Secure Multi-Party Data Management

Stefan Böttcher

University of Paderborn, Computer Science, Fürstenallee 11, 33102 Paderborn, Germany
stb@uni-paderborn.de
http://wwwcs.upb.de/cs/boettcher/

1 Introduction

As data and information is stored, combined and accessed almost everywhere, information security and privacy becomes increasingly important to companies that offer access to data. In comparison to network and hardware security that often focus on the protection of network or hardware from attackers from the outside, information security involves protection of certain information from users that have at least some access rights 'inside' an information system. One key challenge in information security and privacy is to meet two contrary goals at the same time, i.e. to allow users to run applications which access the data that they need, but to prevent the same users from being able to infer sensitive information from the results returned by the application.

While cryptography may be used as a supporting technology, the focus of our research at the University of Paderborn is to solve problems of information security and privacy. Our research involves different aspects of information security and privacy, e.g. access control, policies, key management, role management, inference control, detection of privacy leaks, sovereign information sharing and anonymous union computation. Although there are many great contributions to these problems by other groups in the community, within this paper, we focus on the research contributions by our group in Paderborn.

2 Different approaches to view-based XML access control

We have investigated a variety of techniques for access control in XML databases, ranging from proof techniques to query modification. The proof technique-based approach at first computes the *read set* of query, i.e. the information that has to be accessed in order to answer a query [12]. The read set is usually a superset of the answer returned by the query. Given this read set, access control has to check whether or not the read set is a subset of the access right. This is done by a containment test for XPath expressions, that checks query containment between the read set of a user XPath query and an access right, i.e. whether or not the result selected by the read set of a user query is contained in the result selected by the access right for every XML database state that is valid according to a given DTD [10], [11].

Some approaches use predicate to describe that part of an XML database to which access is prohibited instead of to describe to which part of an XML database access is allowed. In such a case, access control has to check whether the intersection of the read set and an XPath expression describing the forbidden part is empty. We have developed intersection tester for XPath expressions that consider additional structural

Dagstuhl Seminar Proceedings 06431
Scalable Data Management in Evolving Networks
http://drops.dagstuhl.de/opus/volltexte/2007/948

constraints of the XML document given by a DTD, [9], and an intersection tester that considers structural constraints given by an XML schema definition [7].

Furthermore, we have developed a query modification based-approach for XML databases, [8], that is based on query rewriting and substitution. The key idea is to express access constraints as security views and to apply XPath queries to these security views, i.e. users can never access any information that is not contained in the security view. By applying queries to security views, we get nested views or nested XPath queries. A second step of this approach is therefore to optimize these nested queries - which we do by rewrite rules. Furthermore, we have extended the query rewriting and optimization approach to XQuery [2].

3 Anti-inference and finding information leaks

Whenever secret company information that can be accessed by multiple users is illegally leaked to a third party, it is crucial for the company to identify the information leak. We especially focus on scenarios where the information is leaked from a person that has an access right to the leaked information. We call this kind of information leakage an attack from the inside - in comparison to attacks from the outside where people who do not have an access illegally right try to access secret information. While access control helps to prevent attacks from the outside, access control is not applicable to our scenario where multiple users have an access right to the secret information.

Instead our problem is related to inference and anti-inference, i.e., the question is which of the users that have an access right to secret information that has been leaked by someone, did actually submit queries and did retrieve answers that are sufficient to infer the leaked information. In other words, given the knowledge which an attacker can infer from his queries and his answers, can he or can’t he infer the leaked secret information.

In the context of XML databases and XPath queries, we have reduced the problem of finding information leaks to the problem of computing an answer to a database query describing the secret from another answer to a user database query. Furthermore, we have investigated some necessary conditions on the structure of XPath queries that have to be met, such that the answer to a query can be inferred from a previously answered query [4], [6].

4 Sovereign information sharing among malicious partners

The goal of sovereign information sharing is to share common information without a trusted third party and uncovering non-common information to the other party. For example, consider two companies that want to know, who are their common customers, because they assume that customers try to play them off. Of course, no company wants to uncover information about a customer that is not a common customer. Even more, the goal is to have *fair* information exchange, i.e. one company shall get the knowledge of a common customer if and only if the other company gets the same knowledge. Previous protocols that have been proposed for that problem always assume an honest but curious behaviour [1].

However our approach [5] solves a much more challenging scenario, because in addition to the fairness requirement, we assume that participants may act malicious, i.e., our protocol shall solve the following additional problems.

