

Interoperable and Flexible Digital Signatures for E-Government and E-Commerce

Harald Baier and Markus Ruppert*
Darmstadt Centre of IT Security and FlexSecure Ltd.,
Hochschulstr. 10, D-64289 Darmstadt

May 13th 2004

Abstract

The paper at hand presents the concept of a flexible and interoperable public key infrastructure, the so called FlexiPKI. We show how this concept and its realization enables long term security in e-government and e-commerce. As a proof of concept, we describe the implementation of the FlexiPKI concept at the root certification authority in Germany.

Keywords: cryptography, digital signatures, e-government, public key infrastructure, Java Cryptography Architecture

1 Introduction

As of today public key infrastructures become more and more popular because of their fundamental role to achieve security goals like authenticity, integrity, non-repudiation, and confidentiality in open user groups. The security of public key infrastructures mainly depends on the security of digital signatures.

The main purpose of a public key infrastructure (PKI) is to bind a public key to an entity of the infrastructure. The binding is put into practice by a certification authority (CA), which plays the role of a trusted third party. The CA digitally signs a data structure, which contains besides some other data the name of the entity and the corresponding public key. The data

*This paper was written while both authors were working within SicAri, a project funded by the German Ministry of Education and Research

structure together with the CA signature is called a *certificate*. Once, the CA signatures become invalid, the whole infrastructure collapses. All online applications like e-commerce, online banking, or e-government are no longer secure.

In 2001, based on the European Digital Signature Directive [1], European countries established a new German Digital Signature Act [2] to support the further development of e-government and e-commerce. In Germany, the Regulatory Authority for Telecommunications and Posts (RegTP) operates the root certification authority (RCA) in conformance with the Digital Signature Act. Currently, the RegTP is installing a new RCA-software. In order to reduce the total cost of ownership and to ensure long-time security, the main design criteria of the software are interoperability and flexibility of the underlying cryptographic primitive.

The contribution of the paper at hand is as follows: First, in Section 2 we describe in detail general requirements for high security digital signatures as needed in the context of e-government and e-commerce. We are not aware of any comparable catalogue of requirements. Next, in Section 3 we describe our concept of an interoperable and flexible public key infrastructure. It is interoperable as our implementation respects all common certificate profiles (such as PKIX [3] and ISIS-MTT [4]). It is flexible for two reasons. First, it is based on the Java Cryptography Architecture (JCA), a Java based framework for cryptographic algorithms. The JCA only delivers the cryptographic interfaces, not their implementation. The implementation is done within a cryptographic service provider (CSP). Second, our approach makes use of the FlexiProvider ([5]), which is a CSP for the JCA. Currently, the FlexiProvider comprises independent digital signature schemes like RSA, elliptic curve based signatures, and number field cryptographic schemes. Therefore, our PKI is flexible with respect to the underlying mathematical problem. This concept is called the FlexiPKI ([6]). We show that our concept of the FlexiPKI meets the requirements of Section 2.

As a proof of concept we show in Section 4, that the new RCA will make use of the concept of FlexiPKI. We therefore consider our concept of FlexiPKI as a best practice solution. The RCA will be able to support RSA, DSA, and the elliptic curve digital signature algorithms. We conclude that our approach is superior to common approaches in the past. Finally, in Section 5 we discuss future developments of our concept.

2 Requirements on digital signatures for E-Government Use

In this section we describe the requirements, which we impose on digital signatures for use in applications for e-government or e-commerce. First of all, we have to stress that non-repudiation plays a crucial role to establish digital signatures. Non-repudiation of a digital signature has to be provable for a long time. For instance, the German Digital Signature Act enforces qualified certificates to be verifiable for over 30 years ([2]). In Section 2.1 we therefore present requirements to ensure long term security.

In addition, e-government or e-commerce applications will be in use by most of the population. A PKI in this context has to operate for millions of people. For example, the CA must produce millions of certificates or must answer millions of requests about the validity of certificates in a relatively short time. We discuss in Section 2.2 the requirements for such a scalable PKI.

Finally, certificates should be used by a broad variety of applications. The certificates thus have to be interoperable. In Section 2.3 we describe common profiles to ensure interoperability.

2.1 Long Term Security

In order to ensure long term security, we have to take various aspects into account. In this section we present the most important ones. First, the underlying cryptographic algorithm has to be secure for a long time. As there is currently no such algorithm available, long term security is achieved by the ability to use different cryptographic algorithms. We present our approach to this problem in Section 2.1.1. Second, we have to make use of a secure hardware, which actually generates a digital signature. However, this hardware often depends on the cryptographic algorithm in use. Once we switch from one algorithm to another one, we have to switch the hardware, too. This results in growing costs. We turn to the problem of appropriate signature hardware in Section 2.1.2. Finally, even if a signature is valid from a mathematical point of view, it is not clear if the signature will be accepted. The decision of acceptance is dependent on the underlying validity model. Different validity models have been proposed in the past. We discuss them in Section 2.1.3 and explain our choice.

