Scalable and Efficient PKI for Inter-Organizational Communication

Arne Ansper[1*], Ahto Buldas[123*], Margus Freudenthal[1*], Jan Willemson[12*]
[1]Cybernetica, Akadeemia tee 21, Tallinn, Estonia
[2]Tartu University, Ülikooli 18, Tartu, Estonia
[3]Tallinn Technical University, Ehitajate tee 5, Tallinn, Estonia
{arne,ahtbu,margus,jan}@cyber.ee

Abstract

We propose an efficient and flexible system for a secure and authentic data exchange in a multi-institutional environment, where the institutions maintain different databases and provide secure and limited access services to employees of other institutions. The main motivation for building such a system was to organize efficient cooperative use of state registers, in order to increase the efficiency and quality of public services in Estonia. In order to meet high security requirements, several contemporary measures are integrated (using digital signatures, distributing certificate information by means of DNS protocol and linking log files with cryptographic checksums). We give rationale for the design decisions made in the implementation process and conclude with the current state of public use of the resulting infrastructure.

1 Introduction

During recent years, organizations have been making increasingly more use of information technology, especially the Internet in order to simplify their business processes. On the governmental side, most agencies use computerized information systems and some agencies are actively promoting the use of Internet-based services to their customers. Despite considerable progress in the computerization of internal information systems of agencies and communication between citizens and government agencies, much less is done in electronic communication between agencies. Mostly, one of the following methods is used:

- *Paper-based method* – in order to perform a task in agency A, a citizen goes to agency B and receives a written assurance on some fact about her (e.g., that she has no outstanding tax debt). She then presents the assurance (signed by B) to agency A who performs the task. Essentially, agencies use citizens as a "transport protocol".
- *Ad hoc methods* – several agencies have developed solutions for exchanging information between them. The technical solutions vary from custom HTTP-based protocols to manually transferring magnetic tapes or floppy disks. In general, the security of these methods is questionable and the receiving party can get no assurance on the quality of received data.

The methods above are not cost-efficient as new a solution has to be developed for each pair of interacting agencies. Agencies wishing to exchange information with some agencies and wishing to connect to other agencies, must do additional development work for each additional agency, as they all use different protocols and data formats. Lack of standardized solutions prevents small agencies (e.g. regional authorities) from accessing the necessary data electronically.

One way to lower the costs of connecting state agencies is creating a unified framework. Using common protocols and data formats means that once an agency has implemented these protocols, connecting

*All authors were supported by the ESF grant 5568

to an additional agency does not involve much additional work. Standard implementations can be provided to lower start-up costs for small agencies. Additionally, this framework can ensure that a standard set of security measures is implemented uniformly across all agencies.

This paper describes one such technical infrastructure called X-Road (Crossroad), which was developed in accordance with a development project initiated by Estonian government in 2001-2002 [10]. The aim of this project was to provide governmental agencies access to information stored in state registers. State registers are databases containing important information used by the government (e.g. Population Register and Land Cadastre). Major state registers are usually maintained by separate legal entities. Convenient access to state registers by other government agencies improves the efficiency and quality of public services. State officers are granted fast and direct access to all state registers necessary for performing their duties. At the same time, the security measures taken in the system are sufficient for the system being compliant to the restrictions defined by national laws and international agreements. As of today, the system is in active use by 10 different Estonian state registers.

The paper is organized as follows. In section 2, we discuss the main design goals and motivation of the X-Road project and analyze the main use cases that occur in inter-agency communication. We also describe the basics of our solution to user identification and authentication. Section 3 refers to some previous projects of similar nature. Sections 4 to 6 describe technical aspects of the X-Road system: design of the public key infrastructure (PKI) solution, measures taken to solve disputes between parties. A description of the general infrastructure and services offered by the X-Road system follows. Finally, some performance estimates are given based on current implementation.

