Application of Network Smart Cards to Citizens Identification Systems

Joaquin Torres[1], Mildrey Carbonell[1], Jesus Tellez[2], Jose M. Sierra[1]

[1]Carlos III University of Madrid, Computer Science Department,
Avda. de la Universidad, 30, 28911, Leganés (Madrid), Spain.
{jtmarque,mcarbone,sierra}@inf.uc3m.es
[2]University of Carabobo, Venezuela, Computer Science Department (Facyt)
Av. Universidad, Sector Bárbula, Valencia, Venezuela.
jtellez@uc.edu.ve

Abstract. This paper proposes a new authentication and authorization architecture based on a *network smart card* with identification purposes: ID-NSCard. Thus, a citizen who holds this kind of device might be securely authenticated by a remote authoritative server in an identification system. This work shows how the standardized specifications are transparently reused and integrated in the proposed architecture. Details of the protocol and authentication mechanisms are provided for a Case of Study: Spanish National Electronic ID Card.

1 Introduction

Multiple works have attempted to define what a person's identity is. Many of them consider identity as the distinguishing characteristics that determine unequivocally that a person is who that person claims to be. The authentication is the mechanism by which the identity of a person is verified.

Identity clearly is a target of theft. By stealing another person's identity, somebody could gain access to services or facilities to which the thief is not entitled. Stronger ways of reinforcing security and trust are needed in order to avoid undesirable impersonations.

Many countries are starting to issue national identity cards or electronic passports for citizens that include a chip card (ID-card). This is an electronic way to hold a trusted identity credential. Additionally, both logical and biometric identifiers are usually required for authenticating the citizens' credentials during the identification process.

Most of these ID-card solutions provide two main security services: authentication and digital signature. Note that the first one allows the authoritative organisation to determine whether the claimed identity really belongs to the service requester (in this context, identification and authentication terms are commonly used), and the second one guarantees the non-repudiation of an electronic transaction. With these goals, the law [1] and standardization bodies [2, 3] envisage two different qualified digital certificates: citizen's authentication certificate and citizen's digital signature certificate, both of them installed in a SSCD (secure signature creation device).

The present paper is just focused on the identification scheme, by means of the authentication of identity credentials. More concretely, it is focused on a remote authentication procedure, which does not take place between a smart card (in a SSCD role) and an access terminal, but between the first one and a remote authentication server, where the authoritative application is running.

One of the more relevant European references for the implementation and deployment of national ID cards are issued by the European Committee for Standardization (or CEN), which among other specifications is defining an application interface for smart cards used as secured signature creation devices [4, 5] and the European Citizen Card, ECC [6]. Nevertheless, the protocols and schemes derived from these standards might be improved in terms of robustness and security. After analyzing them, we propose a more secure Identification System based on ID-NSCards, which aim to be more autonomous smart cards with identification purposes. With these goals, an atomic implementation of layer 2 authentication protocols within the card and end-to-end communications with a back-end authentication/authorization server, among other aspects, are presented in this work.

In the reminder of this paper, the related work is reviewed and analyzed in section 2 and, afterwards, we describe an authentication architecture based on our network smart card concept, NSCard, which implements ID-card authentication functionalities (ID-NSCard). In section 4, security and trust issues related to such an architecture are discussed. Finally, we treat a case of study for developing the proposed architecture in a real identification system: the Spanish National electronic ID-card.

2 Related works

One of the objectives of our work is, as far as possible, to treat the smart card as a networked host. Several works have been done in this area.

The proposal in [8] was oriented towards establishing a simplified TCP/IP protocol stack. The smart card supports this protocol stack and behaves like a small Web server. The (U)SIM security modules were not absent from this approach. New perspective on business models favoured the creation of generic tools based on smart cards, such as SIM Toolkit [9]. In this frame, these devices are equipped with a certain level of pro-activity and an improved connectivity, through a client-server model, in an over-the-air (OTA) system. This technology is based on Short Message Service, SMS, and is able to update SIM cards as well as downloading and activating new services. Interesting research was done making use of this technology, enabling the SIM card to be used as Web server [10]. Its implementation does not correspond entirely to that established in the standard HTTP protocol, but the result is functional and effective for certain applications.

The aim in [11] was to obtain a TCP-type protocol. This protocol did not fulfil all of the requirements that are established in the standard [12] but it included the concept of agent-based Internet card. In [13] Internet infrastructure extends to include smart cards for the first time, and a specific middleware is defined in order to protect communications between applications and smart cards. A proxy implementation was noted in [14-16]. This allowed cards to be efficiently integrated in distributed environments. The consolidation of Java Card as object-oriented programming favoured this process. Java Card Web Servlet technology was used to transform the smart card into a portable repository for Web objects, including HTML pages with data for a specific application, in [17]. Smart cards continue being integrated into the Web environment, even merely as an element with the capacity to store and transport user's personal information securely. The use of these devices to manage Web session cookies was proposed in [18].