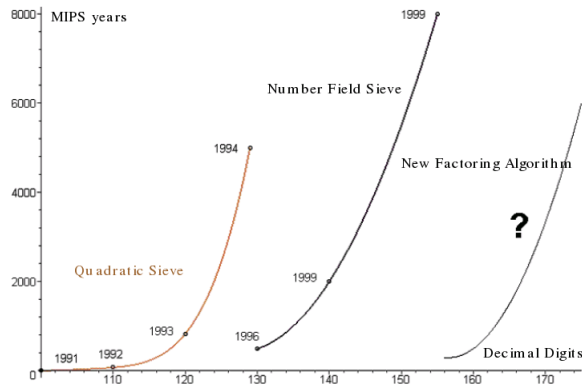


Figure 1: Time to factor an RSA-modulus

2.1.1 Cryptographic Algorithms

We discuss our approach to ensure long term security at the level of the cryptographic algorithm. Each public key algorithm, which is appropriate for use in practice, relies on a difficult mathematical problem. The most popular algorithm is the RSA algorithm. As of today RSA is attacked by solving the factoring problem. However, we do not know how difficult the factoring problem actually is. In the past, different factoring methods have been found, which resulted in a significant speed up in the time to factor an RSA modulus, respectively. This is shown in Figure 1.

As a result, we have to be able to use different public key algorithms. Good candidates are the Digital Signature Algorithm (DSA, [7]) and the Elliptic Curve Digital Signature Algorithm (ECDSA, [8]). Analogously we have to consider different hash functions.

However, in order to be usable, these algorithms have to be efficiently implemented. Our cryptographic library, the open source FlexiProvider ([5]), fulfills these requirements. When we proclaim the paradigm of flexibility of the cryptographic algorithm we mean the ability to flexibly choose the cryptographic algorithm.

2.1.2 Signature Hardware

The generation of qualified digital signatures enforces the use of special hardware. This hardware has to be evaluated on the basis of ITSEC ([9]) or the Common Criteria ([10]). However, as of today, only smart cards

are successfully evaluated. Thus only hardware with restricted resources is available. For this reason, smart cards are often developed for one special use case. A good example is the use of RSA with at most 1024 bit modulus size. This contradicts the paradigm of flexibility as explained in Section 2.1.1.

We therefore propose to be able to easily integrate state of the art cryptographic hardware. Then the total costs of the PKI will be reduced. This becomes more evident once a cryptographic algorithm is broken.

2.1.3 Validity Models

We now turn to the problem of an appropriate validity model. This means the following: The validation of a digital signature first involves a mathematical operation. The result is either **true** or **false**. The validation makes use of the public key of the signer. If the mathematical validation outputs **false**, the digital signature will not be accepted. However, even if the output is **true**, the signature may be rejected. The acceptance of the signature depends on validity of the corresponding public key. Its validity is defined within the underlying validity model.

In all, three different validity models have been proposed: The shell model, the hybrid model, and the chain model. Before we turn to these validity models, we first have to explain the notation of a certificate chain. A PKI has one common trusted anchor: the certificate of the root certification authority. Often a root certification authority issues certificates to further certification authorities. Each such CA is called a second step CA. A second step CA may issue certificates to end users or to further CAs, that is to third step CAs. A certificate chain is a chain from an end user certificate to the common trusted root CA certificate.

The shell model requires *all* certificates in the certificate chain to be valid at the signature *validation* time. However, certificates are valid for at most five years. Once one of the certificates in the certificate chain becomes invalid, all signatures will be rejected. Thus in the context of long term validation, this is not a good choice.

Next, we turn to the hybrid model. It is called the modified shell model, too. The hybrid model requires *all* certificates in the certificate chain to be valid at the signature *generation* time. Thus a valid signature stays valid forever. In order to achieve long term validation, this seems to be an acceptable choice to us. However, we prefer the following chain model.

In the chain model a signature is valid, if the signer possesses a valid certificate at signature generation time. In the context of a signature gen-

erated by an end user, this means that the end user certificate has to be valid at the signature generation time. In the context of a signature for a certificate this means, that the issuer's certificate has to be valid at the certificate production time. To us this seems to be the best choice.

