Protecting a Federated Database Infrastructure Against Denial-of-Service Attacks

Arne Ansper[1,2], Ahto Buldas[1,2], Margus Freudenthal[1], Jan Willemson[1]

[1] Cybernetica AS, Mäealuse 2/1, Tallinn, Estonia
[2] ELIKO Competence Centre in Electronics-, Info- and Communication Technologies, Mäealuse 2/1, Tallinn, Estonia*

Abstract. The need for combining various heterogeneous data sources into a uniformly accessible infrastructure has given rise to the development of federated database systems. Security aspects of such systems have been well-studied, but they have mostly concentrated on privacy and access control issues. In this paper, we take a closer look at the availability problems caused by the network failures, Denial-of-Service attacks, etc. We take the X-Road infrastructure developed in Estonia as the basis of our studies and propose several methods to improve its resilience. We discuss the usage of alternative communication channels, replication of critical databases and replacing the present critical central services with more flexible alternatives.

Keywords: Federated database systems, X-Road, service availability, Denial-of-Service attacks

1 Introduction

There is a growing trend to make governmental services available through the Internet and to interconnect electronic governmental databases and registers. In Estonia, the key factors for such a trend have been the large-scale use of ID-cards (about 1.2 million active cards) and the development of the X-Road infrastructure which acts as a unified access layer to the governmental registers providing state authorities and citizens with efficient lawful access to governmental information. Today, more than 600 state registers are available through the X-Road infrastructure.

The main challenge to overcome during the X-Road development was achieving a unified access mechanism for a diverse ecosystem of separate databases that had been developing since early 1990s. Building one large data warehouse to host all the state databases was not an option from neither the organizational, nor the privacy point of view. Rather it was decided to build an overlay network,

* This research has been supported by European Union through European Regional Development Fund under ELIKO Competence Center (EU30017) and EXCS Center of Excellence in Computer Science.

keeping the amount of changes needed for the existing autonomous database components to minimum.

As such, X-Road can be viewed as an instance of a *federated database system*, a concept known from mid-1980s [7, 13]. By mid-1990s, several of such systems had already been developed and deployed [14, 15]. Originally, a lot of attention was paid to interoperability issues during the X-Road development. In the first versions, specific adapter servers were used to achieve this [2]. However, once the adapter layer was established, the X-Road development focus has mainly been on confidentiality, integrity and access control, central security issues of the federated database management systems in general [5, 4].

Surprisingly, availability issues of the federated databases have not received equally extensive treatment in research community. Some results have been reported on query optimization [6, 11] and dynamically configurable database networks [3, 1], but none of these approaches provide full resilience against massive malicious online attacks.

So far, the X-Road infrastructure has also been positioned as a *peace-time system*, normal operation of which assumes normal functioning of the Internet. However, considering the growing importance and criticality of the X-Road system for the state to fulfil its role, cyber security threats must be taken into account. The X-Road system should at least guarantee that for some high-priority clients certain services remain accessible even in the crisis situations where the normal functioning of the Internet is affected by cyber attacks. For example, in its current implementation, the X-Road central servers are potentially subjects of Denial-of-Service (DOS) attacks, which would in turn threaten the whole infrastructure. In 2007, Estonia was hit by a massive distributed DOS attack [12]. Luckily, the X-Road infrastructure was not hit directly, but given the current state of cyber warfare, it is unreasonable to hope that we would be equally lucky next time.

In this paper, we discuss the possibility of upgrading the X-Road system to a *war-time system*. To achieve this, we propose several ways to improve the X-Road infrastructure so that in case of Internet failures or inaccessibility of central services, the network of important databases would still be operable using alternative communication channels.

The paper is organized as follows. In Section 2, we give an overview of the X-Road system. In Section 3, we set up the main new requirements to the X-Road infrastructure and analyze the achievability of the requirements. In Section 4, we outline the solutions, and finally Section 5 draws some conclusions.

2 X-Road Overview

By early 2000s the level of computerization in the Estonian state databases had reached both the level of sufficient technical maturity and a certain critical volume so that the need for a unified secure access mechanism was clear. To address that need, the development activities on the modernization of national databases started in the beginning of 2001 [10, 8]. The first version of the developed X-Road infrastructure was launched on December 17th 2001. The number of queries and replies mediated through the infrastructure per year exceeded 240 million in 2011 [9]. Today, already the fifth generation of the X-Road is in operation, and the current paper describes part of the effort to produce the sixth generation.

