt model is demonstrated the potential as a suitable candidate for data leakage detection problems. The algorithms on explicit data request，random object allocation and agent selection are analyzed and evaluated with supporting data. Amodification on data allocation design is proposed. The experiment shows that the proposed algorithvs increases the possibility of detecting the guilt agents which leak the data. Consequently，the future works for guilt model are discussed.

【Key words】 Data Allocation Strategy（DAS）；guilt model；Data Leakage Protection（DLP）；data privacy；pseudo data

DOI：10.3969/j.issn.1000-3428.2015.04.026

1 概述

数据泄露保护（Data Leakage Protection，DLP）也称为数据暴露防护或防数据泄露[1]。DLP 系统的主要作用是防止内部和外部人员有意或无意地将敏感信息发送到未经授权的第三方。根据数据流的使用状态，DLP 系统可以用于保护静止的数据（Data-at-Rest，DaR）、传输中的数据（Data-in-Motion，DiM）和使用中的数据（Data-in-Use，DiU）[2]。

当前针对 DLP 的研究方向主要有 2 个：（1）监控数据流防止敏感数据失去控制；（2）敏感数据失去控制后，识别流出渠道。第（1）种研究方向目前主要的技术有模式匹配算法、复杂指纹算法和贝叶斯统计分析算法等[1]。本文主要研究第（2）种方向，即如何使 DLP 系统通过应用数据分配策略（Data Allocation Strategy，DAS）在敏感数据丢失后能够快速检测数据是通过哪个第三方（Agent）流失的。

目前国内外关于 DAS 的最新研究进展，按其研究方法可以分成 2 个阶段，分别称为第 1 代 DAS 和第 2 代 DAS。第 1 代 DAS 算法和技术[3]包括水印技术、标记化算法、诚信机制、信息传输决策点技术、便携式数据绑定算法（Portable Data Binding，PDB）和流模型算法等。第 1 代 DAS 算法的主要缺点是需要对源数据进行一定的修改，而且在检测性能上也不能满足大规模的应用。第 2 代 DAS 算法主要通过应用分配策略分配给每个代理（Agent）不同的数据对象从而在不改变源数据的基础上增加识别泄密者的机会。第 2 代 DAS 技术主要建立在过失模

作者简介：唐昌龙（1980－），男，博士研究生，主研方向：信息安全体系；刘吉强，教授、博士。

收稿日期：2014-04-28 **修回日期**：2014-08-15 **E-mail**：AlanTang.IT@gmail.com

型的基础上[4]。有少数文献也研究了影子模型和数据看守/泄漏检测技术。本文对基于第 1 代和第 2 代 DAS 的检测方法进行研究,优化数据分配算法,并给出针对过失模型的研究方向。

2　相关研究

根据数据分配策略技术在不同环境下的应用,研究者进行了较为广泛的研究。

文献[5]侧重分析了数据分配策略技术与加密技术相结合来防止数据泄漏,研究了利用非对称加密算法(RSA 算法)对伪数据进行加密和揭秘的可能性和大致过程,主要是密钥生成、源数据加密和解密验证。另外,该文还分析了如何优化数据对象分配过程以提高检测成功率。其文章没有提供具体数据来说明加密技术与数据分配策略相结合的实际性能,也没有比较各种不同非对称加密算法与数据分配策略相结合的稳定性和可靠性。

文献[6]研究了数据分配及泄漏检测技术在云计算环境中的应用情况,特别是针对云数据存储环境的保护。该文阐述了关系数据权限保护的机制,比如将数据组拆分成大量的小数据区,然后根据数据区的不同属性添加伪数据以达到泄漏检测目的。同时此文还分析了 K-匿名隐私保护技术,该技术可以确保一个发布的数据/记录能够关联到至少 K 个个体。另外,该文还论述了线性追踪技术在通用数据转换中的应用。

文献[7]研究了数据分配策略技术在 E-mail 过滤中的机制,阐述了一种用伪数据检测过失代理的模型并将其应用到 Email 安全过滤系统中。当未经授权的用户收到来自被检测为过失代理的邮件时,邮件的内容和附件是不可读的,从而保护了组织的敏感业务信息不被泄漏。