2 Design goals

2.1 Security goals

We identify main security goals that should be kept in mind when designing such an interconnected information system:

- *Confidentiality.* State institutions often use confidential (classified) information in their normal working processes. Access to such information has to be restricted. This problem becomes considerably harder if information systems of different institutions are connected. Hence, a strict role-based authentication must be used in the middle-ware that connects the information systems, in order to identify the officers and their access rights.
- *Integrity.* The information received from other institutions should be as reliable as official letters or documents, considering its use in decision-making processes. Hence, there have to be measures used in the system to protect the integrity of data transmitted from one institution to another. The source of the information received has to be detectable in order to prevent unauthorized modification and impersonation by hackers.
- *Availability.* State institutions will increasingly depend on electronic information and simply cannot function when electronic information is unavailable. Hence, the availability of electronic registers is a high priority and its importance continues to grow in the near future. Therefore, it would be good if state registers were available even if a part of the technical infrastructure was out of order. Hence, one has to avoid single points of failure in the system.

Cooperative actions are required from institutions to implement the overall security policy of the system. Confidentiality of information does not depend on a single institution any more but also on those other institutions that use the information via the new interconnected system. Liability has proved itself in practice as one of the most effective mechanism to make people and institutions follow an inter-institutional policy. Hence, before such a cooperation becomes possible in practice, new rules of liability have to be defined.

A monitoring system must be created to detect and identify attempts of abusing confidential information and sending low-quality information, as such events may cause considerable loss for parties relying on this information. We need a logging mechanism that can be used to trace messages, some of which may have been sent long ago.

2.2 Main use cases

2.2.1 Database queries

An employee C_A of agency A makes a query q to agency B. Agency B, before replying to the query, verifies the identity of C_A and checks whether C_A is authorized to receive the information she applies for. After successful identity verification and authorization check, B sends C_A a reply (q, r) (which also includes the original query, see Subsection 6.1 for further discussion). Before accepting the reply, C_A has to verify that it indeed came from B.

Because X-Road only is concerned with computerized databases, employees of agency B do not directly participate in transactions. Agency B as a whole is considered to be the party liable for guaranteeing the quality and legally correct usage of data. Some of this liability is transferred to agency A by a special legal agreement as B can not have full control over A's actions and possible misuses of the data.

2.2.2 Disputes

Suppose that a security incident happened due to communication between an employee C_A of agency A and agency B. Then a dispute may take place between C_A, and a representative of B (which we identify with B). We assume that there is always a third party participating in the dispute, which we refer to as Judge. It may represent a higher state institution, Court, etc. The following claims can form a basis for disputes.

1. *Wrong data*. Suppose C_A has made a wrong decision due to erroneous reply (q, r) created by agency B. In order to prove that she acted correctly, C_A presents the reply r to Judge. Judge checks whether the reply is authentic and was created by B. If the check succeeds, then Judge decides that B is liable for the wrong decision. Otherwise, C_A is decided to be liable.

2. *Leakage of confidential information.* It is claimed that B released confidential information to an unauthorized person. As an evidence, a reply (q, r) is presented to Judge. Judge verifies the authenticity of the reply and checks whether it was indeed created by B. If the check succeeds, then B has to prove that the information was released after successful identity verification and authorization check. If Judge can not verify the proof, B is decided to be liable for leaking the information. Otherwise, B (and, depending on the circumstances, sometimes also C_A) is considered liable for the leakage of data.

Hence, both A and B have to save evidence about the messages received by them – B because of confidentiality requirements and A because of integrity requirements of the overall security policy.

2.3 User identification/authorization

One of the questions that arises during the development of the system is how to authenticate the users of the system. This question has new dimensions in the interconnected system because an information system of agency B has to identify not only its own employees but also employees of other institutions. At the first glance, the solution may seem trivial – give each user a public key certificate and use PKI. Closer analysis of such a solution shows that the problem is deeper: it is not enough for agency B to identify a person C_A (say, an employee of agency A) who tries to use the information resources of B. The main question is whether C_A is authorized to use the information.