In [19] an important number of experts discussed the characteristics, steps and planning of what could be a new generation of smart cards with or without contacts. A clear evolution towards a networked smart card was quite evident. More recent works specify what these cards would be like and what security advantages they would introduce. These cards could fall under the ever-widening concept of network smart cards. More details on the essential contributions of these works could be found in the following paragraphs.

In [20] and [21], the card was already treated as an Internet node which implemented standardised security and communication protocols, to be connected to a network via the host. The card was able to provide services or access Internet resources making use of protocol stacks in the same way as any other node on the network. Its use in security solutions was soon proposed. In this way, the network smart card was able to establish secure direct communication with remote Internet servers, as shown in [22]. This capacity allows the cards to guarantee online transactions. An authentication comparison with OTP devices or with regard to conventional cards can be found in [23]. Other works were focused on lower-level security and they treated packet filtering by the smart card in different stages. This filtering may be produced from interruptions service routines to the actual filtering by the protocol stack [24].

More recently, the advances in networks smart cards have been studied in [25, 26] from different approaches. Concretely in [25], we describe the rationale of our network smart card concept.

Regarding the use of network smart cards in remote identification schemes based on ID-cards there is not much work done. In this paper, we aim to securely integrate this kind of devices in such schemes with a reduced protocol stack and guaranteeing security and trust features.

2.1 Analysis of the closer ID-card solution for user authentication and authorization

Many of the current European ID-card solutions for citizen authentication are based on the standard [5] (e.g. German Electronic ID card, Finnish eID card, Spanish National Electronic ID card, etc.). This specification states the common scenario where the citizen identification remotely takes place: it is named "Client/Server Authentication". This procedure is described in Figure 1 for case of the SSL protocol. Briefly, the citizen's authentication certificate (client certificate) is used during the SSL handshake and afterwards such a citizen is challenged with the value T. For completing the tunnel establishment, she signs T and replies. Obviously, the signature is securely computed inside the ID-card. Once the authentication server verifies the signature, the citizen is authenticated (identified) and the secure tunnel is finally established between the user's computer and the remote server. Consequently, certain on-line services will be then available.

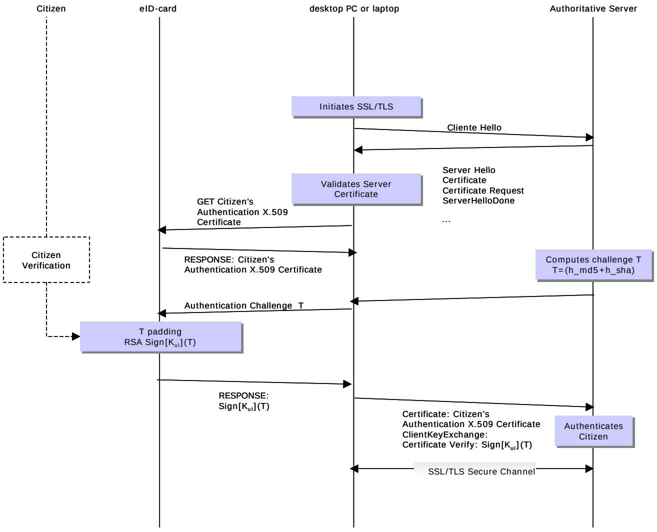

Fig. 1. Client/Server Authentication derived from [5].

But the reader should note in this scheme that:

- It is not applicable to any remote identification scenario: an equipment with a complete TCP/IP protocol stack and secure socket layer is needed. In such a scenario, this equipment is a desktop PC or laptop.
- The secure tunnel is established between PC and remote sever. In a public (unknown) environment, an end-to-end tunnel between the ID-card and the remote server should be desirable.
- Any device authentication does not take place. The ID-card is not explicitly authenticated. In [4], it is specified that a mutual device authentication shall

be used if the operating environment of the ID-card cannot be entirely trusted (untrustworthy environment). This may be the case in public signature terminals or other devices that cannot provide a trusted channel. In the case of requiring a remote authentication process, a *device authentication* should be performed. After successful mutual device authentication, session keys are available on both sides to be used in subsequent transmissions. The appropriate secure messaging should be in compliance with ISO/IEC 7816-4 [7].

