AN OPEN INTERFACE ENABLING SECURE E-GOVERNMENT

The Approach Followed with the Austrian Citizen Card

Arno Hollosi[1], Herbert Leitold[2], Reinhard Posch[1]

[1] *Chief Information Office (CIO) Austria, {Arno.Hollosi, Reinhard.Posch}@cio.gv.at*

[2] *Institute for Applied Information Processing and Communications (IAIK), Graz University of Technology, Herbert.Leitold@iaik.at*

Abstract: When encouraging citizens to approach public administrations by electronic means in order to improve the public services and to avoid costly media transitions from paper-based applies to IT-supported back-office applications, authorities and implementers need to be in particular cautious in two aspects: On the one hand, security is an indispensable guiding principle for concerns of legal certainty, identification and authentication requirements, confidentiality and data protection aspects, and certainly security is needed to achieve broad user acceptance. Electronic signatures based on smartcards represent a state-of-the-art in supporting several of these security requirements. On the other hand, the concepts followed need to be technology-neutral to a large extent to both remain open for future or emerging technologies that may mature to meet these security requirements as well and to avoid discrimination against particular solutions. Otherwise inclusion of upcoming solutions may well turn out a costly experience. In this paper the approaches followed with the Austrian citizen card are discussed – an ambitious project that aims at deploying e-Government on the large scale. By means of an open interface the authorities specify the requirements arising out of the applications in the administrative bodies. This allows the authorities to launch the development of applications based on well-defined interfaces, but not mandating a certain technological instantiation such as a social security card, public identity cards, or private-sector-borne signature cards such as banking cards. By taking up and implementing the interface specification an open market is stimulated that paves the way to a public-private partnerships. The paper gives the rationale of choosing the open interface approach and discusses its actual implementation – the so-called security layer – in detail.

Key words: electronic signatures, open interfaces, citizen card, identity card

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35612-9_23

B. Jerman-Blaži et al. (eds.), *Advanced Communications and Multimedia Security*

1. INTRODUCTION

In a summit on 20th November 2000 the Austrian federal government unanimously decided to employ chip-card technology to simplify approaching public authorities for citizens. This decision resulted in a momentum towards providing the required infrastructures and applications. For instance, the Austrian social security card that will be rolled out to each citizen end of 2002 has been enhanced by electronic signature functionality fulfilling the Austrian signature law [1] and the Austrian signature order [2] which is required for identification and data origin authentication purposes. Similarly, a public identity card has been defined based on smart-card technology. The legal framework has been prepared for e-government by permitting derived and encrypted (one-way hash functions of) public registration numbers as a means of identification [3] or enabling electronic delivery of official notifications [4]. Moreover, numerous pilots and applications have been launched or accommodated for electronic signatures by the various federal ministries such as "finance office online" enabling online tax declarations, "help@gv" an online information platform, or electronic confirmation of payment as a means of prove that administrative fees have been remitted.

However, from an holistic perspective it seems shortsighted to confide the success of e-Government solely on a multitude of technologies or applications, it is apparent that proceedings of authority's processes are interweaved – involving different administrative units and applications – and thus generalized approaches appear as the road to follow. Moreover, two aspects are imperative when considering deploying information and communications technologies (ICT) end-to-end between the citizen and the administration – **security** and **openness:**

- Security measures are required in numerous aspects: Public authority's proceedings require identification of the parties involved in many cases, data origin authentication and requirements of writing are given, or data protection laws are to be followed. In particular, public administrations can not take residual risks to an extent as they are taken in e-commerce scenarios in many cases.
- Regarding openness, applications and infrastructures for e-Government are considered a long-term investment. While electronic signatures, public key infrastructures (PKI), and smartcards are considered appropriate in fulfilling the security requirements listed above, it is well conceivable that off-the-shelf technologies such as cell phones or personal digital assistants (PDAs) will mature in terms of security and will compete with the conventional '*smartcard and PC*' combination. Adapting each application to new technologies as they show up is costly.