Detailed technical descriptions of the whole system can be found in [2, 16]; here we will just shortly cover the main aspects needed in the context of the current paper.

On the high level (Fig. 1), the X-Road infrastructure consists of the organizations providing and requesting data, and some central services. In order to communicate over the X-Road, each participating organization must install a dedicated *security server* that acts as a secure access gateway. Security servers are responsible for encrypting the communication between the organizations, and also for cryptographic authentication and digitally signing the information that is exchanged during the communication. To facilitate this, three central services are needed.

- **Certification Authority, CA** is responsible for providing public key certificates for authentication and signing, together with the Online Certificate Status Protocol (OCSP) responder used to distribute certificate validity information.
- **Time-stamping Authority, TSA** is responsible for providing time-stamps to the digital signatures used to validate the X-Road queries and replies.
- **X-Road Center** is responsible for distributing the service directory and service configuration information. Out of these, access to up-to-date configuration is potentially security critical, since it contains certificate chains together with trusted root certificates and the correspondences between organizations and their security servers.

In the first versions of the X-Road, the CA and TSA were also integrated into the X-Road Center, but in the more recently developed versions these services are separated to allow for better integration with external service providers.

In its present state, the X-Road security servers are, to some extent, capable of communicating with each other even if all the central servers are inaccessible. The OCSP responses and configuration information have a pre-determined life-time, which is currently set to 6 hours. If no updates are possible within this period, the subsequent communication is considered insecure and terminated. However, using potentially insecure communication can under many circumstances still be preferable to having no communication at all. This is one of the main improvement goals of the current paper.

3 Requirements

Our goal is to improve the X-Road infrastructure so that it would become a proper *war-time system*, i.e. it has to stand massive cyber-attacks (e.g. DOS attacks) from the Internet. The main requirement is that the X-Road servers must be able to securely communicate and guarantee the availability of *critical services* even if

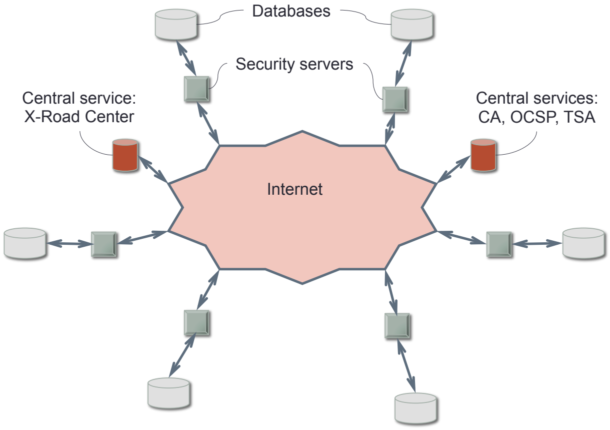

Fig. 1. The peace-time infrastructure of X-Road.

- usual communication through the Internet is impossible or considerably distracted;
- central services (CA, OCSP, TSA) are inaccessible;
- (optionally) the X-Road Center is inaccessible and does not provide configuration information.

This means that the clients must have a better and service-based control over the configuration and evidence collection functions. This enables mediating the services where in certain cases (in crisis situations) the service availability is much more important than the ability of creating evidence, and it is safe to assume that the configuration has not been changed. For example, the 112 (911) services may need critical information from state registers to save human lives. Compared to such important goals, proper collection of evidence for proving the digital signatures is of secondary importance.

This means that in case of cyber-attacks or any other large-scale Internet failure, the system must be able (in time) to switch over to alternative channels (private lines, radio links, etc.) that keep the critical services alive (Fig. 2). Alternative channels are often slower, more expensive, or have some other restrictions, so their required capacity must be carefully determined.

In principle, the choice between different channels can be implemented in the network layer, but this may be awkward and hard to manage, because IP-packets originated from services of different criticality are indistinguishable. Instead, the security servers may support alternative channels on the application level and be able to automatically switch over in case of the main channel malfunction. Priority tags could be assigned to services and customers in order to guarantee a reasonable use of the alternative channels.

