Data & Knowledge Engineering xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

Secure partial encryption with adversarial functional dependency constraints in the database-as-a-service model

Boxiang Dong[a,*], Hui (Wendy) Wang[b]

[a] Department of Computer Science, Montclair State University, Montclair, NJ 07043, USA
[b] Department of Computer Science, Stevens Institute of Technology, Hoboken, NJ 07030, USA

ARTICLE INFO

Keywords:
Database-as-a-service
Data outsourcing
Security, integrity, and protection
Database management
Management of integrity constraints

ABSTRACT

Cloud computing enables end-users to outsource their dataset and data management needs to a third-party service provider. One of the major security concerns of the outsourcing paradigm is how to protect sensitive information in the outsourced dataset. In some applications, only partial values are considered sensitive. In general, the sensitive information can be protected by encryption. However, data dependency constraints (together with the unencrypted data) in the outsourced data may serve as adversary knowledge and bring security vulnerabilities to the encrypted data. In this paper, we focus on functional dependency (FD), an important type of data dependency constraints, and study the security threats by the adversarial FDs. We design a practical scheme that can defend against the FD attack by encrypting a small amount of non-sensitive data (encryption overhead). We prove that finding the scheme that leads to the optimal encryption overhead is NP-complete, and design efficient heuristic algorithms, under the presence of one or multiple FDs. We design a secure query rewriting scheme that enables the service provider to answer various types of queries on the encrypted data with provable security guarantee. We extend our study to enforce security when there are conditional functional dependencies (CFDs) and data updates. We conduct an extensive set of experiments on two real-world datasets. The experiment results show that our heuristic approach brings small amounts of encryption overhead (at most 1% more than the optimal overhead), and enjoys a 10-time speedup compared with the optimal solution. Besides, our approach can reduce up to 90% of the encryption overhead of state-of-the-art solution.

1. Introduction

Recent years have witnessed increasing need for large amounts of digital information to be collected and studied. To address the big data need, the database-as-a-service (*DAS*) model was introduced [26], facilitated by the evolution of cloud computing. It allows end-users with limited resources to outsource their private datasets to a third-party service provider. Since the service provider may not be fully trusted, the *DAS* model raises a few security issues. One of the issues is how to protect the sensitive information in the outsourced data. A general solution is to encrypt the data so that the service provider cannot access the original data without a proper decryption key.

As shown in [28,29,43], in some data publishing scenarios, only a portion of the dataset may be considered as sensitive. For example, consider a hospital that stores its patients' information, including patient's name (NM), gender (SEX), age (AGE), hospital-wide disease code (DC) and disease (DS), in a dataset. An example of data instance is shown in Fig. 1 (a). Consider the following

* Corresponding author.
E-mail addresses: dongb@montclair.edu (B. Dong), Hui.Wang@stevens.edu (H.W. Wang).

https://doi.org/10.1016/j.datak.2018.01.001
Received 10 June 2016; Received in revised form 5 January 2018; Accepted 15 January 2018
0169-023X/ © 2018 Published by Elsevier B.V.

NM	SEX	AGE	DC	DS
Alice	F	53	CPD5	HIV
Carol	F	30	VPI8	Breast Cancer
Ela	F	24	VPI8	Breast Cancer

(a) The original dataset D

NM	SEX	AGE	DC	DS
Alice	F	53	CPD5	α
Carol	F	30	VPI8	Breast Cancer
Ela	F	24	VPI8	γ

(b) The unsafe encrypted dataset $\bar{D}$

Fig. 1. An example of data instance (FD: DC → DS).

patients' security settings: Alice and Ela require that their disease information cannot be shared with any third-party, while Carol agrees to share her disease information with a third-party. Before outsourcing the dataset to a third-party service provider for data management and analysis, the hospital encrypts its data according to the patients' settings. The enforcement of the aforementioned security setting leads to partial encryption of the DS attribute; Carol's disease value is left as plaintext. The encrypted instance is shown in Fig. 1 (b).

