AUTOMATIC AUTHENTICATION BASED ON THE AUSTRIAN CITIZEN CARD

A Reference Implementation

Arno Hollosi, Udo Payer, Reinhard Posch
Institute for Applied Information Processing and Communications

Abstract: The concept of the Austrian citizen card enables the generation of electronic signatures, provides mechanisms to establish confidential communication channels, and supports features for user authentication in public services. This document specifies a mechanism —*based on the Austrian citizen card*— and a trustworthy component —*called security layer*— to fulfil all requirements for authentication processes, suitable for electronic administrations. The additional trustworthy component (security-layer) forms the interface between diverse applications and the smart card (citizen card). But this layer also offers features which can be used very efficiently in conjunction with certificate-based user authentication. Depending on the used technology, three different levels (qualities) of user authentication can be realized. In the following, a short introduction is given to the concept of the Austrian citizen card followed by common descriptions of three mechanisms suitable for usage in the environment of public services.

Key words: citizen card, smart card, identity token, peer-entity authentication, authorization

1. INTRODUCTION

The concept Austrian citizen card is based on smart cards, which are able to generate secure electronic signatures. Today, traditional administrative requests are tightly coupled to the conventional signature of the concerned citizen. Therefore, electronic administration procedures have to offer equivalent possibilities - even in the case of electronic attachments.

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*

The Austrian signature law and signature order form the legal basis of generating so called secure electronic signatures[2]. This enables citizens to enter the electronic administration without prior personal registration. The citizen card concept intends to offer public administration procedures which can be modelled efficiently and economically. This presupposes that electronic administration applications can be automated to a large extent. It also implies that the underlying infrastructure supports such mechanisms.

From all of its diverse forms of appearances, the citizen card concept will be based primarily on the social insurance card, since eight million citizens are under social security and will receive their own social insurance card. Mixing the citizen card concept with mechanisms for social security services is unobjectionable, since different cryptographic mechanisms are in use. An endangerment by cross-references of different ranges is impossible in each case. But other cards (identity card, bank cards, etc.) will be applicable in the citizen card concept as well.

Apart from generation of electronic signatures, the citizen card concept provides the possibility to store additional data elements in so called "information boxes". For example, it can be very useful to store certificates or other information on the card which makes online access to these resources unnecessary. Information boxes can also be used to store electronic documents. As the memory on smart cards is quite limited, it is also thinkable to store references to a repository instead of the documents themselves. Therefore, it would become possible to store documents in arbitrary places – protected by authentication based on the card. Thus, the card owner has full control over content and volume of these information boxes.

Due to the multitude of people involved and their different requirements, the citizen card concept intends to provide roles and mandates. Attribute certificates and other methods of IT-security will be used in order to be able to realise roles and mandates technically.

Public administration entrances have a special need of privacy. This document describes solutions of integrating cryptographic mechanisms exemplarily which are suitable for user authentication in electronic administration. According to different approaches, three stages (qualities) of user authentication can be defined. A description of these solutions is given in section 4.

But before turning to technical details of the implementation, some fundamentals and requirements should be discussed first:

[2] Note that "secure electronic signature" is the term used by Austrian law. In the European directive these signatures are called "qualified electronic signatures".

2. FUNDAMENTAL PRINCIPLES

This part discusses some common authentication mechanisms and points out advantages and disadvantages of current solutions. Moreover, general requirements of e-Government solutions are specified.

Fortunately, the Austrian citizen card will be able to support ECC mechanisms, which is undisputed in connection with smart cards. Therefore, a short introduction into digital signatures – based on ECC mechanisms – is given first.

2.1 ECDSA and the Austrian Citizen Card

The Elliptic Curve Digital Signature Algorithm (ECDSA) [3] is the elliptic curve analogue of the well-known Digital Signature Algorithm (DSA). It was approved in 1999 as an ANSI standard [4], and was accepted in 2000 as IEEE and NIST standards. Moreover, ECDSA was also accepted in 1998 as an ISO standard.

Unlike the ordinary discrete logarithm problem and the integer factorisation problem, no sub exponential-time algorithm is known for the elliptic curve discrete logarithm problem. For this reason, the strength-per-key-bit is substantially greater in an algorithm that uses elliptic curves.

ECC systems over prime fields of at least 161 bits (or 188 bits if one prefers to be more careful) are sufficient until year 2020 [6]. As the social insurance card supports ECDSA, it is clear that these mechanisms will play a major role in the concept of the Austrian Citizen Card.