However, not including appropriate technologies results in the dilemma of being discriminatory with respect to future technologies. Thus, a technology-neutral road needs to be followed that allows the citizen to employ the technology of choice and still keeps the resulting requirements on the administration's application in a reasonable and manageable size.

The approach followed with the Austrian e-Government initiatives was to base the proceeding on a coordinating initiative that resulted in a consolidated view on the requirements from the administration's perspective [5]. This resulted in a list of general demands on the citizen's security token which may show up in a variety of appearances. We refer to the totality of appearances as the "Austrian citizen card" – one might suspect that this mandates smartcards, but in fact a number of alternative technologies are conceivable. However, in its initial phase smartcard systems such as the social security card will represent the vast majority.

The general demands on the functionality of the citizen card are basically (1) creation of so-called secure electronic signatures[4] that are equivalent to handwritten signatures and fulfill the requirement of writing according to the Austrian Signature law [1], (2) a second key pair which can be used for peer entity authentication or for establishing session certificates and session keys for securing the communication, and (3) so-called info-boxes that serve as containers for storing data such as certificates, identifiers with or without access control, or other data. Based on these requirements an open interface has been defined – the so-called security-layer. The security-layer offers a transmission control protocol, internet protocol (TCP/IP) communication interface and extensible markup language (XML) as the basic format of its protocol data units. This interface fulfils the requirement of the e-Government application in a generalized fashion, as well as gives the market a maximum flexibility for joining in the Austrian citizen card project. It therefore is discussed in detail in this paper.

The remainder of the paper is structured, as follows: In section 2 the legal and normative framework in which e-Government operates is discussed. This is mainly the signature law and the signature order. As the European common market asks for cross-border interoperability of national solutions, the European dimension is addressed in terms of the EU Signature Directive [6] and the European standardization efforts carried out by the European Telecommunications Standards Institute (ETSI) and the European

[4] Note, that the term 'qualified electronic signature' is also common for an electronic signature that is equivalent to a handwritten signature in legal terms, such as in the German signature law or in the standards developed in the European Electronic Signature Standardization Initiative (EESSI). However, the notion 'secure electronic signature' is used here, as this is the terminology used in the Austrian Signature law [1].

Committee for Standardization (CEN). Section 3 continues by giving details on general requirements when deploying e-Government. This gives an extended view to the basic requirements **security** and **openness** that have been sketched above. The security-layer as an open interface for e-Government applications is discussed in detail in section 4. Finally, conclusions are drawn.

2. LEGAL AND NORMATIVE FRAMEWORK

In order to contribute to the legal recognition of electronic signatures, the EU Electronic Signature Directive established a framework that required the Member States to adopt its measures and bring the corresponding national laws into force by July 2001 [6]. In its article 5.1 the Directive defined that an electronic signature is admissible as evidence in legal proceedings and satisfies the legal requirements of a signature in relation to electronic data in the same manner as a handwritten signature in relation to paper-based data, if the electronic signature fulfils certain requirements – we refer to such a electronic signature as a 'secure electronic signature'. These requirements for a secure electronic signature are mainly that:

- it is created by a so-called secure signature-creation device (SSCD) which basically is the device getting in touch with the signature-creation data – the subscriber's private key spoken in common PKI terminology.
- it is based on a qualified certificate – the electronic attestation linking signature-verification data (SVD) – the subscriber's public key in PKI terms – to a person. In order to be qualified the certificate and the certification service provider (CSP) needs to fulfill certain requirements.

The requirements stated above are specified in the Annexes of the Directive, where Annex I defines the contents of a qualified certificate, Annex II defines the requirements for CSPs that issue qualified certificates, and Annex III addresses the SSCD. In addition Annex IV states recommendations on secure signature verification.

