Object Security and Verification for Integrated Information and Computing Systems

Claus-Peter Rückemann
Leibniz Universität Hannover (LUH),
Westfälische Wilhelms-Universität Münster (WWU),
North-German Supercomputing Alliance (HLRN), Germany
Email: ruckema@uni-muenster.de

Birgit Frida Stefanie Gersbeck-Schierholz
Leibniz Universität Hannover (LUH),
Certification Authority University of Hannover (UH-CA),
Germany
Email: gersbeck@ca.uni-hannover.de

***Abstract*—This paper presents the results of the development of techniques for implementing communication for complex integrated information and computing systems based on Object Envelopes. This provides means for flexible information exchange, namely objects, in mission critical environments, based on verification methods and cryptography. It covers some challenges of collaborative implementation, legal, and security issues with these processes. A major task is integrating information systems with Distributed and High Performance Computing resources in natural sciences disciplines, like epidemiology information systems, for building integrated public / commercial information system components within the e-Society. The main focus of this paper is on trust in information and how modular system architectures can make use of Object Envelope techniques. It shows that by object envelopes and signing, future security enhanced information and computing systems can be created.**

Keywords–Information Systems; Computing Systems; Security; Verification; Distributed Systems; Public Key Cryptography; High Performance Computing; Object Management.

1. Introduction

Todays information and computing systems are facing challenges from complex environments and heterogeneous content. The associated problems mostly emerge as security and legal issues, resulting in shortcomings for international collaboration management [1]. Over the last years a long-term project, Geo Exploration and Information (GEXI) [2] for analysing national and international case studies, has examined chances to overcome the deficits.

This paper presents the results of these projects using a newly implemented form of envelope regarding content data security for digital objects, Object Envelopes (OEN), in use with integrated information and computing environments in a collaboration framework. These OEN have shown successful content centred solution for various cases integrating the sections High Performance Computing (HPC), Distributed Computing (DC) and services, and natural sciences.

There are numerous situations where the use of information within complex distributed information and computing system environments is subject to security issues and legal regulations, especially if the information is in any way sensitive, highly charged or must be highly reliable [1]. Todays high end resources lack in methods for secure job and object handling. Many environment contexts base of legally binding premises. The information handled within these systems is one of the crucial points of concerns. For "trust in information" and "trust in computing" situations a collaboration framework has been created and tested with various implementation scenarios. The use cases showed two groups of systems, reflected by collaboration matrices [3].

This paper is organised as follows. Section 2 presents the motivation and implementation scenario including the problems with common technology. Sections 3 describes the fundamental implementation architecture for the solution. Sections 4 and 5 explain the requirements for "trust in information" and object signing and verification. Section 6 shows the solution for integrated systems (OEN). Section 7 reports on the evaluation and Sections 8 and 9 summarise the lessons learned and conclusion and outlook on future work.

2. Motivation and implementation scenario

The information and computing system components make use of various technologies, IPC, sandboxing, embedded applications, browser plugins, remote execution, network protocols, computing interfaces as well as public and sensitive data. The major motivation is to create an architecture of system components based on secure, signed, and verified objects in order to press ahead with standardisation for content and object management and reducing complexity. Figure 1 shows some of the basic application scenarios.

There exists a number of scenarios showing how "trust in computing" and "trust in information" can more easily be achieved by reducing complexity for the partners in otherwise very complex systems. The screenshot shows examples of data objects that are subject to protection:

- vector data and multi-dimensional data,
- raster data (aerial, remote sensing, and photographic),
- primary and secondary spatial information,
- calculation, measurement, and processing results,
- meta data and interactive information,
- commercially provided or licensed data, ...