Mostly, the authorization rules are not identity-based but rather role-based – persons who have certain positions in state agencies are authorized to use the information. If C_A goes from one position to another (say higher) position, her rights (authorizations) will change automatically. The reader should keep in mind that a person's rights to use information in B's information system depend on A, and not at all on B. How can agency B obtain up-to-date information about the position of an employee C_A of another agency A?

The most natural way is to obtain such information directly from the source – agency A. Queries may comprise a confirmation of A that C_A has a certain position in A. Moreover, there is no technical reason for B to identify C_A in person. This function may be delegated to agency A. Agency B has just to verify whether the query received came from agency A.

In a closer look, this solution is natural and obvious. First, A is able to give C_A any position (in A) it needs to. Second, authentication mechanisms are under control of A, because even the workstation that C_A uses belongs to (and is controlled by) A, which means A can always impersonate C_A.

Hence, for interconnecting information systems of different institutions we need to organize secure data interchange between agencies, and not between persons. This means that the number of public keys we need is smaller and the PKI can be used in a more efficient way.

Keeping the above in mind, we used a two-level authentication scheme. If an employee C_A of agency A wants to query information from agency B, the following two authentication procedures are performed:

- C_A authenticates herself to agency A,
- agency A authenticates itself to agency B.

Thus, we have the following accountability rules:

- agency A is responsible for authenticating its employees and ensuring that users can perform only queries they are authorized to perform;
- agency B is responsible for authenticating other agencies and ensuring agencies can only perform queries they are authorized to perform.

Both agencies need to store data that can later be used as a proof that they exercised due diligence in performing their duties.

One benefit of this two-level authentication scheme is that there is no direct need to develop unified standards for how the information systems of agencies identify the employees. That again simplifies the technical statement of the problem. Recall that our goal is not to introduce a global and revolutionary user identification scheme, but rather to obtain a working system with minimal efforts and expenses.

To sum up, we have reduced our initial problem to two simpler sub-problems:

1. how to organize a secure and reliable communication between state agencies and
2. how to authenticate employees inside an agency.

Public key cryptography provides suitable technical basis to solve the first sub-problem. As the number of state agencies is relatively small (compared to the number of state officers), it is feasible to issue public key certificates to all agencies and to manage the resulting PKI. Solution to the second sub-problem is solved by each state agency individually. However, operational access control system is a prerequisite that must be fulfilled before an agency can join the X-Road infrastructure.

3 Prior art

In [9], Jason Hackerson discusses experiences from implementing US defense department's public key infrastructure. Originally this project was intended to decrease system administration costs and make networked communications more secure by issuing public key certificates to all military personnel. He concludes that introducing PKI did not solve most of the problems it was expected to solve, but instead created new problems or vulnerabilities. Main problems were related to managing globally accessible directory of users, managing access control lists and operating under limited bandwidth conditions.

SHS project [14, 15] implemented by Swedish government is aimed at producing secure infrastructure that can be used to exchange information between government agencies. Authentication and encryption are performed between organizations, not between individual government officers. All communication is secured using the Secure Sockets Layer (SSL) protocol and optional signing of messages. Standard PKI methods, such as certificate revocation lists (CRL) and LDAP directory are used. SHS does not provide a complete infrastructure and relies on external certification authorities (each set of organizations can decide on a different set of trusted CAs). Security measures to be used by organizations are not standardized at very specific level. Additionally, security mechanisms are not separated from applications and applications are responsible for managing security-related data (such as private keys) and making security decisions.

Canadian government is building a PKI that is meant for use by government agencies [16]. Government agencies manage certification authorities that certify their employees. Central authority cross-certifies these local authorities and acts as a trust root. Secure communication is performed between persons, not between organizations. Standard X.509 protocols are used for certificate management.

S/MIME working group of IETF has developed an experimental specification "Domain Security Services for S/MIME" [4]. This specification describes services that can be applied to E-mail messages passing through gateway from one domain to another. These services include domain encryption and decryption and domain signature, which consist of delegating encryption/decryption and signing functionality to organization's mail server. This specification is applicable only for E-mails and does not cover certificate management issues.