3　基于第 1 代 DAS 的数据泄露检测

在数据没有与第三方共享的环境下,组织一般对敏感数据有充分的控制手段,因此也相对容易防止和检测数据泄露。目前大部分 DLP 安全技术产品和解决方案都建立在组织对数据有绝对控制权的假设或前提下(比如 Symantec,McAfee 等安全公司的数据防护产品)。但也存在很多需要进行数据共享的情况,比如在进行业务流程外包(BPO)的组织中,第三方需要访问组织的一些敏感信息来履行它们的合同与职责。例如一个人力资源 BPO 服务方需要访问企业的员工数据库(可能包括社会保险号等敏感信息),又例如一个市场服务方可能不被完全信任或不能被安全的进行管理。在电子化时代,关系型数据库是非常容易被复制的。而且在很多情况下,服务商在利益的驱动下会将一些保密的商业信息泄露给未授权方。因此,需要有相关的技术来检测和震慑这些不实行为[8-9]。

基于第 1 代 DAS 的研究主要基于 Watermarking 技术、Perturbation 技术和 PITDP 方法等。

(1) Watermarking 技术

Watermarking 技术[10]的主要特点是将可识别的唯一代码嵌入到分发的拷贝中,从而达到可以追踪的目的。但是 Watermarking 技术有 2 个不足之处:1)需要对源数据进行一定的修改(嵌入唯一识别代码)。2)如果数据接收方恶意作为的话,Watermarks 在技术上比较容易被篡改甚至删除[8-9]。

(2) Perturbation 技术

Perturbation 技术是一种非常实用的技术,可用于将敏感信息去敏感化。该技术有很多种实现方式,如 Truncation,Hash,Encryption 和 Tokenization 等。在敏感数据被发送到第三方(Agent)之前,Perturbation 技术可以有效地去敏感化已保护敏感信息不被泄露[11]。

对于一些法规符合性的要求如 PCIDSS(Payment Card Industry Data Security Standard)而言,Perturbation 是非常有效的安全控制手段来缩小符合性评估范围。表 1 总结了各种 Perturbation 技术对 PCIDDSS 审核范围的影响[12]。

表 1　Perturbation 技术与 PCIDSS 审核范围

Perturbation 技术	处理后示例	PCIDSS 范围
Primary Account Number(PAN)	4123 4567 8901 2345	在范围内
Truncated PAN	4123 45XX XXXX 2345	不在范围内
Hashed PAN	2fde4elc67&2d28fc	不在范围内
Encrypted PAN	90f73h3d * 8hh#& HFH&##ED * HD#	在范围内
Tokenized PAN	9483 7266 3928 9819	不在范围内

Perturbation 技术主要存在的问题在于:很多实际的企业应用环境下源数据不能被修改,否则就失去意义。比如,第三方提供员工薪资发放和记账服务,他们需要准确地知道工资数额、账户以及对应的员工信息等[11]。

对于 Tokenization 技术来说,其主要缺点是需要有一个数据库来进行真实数据和其替代值的对应。这种对应增加了对存储和管理的需要,同时不断对新增的数据进行备份以避免数据丢失。另外一个问题是对于跨数据中心的大型应用环境,需要对令牌数据进行持续的同步。这会严重影响实际应用中的性能问题。

(3) PITDP 方法

PITDP(Personal Info Transfer Decision Point)方

法的基本组件和系统流程如图 1 所示[13]。

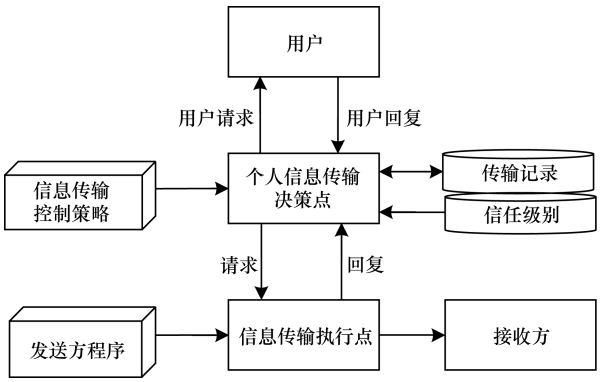

图 1 PITDP 方法组件与流程

PITDP 决定是否所要求的信息传输是安全的。决策基于对传输控制策略,传输日志和接受方信任级别的统筹分析。该方法的缺点是传输控制策略和接受方信任级别非常难以准确设定,导致大比例的 false positive 以及不必要的人为参与(认为决定传输是否安全)。另外,要使发送方和接收方互相不能否认发送或接收,需要额外的控制措施(如签名等),从而增加了过程复杂性和系统开销。