One important issue of partial encryption is that only encrypting the sensitive data may not be sufficient, especially when there exist adversary data dependency constraints [9]. A typical type of data dependency constraints is the *functional dependency* (*FD*). Traditionally, FDs are statements of value constraints between attributes in a relation. Informally, a *FD*: $X \rightarrow Y$ constraint indicates that attribute set X uniquely determines attribute set Y. For instance, a *FD*: *DC* → *DS* exists in the data instances in Fig. 1. Specifically, the FD indicates that all patients with the same disease code (DC) have the same disease (DS). FDs play an important role in database applications. They have been thoroughly researched and applied to improving schema quality through normalization [4,12,44] and to improving data quality in data cleaning [24,6]. When FDs are available to the attacker, it can serve as an important piece of adversary knowledge and bring security vulnerabilities, especially when the outsourced data only needs to be encrypted partially. For example, consider the partially encrypted instance in Fig. 1 (b). Since Ela's and Carol's records have the same *DC* value, the attacker can easily infer Ela's disease from Carol's record if he has the knowledge of FD *F*.

In practice, adversarial FD constraints are easily accessible from common sense or other data sources. Therefore, it is vital to design robust approaches that can defend against the attacks based on adversarial FDs. A straightforward approach is to encrypt all data values in the dataset (possibly at the finest granularity) regardless whether they are involved in any FD. However, as the data owner may only consider a portion of her dataset as sensitive, encrypting all data values may over-protect the data and dramatically reduce the data usability. Indeed, only encrypting a (possibly small) portion of *non-sensitive* data values besides the sensitive values is sufficient to defend against the attack based on adversarial FDs. Our goal is to find the optimal scheme that encrypts the minimal amounts of non-sensitive data while providing strong guarantee against the attack based on adversarial FDs.

1.1. Comparison with existing methods

The problem of using the association between attributes to infer sensitive information has been widely explored in the last two decades. One solution is to block the inference channel by increasing the security level of certain attributes [55,9]. The other is to split the data into multiple fragments to break the sensitive associations between attributes [13,60]. Next, we compare our approach with these two methods.

Inference Control The problem of inference channel via FDs has been investigated in the context of multilevel secure relational database management system (*MDB*) [55,9]. The existing work can be classified into two types: (1) at query time; and (2) at design time. The security mechanisms that enforce protection at query time [9] mainly reject or modify the queries that may incur security violations. Such mechanisms are not suitable for the *DAS* paradigm as the service provider may be compromised and fail to perform the protection when it evaluates the queries. On other hand, the at-design mechanisms (e.g. [55]) detect and eliminate the inference channels via encryption during database design time. However, the encryption is applied at an attribute level (i.e., all values of the attribute are encrypted). Such coarse encryption granularity may bring tremendous encryption overhead. In Example 1.1, we will show that by encrypting a small amounts of non-sensitive data besides the sensitive one can disable the inference based on adversarial FDs with lightweight encryption overhead.

Example 1.1. Consider the dataset D in Fig. 2 (a). Assume it has FD F: $A \rightarrow B$. Consider two security constraint rules S_1: $\Pi_B\sigma_{C=c_1}$ and S_2: $\Pi_B\sigma_{C=c_2}$ on D, which specify that for all records whose value of attribute C is c_1 or c_2, their value of attribute B is sensitive. The encryption scheme $\overline{D}$ that enforces S_1 and S_2 is shown in Fig. 2 (b). Apparently, by the reasoning of F and the (a_1, b_1) pair of Tuple 1001 in $\overline{D}$, the attacker can infer that the cipher value β_1 of Tuple 1, ..., 1000 is indeed mapped to the plaintext value. To fix this, state-of-the-art approach [55] encrypts the A attribute of all the tuples (cells in Fig. 2 (c) that are put in rectangles), which leads to 2000 non-sensitive cells to be encrypted. Our approach (Fig. 2 (d)) only encrypts the A attribute of Tuple 1001, leading to one single non-sensitive data cell to be encrypted. The encryption overhead (i.e., the number of non-sensitive tuples being encrypted) of our approach is only 0.05% of the existing methods [9,55].

Example 1.1 shows that there exists a trade-off between the amounts of encryption overhead and the data owner's computational efforts. The existing methods for the *MDB* model (e.g., [55,9]) do not require the client to traverse the data, with the price of high encryption overhead, while our approach reduces the encryption overhead dramatically but with the owner's efforts to find

Download English Version:

https://daneshyari.com/en/article/6853885

Download Persian Version:

https://daneshyari.com/article/6853885

Daneshyari.com