4 PKI for state agencies

A major task to solve is to broadcast reliably information about public keys that are used by the agencies to encrypt and sign messages during communication. We have to consider that private keys may be compromised and hence there has to be a way of obtaining up-to-date validity information. At the same time, we try to avoid solutions where temporary unavailability of this information breaks the operation of the system as a whole.

Unfortunately, the "classical" PKI has not been designed to meet this availability requirement. In order to make the received the messages verifiable, we need either on-line certificate verification (OCSP) services or recent Certificate Revocation Lists (CRLs). Moreover, to be able to use the message later in disputes, we also need an operable time-stamping Service. By using duplication and synchronization techniques, we can, in principle, improve the availability characteristics of classical PKI solutions. However, there is a simpler and more natural way to meet the availability requirement. Our system will need a name service for resolving agencies' names to their network addresses. A natural choice for implementing that service would be the Domain Name System (DNS) [11, 12]. The same infrastructure can also be used to broadcast public key information [7].

In our solution, DNS is used to distribute security-critical information (addresses and certificate validity information), therefore DNS records need to be protected from unauthorized modification. We use DNSSEC [6] (an extension to the DNS protocol), which uses digital signatures for protecting authenticity of DNS responses. The primary name server signs all the records (each record containing a name-attribute pair) in its database. Having received a response to the DNS query (possibly from secondary or caching server), a client can verify the authenticity of the DNS records returned by using the public key of the primary server.

A PKI solution based on DNS

Figure 1 depicts the infrastructure for distributing certificate information. Certificates are signed by the off-line Certification Authority (CA), which also maintains database of all certificates (both valid and revoked). When CA updates the database, all currently valid certificates are transferred to the Primary DNS server. The primary DNS server generates a new DNS zone (database of DNS records), signs the zone with the private key sk_{DNS} and distributes it to secondary DNS servers that are maintained by the authority who also maintains the primary server.

When agency A is issued a new certificate, A down-

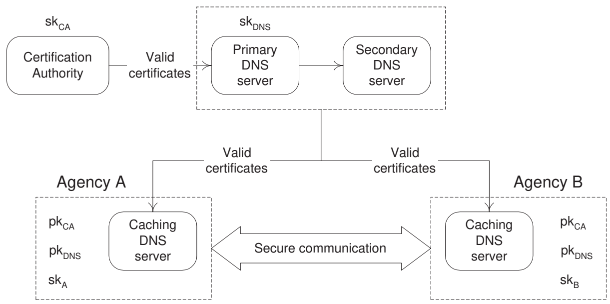

Figure 1: PKI solution by using DNS

loads the certificate by querying the DNS system for the CERT attribute associated with its name. When creating a secure connection, agency A sends its certificate to another party as a part of the SSL handshake. The same certificate is used for verifying signed messages. Hence, there is no need for locating and downloading the other party's certificate. When an agency needs to check the validity of the other agency's certificate, it queries the DNS system for a name which contains a cryptographic hash of the certificate. If the certificate is valid, the hash is contained in the DNS database and the corresponding response is returned. Otherwise, the "not found" error message is returned.

The use of the DNS protocol enables one to use caching features of DNS which improve both the performance and the availability of certificate distribution. When the primary DNS server is unavailable, secondary DNS servers are still ready to answer queries until the signatures on DNS records expire. Currently, signatures are valid for six hours, but the caching DNS servers try to refresh their data much more frequently, in every fifteen minutes. This provides agencies with reasonably up-to-date information, but if both primary and secondary servers are unavailable, agencies can still use information in their caching name servers (until signatures on DNS records expire).

