Addressing Security Requirements within the SPACE System

IMEDIA/02/00

Arve Larsen

February 2000

Norsk Regnesentral
ANVENDT DATAFORSKNING

NR-notat/NR Note

Tittel/Title:
Addressing Security Requirements within the SPACE System

Dato/Date: February
År/Year: 2000
Notat nr: IMEDIA/02/00
Note no:

Forfatter/Author:
Arve Larsen

Sammendrag/Abstract:

The SPACE project developed a fully functional demonstrator of a system aimed at supporting European citizens moving between countries within the EU and the EEA. The project also established several security requirements for a production system. These are only partially addressed in the demonstrator. This paper presents these requirements and their implications as well as possible solutions utilising well-know techniques. A prototype implementing these solutions is also presented.

Emneord/Keywords: SPACE, applied security

Tilgjengelighet/Availability: Open

Prosjektnr./Project no.:

Satsningsfelt/Research field:

Antall sider/No. of pages:

Norsk Regnesentral / Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, 0314 Oslo, Norway
Telefon 22 85 25 00, telefax 22 69 76 60

1 Introduction

The demonstrator developed within the SPACE project [6] implements necessary functionality required to support European citizens moving within the EU and the EEA. However the demonstrator does not fully address all security requirements posed by the involved countries and administrations. The design of the SPACE system [5] provides some initial elements of the required security features to show possible extendibility of the basic design. It is the purpose of this paper to explore the necessary extensions.

The security requirements can be divided into the following groups: citizen data privacy, data authentication, limited disclosure and system security. Citizen data privacy specifies that the citizen (and only the citizen) may access data about himself (and no one else). Data authentication specifies that the source of all citizen data must be reliably specified. Limited disclosure allows administrations to limit disclosure of citizen data information to specific administration. This includes excluding information from the citizen himself. System security specifies that no unauthorised persons or processes may access critical parts of the system. It also includes system stability, availability, robustness etc.

This article has had a two-fold goal. First we wanted to analyse the remaining security requirements within the SPACE system and propose a conceptual design for a security solution. Secondly we wanted to implement this design using standard techniques and services to preserve the philosophy of the SPACE system.

The article continues with a brief overview of the SPACE system in section 2. Section 3 presents the security requirements in more detail. A revised design of the SPACE system satisfying these requirements is presented in section 4. Section 5 describes a prototype implementation of important aspects of this design. Conclusions and directions for further work are presented in section 6.

2 Process and Infrastructure Overview

The SPACE system provides three basic services, advice, portfolio creation, and portfolio delivery. The advice service tailors advice about moving to according a profile of the citizen and his moving situation. The profile is anonymous and is established through a dialogue with the citizen [9] using no sensitive data[1]. The resulting advice consists of tailored public information. There are no specific security requirements for this part of the SPACE system. Some aspects, like reliably establishing the source of a given dialogue fragment could be considered.

Portfolio creation can be divided into several steps. First a profile of the citizen is established through a dialogue with the citizen. In contrast to the advice process this profile is not anonymous and may contain personal information. The destination country uses this profile to specify exactly which data elements are required about the citizen. This includes information about which administrations request which data elements. The departure country uses this list of required data elements to determine which data elements it can provide values for and a set of keys required to retrieve the data.

The set of required keys is presented to the citizen for him to provide values for each key. The valued key set is used by the departure country's citizen data systems (CDS) to retrieve information about the citizen. The data from each CDS is placed into a data package. All data packages for a specific citizen is combined into a portfolio. A special data package, the citizen data package, allows the citizen to provide new or updated information. The demonstrator does not consolidate

[1] Some of the information might have been sensitive if the profile had not been anonymous.

data in different data packages. However the generic design allows different countries and administrations specify different policies for data consolidation. The finished portfolio is stored within the system.

Portfolio delivery allows the citizen to complete his moving scenario by transferring his portfolio and delivering it to the destination country for incorporation into the destination country's citizen data systems.