2.2 Common Solutions

User authentication is a term which is used in a very broad sense. By itself it has no other meaning than the guarantee that users are who they claim to be. User authentication in web-based applications is therefore a much-discussed topic followed by a great number of different solutions:

Almost all browsers support **basic authentication**. When entering a realm, a standard popup window appears on the screen, asking for username and password. The realm value should be considered an opaque string, which can only be compared for equality with other realms on that server. The server will service the request only if it can validate the user-ID and password for the protection space of the Request-URI. The major drawback of the basic authentication scheme is that it is relatively simple for eavesdroppers to spy out the password since it is transmitted as plain text. [10]

An alternative authentication scheme known as **digest authentication** remedies this weakness through the use of cryptographic hashes, usually the MD5 message digest algorithm defined in [7].

Now, while taking username and password, running them through MD5 (as you do with base64 for basic authentication), and sending the result to the server, a potential eavesdropper could record the hashed username and password to initiate a replay attack.

To securely prevent replay attacks, a more sophisticated procedure is obviously necessary - the **digest access authentication** scheme:

The main difference between digest- and digest access authentication is the use of a nonce to prevent replay attacks. But this scheme provides no encryption of message content. The intent is simply to create access authentication methods to avoid the most serious flaws of basic authentication without using SSL [10]

Cookie authentication makes use of functionality at the scripting level to provide user authentication. Cookies can store usernames and passwords independently of locally stored users - this requires that browsers support cookies and that cookies are enabled. Thus, it is not practicable to use this mechanism on publicly accessible computers, such as public portals or university environments.

Beyond standard HTTP authentication schemes, there are a couple of authentication mechanisms, either based on **sending encrypted user information**, or **using digital certificates**.

The **ISAPI** (Internet Server Application Programmer Interface) provides low-level access to the entire web server request and event chain. Because of this it can intercept requests before the web server handles them and provides the greatest authentication flexibility. ISAPI interception plays a major role in some of the proposed solutions.

2.3 General Conditions for e-Government Solutions

This section discusses additional demands caused by heterogeneous environments (web browsers and clients) that are expected in standard e-Government solutions.

Simple processes: The number of used technologies should be limited. Thus, a rapid conversion and adjustment of new applications can be accomplished easily. This also implies openness for all given systems. The people involved (developers of applications for electronic administration) should have to deal with a minimum number of simple structures. These simple structures and interfaces should be simple to learn and deployable without large efforts.

Open Interfaces: Core interfaces should be in the public domain and be built upon international, non-proprietary standards and technologies. This avoids vendor lock-in. Also, these interface should cleanly separate key players and their legal liabilities. The interface to the trustworthy component (security-layer) is implemented as an open interface based on TCP/IP, HTTP, and XML.

Authorization and certificates: Authorization should be based on end-to-end mechanisms. Public access or privately held facilities are not allowed to play a role in the course of user authorization. Therefore, certificates and electronic signatures are well suited for dynamic user authorization. Directory services can be used to share certificates and attribute certificates. These components are forming the essential mechanisms to integrate roles and positions into user authorization. Since usage of certificates in authentication processes is anything but simple, the security-layer is used to simplify all processes by offering an elementary interface.

Authentication: User authentication should be realized by using electronic signatures. Declaration of intention to sign an authentication request should always require the entering of a PIN. For less important authentication scenarios it is also conceivable to use other mechanisms.

Authentication of users is usually limited to the authentication of physical persons. If additional characteristics have to be proven, the best solution would be to fall back on conventional, paper-based mechanisms suitable for conventional administrative authorities. These mechanisms can be based on resident certificates, birth certificates etc.

In a similar way, it can be necessary to enclose these additional certificates in an electronic process. A suitable method to deal with these documents is to store their content in XML structures.

Administration officers – working with electronic documents – have to be authenticated as well. In principle, the same mechanisms as used for authentication of citizens can be applied. Characteristics and roles of administration officers can be managed by attribute certificates. In the case of accepting a role, the officer has to be identified in the context of the application and not in the context of access services.

Single Sign On: Single Sign On is suitable and required for processes in a distributed application framework. Single Sign On can also be realized by using electronic signatures. The main intention is to avoid multiple identifications to act in different roles or deal with different applications following a single authentication. Multiple authentications in the course of a single session are not necessary and desirable. Once authenticated, the user remains authenticated as long as the user does not leave the same security realm.

Confidentiality: A certain degree of confidentiality is also required by some processes. This can be realised by using encryption mechanisms. Methods and key lengths have to be chosen in such a way that claimed levels of confidentiality can be guaranteed.

Beyond this, we have to guarantee secure end-to-end connections. Smart cards are well suited for this purpose, as their private keys are protected from disclosure and they can be used for electronic signatures to prevent unauthorized access to confidential information.