4 基于第 2 代 DAS 的数据泄露检测

Papadimitriou P 等人提出了基于 Agent Guilt 模型[3-4]的数据分配算法。该模型能够大大提高检测泄密者的可能性。此模型定义数据的拥有者为 Distributor,可信第三方(假定)为 Agent。解决的问题是如果 Distributor 的数据被泄露给不可信第三方了,是否能够检测是哪个 Agent 泄露的。

(1) 问题设定

Papadimitriou P 的研究主要基于一种应用场景,Distributor 将数据传输给 Agents 后发现部分数据出现在没有被授权的地方,比如一些网站等。在此种情况下 Distributor 有能力分析和评估是哪个或哪些 Agents 泄露了数据。Panagioti P 等人主要采用了 Agent 过失模型来进行可能性的评估。

该模型假设一个 Distributor 拥有一组有价值的数据 $T=\{t_1,t_2,\cdots\}$。这个 Distributor 希望和一组 $Agents(U_1,U_2,\cdots,U_n)$ 共享其中的一部分数据,同时并期望这些被分享的数据不能被泄露给任何第三方。该模型还提出了 2 种数据分享请求的方法:样本请求和明确请求。

1)样本请求:$Sample\ request\ R_i = Sample(T, mi)$,表示任何 T 中的 mi 的子集可以被传输给 U_i;

2)明确请求:$Explicit\ request\ R_i = Explicit(T, Cond_i)$,$Agent\ U_i$,表示从 *Distributor* 接收所有满足 $Cond_i$ 的 T 数据。

Agent Guilt 模型定义如果一个 AgentU_i 泄露了一组或多组数据给不可信第三方(Target),这个 U_i 就是一个 Guilty。同时 $G_i \mid S$ 表示 U_i 是一个泄露数据 S 的 Guilty。Agent Guilt 模型主要目标是要评估 U_i 是一个泄露数据 S 的可能性($P_r\{G_i \mid S\}$)。

(2) 问题解决模型

为了能够使计算 $Pr\{G_i \mid S\}$ 更简单直接,该模型定义了 2 个前提假设:

1)所有的 T 数据组被泄露的概率(P_t)相同;

2)一个 AgentU_i 泄露一个数据组的决定和其他数据组没有任何联系。

基于上述假设,该模型提出一个 $AgentU_i$ 泄露数据 S 的可能性,可由式(1)计算:

$$P_r\{G_i \mid S\} = 1 \prod_{t\in s\cap Ri}(1-(1-p)/|V_t|) \quad (1)$$

(3) 模拟方法

Panagiotis 等人的工作重心在于研究如何使得一个 Distributor“聪明地”将数据组传输/分配给不同的 Agents 从而增加准确检测是谁泄露了信息的可能性。

基于请求类型和是否加入伪数据,有如图 2 所述 4 种场景。

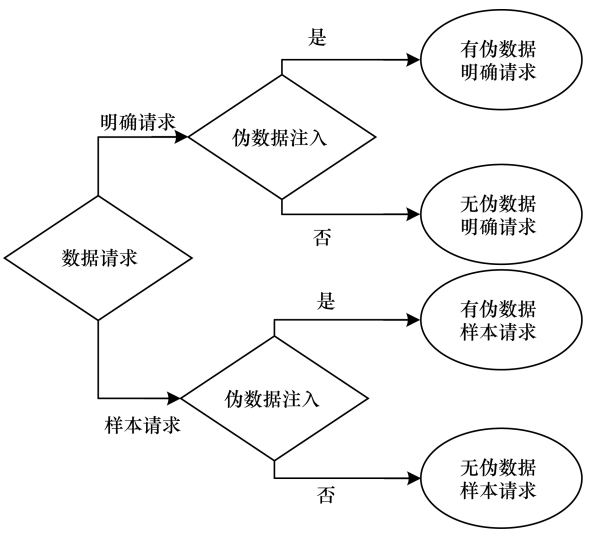

图 2 4 种请求场景

Panagiotis 等人设计了 7 种算法来模拟计算 $P_r\{G_i \mid S\}$:1)明确数据请求分配;2)随机假对象分配;3)优化 Agent 选择;4)样本数据请求分配;5)数据选择(s-random);6)数据选择(s-overlap);7)数据选择(s-max)。