The Austrian signature law [1] has been put in force in January 2000 and the signature order [2] gives a greater detail in particular regarding technical aspects such as cryptographic key sizes. The signature law closely follows the European Directive. However, while the Directive limits mandatory certification of conformity with its provision by designated bodies to the SSCD, the Austrian signature law requires that the technical components and procedures for generating secure electronic signatures must be verified and that the conformance must be certified by a confirmation body. While one might assume this a minor difference to the provisions of the Directive, it yet has an important influence to the system design that is discussed in this paper: Note, that the notion of 'technical components for generating the secure electronic signature' as defined in the Austrian signature law is more general than limiting the scope to the SSCD, as the component for viewing the data to be signed (DTBS), or the component for authenticating the signatory such as a personal identification number (PIN) pad are included. This is further discussed in section 3.

In order to stimulate interoperable solutions, EESSI entrusted CEN and ETSI to develop standards that support the EU Signature Directive [7]. The main provisions that have been developed within EESSI[5] and which basically fulfill the requirements of the Austrian signature order, are:

- For SSCDs, Common Criteria (CC) [8] Protection Profiles – the SSCD-PPs [9] – have been defined.
- Based on cryptographic message syntax (CMS) [10] electronic signature formats have been developed [11], or XML signatures [12] have been specified.
- For CSPs issuing qualified certificates, policy requirements have been defined [13], as well as protection profiles for hardware security modules [14].
- The cryptographic algorithms and its parameters have been specified in a separate document [15]. Signature suites based on Rivest, Shamir Adleman (RSA) [16] and digital signature standards (DSS) [17] with a minimum key size of 1020 bit or signature suites based on elliptic curve cryptography (ECC) [18] [19] with a minimum of 160 bit keys have been specified.

Based on this legal and normative framework, we continue in the following section by discussing what basic requirements can be derived. In particular, organizational aspects and issues regarding forward compatibility are addressed.

3. BASIC REQUIREMENTS OF E-GOVERNMENT

When deploying e-Government on a large scale it is essential to focus on the impact on current processes and on the long-term effect the newly created infrastructure has on organizations. It seems vital that the following principles govern the design of an e-Government infrastructure:

- *Avoidance of vendor lock-in:* The transition of paper-based processes to electronic processes should not create new dependencies or monopolistic situations. Thus the interfaces of the core infrastructure have to be in the

[5] Note that these standards are not yet recognized European standards that in lieu is to be adopted by the member states.

public domain or need to be federal property. This ensures that the administration is never unconditionally bound to a single vendor.

- *Modular design:* The sophisticated electronic processes and transactions, the multitude of participating parties, and the legal responsibilities demand a clear division into atomic components with unambiguously defined interfaces, functions, and responsibilities.
- *Technology neutral design:* Advancements in technology are happening at an ever increasing rate. In order not to be caught in the vicious cycle of perpetual updates or shutting out emerging technologies, the design of the core infrastructure has to be ignorant of underlying technologies to the greatest extend possible.
- *Security and trust:* The basic element providing identification and authentication is the electronic signature. Thus for both the application filed by the citizen and the administration generating official notifications, secure electronic signatures provide the legal certainty. In addition, the PKI-based infrastructure builds a basis for additional certificates for confidentiality.

Taking these design principles into account it is evident that there is need for a high-level interface to the citizen's security token (or smart card). This interface should describe the security token not in terms of algorithms, but in terms of functionality. By encapsulating the functions and responsibilities in a clearly defined component which is accessed through a single interface to security tokens a flexibly integration into the e-Government structure is provided.

Austria's citizen card is therefore a 'requirements profile' for security tokens rather than a specification of smart card features. The requirements of the Austrian citizen card profile are based on the requirements created by e-Government processes. Firstly, there is need for creating secure electronic signatures to be able to submit applications in electronic form. As the signature law confines the way in which secure electronic signatures can be used, the need for a second certified key pair for entity authentication or similar areas arises.

A major advantage of the security-layer concept is that the security-relevant components are viewed as a single entity by the application. I.e., the SSCD, the document viewer component, and the PIN pad are under control of the developer. Replacing components, such as replacing a PIN pad by biometric sensors is transparent to the application. By following the security-layer concept as a requirement catalogue, the market thus gains maximum flexibility in developing solutions. The details of the security layer are given in the following section.