In addition to the above functionality the SPACE project designed a distributed system architecture for this system. The system is divided into the following parts: SPACE Client, SPACE Master object, SPACE Expert object, SPACE CDS object and real CDS. The client part of the system runs on a Java enabled browser on a client machine. The other layers make up the server side of the system. Figure 1 shows the different object types and their relationships.

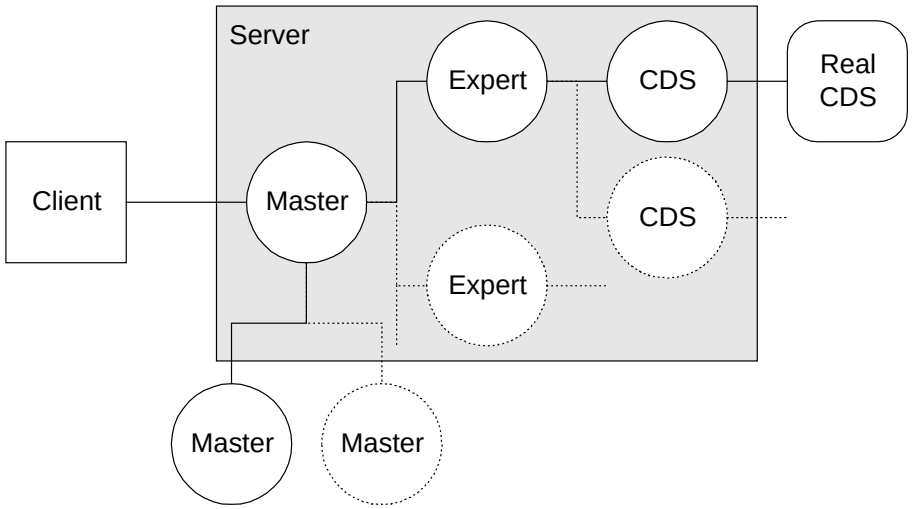

Figure 1: Space Objects

The master level corresponds to a country. A master interacts with clients and experts as well as with other masters. The expert corresponds to sectors. An expert interacts with its master as well as with its CDS objects. The CDS level is the SPACE interface to the real citizen data systems. A CDS object interacts with its expert as well as with a specific real CDS. All the objects for a country constitute a logical server running on one or more machines.

The different layers within the SPACE system can be used to divide the underlying infrastructure into several domains as illustrated in Figure 2. This division is compliant with the IDA Architecture Guidelines [7]. The different SPACE objects acts as gateways between different domains. The SPACE Master performs the same type of work as the EuroGate.

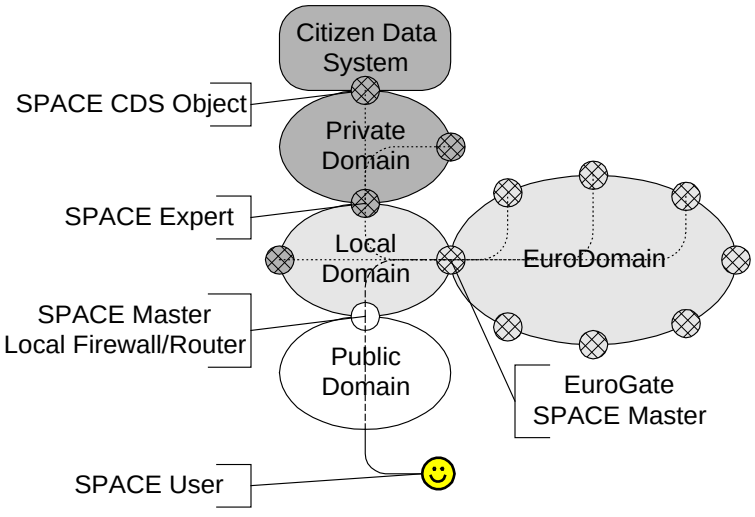

Figure 2: Infrastructure

The EuroDomain is best realised using IDA's TESTA II infrastructure or a similar infrastructure offering the same quality of service. A typical LocalDomain will be a National Network like the Government Secure Intranet in the UK [4]. A typical PrivateDomain will be a network dedicated to a National sector or administration. The PublicDomain will be the citizen's access network such as the Internet or other public access solutions.