(4) 其他相关研究

其他的一些文献也讨论了 Guilt 检测方面的问题。文献[14-15]研究了 Data Provenance 的问题,他们提出跟踪被泄漏数据组的线性关系是提高检测 Guilty Agents 可能性的关键因素。文献[16]提出一些更有针对性的解决方案,比如对数据仓库的线性跟踪。文献[17]对企业实际应用中的数据泄漏与保护进行了研究,并提出了 Data Watcher 和 Leakage Detector 概念模型。如果一个员工试图在公司没有授权的情况下获取一些敏感信息,Data Watcher 模型

用来检测数据泄漏者。在员工将数据泄漏给外部机构的情况下,Leakage Detector 模型用来检测 Guilt 的第三方[18]。

5　算法优化及实验

本文优化了模拟算法 1) ~ 模拟算法 3),并对 Agent Model 进行实验仿真。算法的总体设计思路如图 3 所示。实验环境为:Windows 7 Professional 操作系统,Java 编程语言和 Eclipse IDE 环境。

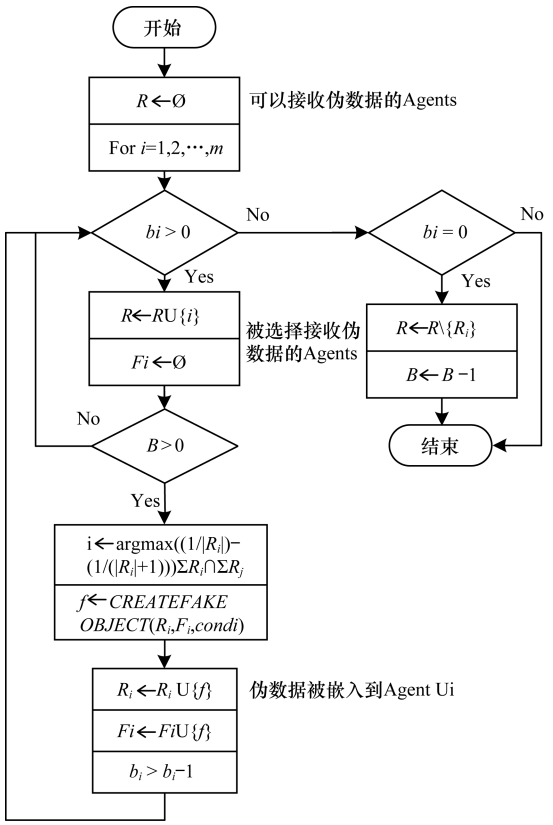

图 3　优化算法流程

对于 Agent 选择有 2 种可能的方式:随机模式和优化模式,表示如下:

(1)随机模式:

$$i_{\text{random}} = SELECTAGENT(R, R_1, R_2, \cdots, R_m)$$

(2)优化模式:

$$i = \text{argmax}((1/|R_i|) - (1/(|R_i|+1)))\Sigma R_i \cap \Sigma R_j$$

对算法的主要优化内容如下:

(1)将原有随机模式 $i_{\text{random}} = SELECTAGENT(R, R_1, R_2, \cdots, R_m)$ 调整为优化模式 $i = \text{argmax}((1/|R_i|) (1/(|R_i|+1)))\Sigma R_i \cap \Sigma R_j$,主要目标是为了提高算法的有效性和速度。

(2)增加了对总体伪数据数量的判断逻辑,即对 $B \leqslant 0$ 的分析以保证算法逻辑上的正确性和稳定性。

(3)将逻辑分析部分 $b_i = 0$ 的位置从算法的尾部转移到算法前部分,紧连 $b_i > 0$ 的逻辑分析部分,以增加算法的连贯性和减少程序的开销。

优化后算法的主要步骤如下:(1)设定/计算伪数据总体数量;(2)如果伪数据数量大于 0,则执行之后步骤,否则重置 Agent 的值;(3)选择最优的 Agent;(4)构造伪数据记录;(5)将伪数据记录添加到 Agent;(6)从总伪数据组中递减伪数据记录。