Taking the previous constraints into account, our work aims to design a complete authentication and authorization architecture for an identification system, which represents a more robust and flexible solution in terms of security, with the following features:

- The authentication protocol will be implemented as an integral part of the ID-card, with the goal of isolating the protocol of the implementation in the access terminal (e.g. laptop, desktop PC, PDA, etc.). Therefore, our approach considers an ID-card with autonomy during the authentication process. In other words, the ID-card participates as stand-alone supplicant or claimant, and not relies on the access terminal (i.e. equipment or host providing the card reader) for this functionality.

- Layer 2 authentication based on a network smart card, NSCard, [25]: we propose the ID-card integration in a layer 2 authentication scheme, which is based on EAP protocol. Therefore, a lightweight networking protocol stack is easily supported by the smart card (TCP/IP and upper layers are not required). We define in this paper an EAP-ID method, which refers to a generic authentication method with identification purposes on our architecture.

- End-to-end mutual authentication scheme: the ID-card and the remote authentication server participate as tunnel endpoints. The individual identification is securely performed through such a tunnel.

Additionally, this work assumes an *a priori* untrustworthy environment, where the access terminal is considered as a potential attacker. Therefore, a previous mutual device authentication has been defined in our identification scheme.

In the following section of this paper, a new proposal of an authentication and authorization architecture along with a network smart card, with specific identification purposes (ID-NSCard), are defined.

3 A new Identification System based on ID-NSCards

This paper proposes a new authentication architecture for ID-cards systems based on our network smart card concept. Under this scope, we consider a remote authentication and authorization scheme, where the ID-card adopts the functionality of stand-alone supplicant instead of split supplicant ("split supplicant" means that ID-card and the access terminal (hereafter referred as Access Control Equipment, ACE) cooperate in the authentication process as an unique device). That is why, in our work, the authentication protocol stack is designed as an integral part of the ID-card (atomic design). With this goal, we propose a specific protocol stack for the chip card that participates as actual end in the authentication process with a remote AAA server. This protocol stack is illustrated in Figure 2. The EAP-ID upper layer represents a generic EAP authentication method [27], specifically here designed with individuals identification purposes. Therefore, the EAP-ID method handles the credentials associated to the duplet individual-domain and the related cryptographic algorithms, during the identification process. Usually, most of the robust identification schemes require a password/PIN for controlling a private key, which is associated to a public key authentication certificate (e.g. X.509v3 certificates) and additionally they may require a biometric token.

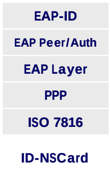

Fig. 2. The protocol stack in the ID-NSCard

Note that in our approach, the goal of a generic EAP-ID method is not to add a new authentication protocol or method but adapts existing authentication protocols used in previous standardized identification schemes. This protocol stack is here defined with a general purpose. Hence, in this section we refer to an "individual", considering that she could be both an user/member registered with an organization and a citizen in a governmental domain. In section 5, the implementation of the authentication method in Spanish National Electronic Identity Card (named DNI-e) as an example of EAP-ID method for citizens is profusely described.

A complete end-to-end architecture is represented in Figure 3. This new architecture introduces significant advantages and requires minimal changes in the network side. Thus AAA proxies keep settings and implementation features. Regarding the Access Control Equipment, ACE, a simple implementation of standardized protocols allows it to behaviour as access point (or NAS, Network Access Server) to the network with pass-through authenticator functionalities. Hence, this equipment must implement EAP and RADIUS client protocols according to [28]. For simplicity's sake, we refer to RADIUS protocol in this paper, but note that a more robust protocol such as DIAMETER [29] could be also implemented in our architecture.

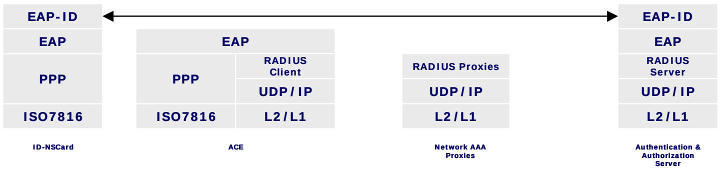

Fig. 3. Authentication and Authorization protocol architecture based on our ID-NSCard

In a first step, the RADIUS server authenticates the Access Control Equipment (ACE) by their own mechanisms. After this step, the functionality of the pass-through authenticator is already shifted to ACE. This reinforces the stand-alone supplicant functionality in the ID-card, since ACE cannot act as supplicant and authenticator at the same time for the same ID-card. One should note the advantages that the ID-card isolation brings with regard to assure the security of the entire scheme in untrustworthy scenarios. More security and trust issues are discussed in section 4 of this paper.

