NETWORK SMART CARD

A New Paradigm of Secure Online Transactions

Asad Ali, Karen Lu, and Michael Montgomery
Axalto, Smart Cards Research, 8311 North FM 620 Road, Austin, TX 78726, USA

Abstract: This paper describes the functionality and practical uses of a network smart card: a smart card that can connect to the Internet as a secure and autonomous peer. The network smart card does not require any special middleware on the host device. It uses standard networking protocols PPP and TCP/IP to achieve network connectivity. Network security is accomplished by an optimized SSL/TLS stack on the smart card. The combination of TCP/IP and SSL/TLS stacks on the smart card enables the smart card to establish a secure end-to-end network connection with any standard (unmodified) client or server on the Internet. This opens the door to seamless, secure and novel applications of smart cards in the most ubiquitous network: the Internet. Some of these applications that use the network smart card in confidential online transactions are explained.

Key words: Internet; smart card; network; TCP/IP; SSL/TLS; secure online transaction.

1. INTRODUCTION

Smart cards have been in use for more than two decades. However, their use has so far been on the fringes of mainstream computing. Smart card advantages such as security, portability, wallet-compatible form factor, and tamper resistance make them increasingly useful in many applications. Smart cards are used as agents in off-line transactions, as tokens for controlling access to physical resources, as SIM units in GSM phones, and as secure tokens for storing confidential and personal information. Current applications of smart cards, though useful in their own right, are hindered by the mismatch between smart card communication standards and the communication standards for mainstream computing and networking. When

smart cards are connected to computers, host applications cannot communicate with them using standard mainstream network interfaces. Specific hardware and software in the form of smart card reader device drivers and middleware applications are needed to access the smart card services.

The network smart card solves this mismatch of communication standards by implementing standard mainstream networking protocols on the smart card. It supports PPP [1] and TCP/IP [2,3,4,5] as the underlying communication layer. It also supports the SSL/TLS [6,7,8] protocol that adds application level network security. Using this combination of TCP/IP and SSL/TLS protocols, which are ubiquitous in the Internet, a variety of mainstream application frameworks can be supported by the smart card. Currently, a working prototype of the network smart card supports a Telnet server and a secure web server using HTTPS. To the host device or a remote application, the smart card appears as another PC on the network. This seamless integration of network smart card with existing PC applications could pave the way for an unprecedented growth in the use of smart cards by enhancing the security of online web transactions.

2. MOTIVATION

Smart cards are extremely secure hardware tokens with a programmable microprocessor chip. Although they are useful in a variety of security critical applications, there are two main impediments to widespread acceptance of smart cards for use in online web transactions. The first is the necessity to install middleware on the host PC where the smart card is physically connected. Without this middleware, smart card services cannot be accessed. The second is the lack of end-to-end security between the smart card and any remote application: for example, an online merchant's web server. A key motivation for developing the network smart card was to address both these drawbacks, and thereby achieve the vision of widespread smart card acceptance.

The root cause of both these drawbacks is the communication protocol mismatch between smart card and the host PC. Current smart cards use APDUs and smart card-specific ISO 7816 standards for communication. Host PCs, on the other hand use standard mainstream network standards like PPP and TCP/IP. Middleware software is installed on the host PC to act as a bridge between these two sets of diverse protocols. This software is not only cumbersome to develop, install and maintain across multiple host platforms,

it also breaks the end-to-end security model when smart cards are accessed from remote applications. Figure 1 illustrates this break in security as a conventional smart card is accessed from a remote PC.

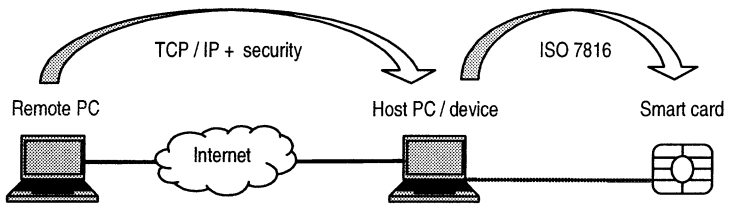

Figure 1. Network connectivity of conventional smart cards.

The remote PC cannot directly communicate with the smart card. It first connects to the host PC using standard mainstream network and security protocols. The host decrypts the information and then passes it to the smart card using ISO 7816 protocols. Due to this protocol conversion, the host PC needs to be trusted. However, PCs are known for their vulnerability to hardware as well as software attacks. As such, even if the ISO 7816 communication is encrypted, the host PC becomes a weak link in the end-to-end security between the remote PC and the smart card.

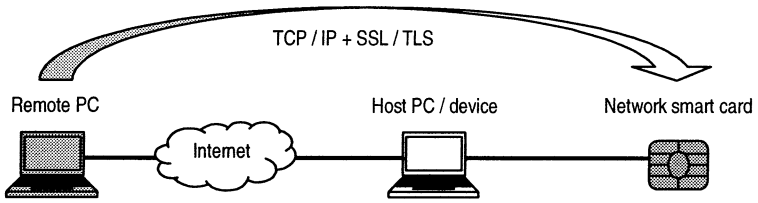

Figure 2. Network connectivity of the Network Smart Card.

Figure 2 describes the same scenario of accessing a smart card from a remote PC using the network smart card. Since the network smart card supports the same mainstream communication protocols as the remote PC, there is no need for any protocol conversion at the Host PC. There is a direct secure connection using the SSL/TLS protocol from the remote PC to the network smart card. The host PC merely acts as a pass-through router and need not be trusted. The OS on the host PC provides services that allow it to be configured as a pass-through router. This eliminates the need for installing any smart card specific middleware. The use of network smart card

in an end-to-end secure connection with a remote client creates a much more secure model of conducting online transactions than is possible with current approaches that either do not use smart cards, or use them as auxiliary tokens connected to the host PC.