The answer, if a public key is valid or not at a fixed time, is given by the directory services or the revocation services of a PKI. The validity model is part of the certification policies. All models are in use in practice. Thus a flexible PKI should be able to switch from one validity model to another one.

2.2 Bulk Generation of Digital Signatures

E-government applications address the whole population. A PKI for e-government thus has to operate for millions of people. The CA, for instance, has to be able to generate millions of certificates in a relatively short time, say some weeks. However, certificate generation is not the most requested service of a PKI. The directory and revocation services must answer millions of queries about the validity of a certificate, again in a relatively short time. Often, the Online Certificate Status Protocol (OCSP, [11]) is used for status queries. The OCSP answers must be signed digitally. Thus a scalable PKI has to be able to address this feature.

In practice, bulk generation of qualified digital signatures is a rather difficult problem for two reasons. First, the German Digital Signature Act only allows to issue qualified certificates to natural persons. Second, it understands a qualified signature as a declaration of intention. As a consequence it is a strong demand that the signer actually sees what he is going to sign before his approval. At a first glance this contradicts the requirements for bulk generation of digital signatures. However, workarounds to solve this problem exist ([12]).

2.3 Interoperability

A certificate is used within a PKI application to ensure that a given public key actually belongs to its supposed owner. As of today, a lot of different use cases are known. In general, different use cases require different certificate fields (the so-called extension fields). Extension fields may be defined by everybody. In this context it is very important that the PKI application is able to interpret all extension fields. If this holds for the certificates generated by a CA, then we call the certificates *interoperable*.

In the past, different working groups have written down various profiles

for qualified certificates. The main task of a certificate profile is to ensure interoperability. The most important profiles are the PKIX profile ([3]) and the ISIS-MTT SigG-Profile ([4]). We prefer ISIS-MTT for the following reasons.

First of all, the ISIS-MTT SigG-Profile addresses all technical requirements of the *German Digital Signature Act* (SigG) and the *Ordinance on Digital Signatures* (SigV). Although this is closely related to SigG, ISIS-MTT will be adopted by most European countries. Second, this standard profiles in detail the IETF standards (for example the RFCs of the PKIX and S/MIME working groups). Thus ISIS-MTT is a standard optimization of PKIX for practical use. Third, in contrast to PKIX it is in conformance with the technical specifications of the European Telecommunications Standards Institute([13]). Finally, it restricts the possible implementation alternatives in order to promote interoperability as well as to reduce the costs of implementation and conformity tests.

3 The concept of FlexiPKI

In this section we present our concept of the flexible Public Key Infrastructure, which we abbreviate as FlexiPKI. We show that our FlexiPKI meets the requirements of Section 2.

The main parts of the FlexiPKI are the FlexiProvider and the trustcenter software FlexiTrust. As stated above the FlexiProvider ([5]) implements various different cryptographic algorithms. Sample schemes are symmetric block ciphers (e.g. AES, 3-DES, Twofish), hash functions (e.g. SHA-1, RIPEMD-160), and digital signature algorithms (e.g. RSA, DSA, ECDSA). A large variety of cryptographic schemes ensures the long term security on the level of the cryptographic algorithm as required in Section 2.1.1.

FlexiTrust is the trustcenter software of the FlexiPKI. Its design and workflow arises from practical experience. We describe it in Section 3.1. The core task of a PKI is the generation of certificates. We come to this area in Section 3.2.

3.1 Design and Workflow

We shortly describe the design and workflow of FlexiTrust. FlexiTrust establishes at least three main components for certificate application processing. It consists of a number of modules to fit perfectly to almost every existing environment. The modules are a result of various projects over the last few