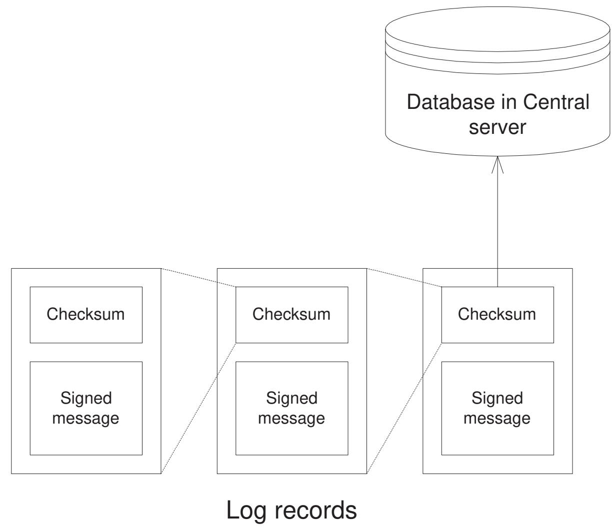

Figure 2: Secure logs used by state agencies

5 Logging and disputes

In order to ensure long-term verifiability of messages, the following measures are taken (see Figure 2):

- Agencies save all the received messages to log files. Records in these logs are connected using cryptographic hash functions. Each record in the log comprises cryptographic checksum of the previous record. Hence, each record depends on the contents of all previous records.

- Agencies periodically send checksums of their most recent log records to the central server. The central server saves these checksums into a database together with current the date/time. Such a commitment procedure prevents agencies from modifying their logs afterwards, and thereby, makes the contents of the logs admissible as evidence to solve further disputes between agencies.

In order to prove that agency A has received a signed message from agency B, first the message together with the included B's certificate is located in A's log file. Cryptographic checksums of the records are recomputed step by step, starting from the located message and if the recomputed value coincides with one of the checksums on the central server, then it is possible to determine the approximate time when the message was sent. Relying on the validity information maintained by the central server, the signature is declared either valid or not.

In the current implementation of the system, the central server is totally trusted and acts as a judge in case of disputes. Therefore no special measures are taken to ensure the integrity of the certificate database and the database of log hashes. In order to make messages provable to external parties (courts etc.) the disputes must be made solvable without participation of the central service. This is possible by using the following techniques:

- Instead of sending checksums of their logs to the central server, agencies may send them to a linked time-stamping service [8, 3].

- Some measures should be taken to prevent the CA from modifying old records in the certificate database. One way to do this would be to use a certificate status protocol where each message is accompanied by a signed validity statement (e.g. OCSP [13]). If this statement is time-stamped, the CA cannot modify old records of certificate database without being verifiably inconsistent with validity statements. Alternatively, the CA may periodically time-stamp contents of the certificate database. More complicated protocols, such as those described in [2], can be used to ensure long-term verifiability of the certificate data.

6 X-Road infrastructure

6.1 General overview

In this section we outline the architecture of the X-Road system (see Figure 3). The whole infrastructure is distributed by its nature. The only central authority is an organization named *X-Road center* that handles certification, secure name service and logging. The main components of the X-Road system are the following:

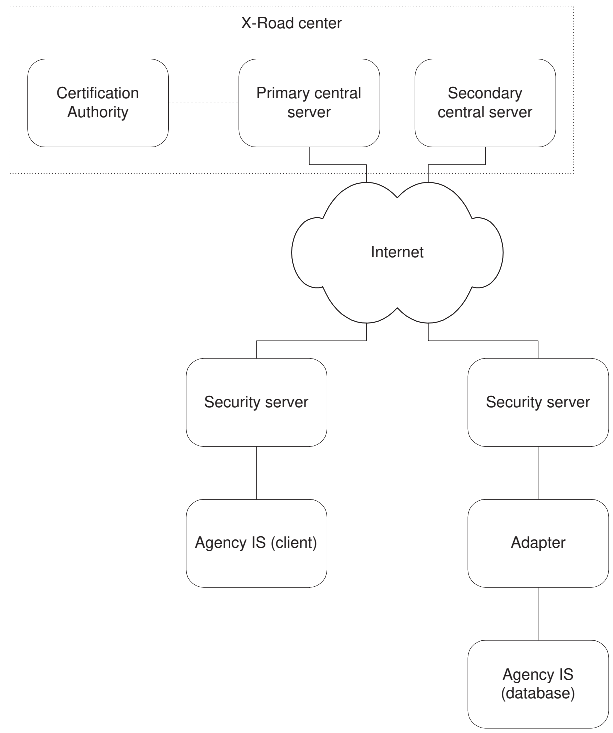

Figure 3: General view of the X-Road architecture

- *Certification authority* – managed by X-Road center. Its duty is to issue public-key certificates to security servers. CA is an off-line entity and communicates with the central server in an off-line fashion (floppy disk).