Our architecture takes advantage of the functions of the LCP protocol provided by PPP [30]. LCP/PPP protocol may be easily accommodated in the ID-card stack. The functions for controlling network included in the NCP sub-protocol are beyond the scope of this work. On the other hand, PPP offers versatility in authentication, thanks to its extensibility. In fact, EAP (Extensible Authentication Protocol) was initially designed for PPP. According to our approach, the EAP Layer must be implemented atomically in the smart card and must allow for exchanging of packets between the EAP methods and LCP frames, as well as, for controlling duplicate and retransmissions.

The EAP-ID method should be designed with the goal of security reinforcing. Hence, three phases should be considered in the authentication process. The first one is regarding the mutual authentication between RADIUS entities. The ACE and AAA server proceed with a previous establishment of shared secret keys and mutual authentication process. After this phase, the ACE is allowed to perform access control functionalities as an authenticator (pass-through) in the EAP scheme and the communication between ACE and the remote server will be protected. Hence, the second phase should be oriented to establish a end-to-end secure tunnel between the actual ID-NSCard and the authentication remote server. Obviously, such a tunnel establishment should take place after a mutual device authentication. That means that both devices (ID-NSCard and server) must posse their own authentication tokens (and independently on user credentials). Basically, two key mechanisms could be applied in this step: shared secret keys or public key certificates. The latter requires the usage of card verifiable certificates (according to ISO 7816). In this work, we describe a common end-to-end tunnel establishment, which uses shared secret keys. A generic EAP-ID must be able to perform the following protocol:

Assume that ID-NSCard (C) and authentication server(S) know the 3DES encryption key k_{ENC}, and the MAC computation key, k_{MAC}

$C \rightarrow S$: SN_C $||RND_C$, the 8-bytes serial number SN_C unequivocally associated to C and a fresh 8-bytes random number RND_C.

$S \rightarrow C$: authentication cryptogram ACG1, as function among others of SN_C and RND_C

C: verifies ACG1 (S is authenticated), generates the send sequence counter SSC_C and derives the session key K_{SK}

$C \rightarrow S$: authentication cryptogram ACG2

S: verifies ACG2 (C is authenticated), generates the corresponding SSC_S, also derives the session key K_{SK}

$C \leftrightarrow S$: further communication is protected by a secure channel (K_{SK} encryption)

Once the mutual device authentication is successful and the secure channel is established, the individual (cardholder) is required to be identified by means of her associated identity credentials in the third phase. Therefore, the procedure through the tunnel continues as follows:

Assume that a X.509v3 public key certificate ($Certificate_I$) and the corresponding private key K_{rI} with authentication purposes have been issued for an individual I.

$S \rightarrow C$: challenge T

I: cardholder is required for entering the corresponding password to sign

$C \rightarrow S$: $Sign[K_{rI}](T)||$ $Certificate_I$

S: verifies the digital signature, $Verify[K_{uI}](Sign[K_{rI}](T))$, and the identity of individual I is authenticated by server S.

Derived from this protocol and our architecture (Figure 3), an authentication message exchange has been designed in our work. An example is described in section 5 of this paper.

4 Notes about the Testbed

A testbed of our authentication and authorization architecture for a identification system has been developed. The back-end authentication server is basically implemented in a computer where freeRADIUS [31] is running, which provides API support both EAP/RADIUS and EAP methods development. Additionally, it implements a set of state machines of EAP (Extensible Authentication Protocol), for an EAP backend authenticator. The EAP API is extended in order to support EAP-ID as a new authentication method including the corresponding method state machine and message parsing. On the other hand, the OpenSSL library includes a general purpose cryptography library, which is partially included in this testbed with the goal of providing well-known cryptographic functionalities.

Multiple network AAA proxies could intermediate between the ACE and the authoritative server. Our testbed considers just one proxy, which simulates one of these entities. The standard RADIUS protocol procedure in a relay version allows us to complete the implementation of the adequate protocol stack in an IEEE 802.11 wireless access point. The Access Control Equipment, ACE, is implemented by a common laptop with an IEEE 802.11g wireless interface. The functionality of RADIUS in this equipment is performed by JRadius-Client [32], a Java version of a NAS Client. The technical challenges for rolling out commercial Access Control Equipments with these features have been easily responded. Note that many of the current PDAs and smart phones with ISO-7816 interfaces are programmable. This protocol functionality that is proposed in our work could be transparently implemented as a library (e.g. dll dynamic library) for the OS. Therefore, the impact in existing terminals is minimized and a potential large-scale deployment is clearly feasible.