1. One participant may fake customer data, e.g. take the yellow pages and claim all the entries are his customers. The protocol shall detect faking and avoid damage to the other participant.
2. One participant may inspect, suppress or invent messages. The protocol shall avoid disadvantages from message inspection by the partner and shall detect all manipulations of messages by the other participant.
3. One partner may stop the protocol at any point in time. The protocol shall avoid disadvantages or damage for the other partner.

We could prove that no fair protocol exists that meets all these requirements. Instead, each protocol requires a minimum amount of risk, i.e. to give some information for which partner may not get the fair information equivalent. However, we could show that it is possible to reduce the damage that one partner would suffer at most by any kind of malicious behaviour (message inspection, message suppression, invention or modification, or invention of faked data) is one information unit, i.e. on customer in our example. Furthermore, there is a trade-off between risk and trust on one side and efficiency on the other side, i.e. the higher the risk and trust the more efficient the data exchange and sovereign information sharing can be implemented.

5 Anonymous union computation

Contrary to the sovereign information sharing problem, in anonymous union computation, the result that shall be computed and shown to all participants is the union of all the shared information. However, the source of each particular piece of information contributed to the union shall remain anonymous in all non-trivial cases[1], and there may be more than two partners contributing to the computed union.

Applications include medical information collection systems that aim to collect and share information about existing diseases (HIV, influenza, etc.) in a group of participants without uncovering which participant suffers from which disease.

Our approach [3] computes the union of multiply encrypted values. Anonymity of the origin of tuples is guaranteed by a data exchange protocol that shuffles encrypted values and distributes them using different paths and by mixing in faked tuples that are later on detected as fakes.

Protocol manipulation can be detected by a decryption-based proof technique, and as for sovereign information sharing with malicious partners, if a participant act malicious or stops the protocol at any point in time, the damage for the other partners can be reduced to be one information unit only.

[1] A trivial case is that we only have two parties and each party knows that information found in the union which it did not contribute itself must be contributed from the other party.

References

1. Agrawal, R., Evfimievski, A.V., Srikant, R.: Information sharing across private databases. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, San Diego, California, USA. (2003) 86–97
2. Sven Groppe, Stefan Böttcher. Schema-based query optimization for XQuery queries. 9th East-European Conference on Advances in Databases and Information Systems (ADBIS 2005). Tallinn, Estonia, September 2005.
3. Stefan Böttcher, Sebastian Obermeier. Secure Anonymous Union Computation Among Malicious Partners.The Second International Conference on Availability, Reliability and Security (ARES), Vienna, Austria, April 2007.
4. Stefan Böttcher, Rita Steinmetz. Information Disclosure by XPath Queries. 3rd International Workshop on Secure Data Management 2006 (SDM) at VLDB 2006. Seoul, Korea, September 2006.
5. Stefan Böttcher, Sebastian Obermeier. Sovereign Information Sharing Among Malicious Partners. 3rd International Workshop on Secure Data Management 2006 (SDM) at VLDB 2006. Seoul, Korea, September 2006.
6. Stefan Böttcher, Rita Steinmetz. Finding the Leak: A Privacy Audit System for Sensitive XML Databases. Second International Workshop on Privacy Data Management (PDM'06) at ICDE2006 - Atlanta, April 2006.
7. Stefan Böttcher, Rita Steinmetz. Embedding XML Schema Constraints in Search-Based Intersection Tests for XPath Query Optimization. 1st International Workshop on Logical Aspects and Applications of Integrity Constraints (LAAIC). Copenhagen, Denmark, August 2005.
8. Stefan Böttcher, Rita Steinmetz. Adaptive XML Access Control Based on Query Nesting, Modification and Simplification. 11th BTW (GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web). Karlsruhe, Germany, March 2005.
9. S. Böttcher. Testing Intersection of XPath Expressions under DTDs. International Database Engineering & Applications Symposium. Coimbra, Portugal, July 2004.
10. S. Böttcher, R. Steinmetz. A DTD Graph Based XPath Query Subsumption Test. XML Database Symposium (XSym 2003) at VLDB 2003, September, 2003.
11. S. Böttcher, A. Türling. Checking XPath Expressions for Synchronization, Access Control and Reuse of Query Results on Mobile Clients. In Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Database Mechanisms for Mobile Applications. Proceedings of the 2003 Spring Workshop of the Working Group on Mobile Databases and Information Sytems within the German Informatics Society (GI). Karlsruhe, April, 2003.
12. Amelie Marian, Jerome Simeon: Projecting XML Documents, VLDB 2003.