Identification: Austria has strict privacy protection laws. In context of the citizen card concept every citizen is assigned a personal ID number which is essential in order to easily and accurately identify a person. However, law forbids that this number is stored in databases. This gordic knot is solved by using one-way functions to transform the PID into a context dependant process identifier which can be stored in databases. Basically, the transformation takes the PID and the application name as input and produces the process identifier (e.g. by using a hash function). Thus a person has different identifiers in different applications. Note that during the authentication process the PID is transmitted, so that the server can derive the context dependant ID.

3. ACCESSING THE CITIZEN CARD TRHOUGH THE SECURITY-LAYER INTERFACE

To access Austrian citizen cards a trustworthy interface is required. The security-layer represents this interface, which is one of the central components in the proposed authentication process.

The security-layer acts as an interface between the browser and the citizen card and can be used to send signature- and verification requests to the citizen card. Moreover, this layer also provides commands for reading and writing data to information boxes (short: info boxes).

Connecting browser applications to the security-layer is quite simple since two different interfaces are supported. Beside a standard TCP/IP and socket-based interface, it is also possible to pass over XML requests [13] in the form of HTTP POSTs [8]. Requests like signing or verifying XML structures can be sent to the security-layer – which is processing this request – and retrieved by the requestor or can be forwarded to a given URL.

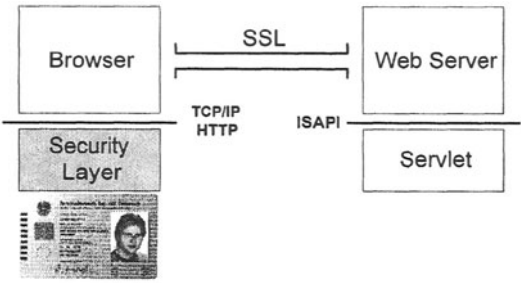

Figure 1: Security-layer offers two interfaces: TCP/IP and HTTP

Another key function is the ability to transform the XML response into HTML using XSLT stylesheets loaded from the server. This HTML, which may contain scripts as well, is then sent to the browser. This mechanism is playing a major role in one of the proposed authentication mechanism, unless further client components are installed.

Just for completeness we have to note, that neither socked based nor HTTP based bindings are inviolable against authentication and confidentiality on the interface to the browser. Thus, a limitation to local host access is mandatory and self-evident. Figure 1 depicts that just local browsers are authorized to bind the security-layer.

4. AUTOMATIC USER AUTHENTICATION

Online applications within the range of the public administration require different stages of security. This also applies to the level of authentication, since applications may exist which do not need qualified certificates[3] [14]. As a function of the authentication level, three stages can be defined:

1. Safe for normal operations
2. Safe within a trustworthy infrastructure
3. Technical end-to-end safety

A common public application requires just a minimum of confidentiality. These services can be realized by a simple server-authenticated SSL connection.

If a trustworthy infrastructure is to be achieved, the use of active components is imperative. These active components can either be loaded

[3] Qualified certificate and advanced electronic signatures are common terms in the European Community framework for electronic signatures.

from a trustworthy side (whenever they are needed) or can be installed on the client's host.

The technical realization of trustworthy end-to-end channels can either be based on certificates, or mechanisms based on advanced electronic signatures, created by using qualified certificates.

These security requirements have to be granted in ordinary technical environments. It can be the fact that some of these requirements have to deal with (1) different client certificates, since no common certificate structures exists. Secondly, the introduction of trustworthy active components can be realized by using the (2) concept of the citizen card in combination with the security-layer. And finally, some of the Austrian citizen cards are based on (3) elliptic curve mechanisms. Thus, ECC mechanisms have to be supported.

4.1 Level I

Level I fulfils only rudimentary demands on the quality of the communication channel. Basically level I guarantees confidential, one-side authenticated communication. Respective requirements can be achieved by simple server-authenticated SSL or TLS connections. In the event of accessing a sensitive realm, secure communication has to be requested by using SSL or TLS. Beyond that, guidelines have to exist, enforcing a minimum key length of at least 100 bits. Apart from well-known problems with server-authenticated SSL connections, there is a theoretical possibility of Man-in-the-Middle attacks.

4.2 Level II

Level II makes higher demands on the trustability of the communication channel. It also requires user authentication. This level is based on signing and verifying authentication information. In doing so, authentication is based on a mutual agreement, very similar to the X509 strong 2-way authentication protocol.