3 Security Requirements

The SPACE project gathered requirements from the involved countries and administrations [18, 19], including security requirements. Our analysis of these given security requirements gives a set of implicit requirements. In addition the European Unions work on Information Security [8] have been used as background information in analysing these requirements, particularly in selecting techniques that may be expected to be recognised as satisfactory in our European context. This section is limited to requirements concerning portfolio creation and delivery.

An important assumption in the SPACE concept is that the citizen has the right to request and receive all pertinent information about himself. This is in accordance with the EU directive on data protection [3] that should be reflected in national legislation. Once in possession of this information he is free to deliver this information to other parties upon his own discretion. One of the security requirements invalidates this assumption but it is still addressed within the SPACE system.

3.1 Given Requirements

The requirements given by the administrations were not very specific, but rather directions to comply with national legislation. The most important aspects mentioned were information security, data privacy and system security.

Information Security In relation to SPACE the most important aspects of information security are source authentication and limited disclosure. Source authentication, including establishing credentials, allows any administration receiving data through the SPACE system to ascertain precisely the source of the data thus establishing the validity of the data. Today bilateral agreements and certified documentation is used to achieve the same. Some countries will only transfer specific information to selected administrations, sometimes even excluding the citizen

himself. The SPACE system must support this type of limited disclosure even though it may violate the assumption of the citizen's right to receive data about himself.

Data Privacy In the SPACE context data privacy means that the citizen and only the citizen may get access to data about the citizen. To facilitate this the SPACE system must provide reliable mechanisms for citizen authentication, authorisation for data retrieval and portfolio protection as well as secure data channels for transfer of citizen information.

System Security In addition to the information related requirements above the authorities require the system itself to be secure. This includes all the mechanisms above as well as mechanisms to limit access to the real citizen data systems and ensure system reliability.

3.2 Implicit Requirements

Based on the given requirements we have further described the implicit requirements for a secure implementation. For each of these requirements we have proposed a solution based on well-know techniques to be implemented in our prototype.

Citizen Authentication The system must be able to reliably ascertain the citizen's identity and make sure that any requests for citizen data comes from the citizen himself.

Solution: The citizen signs his valued key set using his private key. The CDS checks the signature using the citizen's public key obtained using information within the key set.

Citizen Authorisation The system must reliable establish the citizen's authorisation for data retrieval. This mechanism is also basis for non-repudiation.

Solution: Same as ''Citizen Authentication" with the current time as a part of the key set to prohibit replay of captured signed key sets.

Portfolio Protection During its lifetime outside the citizen data systems the portfolio must be protected to ensure that only the citizen may access the information inside the portfolio.

Solution: Requires a secure data channel for unencrypted data from CDS to client. The portfolio is encrypted in the client using a one-time symmetric key. The symmetric key is encrypted using the citizen's private key. The encrypted portfolio and symmetric key are transferred to the SPACE Master where they are stored. To enable portfolio delivery the portfolio is decrypted in the client and transferred through a secure data channel to the destination SPACE server (and ultimately one or more CDS).

Secure Data Channel Transfer of data between different components must be secured so that no one can access the data during transfer.

Solution: This can be achieved either by using middleware that provides this functionality, like CORBA using a secure protocol like SecIOP, or by application level encryption.

Source Authentication The system must allow anyone receiving data from a portfolio to authenticate the source of the data within the portfolio. A special kind of source authentication arises when data from analogue sources is put into an electronic portfolio. We have called this type of source authentication ''credentials" and treated this as a separate requirement.

Solution: Each data package is signed using the source administration's private key.

Credentials When the citizen himself provides values for required data fields, most administrations require some sort of credentials to be established. In a manual system the citizen presenting certified documents to a competent person establishes credentials. The system then trusts this person's judgement of the validity of these documents.

Solution: We have as yet not found a satisfactory solution to this problem. In the long term one may perhaps expect that all required data may be available electronically. Another possible solution is presented in chapter 5.4.

Limited Disclosure The system must allow any administration providing data to limit delivery to specific parties. This includes establishing who is asking for specific information.