The bulk LCP/EAP protocol stack -according to the standardized state machines- has been implemented in a G&D Sm@rtCafé Expert 3.x smart card and it has been enhanced with the corresponding EAP-ID method functionalities.

5 Security and Trust Model discussions

Regarding to the security aspects of our architecture, it should be noted that we are not proposing a completely new authentication protocol in the context of identification systems. Our architecture is designed by well-known protocols that are implemented inside the ID-card with a novel approach.

Nevertheless, this new architecture determines a new way to transport authentication messages between the ID-NSCard and the authentication and authorization server, and where the ID-NSCard takes the control in the user side. Therefore, the security weakness and threats are derived by the actual nature of such standardized protocols and the correctness of their implementation.

Additionally, new secure algorithms, key material or cryptographic techniques are not required. The implementation of the algorithms and authentication mechanisms is transparently reused [4, 5], in both sides. However, one of the more important impacts of our proposal is related to the trust models. If we study the trust model derived from the current scenario detailed in Figure 1, we observe that there exists an explicit trust between the PC and the authentication server (supported by SSL protocol). In any case, the trust relationship in the interface between access terminal (i.e. PC) and ID-card is not questioned and it could be considered as "blind". As we mentioned before, this assumption should not be applied to all scenarios and a more flexible solution is required. With this goal, we have introduced a more robust architecture, which a new trust model is derived from. Therefore, it could be adapted to multiple wired/wireless scenarios, even in mobility situations.

In our trust model, the trust relationship between the access terminal (ACE) and the authentication server is supported and protected by RADIUS protocol and such a trust relationship should be considered as explicit. Here, the ACE is part of the network and it behaves as an access point for the ID-NSCard. The trust relationship between ID-NSCard and ACE should be a priori null (untrustworthy). After an end-to-end successful authentication process (supported by an EAP-ID method) between the ID-NSCard and the authentication server, the trust relationship between them should be then considered explicit, since it is a mutual device authentication process. Therefore, in this step the trust relationship between ID-NSCard and ACE is implicit, since any direct mutual authentication process between them has not occurred. In other words, *iff* ID-NSCard trusts authentication server then the former trusts access terminal. This is a reasonable result in a priori untrustworthy scenarios.

After this step, the environment should be considered as trustworthy and just in these conditions the individual identification should be securely performed.

6 Application of the ID-NSCard to the Spanish Electronic ID card

The application of the ID-NSCard to Spanish National Electronic ID card has been studied in our work. The Spanish National ID card aims to prove digitally the identity and other personal data of the owner by means of an authentication process and validating the integrity and signature of signed documents. Both goals are addressed by a chip card and two different public keys created inside. As result, a citizen X.509v3 authentication certificate and a citizen X.509v3 digital signature certificate are manageable by the owner. The policy requirements for implementing this identification system is based on [33, 34]. The security technical specifications are basically gathered from [4, 5]. Additionally, the envisaged services and scenarios where the Spanish National ID-card might be used are determined by the Police Authority, which depend on the Spanish Home Office. The Spanish case could be considered one of the pioneer experiences in EU.

Nevertheless, other potential scenarios could be considered. Suppose a police control (e.g. dangerous or critical transportation, highroads controls, border areas or government facilities, etc.) requires the identification of the individuals. Hence, the authoritative person (let's say a policeman) carries a reduced-size and portable equipment with a wireless interface (e.g. PDA with a chip card reader). This equipment implements the ACE's functionalities described above in this paper. Afterwards, the authoritative person requests to the citizen the National ID card in order to identify her. Such a citizen shows her National ID card, which is a new version based on our ID-NSCard. Thus, a direct and remote identification process supported by our architecture is carried out between her card and the central authoritative services (Authentication and Authorization server). Once the citizen is remotely authenticated, the portable equipment could receive additional authorization information. This circumstance is out of the scope of this work. In Figure 4, the authentication flow derived from the application of our ID-NSCard to the Spanish National identification scheme is represented.