Right after an authentication request (0), the server has to create and sign a security token (1) – consisting of a timestamp (or nonce), the session ID, and the servers IP address or URL – and has to send this signed security token to the requesting client. The client has to verify the server's signature – has to extract timestamp and URL and has to compose a unique authentication block (2). Uniqueness of the authentication block is based on time stamps and the signature, which is created by using a qualified certificate. This security token has to be sent to the URL which was included in the server's security token. Finally the server has to check the client

signature, derives the client's identity information, and registers the user to the selected application.

0.	C→S:	Auth.Req.
1.	C←S:	certS, SS(tS, URLS, SID)
2.	C→S:	certC, SC(tS, URLS, SID, IDL)

IDL = SBH(KPC1, KPC2, C) ... Identity Link
KPC1 ... public key
KPC2 ... public key of a qualified certificate
C ... personal information (name, date of birth, personal ID)
SBH() ... signed by the public authority
SID ... Session ID
tS ... time stamp or nonce

Figure 2 describes the process of building and signing the security token (1) with the subsequent generation of an authentication block (2), figurative:

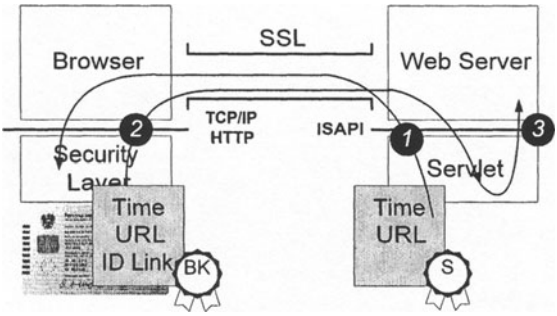

Figure 2: Signing and verifying SCT and authentication block

Level II can be overwhelmed if attackers are in the possession of the server certificate (S) issued to administration servers. However, this is unlikely to happen. A more serious issue arises, if users do not verify the server certificate, which unfortunately is common practice. Assume that a user can be tricked into accessing an arbitrary server (e.g. having a similar URL) that holds a server certificate S'. S' might be issued from a certification authority trusted by the user's browser and thus no security warning appears on the screen. This arbitrary server can then act as man-in-the-middle (until the user verifies the certificate and detects the fraud).

To clarify components and services, section 4.2.1 to 4.2.4 will specify involved data structures and mechanisms in more details.

4.2.1 Security Token

Right after accessing restricted pages, the server has to generate a time stamp. A time frame is started, within the client has to respond by sending the authentication block. The timestamp together with the session ID, IP Address or server's URL forms the security token, which has to be signed by the server and has to be sent to the client. All this has to occur in already established server authenticated SSL connections.

4.2.2 Authentication Block

Initiated by the identification request, the client has to verify the server signature by using the server certificate. If this signature is valid, the client has to check the session ID, and has to memorize the server URL. The next step is to fetch the identity link from the citizen card, which is permanently stored and can be secured by cryptographic mechanisms in one of the citizen cards data bags (info box). These data bags can also be used to store electronic documents in the form of XML structures, certificates, or attribute certificates.

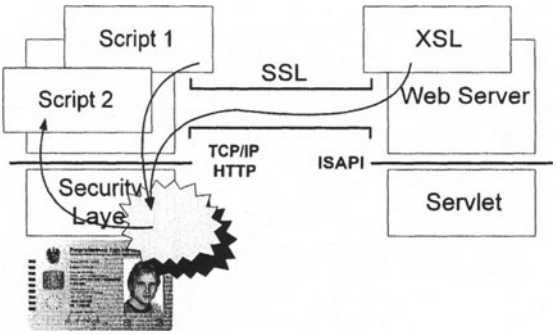

Figure 3. Transforming style sheets (XSL)

Reading the info box is done by a script, which is followed by submitting a form, containing a stylesheet URL. The security-layer is loading the specified stylesheet (XSLT) from the server and transforms it into an HTML page containing a script, which is executed – next to the previous one – by the browser.

This subsequent script is used to compile the authentication block. To be able to generate variable scripts – since scripts have to be generated on a per connection basis – another active component is used at the server side. This component supplies the client with downloadable and modified scripts and pages.

The completed and signed authentication block is sent back to the active server component, where its signature is verified.

4.2.3 User Registration

Right after verification of the authentication block, the active component is extracting the personal ID number (PID) from the identity link to calculate the process identifier. As discussed in section 2.3, one-way functions have to be used to derive a unique process identifier from the unique PID. The derived process identifier can be used as username (and password,) which are used to register users to e-Government applications.

As long as a user remains within the same realm, the user remains registered.

4.2.4 Proxy Services

If the registration of users to applications is based on simple authentication processes, access to protected pages has to be limited to the server. Once registered to the server application, the servlet has to mark this connection as an authenticated connection and has to act as a proxy for the authenticated client.