2.1. State of prominent technology

As an example let us take a look at a method for signing widely used Portable Document Format (PDF) files. Adobe

 ISBN: 978-1-61208-122-9

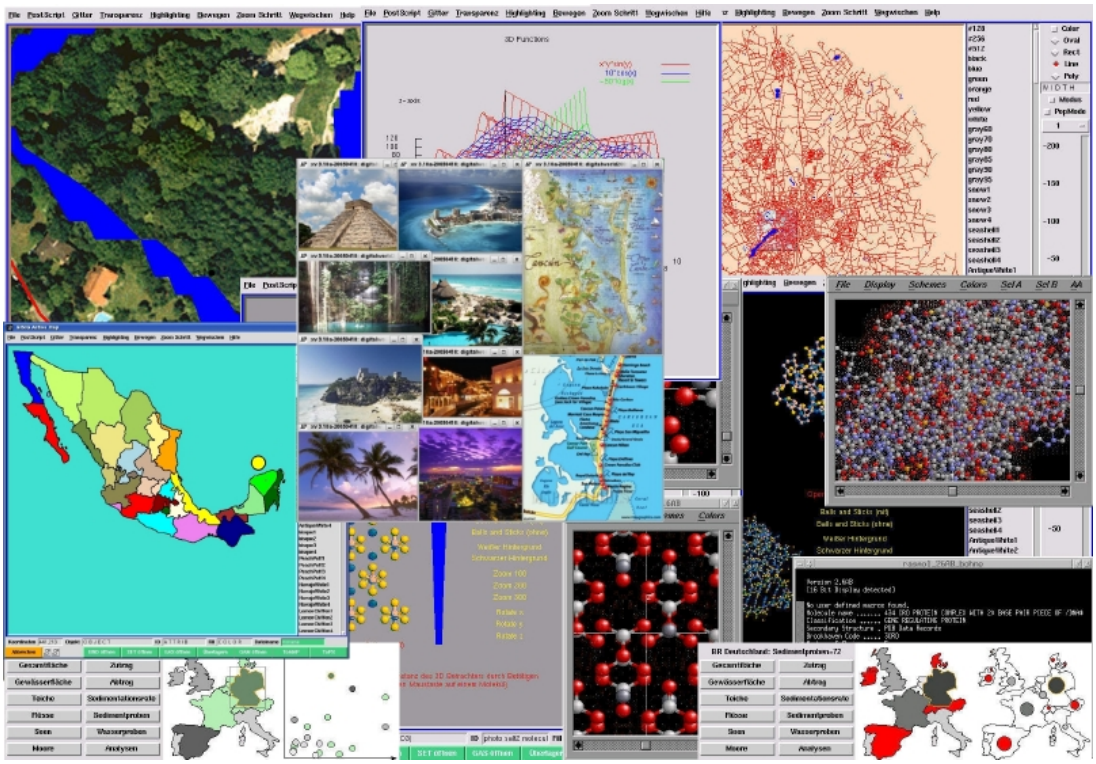

Figure 1. Application scenario from the GEXI case study showing object content scenarios.

uses the Public Key Cryptographic Standard (PKCS) for its proprietary products [4]. A byte stream is built from the PDF document and a digest is calculated. The hash value is encrypted with the private key (signature) and embedded as PKCS#7 into a copy of the document at a defined space. Besides the signed digest the embedded PKCS#7 object includes the full certificate of the signer. Meta data is hold in the signature dictionary. Verification is done using public key and certificate chain using the information from the PKCS#7 object. Possible revocation can be queried via the Online Certificate Status Protocol (OCSP) responder or a Certificate Revocation List (CRL).

2.2. Problems for complex use cases

Although the certificate processing conforms to the X.509-v3 certificate standard (RFC 3280) and standard signature objects are generated as PKCS#7 (RFC 2315), the solution is not appropriate for more complex information system situations. Even the use case portability of this practice does not guarantee for future application. In the case of other file formats the algorithms cannot be implemented due to different properties of “missing” features of these formats. For example a JPEG raster file cannot be signed the way a PDF file can be signed. In some cases the different file formats like JPG, PNG, TIFF or PDF might be embedded into PDF documents but this cannot be implemented for a complex system where hundred thousands of signed objects might have to be embedded into a single context, e.g., into a spatial view. This method based on PDF or other proprietary envelope has been recognised not flexible enough to serve as a generic solution for any complex multi-format information system. The main reasons are, that the algorithm:

- is not portable in between different file formats,
- does not respect meta-data of the information handled,
- does modify the original documents,
- is not intuitively extendable for information systems,
- and there is no flexible and open implementation available, and further on there are
- security issues associated with available products,
- the proprietary solution is not completely transparent,
- the XML has large overhead for huge object collections,
- huge transfer rates for large number of objects, and
- security issues with transfer actions to outer networks.

