Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 29 (2006) 2781–2787
computer communications
www.elsevier.com/locate/comcom

Towards self-authenticable smart cards ☆

Joaquin Torres *, Antonio Izquierdo, Jose M. Sierra, Arturo Ribagorda

Universidad Carlos III de Madrid, Spain

Available online 7 December 2005

Abstract

Traditionally, the smart cards have been seen as security devices, but as soon as they could be integrated into distributed and networked environments their vulnerabilities could be attempted and countermeasures against new security threats in an open-access internet were required. In this work, our target could be represented by an end-to-end mutual authentication scenario where the smart card could authenticate by itself to a Network Access Server by means of link layer protocols and therefore in absence of IP connectivity. Some previous related models based on the Extensible Authentication Protocol are presented. However, in these works the smart card and terminal implement jointly the supplicant functionality (split supplicant). We consider the native EAP multiplexing model specified by the IETF to propose a new approach in order to avoid this split and to achieve an autonomous and highly independent smart card in the authentication scheme: a self-authenticable smart card.

Keywords: Smart card security; Authentication protocol; Non-split supplicant

1. Introduction

Probably motivated by the ubiquitous computing concept, proposed by Mark Weiser [1] at the beginning of 90s, multiple research efforts have been addressed towards the goal for converting the smart card into a device open to world-wide networking, and consider it as a *full-complaint* internet host in accordance with standards [2].

Early cellular networks pushed the chip card technologies by means of the identity modules (SIM) used in an enormous amount of cellular phones. Later, the open systems betted on a wide cross-platform deployment like Java Card [3]. Both of them were important steps in the maturity process of the networked smart card but an exciting roadmap was already expected [4]. Due to progress on semiconductors manufacturing that allowed to augment integration capabilities and upgraded hardware security features [5,6], it was predictable that these devices would adopt a relevant role beyond simple identification tokens or credentials storage: a secure internet node carried in our wallet. Works like [7,8] were focused towards this goal, where a simplified TCP/IP protocol stack was suggested. Around those years, a different approach based on a TCP-like protocol but not standards full-complaint was proposed in [9] and afterwards the proxy approach was revisited [10,11]. From other perspectives, advances and challenges on operating systems [12,13] and on communication protocols [14] were foreseen.

The USIM/SIM technology was not unconnected to these advances. Recently, related technologies along with new business perspectives have arisen providing to some extent a proactive chip card (GSM 11.14, SIM Toolkit) and upgraded connectivity on a client/server over-the-air architecture (OTA). The SIM Application Toolkit extends the communication protocol between the card and the handset. With SIM Application Toolkit, the SIM card has a proactive role and it is able to send commands to the handset, driving its interface and the connection to the network. Over-the-air (OTA) is a technology based on Short Message Services and it is used to communicate with a SIM card inserted in a telephone handset. OTA is built on a client/server architecture where the server-side

☆ This work is supported by the ASPECTS-m Project, which is funded by the CICYT.
* Corresponding author.
E-mail addresses: jtmarque@inf.uc3m.es (J. Torres), aizquier@inf.uc3m.es (A. Izquierdo), sierra@inf.uc3m.es (J.M. Sierra), arturo@inf.uc3m.es (A. Ribagorda).

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.10.029

part is an operator back-end system. OTA applications allow to update data in the SIM card as well as to download and activate new services.

Nowadays, important manufacturers advertise for internet smart cards incorporating secure protocols such as SSL/TLS, applications such as web server and communication protocols such as USB 2.0.

Considering the principles of security design, it seems reasonable to extend security to the whole smart card communication system and not just bounded to transport and network layers but furthermore to physical and data link layers. At the lower layers, the smart cards implement mechanisms in order to authenticate the terminal and vice versa. Recently, numerous works concerning remote authentication with smart card have been and are being developed [15,16], and even questioned [17]. Note that these approaches are focused on user authentication or terminal authentication but not so many efforts have been addressed on authenticated and authorized access to networks from the chip card point of view.