根据式(1),可以计算 $\Delta(i,j)$ 来推测 AgentU_i 比其他 AgentU_j 可能泄露数据的可能性。$\Delta(i,j)$ 表示如式(2)所示,Δ 值越大,越容易确定 U_i 是 Guilty Agent(泄漏了数据)。

$$\Delta(i,j) = P_r\{G_i \mid R_i\} - P_r\{G_j \mid R_i\}$$
$$i,j = 1,2,\cdots,n \qquad (2)$$

同时可以计算平均 Δ(AverageΔ)和最小 Δ(MinΔ),如式(3)所示。

$$\text{Average}\Delta = \Sigma\Delta(i,j)/(n(n-1))$$
$$i,j = 1,2,\cdots,n \qquad (3)$$
$$\text{Min}\Delta = \min\Delta(i,j), i,j = 1,2,\cdots,n \qquad (4)$$

其中,AverageΔ 代表了一个 Agent 被检测为 Guilty Agent 的概率,如果平均 $\Delta = 0.38$,则表示 U_i 是 Guilty Agent 的概率要比 U_i 不是 Guilty Agent 的概率大 0.38。MinΔ 代表了一个 AgentU_i 比另一个 AgentU_j 更有可能是 Guilty Agent 的概率。比如最小 $\Delta = 0.52$,则表示 U_i 是 Guilty Agent 的概率比其他任何 U_j 是 Guilty Agent 的概率高 0.52。

图 4 显示了 AverageΔ 和 MinΔ 模拟数据。模拟试验结果显示添加伪数据能显著提高成功检测 Guilty Agent 的可能性。

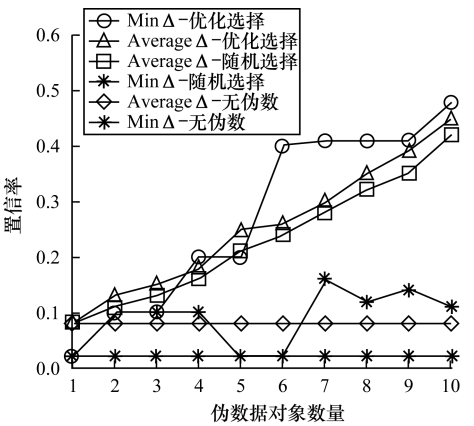

图 4　AverageΔ 和 MinΔ 的模拟数据

较本文提到第 1 代 DAS 和第 2 代 DAS 技术而言,经过优化的算法在添加伪数据后能显著提高成功检测 Guilty Agent 的可能性。但目前 Agent 过失模型仍有以下不足:

(1)现阶段的过失模型设计的不足使得 Agents 可能一起合作来识别哪些是真实的数据,哪些是伪数据。一个 Distributor 可能需要限制传输给每个 Agent 伪数据的数量,以防止引起 Agent 的怀疑或破坏 Agent 的正常操作。因此,伪数据必须被非常小心的构造从而防止 Agents 能够识别它们。

(2)另外,该模型的局限还在于分配策略只适用于事先已知一定数量的 Agents 的情况,不适用于动态数据分配的情况,特别是在线数据分配的情况。

6 未来研究方向

未来针对过失模型将做以下研究:

(1) 改进针对伪数据的设计

添加伪数据进行数据泄漏检测的一个前提是要对用户透明,既不能干扰用户的正常数据处理流程,也不能让用户感知所得业务数据被加入了伪数据。下一步的研究需要关注如何构造一种简单有效和对用户透明的函数来产生伪数据,包括使用加密技术等,使得 Agent 既不能从真数据中识别伪数据,也不会影响 Agent 的正常操作。

(2) 分配策略的扩展和优化

现阶段研究的局限在于分配策略只适用于事先已知一定数量的 Agents 的情况,不适用用于在线数据分配的情况,特别是应用在云计算与 BYOD (Bring Your Own Device)相结合的环境中。扩展分配策略到在线处理的情况非常有实际应用价值。下一步的研究方向需要关注如何解决将数据分配给大量不确定用户的情况。

(3) 非结构化数据保护

当前大部分数据泄漏保护(DLP)解决方法不能够有效解决非结构化数据的保护如 CAD 和 JPG 文件等。如何将过失模型用户保护非结构化数据是未来的一个研究重点[19]。

7 结束语

本文介绍了用于解决数据泄漏检测问题的 DAS 方法,包括 Agent 过失模型、Perturbation 技术、PITDP 过程方法等,阐述了这些方法的特点与不足。分析结果显示 Agent Guilt 模型算法比较适合于解决数据泄漏检测问题,与数据分配策略技术进行结合有较好的应用前景。针对目前数据分配策略技术还不成熟的现状,提出了数据分配优化算法,并进行了相关实验模拟,为今后进行进一步研究和应用节省了时间。本文在分析大量文献和应用环境的基础上,还提出了未来可能的研究方向,包括伪数据设计改进、分配策略的扩展和优化和非结构化数据保护 3 个方面。

参考文献

[1] Lawton G. New Technology Prevents Data Leakage[J]. Computer,2008,41(9):14-17.