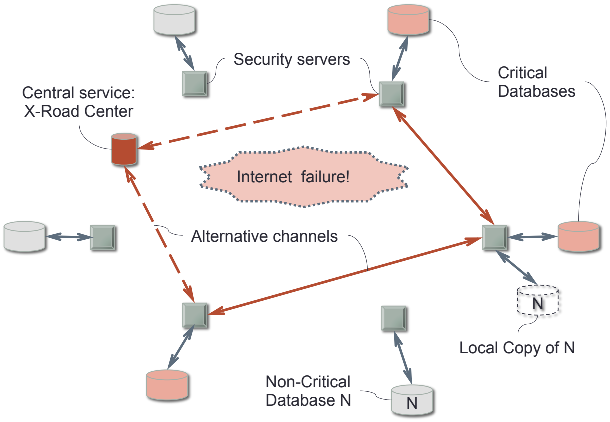

Fig. 2. The war-time infrastructure of X-Road.

Today, the X-Road system supports multiple central servers, which guarantees normal work of the system during a limited period of time, even if one of the central servers of X-Road (maintained by the X-Road Center) is inaccessible. However, the continuous work of the system is not guaranteed after a destruction or a long-term downtime of the X-Road Center. There has to be a possibility of using secondary central servers that keep fresh copies of all the information held in the X-Road Center, including the configuration of the whole system.

It should also be possible to improve the availability of the services when the availability requirements set by the client exceed the availability level guaranteed by the service provider. If it is unreasonable to increase the availability level of the service provider, there is a need of using local copies of the serving database. Local copies of databases are already in use in the Estonian Police and Border Guard Board and in some other institutions, but these are *ad hoc* solutions without any unified framework. It would be desirable to develop a reliable X-Road service for making local copies of databases and for regularly keeping them updated.

In the next Section, we propose solutions to the above-mentioned availability problems.

4 Solutions

4.1 Center-Independent Work

In all the previous X-Road generations, Domain Name System Security Extensions (DNSSEC) directory service has been used to propagate the configuration of X-Road servers, and this solution has worked well this far. However, the use of DNSSEC becomes an obstacle if we require independence from the central services. If the central servers are unreachable and the buffering name servers of the security servers cannot update their data, eventually the names become non-resolving and it will become impossible to check the status of public-key certificates. There is no preventive buffering in DNSSEC and hence if after the service interruption it will be necessary to access one of the security servers, the DNS information would be unusable.

When building the broadcast mechanism for certificate validity information, it is unreasonable to add non-standard extensions to standard protocols. Hence, it is appropriate to abandon DNSSEC altogether and to use an application level solution for the configuration management. It should be ensured, however, that the availability of the system will not decrease.

Hence, to achieve the goal of center-independence, we have to design an application-level high-availability configuration management protocol that ensures an authenticated update of configurations, whereas the configuration update policy should be under control of the owners of security servers.

The requirement of the center-independent operations poses some restrictions to the ways of using the Public Key Infrastructure (PKI) services. Two online services are needed: OCSP for certificate validity information and time-stamping for ensuring the preservation of the evidentiary value of electronic evidence. A recipient of an X-Road message should first check the validity of the corresponding certificate via OCSP and then obtain a timestamp for the signed message.

Later verification of evidence consists of checking that the message is properly signed with a public key listed in the certificate and that the timestamp on the signed message together with a sufficiently fresh OCSP response confirms the validity of the certificate. The signer is responsible for obtaining the OCSP response. Receivers of signed messages determine their policies on how old OCSP responses are acceptable. In principle, in crisis situations where the X-Road center is unreachable, security servers may temporarily abandon all age restrictions to OCSP responses or may choose not to use OCSP at all. Later, if the network operation becomes normal again, OCSP responses can be obtained *post factum*. Note that a positive OCSP response that is dated much later than the signature also confirms the validity of the signature.

The receiver of the signed messages is responsible for the use of the time-stamping service and the receiver is able to define its own acceptance policy for the incoming messages.

In order to support redundant X-Road centers, a protocol must be designed for creating backup copies of the whole X-Road Center (including the Certification Authority) so that backup services can be set up promptly if needed. The copies of the keys used to sign the configuration must be held in backup devices (clones of the original Hardware Security Module (HSM)). This would enable to immediately switch over to one of the secondary X-Road centers if necessary.

4.2 Alternative Channels

A flexible use of redundant channels would extend the X-Road's scope of usability even more, because the critical services would stay accessible even in case of Internet malfunction.