Solution: The source CDS object uses the list of required data elements to establish which administration is requesting which data elements. The data in a portfolio is structured according to which administration requests data. The source CDS object encrypts the parts it wants to limit using a one-time symmetric key. This key is encrypted using target's public key and included in the data package. The target decrypts the symmetric key using its private key. The symmetric key is then used to decrypt the data itself.

CDS Access Access from a CDS object to the real CDS system must be secure.

Solution: There are two basic types of solution to this problem.

1. Citizen authentication takes place within the real CDS.
2. One of the SPACE objects handles the actual authentication. A secure protocol including private domains, secure data channels and trusted connections are set up, allowing the real CDS to trust the request it receives from the SPACE system

To accommodate for a variety of different real citizen's data systems, the data retrieval part of the CDS object, including the interface to the real CDS is not part of the generic SPACE design. Thus the actual solution to this requirement is outside the scope of this paper.

System Reliability In addition to the information security requirements above the system should be reliable, secured from denial of service attacks etc.

Solution: We consider this a well-know problem with well-known solutions outside the scope of this paper.

4 Conceptual Design

The SPACE demonstrator was built using ICL's DAIS[2], an implementation of CORBA 2.1 [11]. CORBA was chosen to allow different countries or administrations to implement their own components, perhaps using several different CORBA implementations. A main goal when designing the new security features within this system has been to preserve this flexibility.

The CORBA security service [12] provides the necessary lower level security, such as secure data channel, in particular the SecIOP-protocol which enables components implemented using different ORBs to interoperate in a secure fashion. Although few implementations currently offer the necessary services we expect them to be available in the future.

Designing support for the other security requirements implies implementing several well-known mechanisms, such as digital signatures, public and private keys, digital certificates and symmetric

[2] Now available as LiveContent BROKER from PeerLogic [14].

key encryption. To follow up our choice of CORBA as a platform independent midleware we wanted to find a platform independent security environment. Although the current SPACE demonstrator is implemented in C++ we chose to base our design on the Java Cryptography Environment, JCE [20].

No substantial changes were needed in the SPACE design itself. The SPACE infrastructure already provides the necessary structure by defining the proper domains and restricting object interaction into specific patterns. In addition the object roles and their interaction patterns were design specifically to support this kind of extension.

4.1 Data Types and Structures

The design for the SPACE system [5] specifies highly structured data types and structures in IDL. To simplify the internal handling of these data structures we substituted the IDL data types with an XML-based data format.

For our purposes the most important data types are key sets, data packages and portfolios. These are all hierarchical structures comprised of IDL structures and sequences.

Key Sets and Valued Key Sets A key set is a structure describing the set of keys required to retrieve values for a set of data fields within a given CDS. A valued key set includes a key set as well as values for the required keys.

Data Packages and Citizen Input Data Packages A data package contains data retrieved from one CDS. It consists of a set of named data fields, their values and source, as well as a filter and a timestamp. The data source identifies reliably the real source of the data. The timestamp denotes the time the value was retrieved (and thus known to be valid) from the data source. The filter is used to specify valid targets for a specific data field, thus giving crude support for limited disclosure.

A citizen data package is special in that it contains data given by the citizen himself, as well as a profile of the citizen obtained using the dialogue mechanism [9]. This is treated as a separate data type because of the inherent difference in data retrieved from trusted systems and directly from a person.

Portfolios A portfolio contains all relevant information about a citizen and his moving situation. It consists of a set of data packages, a key set, a citizen input data package as well as a timestamp.

For our purposes we started out with a simple name based mapping from IDL to XML. To replace the key set IDL-type we defined a <KEYSET> tag, which includes <KEY>, <DATE> and <VALUE> tags similar to the IDL-based structure. The same was done for the other data types. Due to the more flexible nature of XML, some of our XML data types, like valued key sets, were much simpler than their IDL counterparts. The new security design replaces some aspects of the original design. This includes the data source and filter fields of data packages.