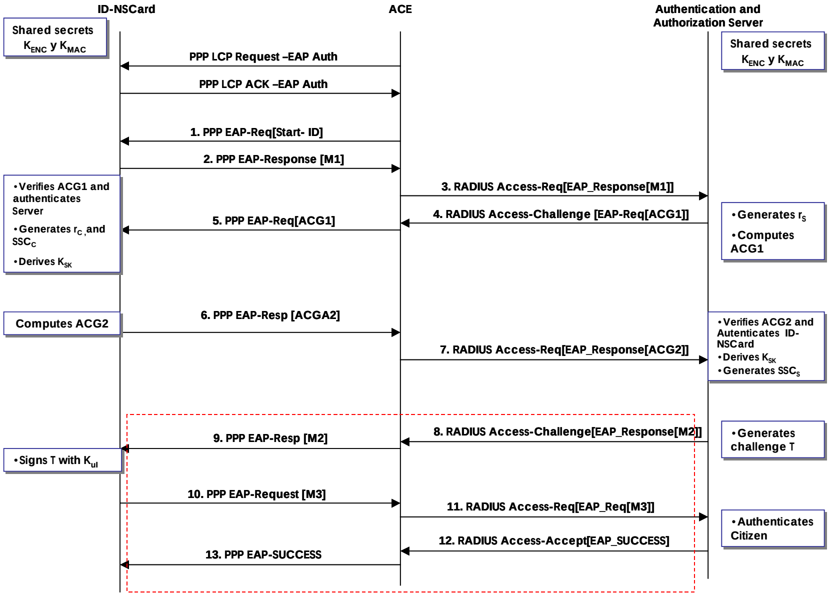
ID-NSCard
ACE
Authentication and Authorization Server
Shared secrets K_{ENC} y K_{MAC}
Shared secrets K_{ENC} y K_{MAC}
PPP LCP Request –EAP Auth
PPP LCP ACK –EAP Auth
1. PPP EAP-Req[Start- ID]
2. PPP EAP-Response [M1]
3. RADIUS Access-Req[EAP_Response[M1]]
•Verifies ACG1 and authenticates Server
•Generates r_C, and SSC_C
•Derives K_{SK}
5. PPP EAP-Req[ACG1]
4. RADIUS Access-Challenge [EAP-Req[ACG1]]
•Generates r_S
•Computes ACG1
Computes ACG2
6. PPP EAP-Resp [ACGA2]
7. RADIUS Access-Req[EAP_Response[ACG2]]
•Verifies ACG2 and Autenticates ID-NSCard
•Derives K_{SK}
•Generates SSC_S
9. PPP EAP-Resp [M2]
8. RADIUS Access-Challenge[EAP_Response[M2]]
•Generates challenge T
•Signs T with K_{ul}
10. PPP EAP-Request [M3]
11. RADIUS Access-Req[EAP_Req[M3]]
•Authenticates Citizen
12. RADIUS Access-Accept[EAP_SUCCESS]
13. PPP EAP-SUCCESS

Fig. 4. Authentication flow with ID-NSCard as Spanish electronic ID-card

In the following paragraphs, consider the nomenclature used in the authentication protocol architecture in Figure 3.

Assume that the ACE has been correctly authenticated by the RADIUS infrastructure and that ACE and authentication server share a static key with the goal to protect their communications. Firstly, the EAP layer is activated both in the ID-NSCard and ACE by means of PPP/LCP configuration messages. Afterwards, the Spanish citizen authentication based on the ID-NSCard should take place as follows (for simplicity's sake, phase 1 is skipped):

Phase 2:

1. The ACE sends a PPP-EAP Start ID message to the ID-NSCard, in order to initiate an identification procedure based on EAP-ID.

2. The ID-NSCard returns the PPP-EAP_Response [M1] packet to the ACE, such that

$M1 := SN_C \,||RND_C$

3. The ACE encapsulates this message into a RADIUS Access-Request packet and afterwards sends it to the Authentication Server, in the back-end network.

4. Upon received M1, the Authentication server generates a random number r_S and she initiates the device authentication process by responding to the ACE with a RADIUS Access-Challenge [EAP_Response[ACG1]], such that

$r_S :=$ *32-bytes random number*
$S := RND_S \,||\, SN_S \,||\, RND_C \,||\, SN_C \,||\, r_S$
$ACG1 := E[K_{ENC}](S) \,||\, MAC[K_{MAC}](\, E[K_{ENC}](S)\,)$

5. The ACE processes the RADIUS headers and sends the received EAP packet to the ID-NSCard, encapsulated into a PPP frame.

6. After a successful verification of the authentication cryptogram ACG1 (server is authenticated), ID-NSCard generates a random number r_c and *SSCc*, and derives session key K_{SK}. Afterwards, she returns the PPP-EAP _Response [ACG2] packet towards the server, such that

$r_C :=$ *32-bytes random number*
SSCc := send sequence counter
$R := RND_C \,||\, SN_C \,||\, RND_S \,||\, SN_S \,||\, r_C$
$ACG2 := E[K_{ENC}](R) \,||\, MAC[K_{MAC}](\, E[K_{ENC}](R)\,)$
$K_{SK} := r_S \oplus r_C$

7. The ACE builds the corresponding RADIUS-Access Request packet and sends it to the authentication server.

After a successful verification of authentication cryptogram ACG2 (ID-NSCard is authenticated), the authentication server generates *SSCs* and derives the session key K_{SK}, such that

$K_{SK} := r_S \oplus r_C$

In this step, both devices are mutually authenticated (without cardholder participation) and they posse the session key K_{SK}, which allows them to encrypt the following communication in an end-to-end secure tunnel.