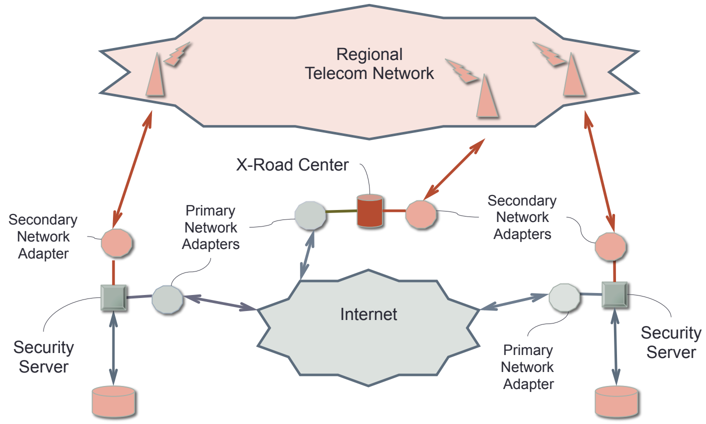

Fig. 3. An implementation of alternative channels through local telecom networks.

Alternative communication networks can be used at several different layers. At the network layer, it is possible to add routers between every security server and the Internet, so that the messages can be routed through alternative network devices such as radio modems. These modems should communicate by using a common routing protocol and should ideally be maintained by a single organizational entity. The main drawback of the network-layer solutions is that since the communication between the security servers is encrypted, it would be nearly impossible for ordinary network routers to selectively direct certain services to alternative channels.

A more flexible solution is making the security servers able to differentiate between different networks. Already the presently deployed security servers have a built in mechanism for using redundant security servers. This mechanism can be extended with the support for alternative networks. In the case of the main network (Internet) malfunction and if certain services are allowed to use alternative channels, the security servers are accessed via an alternative network (using different IP addresses). The credentials for using alternative networks can be defined at the service layer, enabling thereby differentiation between critical services (that must be accessible in crisis situations) and ordinary services that, for economical reasons, will not be offered through lower-bandwidth and/or more expensive channels.

The solution we propose is the following (Fig. 3). The security servers of the databases that have been declared to be critical must be equipped with secondary network adapters that are connected to the network of a regional telecom operator. We assume that a regional telecom network can be protected against external DOS attacks by separating it from the Internet. A separate agreement for the local telecom operator is required so that the regional part of the network stays accessible even if the rest of the Internet is cut off from it. Usually, such connections are priced per transmitted bytes plus some constant monthly fee that depends on the channel capacity. From customers' point of view, such connections look just like ordinary Internet connections.

Service level applications (Fig. 4) should distinguish between the critical queries/responses and the ordinary ones. There should also be a service quality measurement system and a smart decision-making mechanism that is able to send messages through the secondary network adapter if needed. Note that the views of different security servers may lead to different decisions about the emergency situations and hence, one server may send a message through alternative channels while for the rest of the servers, the Internet seems working properly. This means that all critical servers must always be ready to receive messages through alternative channels, even if to the best of their knowledge the Internet is working properly.

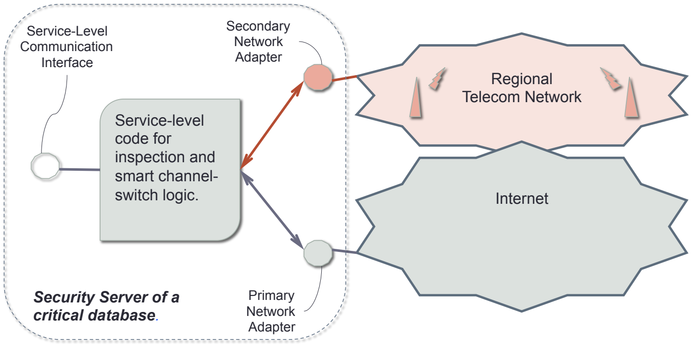

Fig. 4. A security server that supports alternative channels.

Though the alternative channels are mostly (in peace time) not used, one has to be sure that they are operable when needed. This means that alternative connections cannot be entirely passive even if they are not directly needed. Constant pinging is necessary for monitoring their health (see Section 4.4 for a more detailed discussion).

The X-Road Center can also be considered as a critical on-line database and may also be equipped with alternative channels support. This would enable to overcome the need of solving the center-independent work problem.

4.3 Replication of Databases

Replication of databases is actually a controversial issue for the X-Road. On one hand, it is required to provide data availability in case of DOS attacks or other Internet malfunction scenarios. On the other hand, it contradicts the basic principle of X-Road that each data source should be kept only in one place to guarantee its freshness and have a clearly defined party responsible for its quality. For this reason, database replication has not been supported nor encouraged in the previous generations of X-Road.