3. Fundamental implementation architecture

The fundamental architecture is based on a layered concept for the implementation and operation of information and computing systems [1]. The “trust in information” is twofold, regarding the content information domain and the utilisation information domain. It has been possible to transparently separate nearly all of the implementation aspects for the three columns and layers.

The case study showed that for the application within the integrated information and computing system the flexibility of content handling largely profits from object envelopes. Objects have to be signed with digital signature and timestamps of the originating authors and manipulation.

 ISBN: 978-1-61208-122-9

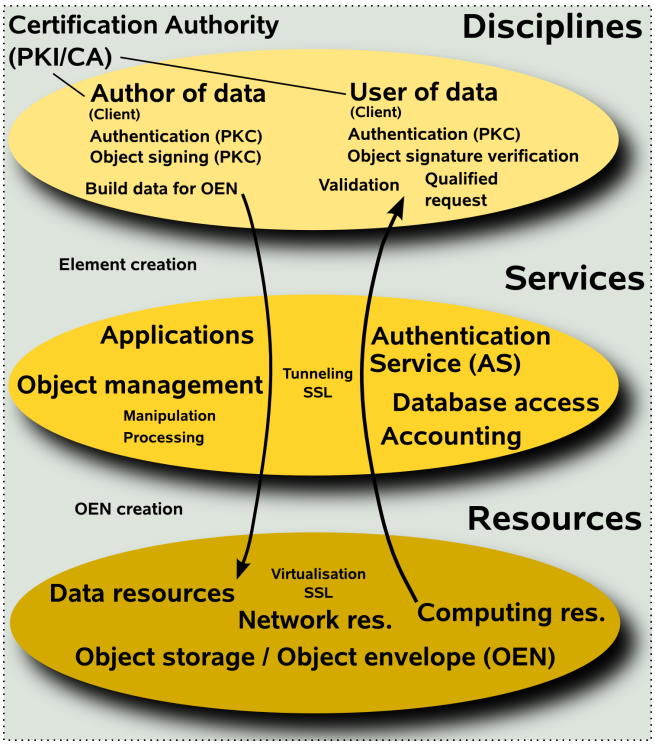

Figure 2. Object handling for integrated systems.

Figure 2 shows the object handling for integrated information and computing systems. The following sections give a detailed description on requirements and the processes depicted in this illustration. For most information systems used in mission critical application environments it is essential to assure accurate data objects all over the life-cycle of objects, thus guaranteeing "trust in information". Cryptographic techniques specified as public key cryptography in Public Key Infrastructure (PKI) environments [5] provide a framework for addressing important security considerations of authentication and data integrity. In this regard the authority (CA) signs the public user keys in order to maintain the integrity of the public key, expiration information and other important information contained within the user Public Key Certificates (PKC).

4. Requirements for trust in information

Digital signature capabilities allow object authors to set up a secure signing environment and allow the consumer of the data object to validate the object concerning integrity and authentication of the signer. The following passages describe the certificate requirement for trust in information.

4.1. Object cryptography

Asymmetric public key cryptography is based on the use of public / private key pairs [6]. A public key is typically disseminated in the form of a certificate, whereas a private key is a separate and distinct data structure always protected from unauthorised disclosure in transit, use and storage. The keys of a key pair are different, but related according to the circumstances that one has to decrypt what the other encrypts. Given an encryption key it is computationally not feasible to determine the decryption key and vice versa.