Phase 3:

8. The Authentication Server continues with this phase by sending RADIUS Access-Challenge[EAP_Response[M2]], where T is the actual protected value for challenging to the citizen. In our proposal T is a 32-bytes challenge.

$T{:=}SN_C||RND_C||RND_S$
$M2{:=}E[K_{SK}](T)$

In order to provide end-to-end EAP per-packet integrity protection, note that M2 should also include the encryption of the 4-octet EAP headers (i.e. Code, Identifier and Length) and not only the encryption of the challenge T. All this information is carried in the Data field of the corresponding EAP packet. Consequently, ID-NSCard could check if the EAP headers re-transmitted by ACE correspond to the EAP headers sent by the remote authentication server.

9. The ACE processes the RADIUS headers and transmits the received EAP packet to the ID-NSCard, which is able to decrypt the message and to obtain the challenge T.

10. Once the ID-NSCard checks the freshness of T, the cardholder's password (and optionally the biometric token) is required with RSA digital signing purposes. The ID-NSCard responds to the previous challenge in a PPP-EAP Request [M3], such that

$M3{:=}\ E[K_{SK}\ ](Sign[K_{uI}\ ](T))||\ Certificate_I$

As in step 8, encryption of the 4-octet EAP headers should be also included in the EAP Data field.

11. The ACE builds the RADIUS Access Request message and forwards it towards the Authentication server.

Authentication server validates the citizen's certificate and verifies her RSA signature. In this step, such a citizen is identified.

12. In case of a successful authentication, a information message is sent to the ACE and afterwards (step 13) to the smart card.

Further authorization decisions and potential services (e.g. re-authentication procedures) are out of scope of this work. With this case of study, we have shown how our authentication architecture with ID-NSCards is applied in standardized identification systems. Obviously, this architecture is easily applicable to similar national or international identification schemes, as well as, to many organizational identification systems.

7 Conclusions

Many countries are starting to issue national identity cards or electronic passports that include a chip card. This is an effective electronic way to hold a trusted identity credential by their citizens. Our work has proposed a new approach based on network smart cards with specific identification purposes, ID-NSCards. This device participates in an authentication architecture, which allows us to transport securely authentication messages between such a device and the remote authoritative server. This solution provides flexibility and robustness versus the common scheme, since the smart card behaves as an autonomous authentication supplicant, independently on the access terminal and on the characteristics of the scenario. Additionally, this solution transparently reuses the envisaged standardized authentication mechanisms for European electronic ID-Cards. Some notes about our testbed are provided and as example of application, our architecture is applied to a version of the Spanish electronic ID-Card based on our ID-NSCard. As result, multiple wired/wireless practical scenarios of utilization (both organizational and governmental) are foreseen for next future work.

References

1. EU Directive 1999/93/EC of the European Parliament and the Council of 13 December 1999 on a Community framework for Electronic Signatures, December 1999.
2. ETSI TS 101 862 v. 1.3.2: Qualified Certificate Profile, June 2004.
3. Santesson, S., Nystrom, M., Polk, T., Internet X.509 Public Key Infrastructure: Qualified Certificates Profile, IETF RFC 3739, March 2004.
4. CEN/CWA 14890-1, Application Interface for smart cards used as Secure Signature Creation Devices - Part 1 - Basic requirements, 2004.
5. CEN/CWA 14890-2:2004; Application Interface for smart cards used as Secure Signature Creation Devices - Part 2 - Optional Features, 2004.
6. CEN/CWA 15264:2005, Architecture for a European interoperable eID system within a smart card ; User Requirements; Best Practice Manual for Card Sheme Operators Part 1 to 3, 2005.
7. ISO/IEC 7816-4: Identification cards - Integrated circuit(s) cards with contacts, Part 4: Interindustry commands for interchange, 2005.
8. Rees, J. and Honeyman, P., Webcard: a Java Card web server. In Proc. of 4th IFIP Smart Card Research and Advanced Application Conference, CARDIS '00, Bristol, U.K., 2000.
9. 3GPP TS 31.111 V7.5.0, Specification of the SIM Application Toolkit (SAT) for the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface, September 2006.
10. Guthery, S, Kehr, R. and Posegga, J., How to Turn a GSM SIM into a Web Server. Projecting Mobile Trust onto World Wide Web. In Proc. of 4th IFIP Smart Card Research and Advanced Application Conference, CARDIS '00, Bristol, United Kingdom, 2000.
11. Urien, P., Internet card, a smart card as a true Internet node. Computer Communications, Vol. 23, Issue 17, pp. 1655-1666, 2000.
12. Postel, J., Transmission Control Protocol, IETF RFC 079, September 1981.
13. Itoi, N., Fukuzawa, T. and Honeyman, P., Secure Internet Smartcards. Java on Smart Cards: Programming and Security, First International Workshop, JavaCard 2000, LNCS 2041, Cannes, France, September 14, 2000.

