Orchestrating Access Control in Peer Data Management Systems

Christoph Sturm

Department of Informatics
University of Zurich
sturm@ifi.unizh.ch

Abstract. This paper describes an approach to establish access control mechanisms in a peer data management system (PDMS). Based on the research on security in Peer-to-Peer networks, we develop a decentralized access control component for PDMS. For this purpose, information resident in local access control components in the peers is used, and mappings between the peer access control policies are established. A client side access control mechanism enforces access rights in the whole PDMS.

1 Introduction

The use of Peer-to-Peer (P2P) networks introduced new challenges for database management systems. These new databases, called Peer Data Management Systems (PDMS), are defined as follows: A PDMS is a Peer-to-Peer network where every peer has its own database management system and intends to share parts of its database with other peers. To share data, the peers need to establish data mappings between their schemas [10, 13, 9, 16]. Query processing is done by traversing these mappings, rewriting the queries, executing them on the peers and gathering the results at the peer that requested data. Because every peer can leave and join the network at its will, there is no permanent global schema. In fact, a kind of global schema is only established during query execution. Many security problems arise in such an environment, and, therefore, many projects deal with security in P2P networks (a good overview is given in [18]). But to our knowledge there is no approach which considers the creation of an access control mechanism in the special case of PDMS.

To illustrate the need for such an access control component, let us consider the following scenario. Several databases store health information about a person A. Database db_x holds data on the doctor, database db_y has details of health insurance, and database db_z is at a hospital which provided medical treatment for A. In an emergency, this information should be combined to give A the best possible medical care. A PDMS might provide such a service, because the mappings between those databases can be established fast and remain flexible. Let us assume that we have established all mappings between relevant data sources. Without access control in the newly established PDMS, every user can see all data. This is definitely not appropriate. We need a fine grained PDMS access control, similar to what is common in relational databases.

2 The Research Question

The following questions are being addressed in my research:

- How can a fine grained access control for PDMSs be established?
- How can the PDMS access control component be distributed in the network?
- How can the information inside local access control components be used for the PDMS?
- What is the relationship between local and global access control rules?
- How can one prevent PDMS access control bypassing?

The intention is to build on existing research work in P2P security mechanisms. With these mechanisms we can guarantee a secure authentication and communication inside PDMSs.

The next step is to establish the PDMS access control component. Here, the solution can be based on the security mechanisms of loosely coupled federated databases. But the mechanism proposed by Heimbigner and McLeod [11] seems to be insufficient for a PDMS. It is only based on peers and not on users, and it depends on access lists. Each database item which needs to be protected has its own access list where all authorized peers are recorded. This approach will also work in a PDMS; however, it is very costly and scales poorly. Therefore, we need a new approach that matches the requirements of a PDMS.

3 Significant Issues in the Field of Research

The basic problem in the research field is the missing central authority. Access control in data integration systems is a well studied research topic, see for example [2, 20]. But up till now, every control mechanism relies on such a central authority. Furthermore, the high dynamics of PDMSs is a major problem. Peers leave and join the system at arbitrary times. Moreover, peers normally belong to different organizations and establish cooperations for a short period of time. Therefore, trust between peers is very important. Finally, if obstacles to join the PDMS are too high, the flexibility that P2P systems are famous for is going to disappear.

4 State of the Art

Before we can think of access control in PDMSs, some basic requirements need to be fulfilled. These are secure authentication and communication, and client based access control. A lot of research has been done to enable these services in a PDMS, as detailed below.

4.1 Secure Authentication and Communication

The basis of every security consideration is secure authentication of users and peers. Every user must own a single, specific and distinct ID. But without a central authority there is no instance that gives the guarantee that a newly generated ID is distinct. As shown by Berket et al. [3], this problem can be solved via a public key infrastructure (PKI) and certificates. A central certification authority guarantees distinct user IDs. Nevertheless, one problem remains. Normally, every peer should be able to have only one identity. Otherwise, the P2P network is vulnerable to "Sybil" attacks [8]. This can be prevented through the assignment of peer certificates from a certification authority, guaranteeing PDMS-wide distinct peer IDs. Peer hardware information, for example the MAC address of the network card in a certificate, can preclude multiple peer identities. Such a solution gives us the possibility, besides secure authentication, to establish secure communication between the peers and between users.