In our approach, we consider the potential scenario where the smart card could authenticate itself to the Network Access Server by means of layer 2 protocols and therefore in absence of IP connectivity. Thus, the smart card behavior could be seen, for instance, like a laptop trying to get authenticated and authorized access to wired or wireless LANs, and the terminal could be seen like an access point (AP) to such network. Motivated by the massive deployment of WLANs and mobile systems (3rd and 4th generation), authentication protocols and related mechanisms and infrastructures are being widely studied in order to guarantee secure access networks [18,19]. As shown in these works, one of these emerging frameworks is the Extensible Application Protocol (EAP) [20] of which goal is to extend and improve the authentication spectrum carried out by link layers, originally PPP (PAP, CHAP, MS-CHAPv2) and later adapted to provide service to IEEE 802 networks.

EAP isolates the authentication tasks at the lower layers and allows to improve these tasks by means of additional standardized methods without IP connectivity. Therefore it is well-suited to authenticate devices. EAP is not a protocol to transport data but an architecture to transport authentication protocol packets. For instance, the TLS protocol provides an authentication service based on digital certificates. This service is provided by the EAP-TLS protocol [21]. In this case, both ends agree about an authentication method by means of EAP messages. In Fig. 1, the EAP Multiplexing Model described in RFC 3748 is shown. More details about this model can be found along such document.

Our work aims to provide a *networked* smart card with an adequate EAP implementation according to the model in RFC3748. Our target layout could be represented by an end-to-end mutual authentication scenario, compounded by a self-authenticable smart (host), which wants to gain access to a network. At the opposite end, an authentication server completes the scheme.

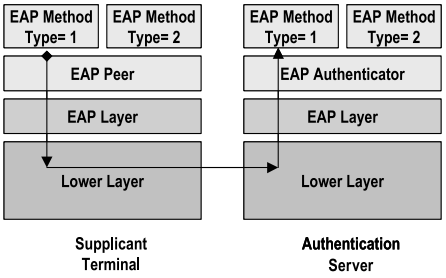

Fig. 1. EAP multiplexing model in RFC3748.

Next section introduces the related works where the EAP models on smart cards are considered from different approaches. Applied model deficiencies and inconveniences to suite them to our goal are identified. In Section 3, our better suited model is proposed. Section 4 is devoted to some considerations about our model implementation. In Section 5, threats analysis on the model are raised, and finally conclusions of our work are exposed.

2. Related work

2.1. EAP support in smart card

The work detailed in [22,23] aims to integrate smart cards into a mutual authentication process for wireless LAN access, based on EAP over 802.1X and RADIUS. It takes as reference a slightly new EAP multiplexing model which is described in the internet draft "EAP-Support in Smartcard"[1] realized by the same author, P. Urien. This variant of the model is clearly inspired of the original one in RFC 3748.

Urien's work proposes to include a new entity that communicates with the smart card in the EAP Multiplexing Model depicted in Fig. 1. This smart card interface cooperates with an EAP method. This interface could be seen like a software that transparently exchange EAP messages between the terminal and the smart card. Additionally, this software could be in charge of some computation tasks (non-security related) giving support to the smart card which could store multiple identities and adequate credentials associated to each EAP method. Therefore, according to RFC3748 the most important difference in this proposal is that the supplicant is partially implemented in the smart card (split supplicant), since RFC3748 allows an flexible and non-mandatory implementation. The EAP Multiplexing Model taken as reference in [22] is shown in Fig. 2. Note that for simplicity just the EAP peer layer in the terminal is represented here.

Regarding to the scenario proposed in Urien's work, the smart card does not aim to be an internet host and highly independent of the terminal in the authentication process. Here, the Card Acceptance Device or terminal is presented

[1] At the time to write down this paper, the last "EAP-Support Smart Card" release is *draft-urien-eap-smartcard-08.txt.*