Technically, however, this is possible already today. If a critical database B needs some view of database A to maintain its operations in the time of crisis, its developers can obtain this view just by performing regular X-Road queries. In case the database A is large, this may be a tedious task. It may also be the case that the entire database A is not even needed for B, but it is easier for the developers not to care about this and download entire A, putting unnecessary load to A. Also, the task of keeping the local copy of (the local view of) A up-to-date is a non-trivial task.

Hence, the least we can do to improve the situation is to define a set of templates and best practices for the developers to follow when replicating the databases. The good aspect of such a solution is that it requires minimal intervention into the existing systems. However, it is very difficult to guarantee that the developers would actually follow these best practices, so potentially this solution would not improve the current state of affairs too much.

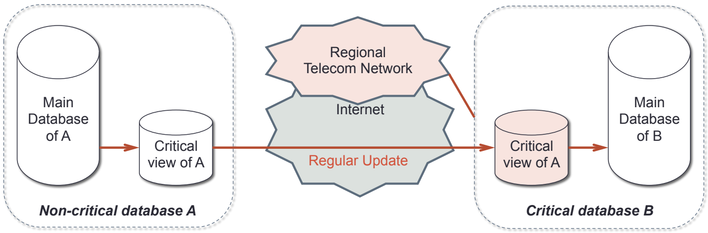

Fig. 5. A view of a non-critical database A saved at a critical database B.

As a more advanced solution, a special component can be added to the security servers of the clients. This component directly implements the best practices of creating local copies of the client views of the databases (Fig. 5). This component is configured with the specifics of the database and the anticipated use patterns. Compared to the first solution, more work must be done with the X-Road system, but the result would be a much better unified and easy-to-use replication mechanism.

4.4 Channel Switching Logic

For reasonable use of alternative channels, every security server must be able to automatically determine when exactly is it necessary to use alternative connections instead of the primary ones. It would be hard for the whole infrastructure to synchronously switch to alternative connections and hence, such decisions must be made separately for every point-to-point connection. Note that the databases may be subjects to selective DOS attacks, so that server A does not necessarily know if another server B is under attack or not. It may also be the case that neither A nor B is under a direct attack, but there is just no proper Internet connection between these two servers. For example, server B might be down because of regular maintenance. In this case, it would be unreasonable for A to try to connect with B via an alternative channel. So, A must be able to distinguish between the situations where B is under attack, and where B is just down for some temporary technical reasons.

We will assume that A will recognize if there is a DOS attack against A itself and is able to shut down the regular connection in this case. We will also assume that even if the regular channel has been shut down, A would be able to recognize the end of the DOS attack. This check can be implemented by A periodically trying to re-open the regular Internet connection and shutting it down again if the attack continues.

In this Section, we propose a channel switching logic for server A that has a connection with server B (either for providing the data to or querying the data from B). This logic is applied on the service level and the transactions between the servers are in the form of query/response. The same logic must be applied by A for every B that it is connected to via an alternative channel.

The logic consists of the following main items.

- A has to regularly ping every connected server B through the alternative connection in order to determine whether the connection is healthy and whether B is up.
- A has to make regular *dummy queries* to every connected server B (answered by B with *dummy responses*), the purpose of which is to determine, whether the regular connection works between A and B.
- If A is queried by B via the alternative connection, the query must be answered via the alternative connection.
- If A is queried by B via the regular connection, the query must be answered via the regular connection.
- If A is under a DOS attack, it will shut down the regular channel and start using the alternative channel for every partner B, until the DOS attack ends.
- If certain number of (dummy) queries by A to B are not answered by B, but at the same time, B pings via the alternative channel then A decides that B is up but is unable to answer to A's queries via the regular channel. This means that the regular channel is down, and hence A starts using the alternative channel for sending requests to B. Still, A continues to send the dummy queries to B via the regular channel in order to be able to change back to the regular channel later.

- If A uses an alternative channel with B and a certain number of A's dummy requests have been successfully answered by B then A decides that the regular channel between A and B is operational again and A changes over to the regular channel with B.

5 Conclusions and Further Work

In this paper, we studied the methods for increasing availability of federated database systems in case of severe network failures (e.g. due to Denial-of-Service attacks). As the basis of our studies we chose the X-Road infrastructure, developed in Estonia to allow unified and controlled access to various governmental databases. In its core, the X-Road infrastructure is rather generic to allow generalization of our proposals to other similar infrastructures.