- *Central server* – managed by the X-Road center, contains a DNS server (either primary or secondary) and a logging server. The logging server receives hashes from security servers and saves them in the database. Central servers use redundancy to improve availability.

- *Security server* – managed by an agency that uses the X-Road system. Security servers encapsulate all security-related functionality, such as signing and verification of messages, sending messages over secure channels, storing evidence for later disputes, etc.

- *Client information system* – this is the information system run by the organization that uses the data. The client information system sends requests to the security server, which secures the requests and forwards them to the other party's security server.
- *Adapter server* – managed by the data provider. Since most data providers use existing databases, they need adapter servers that receive queries from security servers and translate these queries into a database-specific format.
- *Database* – information system that provides actual services used by other agencies.

Agencies exchange information using XML-based query-response protocols XML-RPC and SOAP. It is possible to turn this into one-way communication by creating queries like “accept message or document X” and responses that simply acknowledge receipt. All communication between agencies is secured by using the Secure Sockets Layer (SSL) protocol [5] with both client and server authentication. All messages are signed and logged so that in case of a dispute it is possible to prove that they were indeed sent. Each response contains the corresponding query. That makes the response a “full sentence” of the form “The answer to your query 'is person X allowed to cross the border?' is 'yes'.”. This prevents presenting the answer to one query as the answer to a different query.

6.2 Security servers

All agencies connected to X-Road must implement the standard X-Road communication protocol and security measures. These measures are quite complex and it is not realistic to expect every agency (especially the smaller ones, employing only a few people) to correctly implement them. Administering security-critical systems requires expertise and resources to configure systems without unnecessary services and to continuously keep track of security advisories and install patches. Therefore, it was decided that all security-related functionality should be concentrated into self-contained easily manageable components – Security servers.

A security server acts as a firewall between agency's information system and the Internet. It routes messages to their recipients and secures messages both for transport (by encryption) and for long-term validation (by signing and logging). In terms of the information system the security infrastructure is completely transparent – the application level protocol used between the information system and its security server matches exactly the protocol used between an adapter server and its security server. If we connect the information system (that makes a query) directly to the adapter of the database servicing that query, the two information systems operate in the same way as if they were connected through the security infrastructure.

Security server has very small administration needs. It is a self-contained system which offers a minimal user interface that lets the administrator only to modify basic configuration parameters, manage the keys and perform day-to-day managing tasks, such as archiving message logs. Besides placing low demands on the skills of systems administrator, this approach also makes it easy to restore the server in case of a failure: reinstalling software and restoring backed up configuration takes just about 15-20 minutes, thus reducing downtime. Since all security servers run almost identical configurations, providing technical support is also easier.

6.3 Key management

X-Road system contains a quite shallow certification hierarchy, where all agencies are certified by a single Certification Authority. Standard X.509 format is used for all certificates. Keys are generated by the security servers of the agencies. Certification requests are stored on a floppy disks and transported to the X-Road center, either personally or by secure conventional mail.

Although authentication-encryption and digital signatures are usually considered separate applications, the same key is used both for the SSL protocol and for signing requests and responses. Both operations (encryption and signing) are done in the same application with identical security properties, and hence using two different keys gives no extra security.

Top-level public keys (pk_{CA} and pk_{DNS}) are stored in central servers where they can be fetched by security servers. Cryptographic fingerprints of these keys are either communicated to agencies by telephone or sent by secure conventional mail. The digest is entered into the security server via the configuration interface, which fetches the key from the central server and verifies its authenticity. The requirement to type in the digest prevents systems administrators from fetching the key first and then simply pressing "Next" when the authenticity of the downloaded key needs to be verified.