4.2 Trust Management

There are several proposals for trust management systems in P2P networks, e.g. [1] and [19]. Without trust between the participants, there will be no collaboration, and, without this, no working network. Besides, trust management systems that are based on peer reputation are able to detect malicious peers and exclude them from the network, if the authentication problem described in Sect. 4.1 is solved. Trust information can further be used to optimize the selection of peers and therefore network performance. These trust management systems must be immune to attacks of malicious peers.

4.3 Client Based Access Control

In a P2P network, one needs a new access control approach. As stated by Miklau and Suciu [14], trust domains in PDMSs differ from domains in traditional client server databases. The data owner trusts and controls only the data, whereas the execution of the query, and the query itself, may be beyond the control of the data provider. Hence, a PDMS peer is forced to give away its raw data to enable other peers to execute their queries and establish their mappings. When a peer gives away its data it also gives away the control over this data. It cannot restrict access to the data given away or protect it from changes. Even worse, one cannot prove where the data originates from.

One solution to solve this problem is to perform access control on the client and not on the data provider side. This can be done via trusted software on the client. The software enforces access and distribution restrictions of the data provider, and the client can only operate on the data through this trusted software. The data provider therefore encrypts the content and gives the encrypted data, together with the information needed to decrypt the data, to the data requester. Only the trustful software from the requestor is able to decrypt and display the data.

Another approach to enforce client based access control can be a solution based on the encryption and distribution of keys [14, 4, 5]. In the current solution, we opt for the trusted software approach, because we need this technology to enable the distribution of access control anyway (see Sect. 5.2). Besides, trusted software can make it much harder for a malicious peer to gain raw unencrypted data from the system.

4.4 Access Control in Peer-to-Peer Networks

Recent papers consider access control in P2P networks. Sandhu and Zhang [17] present a general framework for the use of trusted computing technology for access control in P2P networks. The work of Berket et al. [3] presents an access control mechanism for a P2P network. Secure communication and authentication of peers is provided via PKI. In addition, every peer establishes its own rights management policy for its own data and enforces this policy through a special authorization manager. A related approach from Crispo and others [6] uses more flexible policies. These two approaches do not address the problem of client based access control that causes problems as stated before (see Sect. 4.3). None of these approaches considers existing access control components and information residing on the peers, which is essential for a PDMS.

5 Problem Solution

As a basis, every peer and every PDMS user requires a certificate from a central certification authority. This enables secure authentication and communication between peers and PDMS users. Furthermore, we postulate that every peer has a fully fledged DBMS with an access control component that offers fine grained access control including roles, users, access rights and grant rights. Access control information contained in these components can be considered as a kind of meta data that can be connected through mappings to other peers. Of course, access control data is special and so are the mappings. What we need is a general data exchange format for access control information. Afterwards, we need to design a mapping language and appropriate transformation rules that can map/transform this information between different peers. This situation is illustrated in Fig. 1.

The creation of authorization mappings is a collaborative task, because two peers, between whom an access control mapping should be established, need to coordinate their access control mechanisms. That might cause changes to the local access control rules of both peers.

To make a valid and secure authorization mapping (AM) contract, the mapping should be encrypted and signed by both peers. The encryption ensures that only the two contractual partners can see the mapping rules and the signatures guarantee that changes to the mappings are done collaboratively. It is clear that the AMs and the certificates increase the effort to join a PDMS, but we believe it is worth doing, due to the additional security control achieved in this way.