4.2. Object PKI

A PKI provides a trusted and authenticated key distribution infrastructure [7], [8], [9]. Its primary purpose is to strongly authenticate the parties communicating with each other, though the use of digital signatures, where the CA is an independent authority that issues PKC for binding the identity of a user to a public key by means of CA's digital signature. Furthermore, the CA will record and track the issued PKC and will schedule expiry dates for certificates and ensure certificates are revoked. There exist solutions for special use, like the PKCS. These solutions comprise fundamental definitions for special data structures and algorithms, providing the base of common PKI implementations. They define the syntax/format for a digital signature [10] and provide means for distributing certificates and revocation lists. A common trust model in a PKI is a strict hierarchy of CA institutions where all entities in the hierarchy trust the single root CA. The root is the starting point for trust. A certification hierarchy forms the certification path (chain of trust), from the certificate back to the root CA. To verify the trustworthiness of user certificates signed by a CA, the certificate chain of trust will be built by mapping the issuer name of the subordinate certificate to the subject name of the certificate higher up the chain and verifying it is signed with a valid signature, that has not expired. The Object Envelope (OEN) is able to describe any form of object PKI data.

4.3. Meta data

Various meta data is necessary to describe the signed object data. For chronology as well as for plausibility the security of the time and data association is important. Integrated system components as well as interested parties must be able to use these meta data as well as for example must be provided with means to verify that the time stamps associated with an object are authentic and hold integrity. Trusted time stamp authorities are required for this service. This may be a function of the CA, respective a dedicated time server service. The Object Envelope (OEN) is able to describe any form of embedded or referred meta data.

5. Object signing and verification

The following workflow considers the distributed implementation of the respective system components (Figure 2) within the layers.

5.1. The signing process

The following sequence describes the signing process for the implementation in operative context (case study on environmental sciences / epidemiology). The signing

 ISBN: 978-1-61208-122-9

process consists of single operation actions requiring some prerequisites:

- **Disciplines layer:** The author of data objects, the originator, e.g., co-worker of a human health organisation, signs the created objects, e.g., disease case numbers, with his private key and according meta data.

The signing requirements for this process are:

- Asymmetric key pair / PKC.
- PKI-enabled application. A special client for communicating with the information system services is desirable.

During signing procedure, the data object is digested with a hash algorithm and then the hash value is encrypted with the signer's private key. If the object changes, the message digest changes. Though, a message digest is simply a unique number created at signing time that identifies the object data that was signed. Containing this information, a signature element for the OEN is generated, including the signer's public key, data content, CA Certificates, and element meta data. The signature object elements are passed on to the service layer, including the object data. Within the disciplines layer object signing requires a client, able to handle the services that the PKI has enabled. Specifically, encryption / decryption and digital signature generation is requested by OEN envelopes. In addition, the services and client software must be able to access the data and key / certificate life-cycle management functions. Software provided with those features, is said to be PKI-enabled. Widely used applications are already PKI-enabled, like Web-browsers and more popular e-mail clients and electronic forms packages. For future integrated information and computing components, PKI support will be an essential feature.

- **Services layer:** Objects are processed by the services.
- **Resources layer:** Processed objects are stored to the distributed storage.

5.2. The verification process

The following sequence describes the verification process for the implementation in operative context (case study on environmental sciences / epidemiology).

- **Disciplines layer:** An user, e.g., a member of a research team at an university, requests objects.
- **Services layer:** The user is authenticating at the authentication service (AS). The AS requests the objects or service operation.
- **Resources layer:** Objects are collected from the distributed resources.
- **Services layer:** Objects are calculated, validated, accounted, and provisioned via services for the user client.
- **Disciplines layer:** Consumer's client application retrieves the signed data objects and performs the desired validation and verification procedure.

The requirements for the verification process regarding PKI and object infrastructure are:

- PKI-enabled application, a special user client interface is necessary for using the information system services.
- Object Envelope including the signed object, containing signer's PKC and CA certificates.
- OCSP responder or compatible revocation system.

