

Chapter 3.12

Ensuring Correctness, Completeness, and Freshness for Outsourced Tree-Indexed Data

Tran Khanh Dang

National University of Ho Chi Minh City, Vietnam

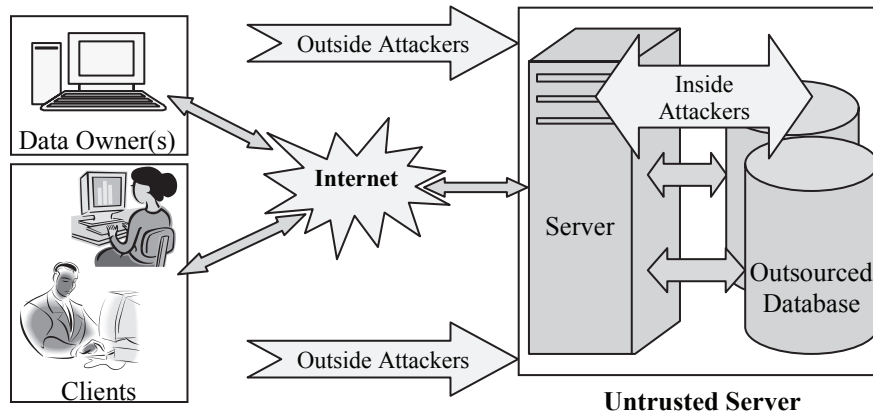
ABSTRACT

In an outsourced database service model, query assurance takes an important role among well-known security issues. To the best of our knowledge, however, none of the existing research work has dealt with ensuring the query assurance for outsourced tree-indexed data. To address this issue, the system must prove authenticity and data integrity, completeness, and freshness guarantees for the result set. These objectives imply that data in the result set is originated from the actual data owner and has not been tampered with; the server did not omit any tuples matching the query conditions; and the result set was generated with respect to the most recent snapshot of the database. In this paper, we propose a vanguard solution to provide query assurance for outsourced tree-indexed data on untrusted servers with high query assurance and at reasonable costs. Experimental results with real datasets confirm the efficiency of our approach and theoretical analyses.

INTRODUCTION

Outsourcing database services is emerging as an important new trend thanks to continued growth of the Internet and advances in the networking technology. Organizations outsource their data management needs to an external service provider, thereby freeing them to concentrate on their core business. In this outsourced database service (ODBS) model, organizations rely on the premises of external service providers, which include hardware, software, and manpower, for the storage and retrieval management of their data, and they operate other business applications via the Internet without having to maintain applications *in-house*. Figure 1 depicts key “actors” in the most general and complicated ODBS model (Mykletun, Narasimha, & Tsudik, 2004),¹ where multiple data owners (say, separate departments of an organization) outsource their data to a certain database server (which may be untrusted) and allow users (may be other departments, partners

Figure 1. The ODBS model and security threats at the server side



of the organization, or even themselves) to access the outsourced data. This service model is a recent and important manifestation of the outsourcing trend of different information technology services. As we can see, however, among issues needing to be addressed in order to make this model reality, security-related issues must be of crucial concern due to the fact that the server may be untrusted, and both data as well as users' queries can now be exposed to the server and hackers/malicious users (corresponding to inside and outside attackers as shown in Figure 1, respectively). This means that, in this ODBS model, apart from secure network communication channels and other necessary security procedures at the user side (Axelrod, 2004), efficient and effective solutions to security threats inside the server are indispensable. We discuss in more detail these server-side security-related issues below.

Security Issues in the ODBS Model

Since a service provider is typically not fully trusted, the ODBS model raises numerous interesting research challenges related to security issues. First of all, because the life-blood of every organization is the information stored in its databases, making outsourced data confidential is therefore one of the foremost challenges in this model. In

addition, privacy-related concerns must also be taken into account due to their important role in real-world applications.² Not less importantly, in order to make the outsourced database service viable and really applicable, the query result must also be proven qualified. This means the system has to provide users with some means to verify the query assurance claims of the service provider. Overall, most crucial security-related research questions in the ODBS model relate to the below issues:

- **Data confidentiality:** Outsiders and the server's operators (database administrator—DBA) cannot see the user's outsourced data contents in any cases (even as the user's queries are performed on the server).
- **User privacy:** Users do not want the server and even the DBA to know about their queries and the results. Ensuring the user privacy is one of the keys to the ODBS model's success.
- **Data privacy:** Users are not allowed to get more information than what they are querying on the server. In many situations, users must pay for what they have got from the server and the data owner does not allow them to get more than what they have paid for, or even users do not want to pay for

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:
www.igi-global.com/chapter/ensuring-correctness-completeness-freshness-outsourced/54510

Related Content

Research on Animation Character Expression Generation Based on Attention Conditioned CycleGAN

Chunling Kong (2024). *Journal of Cases on Information Technology* (pp. 1-22).

www.irma-international.org/article/research-on-animation-character-expression-generation-based-on-attention-conditioned-cyclegan/356409

E

(2007). *Dictionary of Information Science and Technology* (pp. 200-253).

www.irma-international.org/chapter//119566

Usable M-Commerce Systems

John Krogstie (2005). *Encyclopedia of Information Science and Technology, First Edition* (pp. 2944-2948).

www.irma-international.org/chapter/usable-commerce-systems/14723

Leveraging Knowledge Reuse and Systems Agility in the Outsourcing Era

Igor Crk, Dane Sorensen and Amit Mitra (2008). *Journal of Information Technology Research* (pp. 1-20).

www.irma-international.org/article/leveraging-knowledge-reuse-systems-agility/3694

Bridge Health Status Detection Based on Deep Learning: Integrating Attention Mechanism and Hybrid Feature Extraction Technology

Qingping Wang (2025). *Journal of Cases on Information Technology* (pp. 1-24).

www.irma-international.org/article/bridge-health-status-detection-based-on-deep-learning/382564