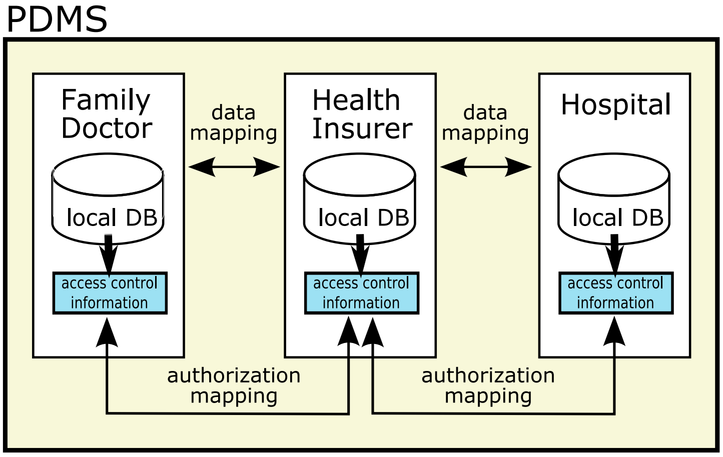

Fig. 1. PDMS with authorization mappings

5.1 Two Level Access Control

A basic principle of our approach is that there are two levels of access control inside a PDMS. On the one hand, we have access control inside the individual peer. On the other hand, we have authorization mappings between single peers. These mappings are established in such a way that users have appropriate rights on the relevant databases. To make things easier, especially to avoid the need to map each individual user, it may be useful to map roles or groups of users. It is important to note that the individual peer has full control over its local user management component. So it can always change local access rights, which ensures peer autonomy. In contrast, mappings between the peers can only be composed or changed in cooperation. An exception is the dissolution of an AM contract, which can be done without the other partner.

5.2 Indirect Mappings

This problem results from the underlying assumption of data distribution and data processing in a PDMS. When we look at mappings from the point of view of authorization, we deserve an “indirect mapping” problem, shown in Fig. 2. A peer which has access to data can grant this access privilege to other peers without asking the originating peer. This is the fundamental “share your data” principle of a PDMS. However, we think that this is not a good solution for access rights. There might be situations where the sharing of access rights is exactly what we want. But for the majority of cases, we need a mechanism which restricts these indirect mappings.

A solution to this problem is the trusted PDMS software mentioned in Sect. 4.3. This must be installed on every contributing peer and is distributed by the central certification authority. The software guarantees several things:

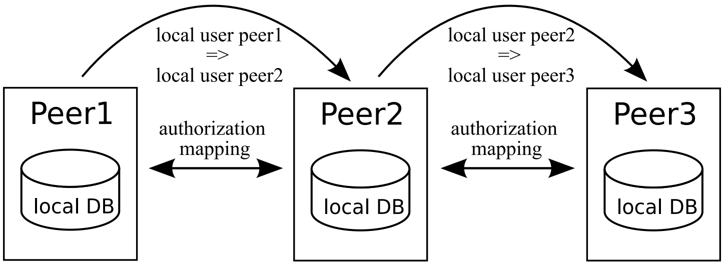

Fig. 2. The indirect mapping problem

- The data provided by a particular peer cannot be rewrapped as someone else's data. That means, the origin of the data is clearly announced.
- Only data exactly addressed to the user is shown to the user.
- Data addressed to a user cannot be redirected to another user.
- In case of indirect AMs, the peer and user ID has to be added to the request to show the authorization path.

That means that the middle peer, *Peer2* from Fig. 2, is not able to redirect queries from *Peer1* with other user rights than the user rights of the originating requestor. *Peer2* can execute queries over data provided by *Peer3* but it is not able to republish the data as its own. This is the default mode for AMs between peers. However, a peer can explicitly assign so called "indirect authorization mapping rights" to another peer. If *Peer3* has granted such a right to *Peer2*, *Peer2* is allowed to give *Peer1* access to data of *Peer3*.

In addition, the trusted PDMS connection software guarantees that the data owner has always full control over his data.

5.3 Diversity of Access Control Methods

Another problem to be considered is the high diversity of access control methods (positive or negative access rights, open or closed world assumption, etc.) residing on different peers. This is a well known problem in federated databases [12, 7]. To solve these inequalities, one can think about conversion peers that convert one access control method into another. This is only possible if indirect AMs are allowed. Without indirect mappings, the conversion between the different methods must be done inside/through the mapping. This approach is more flexible but requires a more powerful mapping language.