To facilitate the new security features, we needed new structure for handling signatures and encrypted data. We considered using existing security formats such as PKCS#7 (see for example [17]) but chose to stick to a much simpler XML-based format. In a complete design this simple format should be tailored to fit within the general framework of the surrounding security infrastructure.

Signed Item For signatures we defined a <SIGNED ITEM>. Each signed item consists of tree different blocks. <DATA> contains the actual data to be signed, such as a <KEYSET> or a <DATA PACKAGE>. <SIGNATURE> contains the signature of the <DATA> part in a simple hexadecimal format. <CERTIFICATE> contains the data needed to verify the signature. The design also requires that all signed items should have a <DATE> item within its <DATA> item. This is used to limit the lifetime of signed requests.

Encrypted Item For encrypted data we defined a <ENCRYPTED ITEM>. Each encrypted item consists of two different blocks. <DATA> contains the actual encrypted data. <KEY> contains the encrypted one-time key. Both items are stored in a simple hexadecimal format.

4.2 Required Components

An essential part of the SPACE concept is to build on existing technology as much as possible. This conceptual design for integrating security requirements requires new components in the implementation.

One vital component is one providing secure data channels. As mentioned earlier we believe that this can be provided by CORBA implementations providing security services and SecIOP or similar protocols.

Most of the solutions to the implicit requirements build on some sort of cryptography. This means that a cryptographic infrastructure with satisfactory functionality must be available. This also includes an infrastructure for key exchange, such as a certification authority providing a catalogue of digital certificates. This is not so much a technical as a political challenge. There is still some way to go before certified electronic documents are equal to their paper based precursors.

Another aspect of this infrastructure is a suitable data format. Our design is based on CORBA interfaces handling text structured using a simple XML syntax. The surrounding cryptographic infrastructure may demand a certain data format. Our design is not heavily dependent upon using XML for structuring text. Thus our data format could be substituted by another text based data format. With minor changes any byte-oriented data format could be utilised.

5 Prototype Implementation

We based our prototype upon Java 1.1 and the Cryptix [11] implementation of JCE [20]. Once suitable implementation of the JCE of Java 1.2 (or higher) becomes available the prototype should be easily portable. For handling our XML based data format we used a SAX [10] implementation.

To test out CORBA using SecIOP we did some brief experiments with ORBAsec SL2 [13]. We could also have used the current version of our original orb, which also offer the necessary services. We did not, however seek out other relevant implementation to test interoperability. Nor did we include the security service in the actual prototype.

In this prototype we only implemented a subset of the complete SPACE functionality. This subset represents all the functionality affected by the new security design and covers portfolio creation and portfolio delivery.

5.1 Portfolio Creation

In the original SPACE scenarios portfolio creation is done within the origination country. This was based on common requirements that applications for working permits, right of stay etc. should be

done in the originating country. However, there is nothing within the design that prohibits portfolio creation from the destination country or indeed a 3rd country.

The list below gives a brief description of the algorithm implemented in our prototype.

Client: Portfolio Preparation

1. Use dialogue mechanism [9] to establish citizen profile (simulated).
2. Use profile to retrieve set of required keys (simulated).
3. Provide values for key set (simulated).
4. Add current time to key set.
5. Create XML-document (or document part) for key set.
6. Create digest of key set.
7. Encrypt digest using citizen's private key.
8. Append encrypted digest as signature to key set.
9. Transfer signed key set (XML-document) to Master for data retrieval.

Server: Retrieval

10. Decrypt digest using citizen's public key.
11. Create digest of key set.
12. Compare new digest with decrypted digest (match=signature OK)
13. Use key set to retrieve data package.
14. Sign data package (same procedure as key set).
15. Return signed data packages to client for citizen review.

Client: Citizen Review and Portfolio Encryption

16. Data Packages displayed.
17. Citizen provides new data if necessary, placed in a separate citizen data package (simulated).
18. Citizen Data Package signed.
19. Portfolio encrypted using a one-time key.
20. One-time key encrypted using the citizen's private key.
21. Encrypted portfolio returned to server for storage.