We analyzed the requirements for improvement under the network-failure scenario. As a result of the analysis, we proposed three main categories of measures to achieve better service availability in this scenario.

First, in order to have continuous access to certificate validity information and other secure configuration data, the current DNSSEC-based solution needs to be replaced by more flexible protocols (e.g. OCSP allowing the response caching as a standard measure).

Second, alternative communication channels between critical ervices need to be introduced. These alternative channels may be considerably more expensive, hence the security servers need to support better filtering of services based on their priority.

Third, selected views of critical databases may be replicated at the service consumer's site. Specific extra components need to be developed for security servers in order to achieve such a functionality.

Finally, we stated the main procedures and criteria to decide when to switch over to alternative channels and back. Still, it has to be noted that the way channel switching logic is presented in the current paper is just a theoretical proposal. Before practical implementation it should be validated (e.g. using some simulation framework), but this effort remains out of the scope of this paper.

All of the above-described solutions are planned to be implemented for the next generation of X-Road. After the implementation phase, a thorough analysis on the quality of service and the provided level of resilience against the network problems and Denial-of-Service attacks must be conducted. Conducting such a study will also remain a subject for future work.

Acknowledgements. Authors are grateful to Viljar Tulit and Alar Jõeste for helpful discussions.

References

1. Apache Hadoop project. `http://hadoop.apache.org/`.

2. Arne Ansper, Ahto Buldas, Margus Freudenthal, and Jan Willemson. Scalable and Efficient PKI for Inter-Organizational Communication. In *Proceedings of the 19th Annual Computer Security Applications Conference ACSAC 2003*, pages 308–318, 2003.
3. Graham Bent, Patrick Dantressangle, David Vyvyan, Abbe Mowshowitz, and Valia Mitsou. A dynamic distributed federated database. In *Proc. 2nd Ann. Conf. International Technology Alliance*, 2008.
4. Steven Dawson, Shelly Qian, and Pierangela Samarati. Providing security and interoperation of heterogeneous systems. In *Security of Data and Transaction Processing*, pages 119–145. Springer, 2000.
5. 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.
6. Georges Gardarin, Fei Sha, and Zhao-Hui Tang. Calibrating the Query Optimizer Cost Model of IRO-DB, an Object-Oriented Federated Database System. In *VLDB*, volume 96, pages 3–6, 1996.
7. Dennis Heimbigner and Dennis McLeod. A federated architecture for information management. *ACM Trans. Inf. Syst.*, 3(3):253–278, July 1985.
8. Ahto Kalja. The X-Road Project. A Project to Modernize Estonia's National Databases. *Baltic IT&T review*, 24:47–48, 2002.
9. Ahto Kalja. The first ten years of X-road. In *Estonian Information Society Yearbook 2011/2012*, pages 78–80. Department of State Information System, Estonia, 2012.
10. Ahto Kalja and Uuno Vallner. Public e-Service Projects in Estonia. In Hele-Mai Haav and Ahto Kalja, editors, *Databases and Information Sustems, Proceedings of the Fifth International Baltic Conference, Baltic DB&IS 2002*, volume 2, pages 143–153, June 2002.
11. Ee-Peng Lim and Jaideep Srivastava. Query optimization and processing in federated database systems. In *Proceedings of the second international conference on Information and knowledge management*, CIKM '93, pages 720–722, New York, NY, USA, 1993. ACM.
12. Rain Ottis. Analysis of the 2007 Cyber Attacks Against Estonia from the Information Warfare Perspective. In *Proceedings of the 7th European Conference on Information Warfare and Security*, pages 163–168, 2008.
13. Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Comput. Surv.*, 22(3):183–236, September 1990.
14. Marjorie Templeton, Herbert Henley, Edward Maros, and Darrel J. Van Buer. InterViso: dealing with the complexity of federated database access. *The VLDB Journal*, 4(2):287–318, April 1995.
15. A. Tomasic, L. Raschid, and P. Valduriez. Scaling access to heterogeneous data sources with DISCO. *IEEE Transactions on Knowledge and Data Engineering*, 10(5):808 – 823, 1998.
16. Jan Willemson and Arne Ansper. A Secure and Scalable Infrastructure for Inter-Organizational Data Exchange and eGovernment Applications. In *Proceedings of The Third International Conference on Availability, Reliability and Security ARES 2008*, pages 572–577. IEEE Computer Society, 2008.