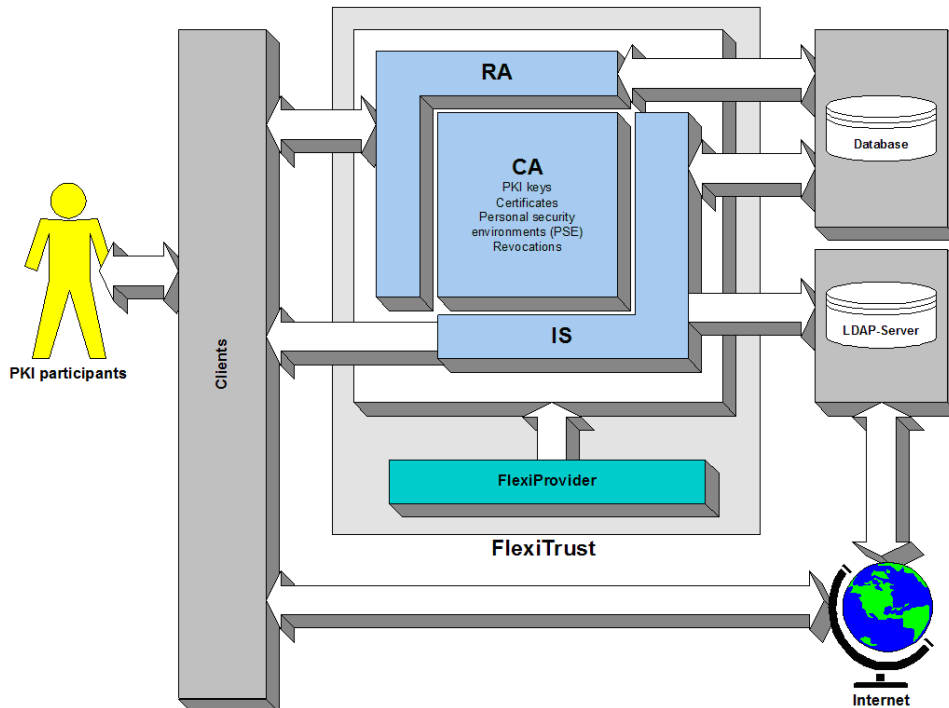


Figure 2: FlexiTrust core design

years. The demands on the certification processes in different application contexts have been put into practice, respectively.

The three core components of FlexiTrust are:

- Registration authority (RA).
- Certification authority (CA).
- Infrastructure services (IS).

The core components and their relationships are given in Figure 2. Let us first describe the registration module. Its main task is to process the registration and application requests. Import of registration data is put into practice using strong authentication. The RA component is scalable: It offers both individual request processing and bulk processing using databases or XML interfaces. The bulk processing is important in context of bulk rollout of certificates as required in Section 2.2. Processing and access rules can individually be defined for all request types.

We next turn to the CA module. It supports both signing with smartcards and hardware security modules. The CA module is compatible with smartcards and hardware security modules of various suppliers and thus meets the requirement of Section 2.1.2. Furthermore, the CA module allows the two common ways of smartcard personalization. First, it allows both key *and* certificate generation at the certification authority. In this case the module allows to load additional applications on the smartcard. Second, the CA module supports signature token delegation, that is the key generation takes place at the smartcard supplier on the smartcard itself. The CA only handles the certification request (PKCS#10). In this case it is important to use strong authentication (for instance using PKI methods or shared secrets). Last but not least, the visualization of the data to be signed and an interactive signature confirmation are implemented.

Finally, we describe the IS module. Roughly speaking the IS module is responsible for the whole PKI workflow besides registration and certification, that is it handles the whole certificate life cycle after its generation. The IS module handles different secure interfaces to a number of well known PKI services. For instance, the IS module may establish a secure connection to the directory or status information services. These services are not necessarily part of the core components of FlexiTrust, as for these directory services a lot of solutions are available. FlexiPKI supports all validity models presented in Section 2.1.3.

All certificate and workflow profiles are defined within XML structures. The profiles are therefore easy to configure and available for further processing. The whole trustcenter solution has the ability to independently host a number of trustcenter instances and hierarchies using the same hardware with exclusive roll based access rules and profiles. The FlexiPKI is therefore flexible and able to process bulk generation of digital signatures.

3.2 Certificate Generation

The requirements of the European Signature Directive ([1]) affects the workflow of the certificate generation process. In addition, the existence of special profiles for qualified certificates ensures interoperability as described in Section 2.3. The most important security requirements are:

1. Key pairs are unique.
2. The hardware for storage and usage of the private key has to be evaluated on basis of the Common Criteria ([10]) or ITSEC ([9]).

3. No copy out of this environment is allowed.
4. The private key token has to be handed over to the certificate owner in a secure manner. The owner gives a receipt.
5. Key and identity will be assigned for about 30 years and the assignment can be verified in the entire period. Thus certificates and all corresponding data have to be stored and be available for a long time. If necessary, this requires re-signatures and time stamping over the data.
6. Every qualified certificate identifies exactly one natural person.
7. A certificate is valid only, if the responsible trustcenter declares the certificate as valid.

The trustcenter software FlexiTrust meets these requirements. It is in conformance with the ISIS-MTT profile and thus guarantees interoperability.

We mention a common problem in the context of certificate generation. As mentioned above, qualified certificates are issued to a natural person. However, often this person does not sign himself, but he delegates the signature process to other persons. From a technical point of view it is not possible to assign such signatures to its signer. Thus a special workflow has to be established to ensure such an assignment.

Typically delegated signatures are used within trustcenter processes. The owner and responsible person for the root or CA keys will never use these keys himself. The delegation is then established by separation of functions.