Server: Portfolio Storage

22. Portfolio stored (persistent).
23. Portfolio id returned to citizen.

Client: Portfolio Id Signature

24. Portfolio id encrypted using citizen's private key.
25. Signed portfolio id returned to server for storage.

Server: Signed Portfolio Id Storage

26. Check portfolio id signature? (All citizen signatures should identify same citizen!)
27. Signed portfolio id stored together with portfolio.

5.2 Portfolio Delivery

Similar to portfolio creation, the original SPACE scenarios limited portfolio delivery to the destination country. Again there is nothing within the system itself which prohibits other solutions.

Client: Portfolio Retrieval

1. Portfolio id encrypted using citizen's private key.
2. Signed portfolio id send to server for portfolio transfer.

Server: Portfolio Retrieval

3. Portfolio id signature checked using citizen's public key.
4. Encrypted portfolio returned to client.

Client: Portfolio Decryption and Delivery

5. One-time key decrypted using citizen's private key.
6. Portfolio decrypted using one-time key.
7. Current time added to portfolio.
8. Portfolio signed using citizen's private key (citizen authorisation).
9. Portfolio returned to server for delivery.

Server: Portfolio Delivery

10. Data Package signatures checked.
11. Data Packages delivered to target citizen data systems[3].

5.3 Algorithm Usage

Our design does not specify specific algorithms to use. The Cryptix platform and the JCE allows for several different solutions and combinations. For our prototype we ran several different algorithms.

[3] The actual data delivery was outside the scope of the SPACE project. The current SPACE design is limited to describing three types of solutions; push, pull and paper. Push allows the SPACE system to actively enter data within a real CDS. Pull allows a real CDS to extract data from the SPACE system. Paper involves printing the portfolio and delivering the paper to the involved administrations.

Message Digest Our main choice of message digest algorithm was RIPEMD-160 [2]. We also tested MD5 [16] and MD4 [15].

Public/private key For public and private key encryption we employed RSA.

Symmetric key For symmetric key encryption we used CAST5, IDEA and SAFER.

5.4 Complications and Alternative Solutions

Like every other computer system, the SPACE system is based on a simplified model of the real world. Sometimes the inherent imperfections within this model foster complications. Furthermore the real world itself may contain incompatible requirements from different sources, itself a source of complications.

A much voiced complaint against the SPACE system is that it is build around a much simpler process than what is currently required for moving between most countries. We have chosen our simple model for two important reasons. Firstly, this simple model shows how simple the moving process can be given harmonised legislation. The Nordic countries have had this simple process since through the Inter-Nordic Movement Form since 1947. Secondly, the generic design of the SPACE system can easily be extended to cater for more complicated processes, such as the need to apply for the right to settle in the destination country. We feel that our solution highlights the benefits of a simple process as well as the complexity inherent in less harmonised processes.

Another possible problem involves combining Limited Disclosure and rules and policies for data retrieval and data package consolidation. This is because the originating CDS, i.e. the responsible administration, limits disclosure while Expert/Master enforces rules and policies. If two CDS can deliver the same data item and one of them limits disclosure of this item while another do not, the expert or master level must choose which of these (or both) to include in the final portfolio. Since the CDS may in theory choose not to disclose particular data items to experts or masters, consolidation may not be possible. Again we feel that our design highlights the advantages of simple and harmonised legislation whilst allowing for more complicated situations.

Another complication is inherent in data validation. How do we establish a reliable correspondence between values for keys (citizen id within a CDS) and citizen's digital certificate (citizen id within public/private key scheme). Incorporating the necessary relationships in every real CDS, thus giving the real CDS the complete responsibility for handling data validation can solve this. The necessary relationships may also be included in the CDS objects. This means that in addition to limited meta-data, the CDS object must also contain real data for every object within the real CDS it encapsulates. This will lead to much more complex maintenance of the CDS objects. Another solution may be to include the necessary relationships within a digital certificate or a TTP service. This is a simple and elegant solution for countries employing a single citizen ID. It may be more complex for countries without a single ID.