4. SECURITY LAYER – AN OPEN INTERFACE

In this section we are taking a closer look at the security-layer interface. The security-layer is the interface offered by the so-called security-capsule – the entity containing the citizen card and implementing its immediate environment which applications communicate with. The distinction between the interface and entities implementing its functionality is important, as an open interface can be defined technology-neutral to a large extent. Thus, in a public-private partnership a non-discriminatory approach can be followed where the public authorities define the interface – the security-layer – and the industry is free to implement the citizen card – the security-capsule.

As stated in the previous section, among the major goals of the security-layer concept are forward compatibility and independence from underlying technologies. Therefore, the interface uses TCP/IP connections as communication channel and XML encoding for the communicated protocol elements. This choice appears reasonable, as TCP/IP stacks are available even for the smallest devices and for fringe operating systems. The choice of XML is straight-forward, as it is aimed that documents to be signed are primarily XML documents, and that the applications utilizing the security-capsule will be XML-aware in most cases. Furthermore, XML is human-readable and can be easily parsed; the verbosity of XML is a negligible drawback.

The security-layer uses a straight-forward request/response protocol scheme. The application opens a connection to the capsule, sends its request, and waits for the answer. The capsule receives the request, processes it, sends a response, and closes the connection. The protocol and its XML encoded data can utilize different transport layers. Among the layers currently specified are plain TCP/IP, hypertext transfer protocol (HTTP), transport layer security (TLS), and HTTP over secure socket layer (HTTPS). Simple object access protocol (SOAP) and XML remote procedure calls (XML-RPC) are currently being investigated.

4.1 Protocol, Functions, and Commands

The security-layer provides the following high-level functions to applications:

- signing documents according to CMS [10] or XMLDSIG [20]
- verifying CMS or XMLDSIG signed documents
- storing and retrieving data
- utility functions such as creating a session certificate or creating a symmetric session key based on Diffie-Hellman
- querying properties of the capsule and of the cryptographic token.

Providing not only functions for creating electronic signatures, but also for verifying signatures relieves applications of the burden of dealing with this complicated subject area for the lifetime of the document. Applications are even ignorant to which algorithm (e.g. RSA, DSA, ECDSA) or which certificate format (e.g. X509, PGP, SPKI) is being used. Thus, not only can applications be light-weight, they also need not be updated when a new signature algorithm, a new signature format, or a new certificate format is introduced. This forward compatibility is one of the key benefits of the security-layer.

Access rights and their management are another area of critical importance. Again, the design relieves applications of dealing with this issue. Instead, applications issue their request through the interface ignorant of any access rights. The security capsule then either grants or denies access to the function according to its own security policy. This policy may involve the user, such as entering a PIN code, but it can also make automated decisions, e.g. based on the certificate of the application when using TLS or HTTPS as transport layer. Again, new technologies such as using biometric data instead of PIN codes can be introduced without the need to update applications.

It is important to note that the security-layer interface allows specifying Internet references (unique resource identifiers, URIs) instead of data itself making full use of the network wherever it makes sense. Thus it is not necessary that all data is transmitted through the security-layer interface. Instead the security-capsule resolves the references and downloads the data from the specified remote resource when desired.

4.1.1 Creating and Verifying CMS Signatures

The command for creating a CMS signature [10] takes a single file as input. The application can specify which key should be used for signing, and it also provides information on the file's MIME-type [21] [22], so that the capsule can invoke the appropriate trusted viewer for the file. The function returns a CMS object, which additionally contains information according to ETSICMS [11]. In particular, it contains a signed certificate reference. This is necessary, because otherwise one has no guarantee that the certificate used for signing is the one accompanying the CMS signature object.

CMS signatures can be created as detached signatures (data and signature in different files) or as enveloping signatures (CMS object encapsulating the data file).

The verify command takes a CMS signature object as parameter and optionally a data file in case of a detached signature. It returns two results:

one for the validity of the signature itself, and one for the validity of the signing certificate at a specified point in time.