In the X-Road architecture, special attention was paid to the key change procedure. Instead of using traditional X.509 tricks (certificate suspension, very long validity periods) to avoid key change, it was decided to make changing entity's key as simple as possible and use it for all occasions. In the X-Road system, even the top-level keys can be changed relatively frequently.

In order to change some entity's key, first a new key is created and certified, but for some time, the entity continues to use the old one; other parties trust both the new and the old keys. When all parties have received the new key, entity deletes the old key and starts to use the new one. After some time, certificates associated with the old key are revoked and other parties trust only the new key. Delays between these steps must exceed the DNS update period so that key information is distributed between all the involved parties.

This procedure for key exchange ensures that parties always have at least one valid key and are therefore able to process requests. For example, the CA key can be changed while all security servers are processing requests at their full capacity. During that process, all messages are successfully delivered and there are no interruptions in service.

6.4 Efficiency

Central server

The central server has to (1) answer the DNS queries, (2) record log checksums from security servers, (3) generate and sign DNS zones (primary server only). The workload resulting from these tasks does not depend on the amount of queries forwarded through the X-Road system. It is proportional to the number of agencies and security servers.

The only resource-consuming task is generating and signing DNS zones on the primary DNS server. Answering DNS queries is not a very demanding operation and the use of caching DNS servers reduces the load even more. Recording cryptographic checksums simply consists of receiving about 40 bytes from the network and saving it to database. Considering that security servers send out these checksums every five minutes, this kind of traffic is not a problem even when hundreds of thousands of agencies join the X-Road system. The signatures on DNS records are pre-computed in batch mode, and hence, they do not cause computational bottlenecks.

Security server

Security server's load is directly dependent on the number of queries forwarded through this server. For each query, the security server must

- Sign the query or query response.
- Verify the other party's signature on query or query response.
- Initiate SSL connection to the server or accept incoming connection. Both client and server cache SSL sessions and therefore public key operations are used very rarely.
- Check the validity of other party's certificate. This requires one DNS query; DNS server's signature is verified once and later the cached response is used.
- Log an incoming query or a query response. This involves one disk operation and one hash computation.

Therefore, the most resource-consuming task for a security server is signing and verifying messages. This observation was confirmed in load testing where the performance of security servers was measured on different hardware platforms. The best indication of server's performance turned out to be the processor speed measured in RSA signatures per second (this measurement was obtained using `openssl speed rsa1024` command). Processing a query took only 1.26 times more time than one RSA signature. Signature verification, message decoding/encoding and other maintenance functions consumed about one fifth of the computing time. Thus, it can be seen that a regular 2.4 GHz P4 PC (capable of 204 RSA signatures per second) can serve about 160 queries per second.

Security servers can make use of load balancing between multiple servers. On the client's side, agency's information system is responsible for dividing queries between several security servers. On the server's side, agencies may have several security servers, each one using separate external IP address. When making a connection, client picks server's IP address at random from all addresses associated to a given agency. Therefore, agency's throughput scales linearly in the number of computers. Redundant servers approach also improves the availability of the system.

6.5 Standardized information systems

Although most of the agencies are expected to integrate X-Road related functionality into their information systems, the X-Road center also offers a generic information system called MISP (mini information system/portal) for free. MISP is a web-based system that allows users to make queries using X-Road. MISP complies with all the requirements placed upon agency information systems and it is mainly used for testing purposes and by small agencies who cannot afford building their own customized systems.

MISP is not tied to specific queries, it uses XSD schemas [1] describing queries and query responses to automatically generate forms for entering query parameters and for displaying responses. Additionally, MISP uses special queries for determining the list of all available databases and determining which queries it is allowed to make.

For end-user authentication, MISP uses a mix of password-based and certificate-based methods. The user uses the password to associate certificate (issuer of the certificate is not important) with her account and uses this certificate for day-to-day operation. After this, the system generates a new password for next use time when a change of a certificate is needed.