To validate a signature, the consumer's software client first retrieves the author's certificate from the OEN and generates a digest hash of the document using the same hash algorithm the signer used (for example SHA-256). Then the hash value encrypted with the author's private key during signing process is decrypted using the author's public key, if successful, signer's authentication is valid (verifying signer's identity). Then the decrypted hash value is compared to the even locally generated hash value. If they are identical, the integrity check is valid (verifying object's integrity). Furthermore, as well as the signing process, the verification workflow requires a PKI enabled client on the disciplines layer. In particular, encryption/decryption and digital signature verification must be supported. To establish certificate trust, the application builds and validates the certificate chain as described above. To facilitate the verification workflow the CA certificates are embedded in the OEN. After the chain is validated, and the trust anchor is found from the certificate trust list, the client determines whether any of the certificates in the chain have been revoked. The client software looks for a valid revocation response like an OCSP response or an embedded CRL reference for the OEN object.

6. Solution for use with integrated systems

What we needed, was not only a signature standard and an envelope technology but a generic extensible concept for information and computing system components. The benefits for development, configuration, and use of complex information and computing systems are:

- no overhead, minimising communication,
- transparent handling,
- no proprietary algorithms.

Future objectives, combined with client components are:

- channels for limiting communication traffic,
- qualified signature services and accounting,
- using signed objects without verification,
- verify signed objects on demand.

The tests done for proof of concept have been in development stage. A more suitable solution has now been created on a generic envelope base. The current solution is based on OEN files (extension used is `.oen`) containing element structures for handling and embedding data and information. Listing 1 shows a small example for a generic OEN file.

```
1 <ObjectEnvelope><!-- ObjectEnvelope (OEN)-->
2 <Object>
3 <Filename>GIS_Case_Study_20090804.jpg</Filename>
4 <Md5sum>...</Md5sum>
5 <Shalsum>...</Shalsum>
6 <DateCreated>2010-08-01:221114</DateCreated>
7 <DateModified>2010-08-01:222029</DateModified>
```

```
<ID>...</ID><CertificateID>...</CertificateID>
<Signature>...</Signature>
<Content><ContentData>...</ContentData></Content>
</Object>
</ObjectEnvelope>
```

Listing 1. Example for an Object Envelope (OEN).

An end-user public client application may be implemented via a browser plugin, based on appropriate services. With OEN instructions embedded in envelopes, for example as XML-based element structure representation, content can be handled as content-stream or as content-reference. The way this will have to be implemented for different use cases depends on the situation, and in many cases on the size and number of data objects. Listing 2 shows a small example for an OEN file using a content DataReference.

```
<ObjectEnvelope><!-- ObjectEnvelope (OEN)-->
<Object>
<Filename>GIS_Case_Study_20090804.jpg</Filename>
<Md5sum>...</Md5sum>
<Sha1sum>...</Sha1sum>
<DateCreated>2010-08-01:221114</DateCreated>
<DateModified>2010-08-01:222029</DateModified>
<ID>...</ID><CertificateID>...</CertificateID>
<Signature>...</Signature>
<Content><DataReference>https://doi...</DataReference><
 /Content>
</Object>
</ObjectEnvelope>
```

Listing 2. OEN referencing signed data.