4.1.2 Creating and Verifying XMLDSIG Signatures

The command for creating XMLDSIG signatures [20] is more sophisticated than its CMS counterpart, as the XMLDSIG recommendation has more features. It can take more than one data file as parameter. Each data file can be transformed by algorithms specified in XMLDSIG before the signing process. Optionally applications can provide additional supplements needed for these transformations.

Among the transformations allowed are XML canonicalization [23], XPath transformations [24], and XSLT stylesheet transformations [25]. The latter two transformations play an important role for usability and generalized data handling. The resulting XMLDSIG signature includes signed properties according to ETSI's "XML Advanced Electronic Signatures" [12], again in particular a reference to the signing certificate. Furthermore, the signature contains a manifest with references to all input data used for the transformations. This allows applications to make reasonable assumptions about the correlation between original input data and the transformed data actually signed.

The verify command takes a document containing an XML signature, an XPath describing the location of the signature inside the document, and optionally supplement data used for transformations. The function returns results for the validity of the signature, the validity of the signing manifest, and the validity of the signing certificate at a specified point in time.

4.1.3 Storing and Retrieving Data

Applications may store or read data from the security capsule. In some cases this data may be stored on the cryptographic token itself, in other cases this data may reside inside the capsule, or even somewhere on the net. The security-layer interface defines so called info-boxes, which are containers for data. Each info-box has an identifier (analogous to a file name) and contains data according to its type. There are two different types of info-boxes:

- a binary file type and
- an associative array type.

As the name already suggests, the binary file info-box behaves like a file. Applications can read the info-box, or overwrite the info-box with data. It is not possible to read or overwrite parts of the info-box, the command always affects the whole info-box content.

The associative array stores data in key/value pairs. E.g. the certificate info-box is an associative array, where certificates are stored in the pair-value and where the corresponding pair-key is set to the name used for selecting the certified signature key to be used when creating signatures. The interface has functions for creating, updating, and deleting pairs and for renaming and searching keys.

4.2 Transport Layer Binding

The security-layer protocol itself may utilize different transport layers. Transport layers currently defined are TCP/IP, TLS [26], HTTP [27], and HTTPS [28]. TCP/IP and TLS just transmit the XML requests and responses as they are. In case of TLS the capsule may evaluate the application's certificate when deciding whether to grant access privileges or not.

The HTTP and HTTPS bindings are designed so that Web browsers do not need any active components (not even scripts) to access the security-layer. To that end, the HTTP binding uses the standard mechanisms used by HTML forms. The binding defines a set of input fields:

- *XMLRequest:* this field holds the XML coded security-layer request itself.
- *DataUrl:* specifies an URL where the security-capsule should send the resulting response.
- *RedirectUrl:* specifies an URL to which the browser should be redirected in response to its request.
- *StylesheetUrl:* if no RedirectUrl is specified, the browser would directly receive the XML encoded response. In most cases however, one would rather not see the user receiving the XML as such. By specifying a XSLT compliant stylesheet the security-capsule uses this stylesheet to transform the XML response (for example into HTML) before sending it to the browser.

Among the fields listed above, only the first one (XMLRequest) is mandatory. By having the option to send the response directly to the server and send a formatted reply to the browser this binding offers utmost flexibility in design of the data flow and user experience.

CONCLUSIONS

The paper has given an overview to the approach followed by the Austrian e-Government initiatives. Two aspects have been identified as crucial in order to reach acceptance: On the one hand, security is a must to both achieving citizen's acceptance and to fulfill legal requirements. On the other hand, discriminatory situations ruling out solutions that may show up in the market in the future need to be avoided.

Both aspects have been fulfilled by means of an open interface – a so-called security-layer. The interface de-couples the application from the security-relevant functional blocks, such as the signature-creation process, by defining a general requirement catalogue. The market then can take up the approach by implementing the requirements with the technologies of choice. This results in the forward-compatibility aimed.