4 Best Practice and Proof-of-Concept

In this section we describe, how we put into practice our concept of the FlexiPKI. To our mind, the project serves as a best practice example for an interoperable and flexible PKI. In addition, it is a best practice for the cooperation between a research group (Darmstadt University of Technology) and industry (FlexSecure Ltd.).

In the first quarter of 2003, the German Regulatory Authority for Telecommunications and Posts (RegTP) published a call for tender to replace the old root certification authority (RCA) system. T-Systems, a leading provider of information and communications technology services, and FlexSecure, a

spin off of Darmstadt University of Technology, submitted a joint offer. The key point of our proposal was the use of FlexiPKI and FlexiTrust within the new RCA instance. Our consortium won the contest.

The customer pointed out that the choice of FlexiTrust was well founded in its ability to establish fail safe concepts and to be updated to new security requirements with almost no effort to enable longterm security. In less than nine months the new solution was specified, adapted to the special needs of the RCA, evaluated and installed.

It was designed to carry out the requirements of the RegTP as well as the requirements of ISIS-MTT. The practical experience during the specification and implementation period ended up in redesigned parts of the ISIS-MTT profiles. For example, a new certificate extension **Validity Model**¹ was defined to be able to identify the underlying validity model without reading the certification policy.

An entire Common Criteria protection profile meeting the requirements of the German Digital Signature Act had to be specified during the evaluation process with respect to high security achievements as well as basic rules to establish delegated bulk signatures for OCSP services and signature renewal. As a result, the current version of FlexiTrust 3.0 Release 0347 is evaluated according to the Common Criteria, Evaluation Assurance Level (EAL) 3+ high. We point out that FlexiTrust is the only available trust-center software with such an evaluation.

The development of FlexiTrust started 1999 at Darmstadt University of Technology. The design of FlexiTrust was object oriented ([6]). Two main design criteria determined the development: First, the demand on best integration practice in existing environments. A fundamental basis for this demand is interoperability. Second, the paradigm of long term security based on the flexible use of cryptographic schemes.

5 Forecast

We mention future developments of our FlexiPKI concept. First, as soon as new qualified signature tokens like TCOS 3.0 will be available, FlexiTrust will be re-evaluated. The new evaluation enables signatures with different cryptographic algorithms. Again we point out that this feature is indispensable for the needs of e-government and e-commerce.

¹<http://www.informatik.tu-darmstadt.de/TI/Forschung/FlexiPKI/validitymodel/index.html>

Second, we integrate the concept of a fail safe PKI as proposed by Maseberg [14]. Roughly speaking, the fail safe concept is based on a redundant second PKI in the background. Once the operational PKI collapses, the second PKI will stand in to ensure security.

References

- [1] Directive of the European Parliament and of the Council on a Community framework for electronic signatures . Directive 1999/93/EC, 1999.
- [2] German Digital Signature Act: Gesetz über Rahmenbedingungen für elektronische Signaturen und zur Änderung weiterer Vorschriften, Bundesgesetzblatt Nr 22, 2001, S876ff.
- [3] PKIX: Internet X.509 Public Key Infrastructure Qualified Certificates Profile. RFC3039, 2001.
- [4] ISIS-MTT: Common ISIS-MailTrust Specifications for Interoperable PKI Applications, Optional Profile. SigG-Profile, 2002.
- [5] FlexiProvider, A Provider for the Java Cryptography Architecture. <http://www.flexiprovider.de>, 2004.
- [6] Buchmann, J., Ruppert, M. & Tak, M., FlexiPKI - Realisierung einer flexiblen Public-Key Infrastruktur. *P. Horster: Systemsicherheit*, Vieweg, 2000.
- [7] FIPS186: Digital Signature Standard. Federal Information Processing Standards Publication 186, 1994.
- [8] X9.62 - Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA). ANSI, 1998.
- [9] ITSEC: Information Technology Security Evaluation Criteria. 1991.
- [10] Common Criteria for Information Technology Security Evaluation (CC), Version 2.0. ISO/IEC-Standard 15408, 1998.
- [11] X.509 Internet Public Key Infrastructure Online Certificate Status Protocol. RFC2560, 1999.
- [12] Huehnlein, D. & Knosowski, Y., 1000mal signiert: Aspekte der Massensignatur. *Zeitschrift fuer Kommunikationssicherheit*, **02**, 2003.

- [13] ETSI TS 101 862 v1.2.1 (2001-06): Qualified Certificate Profile, Technical Specification. 2001.
- [14] Maseberg, S., *Fail-Safe-Konzept für Public-Key-Infrastrukturen*. Ph.D. thesis, Technische Universität Darmstadt, 2002.