One benefit of content-reference with high performant distributed or multicore resources is that references can be processed in parallel on these architectures. The number of physical parallel resources and the transfer capacities inside the network are limiting factors. Whereas the XML signature standard (RFC 2807) [11] proclaims the feasibility that XML signatures can be applied to arbitrary digital content via indirections, this only answers the problem of huge data regarding quantity or size theoretically. For practical use in real-life use cases one would prefer solutions matching to the situation, being flexible, transparent, open, portable, and using general modular components. For qualified requests signatures and signature groups can be verified. For non-qualified requests signatures can be ignored. All OEN can be embedded into existing information and computing system components. Listing 3 shows a small example of an OEN embedded into a GISIG Active Source component.

```
#BCMT-------------------------------------------------
###EN \gisigsnip{Object Data: Country Mexico}
#ECMT-------------------------------------------------
proc create_country_mexico {} {
global w
#  Sonora
$w create polygon 0.938583i 0.354331i 2.055118i ...
#BCMT-------------------------------------------------
###EN \gisigsnip{Object Data: Object Envelope (OEN)}
#ECMT-------------------------------------------------
#BOEN <ObjectEnvelope>
##OEN <Object>
##OEN <Filename>mexico_site_name_tulum_temple.jpg</
 Filename>
##OEN <Md5sum>251b443901d87a28f83f8026a1ac9191
 *mexico_site_name_tulum_temple.jpg</Md5sum>
##OEN <Sha1sum>f0eb9d21cfe2c9855c033be5c8ad77710356c1eb
  *mexico_site_name_tulum_temple.jpg</Sha1sum>
##OEN <DateCreated>2010-08-01:221114</DateCreated>
##OEN <DateModified>2010-08-01:222029</DateModified>
##OEN <ID>...</ID><CertificateID>...</CertificateID>
##OEN <Signature>...</Signature>
##OEN <Content><ContentDataReference>http://.../
 mexico_site_name_tulum_temple.jpg</ContentReference></
 Content>
##OEN </Object>
#EOEN </ObjectEnvelope>
...
proc create_country_mexico_autoevents {} {
global w
$w bind legend_infopoint <Any-Enter> {set killatleave [
 exec ./mexico_legend_infopoint_viewall.sh $op_parallel
  ] }
$w bind legend_infopoint <Any-Leave> {exec ./
 mexico_legend_infopoint_kaxv.sh }
$w bind tulum <Any-Enter> {set killatleave [exec
 $appl_image_viewer -geometry +800+400 ./
 mexico_site_name_tulum_temple.jpg $op_parallel ] }
$w bind tulum <Any-Leave> {exec kill -9 $killatleave }
}
...
```

Listing 3. OEN embedded with Active Source.

Additionally, algorithms like check sums (MD5, SHA or others) or encryption for content or meta data can be handled very flexible. Common modules for these algorithms are md5sum, sha1sum, sha512sum, gpg, and many other tools supporting functions and features like authentication, integrity, reliability, confidentiality, and authorisation.

7. Evaluation

The primary benefits of the presented solution using OEN with signed objects are that the algorithm is

- portable in between different object and file formats.
- It respects meta-data for the objects.
- Original documents can stay unmodified.
- The solution is most transparent, extendable, flexible, and scalable, for security aspects and modularisation.
- Guaranteed data integrity and authentication derived from the cryptographic strength of current asymmetric algorithms and digital signature processes.
- Flexible meta data association for any object and data type, including check sums and time stamps.

Main drawbacks are:

- Requirements for use outside the case studies: Interoperability between multiple PKIs, a global cryptosystem on the internet (Global PKI), special PKI-enabled software clients to generate, store and manage certificates and associated data is not already implemented.
- Risks: Lost, destroyed, or compromised private keys and loss of primary verification for keyed object data.
- Inconveniences: Authors have to register at a CA and request digital certificates.

With modern information and computing systems object management is a major challenge for software and hardware infrastructure. Resulting from the case studies with information systems and computing resources, signed

 ISBN: 978-1-61208-122-9

objects embedded in OEN can provide a flexible solution. PKI technology offers means to attest, identify and manage the exchange of encryption keys and secure transmission between parties. Although PKI technology has not already been broad-based adopted by public and private organisations as it mostly only is supported for optional use with single processes like e-mail communication, it is a valuable support for creating a secure object life-cycle for mission critical high end information systems.

8. Lessons learned

The OEN solution has been found to be a flexible and extensible solution for creating a secure environment for integrated information and computing systems. The case study showed that nearly any data structure can be handled with object envelopes in embedded or referred use. Signatures, check sums, and meta data can be used in various ways for the purpose of the information and computing system. Key loss is not critical for the data itself. Service providers and users can ensure the integrity and re-keying. For scalability, e.g., for different object sizes from some bytes up to several Giga-bytes, it is preferable to have more than one fixed method. Therefore embedded and referred data has to be supported. This leads to the conclusion that in the future of integrated information and computing systems we will need to create means of securely submitting modular application components into the services pipeline.