REFERENCES

[1] Austrian signature law: "Bundesgesetz über elektronische Signaturen (Signaturgesetz - SigG)", BGBl. I Nr. 190/1999, BGBl. I Nr. 137/2000, BGBl. I Nr. 32/2001.

[2] Austrian signature order: "Verordnung des Bundeskanzlers über elektronische Signaturen (Signaturverordnung - SigV)", StF: BGBl. II Nr. 30/2000.

[3] Administration reform law: "Verwaltungsreform Gesetz", 2001 amending "Allgemeines Verwaltungsverfahrensgesetz (AVG)" BGBl. Nr. 51/1991.

[4] Notification delivery law: "Bundesgesetz vom 1. April 1982 über die Zustellung behördlicher Schriftstücke", BGBl. I Nr. 137/2001.

[5] Posch R., Leitold H.: "Weissbuch Bürgerkarte", Bundesministerium für öffentliche Leistung und Sport, IT-Koordination des Bundes, June 2001.

[6] Directive 1999/93/EC of the European Parliament and of the Council of 13. December 1999 on a community framework for electronic signatures.

[7] European Electronic Signature Standardization Initiative: "EESSI explanatory document: Description of deliverables", EESSI Steering Group, 2000.

[8] International Organization for Standardization: "Information technology - Security techniques - Evaluation criteria for IT security", ISO/IEC 15408-1 to 15408-3, 1999.

[9] CEN/ISSS WS/E-Sign Workshop: "Security Requirements of Secure Signature Creation Devices (SSCD-PP)", CWA 14168 and CWA 14169, 2002.

[10] Hously, R.: "Cryptographic Message Syntax (CMS)", IETF Request For Comment RFC 2630, 1999.

[11] ETSI SEC: "Electronic Signature Formats, v.1.2.2", Technical Specification ETSI TS 101733, 2000.

[12] ETSI SEC: "XML Advanced Electronic Signatures (XAdES)", Technical Specification ETSI TS 101903, 2002.

[13] ETSI SEC: "Policy requirement for certification authorities issuing qualified certificates, v1.1.1", Technical Specification ETSI TS 101456, 2000.

[14] CEN/ISSS WS/E-Sign Workshop: "Cryptographic Module for CSP Signing Operations – Protection Profile (CMCSO-PP)" , CWA 14167-2, 2002

[15] European Electronic Signature Standardization Initiative: "Algorithms and Parameters for Secure Electronic Signatures, v2.1", EESSI algorithm group, 2001.

[16] RSA Laboratories: "RSA Cryptography Standard", PKCS #1 v2.1 draft 2, 2001.

[17] National Institute of Standards and Technology, "Digital Signature Standard (DSS)", NIST FIPS Publication 186-2, 2000.

[18] American National Standards Institute, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", ANSI X9.62-1998, 1998.

[19] International Organization for Standardization, "Information technology - Security techniques - Cryptographic techniques based on elliptic curves - Part 2: Digital signatures", ISO/IEC FCD 15946-2, 1999.

[20] Eastlake D., Reagle J., and Solo D.: "XML-Signature Syntax and Processing", W3C Recommendation, 2002.

[21] Freed N., Borenstein N.: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", IETF Request For Comment RFC 2046, 1996.

[22] Murata M, Laurent S. St., and Kohn D.: "XML Media Types", IETF Request For Comment RFC 3023, 2001.

[23] Boyer J.: "Canonical XML", W3C Recommendation, 2001.

[24] Clark J., DeRose S.: "XML Path Language", W3C Recommendation, 1999.

[25] Clark J.: "XSL Transformations (XSLT)", W3C Recommendation, 1999.

[26] Dierks T., Allen C.: "The Transport Layer Security (TLS) Protocol, Version 1.0", IETF Request For Comment RFC 2246, 1999.

[27] Gettys, Mogul, Frystyk, Masinter, Leach, and Berners-Lee: "Hypertext Transfer Protocol, HTTP/1.1", IETF Request For Comment RFC 2616, 1999.

[28] Rescorla: "HTTP over TLS", IETF Request For Comment RFC 2818, 2000.