6.6 Citizen portal

Citizen portal is a special kind of agency information system that can be used by all citizens to access information about themselves. All queries that are accessible to Citizen portal take citizen's personal code as an argument (usually it is the only argument) and return information stored in the registry concerning this particular citizen. Although almost all Citizen portal queries are informational, some queries are planned which allow citizens to send documents to state agencies.

Citizen portal can use two methods for authenticating persons:

- Using the authentication certificate stored on the Estonian national ID card.
- Using Internet banks – citizen logs on to her Internet bank account (more than third of population has an Internet bank account) and clicks on the Citizen portal link. She is then forwarded to the portal together with authentication information.

7 Conclusions and further directions

The X-Road system was initially deployed in December 2001 and has passed the piloting phase by now. By May 2003 there were 10 state registers involved in the infrastructure, including Business Register, Passport Register, Land Register, Register of Buildings, Population Register and others.

The range of applications of the X-Road system is probably much broader than connecting state agencies. It is a general framework for exchanging messages between different organizations (although the system was originally designed for query-response protocol, it can also be used for document exchange). The ideas used in the X-Road architecture can be fruitful in many other PKI applications as well.

To extend the variety of possible applications of the X-Road system (legal-grade electronic documents, very small agencies, etc.), the following developments would be suitable:

- In order to make the logs of the X-Road system legally admissible evidence, time-stamps have to be obtained for the cryptographic checksums (using a linked time-stamping service).
- Currently, the security servers are separate computers. To lower the setup costs for very small agencies, a software version of the security server should be created that is capable of sharing a computer with application programs.

References

[1] W3C XML schema. http://www.w3.org/XML/Schema.

[2] Arne Ansper, Ahto Buldas, Meelis Roos, and Jan Willemson. Efficient long-term validation of digital signatures. In *Public Key Cryptography - PKC'2001*, volume 1992 of *LNCS*, pages 402–415. Springer-Verlag, 13 February 2001. Cheju Island, Korea.

[3] Ahto Buldas, Peeter Laud, Helger Lipmaa, and Jan Villemson. Time-stamping with binary linking schemes. In *Advances in cryptology – CRYPTO'98*, volume 1462 of *LNCS*, pages 486–501, Santa Barbara, 1998. Springer-Verlag.

[4] T. Dean and W. Ottaway. Domain Security Services for S/MIME. RFC 3183, October 2001.

[5] T. Dierks and C. Allen. The TLS Protocol Version 1.0. RFC 2246, January 1999.

[6] D. Eastlake. Domain Name System Security Extensions. RFC 2535, March 1999.

[7] D. Eastlake and O. Gudmundsson. Storing Certificates in the Domain Name System (DNS). RFC 2538, March 1999.

[8] Stuart Haber and W. Scott Stornetta. How to time-stamp a digital document. *Journal of Cryptology*, 3(2):99–111, 1991.

[9] Jason X. Hackerson. Rethinking Department of Defense Public Key Infrastructure. In *23rd National Information Systems Security Conference, Baltimore*, October 2000. MD, USA.

[10] A Kalja and U. Vallner. Public e-Service Projects in Estonia. In Hele-Mai Haav and Ahto Kalja, editors, *Databases and Information Systems, Proceedings of the Fifth International Baltic Conference, Baltic DB&IS 2002*, volume 2, pages 143–153, June 2002.

[11] P. Mockapetris. Domain Names – Concepts and Facilities. RFC 1034, November 1987.

[12] P. Mockapetris. Domain names – Implementation and Specification. RFC 1035, November 1987.

[13] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP. RFC 2560, June 1999.

[14] The Swedish Agency for Public Management. *Description of SHS*, 2000. Available online at http://www.statskontoret.se/gel/shortdesc.pdf.

[15] The Swedish Agency for Public Management. *SHS version 1.0 and version 1.1 documentation*, 2003. Available online at http://www.statskontoret.se/shs/spec.htm.

[16] Treasury Board of Canada Secretariat. *Government of Canada Public Key Infrastructure*. http://www.cio-dpi.gc.ca/pki-icp/.