14. Donsez, D., Jean, S. And Lecomte, S., Turning Multi-Applications Smart Card Services Available from Anywhere at Anytime: a SOAP/MOM approach in the context of Java Cards. In Proc. of Smart Card Programming and Security Conference. e-Smart 2001, Cannes, France, 2001.
15. Chan, A. T., Tse, F., Cao, J., and Leong, H. V., Distributed Object Programming Environment for Smart Card Application Development. In Proc. of the Third international Symposium on Distributed Objects and Applications (September 17 - 20, 2001), IEEE Computer Society, 2001.
16. Chan, A., Tse, F., Cao, J. and Leong, H.V., Enabling Distributed Corba Access to Smart Card Applications. IEEE Internet Computing, Vol. 6, No. 3, pp. 27-36, May-June 2002.
17. Chan, A.T.S, Cao, J., Chan, H.and Young, G.H., A web-enabled framework for smart card applications in health services. Communications of the ACM, Vol. 4, No.9, pp. 76-82, 2001.
18. Chan, A.T.S, Mobile cookies management on a smart card, Communications of the ACM, Vol. 48, No. 11, pp. 38 - 43, 2005.
19. IST Project RESET, Roadmap for European Research on Smartcard related Technologies, IST-2001-39046: Final Roadmap v.5, May 2003.
20. Montgomery, M., Ali, A. and Lu H.K., Secure Network Card. Implementation of a Standard Network Stack in a Smart Card, In Proc. of 4th IFIP Smart Card Research and Advanced Application Conference, CARDIS '04, Toulouse, France, Kluwer Academic Publishers, August 23-26, 2004.
21. Lu, H.K. New Advances in Smart Card Communications, International Conference on Computing, Communications And Control technologies (CCCT), Austin, TX, USA, August 14-17, 2004.
22. Lu H.K. and Ali A., Prevent On-line Identity Theft - Using Network Smart Cards for Secure On-line Transactions. In Proc. of 7th International Conference on Information Security, ISC '04, Palo Alto, CA, USA, September 27-29, 2004.
23. Ali, A., Lu, K. and Montgomery, M., Network Smart Card: A New Paradigm of Secure On-line Transactions, In Proc. of Security and Privacy in the Age of Ubiquitous Computing, IFIP TC11 20th International Conference on Information Security (SEC 2005), May 30 - June 1, 2005, Chiba, Japan.
24. Lu, H.K., Multi-stage Packet Filtering in Network Smart Cards, In Proc. of 6th IFIP Smart Card Research and Advanced Application Conference, CARDIS '06, Tarragona, Spain, LNCS 3928, pp. 192-205, 2006.
25. Torres, J., Izquierdo, A. and Sierra, J.M., Advances in network smart cards authentication, Computer Networks, Vol. 51, Issue 9, pp. 2249-2261, June 2007.
26. Lu, H.K., Network smart card review and analysis, Computer Networks, Vol. 51, Issue 9, pp. 2234-2248, June 2007.
27. Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., Levkowetz, H., Extensible Authentication Protocol (EAP), IETF RFC 3748, Standards Track, June 2004.
28. Aboba, B., Calhoum, P., RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP), IETF RFC 3579, September 2003.
29. Eronen, P., Hiller, T., Zorn, G., Diameter Extensible Authentication Protocol (EAP) Application, IETF RFC 4072, August 2005.
30. Simpson, W., The Point-to-Point Protocol (PPP), IETF RFC 1661, Standard Track, July 1994.
31. FreeRADIUS, available on http://www.freeradius.org, GNU General Public License.
32. JRadius-Client, SourceForge Project, available on http://jradius-client.sourceforge.net
33. ETSI TS 101 456 v.1.2.1, Policy Requirements for certification authorities issuing qualified certificates, April 2002.
34. ETSI TS 102 042 v.1.1.1, Policy Requirements for certification authorities issuing public key certificates, April 2002.