9. Conclusion and future work

The security and verification of information content is an essential part of the challenge to build future integrated information and computing systems. Object Envelope (OEN) techniques can help to establish a flexible and portable way for using content data. Further on with implementation and legal issues, the security aspect are on the rise for any complex system. Even though PKI technology offers means to attest, identify, manage the exchange of encryption keys and secure transmission between parties, there has not been broad-based adoption of PKI technology by public and private organisation. After all, a significant number of countries recognise digital signatures as legally binding. In case of security enhanced integrated information and computing system components object signing provides a robust solution to facilitate "trust in information" and to overall support "trust in computing". In order to put this implementation into international public practice there is a need for future PKI development and deployment offering a global public key cryptosystem for the Future Internet. This work showed that it is possible to bring complex information and computing systems to life, being able to create interfaces that can also be interfaces between the logical columns and interest groups.

Acknowledgements

We are grateful to all national and international academic and industry partners in the GEXI cooperations for the innovative constructive work and to the colleagues at the LUH, IRI, HLRN, WWU and the participants of the postgraduate European Legal Informatics Study Programme (EULISP) for prolific discussion of scientific, legal, and technical aspects.

References

[1] C.-P. Rückemann, "Legal Issues Regarding Distributed and High Performance Computing in Geosciences and Exploration," in *Proceedings of the International Conference on Digital Society (ICDS 2010), The International Conference on Technical and Legal Aspects of the e-Society (CYBERLAWS 2010), February 10–16, 2010, St. Maarten, Netherlands Antilles / DigitalWorld 2010, International Academy, Research, and Industry Association (IARIA)*. IEEE Computer Society Press, IEEE Xplore Digital Library, 2010, pp. 339–344, Berntzen, L., Bodendorf, F., Lawrence, E., Perry, M., Smedberg, Å. (eds.), ISBN: 978-0-7695-3953-9, URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5432414 (PDF) [accessed: 2010-03-28], (Best Paper Award).

[2] "Geo Exploration and Information (GEXI)," 1996, 1999, 2010, URL: http://www.user.uni-hannover.de/cpr/x/rprojs/en/index.html#GEXI (Information) [accessed: 2010-05-02].

[3] C.-P. Rückemann, "Legal Base for High End Computing and Information System Collaboration and Security," *International Journal on Advances in Security*, vol. 3, no. 3&4, 2010, (to appear) ISSN: 1942-2636, URL: http://www.iariajournals.org/security/ [acc.: 2010-08-18].

[4] "Acrobat Digital Signatures, Digital Signature User Guide for Acrobat 9.0 and Adobe Reader 9.0," Adobe, 2010, URL: http://www.adobe.com/devnet/acrobat/ [accessed: 2010-08-10].

[5] "ITU-T Recommendation X.509 ISO/IEC 9594-8, The Directory: Authentication Framework," 2000, URL: http://wwwitu.int/itu-t/recommendations [accessed: 2010-05-29].

[6] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, 1976, pp. 644–654, URL: http://www.cs.purdue.edu/homes/ninghui/courses/Fall04/lectures/diffie-hellman.pdf [acc.: 2010-09-18].

[7] C. Adams and S. Lloyd, *Understanding PKI: Concept, Standards, and Deployment Considerations*, 2nd ed. Addison Wesley Longman Publishing Co., Inc. Boston, MA, USA, 2002, 322 pages, ISBN: 0672323915.

[8] A. Nash, W. Duane, C. Joseph, and D. Brink, *PKI: Implementing and Managing E-Security*. McGraw-Hill OsborneMedia, New York, USA, 2001, 513 pages, ISBN: 0072131233.

[9] B. F. S. Gersbeck-Schierholz, "Trustworthy Communication by Means of Public Key Cryptography," 2010, URL: http://www.rrzn.uni-hannover.de/fileadmin/it_sicherheit/pdf/pki2010_gersbeck.pdf [accessed: 2010-05-29].

[10] B. Kalinski, "PKCS#7: Cryptographic Message Syntax Version 1.5," March 1998, Internet Request for Comments: RFC 2315, RSA Laboratories, East (ed.).

[11] "XML-Signature standard," W3, 2010, URL: http://www.w3.org/TR/xmldsig-core/ [accessed: 2010-08-10].