[2] 冯 梅,蒋鲁宁. DLP 产品的作为与不作为[J]. 中国信息安全,2012,(2):84-86.

[3] Papadimitriou P, Garcia-Molina H. A Model for Data Leakage Detection [C]//Proceedings of the 25th International Conference on Data Engineering, March 29-April 2, 2009, Shanghai, China. Washington D. C., USA:IEEE Press,2009:1307-1310.

[4] Papadimitriou P, Garcia-Molina H. Data Leakage Detection[J]. IEEE Transactions on Knowledge and Data Engineering,2011,23(1):51-63.

[5] Koneru A,Rao G S N,Rao J V. Data Leakage Detection Using Encrypted Fake Objects[J]. International Journal of P2P Network Trends and Technology,2013,3(2):104-110.

[6] Shobana V, Shanmugasundaram M. Data Leak Data Detection Using Cloud Computing[J]. International Journal of Emerging Technology and Advanced Engineering,2013,3(Special Issue 1):111-115.

[7] Ansari Z S, Jagtap A M, Raut S S. Data Leakage Detection and E-mail Filtering[J]. International Journal of Innovative Research in Computer and Communication Engineering,2013,1(3):565-567.

[8] Czerwinski S, Fromm R, Hodes T. Digital Music Distribution and Audio Watermarking [EB/OL]. [2014-03-14]. http://www.scientificcommons.org/43025658.

[9] Agrawal R, Kiernan J. Watermarking Relational Databases[C]//Proceedings of the 28th International Conference on Very Large Data Bases, August 20-23, 2002, Hong Kong, China. [S. l.]: VLDB Endowment Inc.,2002:155-156.

[10] 赵 耀. 基于小波变换的抵抗几何攻击的鲁棒视频水印[J]. 中国科学 E 辑:信息科学,2006,36(2):137-152.

[11] Sweeney L. Achieving k-anonymity Privacy Protection Using Generalization and Suppression[J]. International Journal on Uncertainty, Fuzziness and Knowledge-based Systems,2002,10(5):571-588.

[12] Conway W. How Enterprises Can Use Tokenization to Reduce Risk and Minimize the Cost of PCI DSS Compliance [EB/OL]. [2014-03-14]. http://blog.403labs.com.

[13] Choi D, Jin S, Yoon H. A Method for Preventing the Leakage of the Personal Information on the Internet[C]//Proceedings of ICA0T'06. Washington D. C., USA:IEEE Press,2006:20-22.

[14] Buneman P, Khanna S, Tan W C. Why and Where: A Characterization of Data Provenance[C]//Proceedings of the 8th International Conference on Database Theory, January 4-6, 2001, London, UK. Berlin, Germany: Springer,2001:316-330.

[15] Buneman P, Tan W C. Provenance in Databases[C]//Proceedings of ACM SIGMOD International Conference on Management of Data, June 11-14, 2007, Beijing, China. New York, USA: ACM Press,2007:1171-1173.

[16] Cui Y, Widom J. Lineage Tracing for General Data Warehouse Transformations[J]. The VLDB Journal, 2003,12(1):41-58.

[17] Jagtap N P, Patil S J, Bhavsar A K. Implementation of Data Watcher in Data Leakage Detection System[J]. International Journal of Computer& Technology, 2012, 3(1):318-322.

[18] Capizzi R, Longo A, Venkatakrishnan V N, et al. Preventing Information Leaks Through Shadow Executions[C]//Proceedings of ACSAC'08, Augest 4-6, 2008, Hsinchu, China. Washington D. c., USA: IEEE Press,2008:322-331.

[19] 张红艳. 强化非结构化数据管理深度挖掘企业信息价值[J]. 电子技术与软件工程,2013,(17):261-262.

编辑 金胡考