5.4 Authorization Mapping Content

The AM consists of both peer IDs and a mapping of the users and roles (in our scenario from the introduction: $user\,f$ of $databasedb_x \Leftrightarrow user\ g$ of $databasedb_y$, or $role\ w$ of $database\,db_x \Leftrightarrow role\ v$ of $database\,db_y$). In addition, it needs to be specified whether indirect mappings to a particular user or role are allowed. Note that only mappings between local users or roles of the partners are allowed. Therefore, an indirect (inherited) AM always needs to be resolved through the granting middle peers. Only a direct AM, which can be derived semi-automatically from the indirect ones, can shorten the detour. To make the mapping secure, the signature of the two peers has to be added. The content of the mappings should be minimal to reduce the complexity and performance impact of access control.

5.5 Decentralized User Management Component

Up till now we only have a mechanism to connect local access control components of peer databases. The missing link is an authority that manages these connections for a number of peers. Here the idea of islands of trust comes into play. Due to the high trust barrier in a PDMS, the best starting point is to establish a small group of peers that highly trust each other. Such a group is called an island of trust. Referring to our scenario, a health insurance company can establish a trust island for hospitals. In the next step, this island can be connected to the other islands established by other health insurers, etc. If every participant in an island trusts the others, they might grant each other further rights, especially indirect AMs.

Highly reliable and trustful peers (e.g., the health insurance company) in the island of trust will hold many indirect AM rights. This makes them responsible for connecting access rights of the island of trust to the rest of the PDMS. In fact, every peer with the right to grant access to data of other peers (indirect AM right) is a kind of substitute for all peers connected through indirect mappings. The most reliable and trustful peers establish therefore something like a decentralized user management component. The more trust there is inside the PDMS, the more centralized access control will be. The drawback is that as access control becomes more and more centralized, it will become increasingly vulnerable to attacks. Therefore, it is a good idea for each peer to grant more than one peer indirect AM rights. Because we treat access rights as data and establish mappings between them, we can use similar methods as we use for data mappings to make them more reliable or scalable.

With our approach we dynamically connect already available access control components to establish a PDMS wide access control system. Due to these mappings, that are as flexible as the data mappings, we are now able to support a range of solutions, from centralized to completely decentralized PDMS access control, exactly arranged to the requirements of each individual peer.

5.6 Correlation Between Data and Rights Mappings

There is a strong correlation between data and authorization mappings. Data mappings are established at peer level. That is, mappings are shared by all users on a peer. Of course, it makes sense to hide the mappings from the user who has no access; nevertheless, every peer user can see all mappings in principle. It is clear that such data mappings should coincide with the corresponding rights to access the mapped data. Therefore, an AM always accompanies a data mapping. As a result, the authorization path can easily be found.

6 How PDMS Access Control Works

When an access request from a connected peer arrives, several things need to be checked. First of all, the sender of the request has to be proven through PKI decryption. Additional IP address and challenge response tests can increase security. Then the rights mappings have to be considered. If there is a direct rights mapping between the two peers, the user sending the request has to be tested via challenge response. Otherwise, we need to check all peers and users of the current authorization path. If everything is all right so far, we can start with real access control. First, the corresponding local user rights are derived from AMs. Afterwards, the request has to be executed using the permissions of the according local user. During the execution, the rights of the local user have to be considered as usual. Afterwards, the result is encrypted with the public key of the requestor and sent to the requestor. If the peer is not the endpoint of the request, we rewrite the query using the data mapping, add the local user/role the rights mapping corresponds to and forward it to connected peers.

7 Future Work

First, we are going to specify the export schema of the access control information. The international standard XACML [15] can be such an export schema. In that case mappings will be established between XACML documents. Next, the design of the mapping and transformation language of the AMs will be developed. Here we may benefit from former research in federated databases [12]. In addition, tools to assist the creation of AMs are essential.

As a proof of concept, we are going to implement our security framework on top of a P2P enhanced version of SIRUP [21]. The implementation will focus on performance and scalability of access control mechanisms, because these are central issues in a PDMS.