Using digital signatures is another possible problem. Our design assumes that a reliable infrastructure exist for every moving citizen. So far the necessary keys and certificates are not available to everyone. Still, we feel that most countries are either evaluating or developing solutions that will furnish every citizen with the necessary digital ids. An intermediate solution may be to limit the relevant functionality to specific offices, where the digital signatures of competent persons signifies that the citizen has presented certified documentation to this person.

Using competent persons to certify data also solves the problem of credentials. A “credentials officer” or “office” certifies that the citizen has presented certified documents that establish the validity of the electronic data.

6 Conclusion

Our analysis of the security requirements in the SPACE project shows that available technology can be used to build a system satisfying all the given requirements. We have also established that the infrastructure needed to allow citizens to perform all the necessary steps themselves is not yet available. This situation may eventually be remedied. In the meantime competent persons may handle some of the steps, limiting the availability to specific locations.

Acknowledgements

The work described here was funded by NR. It is based upon results from the SPACE (Single Point of Access for Citizens of Europe) project [6], partially funded by EU's Telematics Applications Programme.

References

[1] Cryptix: *Cryptix – A cleanroom JCE implementation*. http://www.cryptix.org

[2] H. Dobbertin, A. Bosselaers, and B. Preneel: *RIPEMD-160, a strengthened version of RIPEMD.* In volume 1039 (Fast Software Encryption) of Lecture Notes in Computer Science, pages 71-82. Springer Verlag. 1996. ftp://ftp.esat.kuleuven.ac.be/pub/cosic/bosselae/ripemd/ripemd160.ps.gz

[3] EU directive 95/46/EC: *Data Protection*, Oktober 1995. http://www2.echo.lu/legal/en/dataprot/directiv/directiv.html

[4] Government Secure Intranet. http://www.ccta.gov.uk/services/gsi/gsibriefing.htm

[5] P. D. Holmes, A. Larsen, M. Gritzman, L. Lundsgaard, H. Skardhammar, and R. Pohjosmäki: *SPACE Client and Server Specification: Infrastructure Specification for the SPACE Technical Platform.* EU Project Report. SPACE Deliverable D902/D903 (Confidential), Norwegian Computing Center (NR), 1997. Also reprinted as NR Technical Note IMEDIA/06/97, 1997.

[6] P. D. Holmes, A. Larsen, S. Myrseth, and M. Gritzman: *SPACE: An Architecture for Coordinated Intra-European Assembly and Exchange of Citizen Data.* NR Note IMEDIA/01/98, Norwegian Computing Center (NR), 1998.

[7] IDA: *IDA Architecture Guidelines.* http://www.ispo.cec.be/ida/text/english/dissemination/idagltoc.htm

[8] INFOSEC: Security of Telecommunications and Information Systems. http://www.cordis.lu/infosec/src/ets.htm

[9] A. Larsen and P. D. Holmes: *An Architecture for Unified Dialogue in Distributed Object Systems.* In TOOLS 26, pages 244-258. IEEE, 1998. Also available as NR Note IMEDIA/01/98.

[10] D. Megginson: SAX: *The simple api for XML*, May 1998. http://www.megginson.com/SAX/index.html

[11] OMG: *The Common Object Request Broker: Architecture and Specification, Revision 2.1.* OMG, 1997.

[12] OMG: *CORBAservices Security Service v1.2 specification.* OMG, 1998.

[13] ORBAsec SL2. http://www.adiron.com/orbasecsl2.html

[14] PeerLogic: LiveContent Broker. http://www.peerlogic.com/products/products.html

[15] R. Rivest: *The MD4 message-digest algorithm.* RFC-1320, April 1992. http://www.rfc.net/get2.php3/rfc1320.txt

[16] R. Rivest: *The MD5 message-digest algorithm.* RFC-1321, April 1992. http://www.rfc.net/get2.php3/rfc1321.txt

[17] B. Schneier: *Applied Cryptography.* John Wiley and Sons, March 1995.

[19] SPACE D0204: *User requirements analysis report.* Confidential Project Report, 1996.

[20] Sun: *JCE – Java Cryptography Extension.* http://java.sun.com/products/jce/index.html