4.3 Level III

The policy of level III is in principle based on the same mechanisms as describe in section 4.2. The only difference is the use of the SSL server certificate, which is tightly bound to the used SSL connection, to form the security token (SCT).

After generating and sending the SCT to the client, the client can extract the server certificate from the corresponding SSL connection and can use this certificate to verify the SCT signature. After successfully verifying the SCT, a trustworthy server authentication on the client side can be assumed.

This mechanism can successfully prevent man-in-the-middle attacks and spoofs relying on "lazy" users not verifying certificates. But realization of level III authentication requires the integration of a further trustworthy active component at the client, to obtain the required server certificate.

Browser Helper Objects [1] can be used to get access to any running instance of an Internet Explorer by attaching itself to every new instance. By using this feature, it is possible to gain easy access to the object model as well as to receive all events coming from the browser. A BHO is therefore an excellent device to intercept and modify HTTP data streams, and well suited for level III implementations.

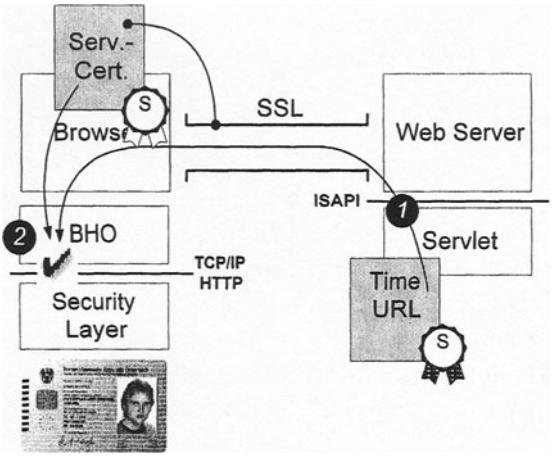

Figure 4: Requirement of a trustworthy instance (BHO) to obtain the SSL certificate

Once the signature of SCT was verified, an authentication block is built on the same way as described in layer II.

5. CONCLUSION

The concept "Austrian citizen card" takes into account mechanisms for electronic signatures, describes interactions between the citizen card and e-Government applications, which may make use of preliminary portals or market places. Substantial characteristics of these mechanisms are directory services, certificates and their attributes. This paper describes some features of the concept of the Austrian citizen card (which are of interest in context of authentication) and how this concept can be used for user authentication processes using standard web technologies. It should be noted that the discussed procedures, applications, and application structures are suitable not only for e-Government but for generic web applications as well. Furthermore, we discussed how the available underlying technologies allow different qualities (levels) of user authentication. From the offered procedures, application developers are free to choose a suitable one, which either is well suited for a certain application or offers a maximum of security.

REFERENCES

[1] Esposito: Browser Helper Object: "The Browser the Way You Want IT", Microsoft Corporation, January 1999.
[2] Certicom, "Elliptic Curve Cryptosystem for Smart cards", Certicom White Paper, 05/1998.
[3] ANSI X9.62, "Public Key Cryptography for the Financial Services Industry": The Elliptic Curve Digital Signature Algorithm (ECDSA), 1999.
[4] American National Standard Institute, "Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA)", ANSI X9.62-1998, 1998.
[5] Karlinger: "XML Electronic Signatures Application according to the international standard XML Signature Syntax and Processing", CMS 2001 Darmstadt, Germany, 2001.
[6] Lenstra: "Selecting Cryptographic Key Size", The Journal of the International Association for Cryptology Research, Vol.14 Number 4, 2001.
[7] R. Rivest: "The MD5 Message-Digest Algorithm", RFC1321, April 1992.
[8] L. Daigle, D. van Gulik, R. Iannella, P. Faltstrom: "URN Namespace Definition Mechanisms", RFC2611, June 1999.
[9] Gettys, Mogul, Frystyk, Masinter, Leach, and Berners-Lee: "Hypertext Transfer Protocol HTTP/1.1",RFC2616, June 1999.
[10]Franks, Hallam-Baker, Hostetler, Lawrence, Leach, Luotonen, and Stewart: "Basic and Digest Access Authentication", RFC2617, June 1999.
[11]Reagle: "XML Signature Requirements", RFC2807, July 2000.
[12]Reagle, Eastlake, Solo: "XML Signature Syntax and processing", RFC3075, March 2001.
[13]IETF W3C: "XML-Signature Syntax and Processing".
[14]The European Parliament and the Council of the European Union: "DIRECTIVE 1999/93/EC OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL", Official Journal of the European Communities, Article5, December 1999