References

1. Karl Aberer and Zoran Despotovic. Managing Trust in a Peer-2-Peer Information System. In *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM '01)*, pages 310–317, 2001.
2. Christian Altenschmidt, Joachim Biskup, Ulrich Flegel, and Yücel Karabulut. Secure Mediation: Requirements, Design, and Architecture. *Journal of Computer Security*, 11(3):365–398, 2003.
3. Karlo Berket, Abdelilah Essiari, and Artur Muratas. PKI-Based Security for Peer-to-Peer Information Sharing. In *Proceesings of the Fourth International Conference on Peer-to-Peer Computing (P2P 2004)*, pages 45–52, 2004.

4. Elisa Bertino, Barbara Carminati, Elena Ferrari, Bhavani Thuraisingham, and Amar Gupta. Selective and Authentic Third-Party Distribution of XML Documents. *IEEE Transactions on Knowledge and Data Engineering*, 16(10):1263–1278, 2004.
5. Luc Bouganim, François Dang Ngoc, and Philippe Pucheral. Client-Based Access Control Management for XML documents. In *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB 2004)*, pages 84–95, 2004.
6. Bruno Crispo, Swaminathan Sivasubramanian, Pietro Mazzoleni, and Elisa Bertino. P-Hera: Scalable Fine-grained Access Control for P2P Infrastructures. In *Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS'05)*, pages 585–591, 2005.
7. Sabrina De Capitani di Vimercati and Pierangela Samarati. Authorization specification and enforcement in federated database systems. *Journal of Computer Security*, 5(2):155–188, 1997.
8. John R. Douceur. The Sybil Attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS '01)*, pages 251–260, 2001.
9. Enrico Franconi, Gabriel M. Kuper, Andrei Lopatenko, and Ilya Zaihrayeu. The coDB Robust Peer-to-Peer Database System. In *Proceedings of the Twelfth Italian Symposium on Advanced Database Systems (SEBD 2004)*, pages 382–393, 2004.
10. Alon Y. Halevy, Zachary G. Ives, Dan Suciu, and Igor Tatarinov. Schema Mediation in Peer Data Management Systems. In *Proceedings of the 19th International Conference on Data Engineering (ICDE 2003)*, pages 505–516, 2003.
11. Dennis Heimbigner and Dennis McLeod. A Federated Architecture for Information Management. *ACM Transactions on Information Systems (TOIS)*, 3(3):253–278, 1985.
12. Dirk Jonscher and Klaus R. Dittrich. An Approach for Building Secure Database Federations. In *Proceedings of 20th International Conference on Very Large Databases (VLDB 94)*, pages 24–35, 1994.
13. Anastasios Kementsietsidis, Marcelo Arenas, and Renée J. Miller. Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 325–336, 2003.
14. Gerome Miklau and Dan Suciu. Controlling Access to Published Data Using Cryptography. In *Proceedings of 29th International Conference on Very Large Databases (VLDB 2003)*, pages 898–909, 2003.
15. Tim Moses. eXtensible Access Control Markup Language (XACML) Version 2.0. OASIS Standard, February 2005.
16. Wee Siong Ng, Beng Chin Ooi, Kian-Lee Tan, and Aoying Zhou. PeerDB: A P2P-based System for Distributed Data Sharing. In *Proceedings of the 19th International Conference on Data Engineering (ICDE 2003)*, pages 633–644, 2003.
17. Ravi Sandhu and Xinwen Zhang. Peer-to-Peer Access Control Architecture Using Trusted Computing Technology. In *Proceedings of the Tenth ACM Symposium on Access Control Models and Technologies (SACMAT '05)*, pages 147–158, 2005.
18. Dan S. Wallach. A Survey of Peer-to-Peer Security Issues. In *Software Security – Theories and Systems, Mext-NSF-JSPS International Symposium (ISSS 2002)*, pages 42–57, 2002.
19. Li Xiong and Ling Liu. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.

20. Jacqueline Yang, Duminda Wijesekera, and Sushil Jajodia. Subject Switching Algorithms for Access Control in Federated Databases. In *Proceedings of the 15th Annual Working Conference on Database and Application Security (DBSec '01)*, pages 61–74, 2002.
21. Patrick Ziegler and Klaus R. Dittrich. User-Specific Semantic Integration of Heterogeneous Data: The SIRUP Approach. In *First International IFIP Conference on Semantics of a Networked World (ICSNW 2004)*, pages 44–64, 2004.