3. RELATED WORK

For years, efforts have been underway to connect smart cards to the Internet. Early pioneers [9,10,11,12] have made major contributions to enhancing the connectivity paradigm of smart cards. However, they did not make the smart card a truly independent node on the Internet. These developments relied on some form of host middleware or software service to be installed on the host machine. In addition, they did not support any mainstream network security protocol on the smart card. The network smart card addresses these shortcomings in the earlier efforts by implementing a standard network protocol stack on the smart card. With this approach the smart card can establish an end-to-end secure connection with standard unmodified remote applications. In addition, the use of smart card does not require any smart card specific middleware to be installed on the host PC. The freedom to use the smart card on any PC without requiring smart card specific software on the PC is one of the major contributions of the network smart card.

4. ARCHITECTURE & DEPLOYMENT

The architecture of the network smart card is based on two key principles: the use of standard TCP/IP and SSL/TLS protocol stacks inside the smart card; and utilizing standard interfaces and drivers built into most operating systems, thereby eliminating the need to install smart card specific software. This allows the network smart card to seamlessly integrate in the existing mainstream computing infrastructure. The deployment of smart cards is no longer encumbered by the middleware. Figure 3 shows some of the ways a network smart card can be connected to the network where unmodified clients can seamlessly access it. The card can be used in the standard credit card size form factor when connected to a host computer or a smart card hub. It can also be used in the SIM form factor when placed in GSM phones and wireless PDAs.

Figure 4 shows details of the network smart card protocol stack. The network smart card is connected via the "direct connection to a host

computer" deployment option. It contains a complete network stack consisting of PPP, TCP/IP, and SSL/TLS, and various network applications.

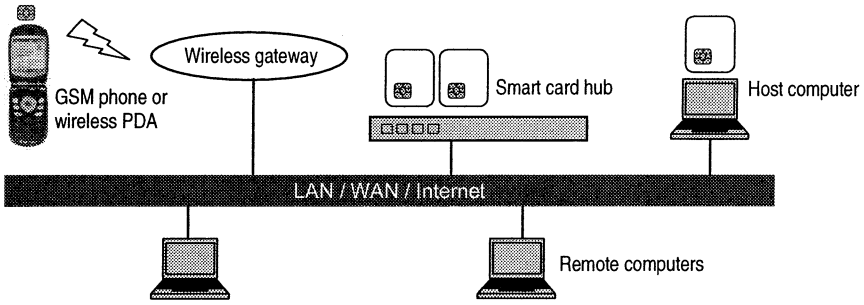

Figure 3. Deployment options for network smart card in various form factors.

The host computer can be any platform that is configured to permit network access from a serial or USB port. This includes most workstation, desktop, and laptop platforms including Windows, Mac, Linux, and Unix platforms, as well as some mobile devices. In the case of Windows platforms, configuration is a simple task requiring a few simple steps via the New Connection Wizard (a standard utility that comes with all Windows operating systems). This establishes a direct connection to another computer. The host is unaware that the computer being connected is a smart card: it treats the smart card as any other computer requesting a connection. Specific details of the network smart card architecture and various

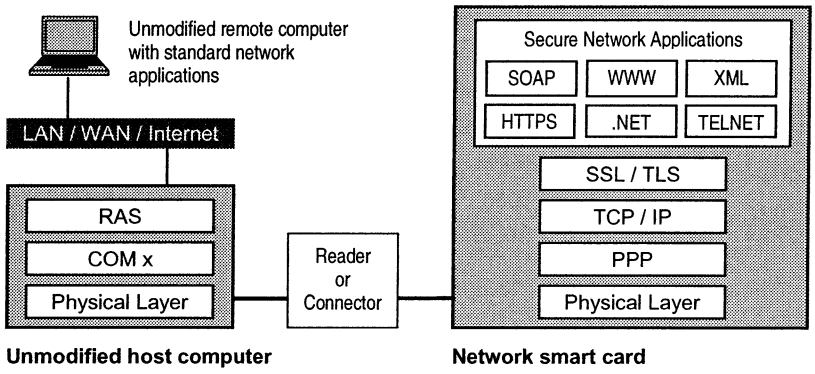

Figure 4. Network protocol stack and connection through a host computer.

connectivity models are described in a separate paper [13].

The host computer functions simply as a router to connect the smart card to the network, where other remote computers may access it. Since the smart card has its own IP address, any remote client or server anywhere on the network can securely communicate directly with this card using standard network applications. As far as the remote computer can tell, the smart card is just another standard computer on the Internet. This kind of seamless network interaction with unmodified remote applications is not possible with conventional smart cards. In figure 4, the solid line boxes (e.g. TCP/IP, SSL/TLS) represent components and services that have already been implemented in the network smart card, whereas the dotted line boxes (e.g. SOAP) are application frameworks that can easily be added using the underlying communication layers.

5. APPLICATION FRAMEWORKS

A variety of standard application frameworks can be supported on top of the underlying implementation of TCP/IP in the network smart card. Applications running on the smart card do not communicate directly with the TCP/IP layer. Instead they go through a socket interface layer. Since this is how mainstream applications on a PC connect to the network, applications on the smart card can seamlessly integrate with other remote applications on the network. Standard unmodified client applications can connect to the smart card services as if they were connecting to another computer on the network.

The utility of this seamless integration is further augmented by the SSL/TLS layer, which adds application level network security to all remote connections to the smart card. SSL and TLS are the de-facto standards for securing communication between web servers and web clients (the browsers). These protocols have been in use for several years and no critical flaws have been discovered in them. Adding these protocols to the smart card substantially increases the security of network access to smart cards. The SSL/TLS library developed for the network smart card provides a simple API that can be used by on-card applications to establish secure end-to-end network connections with any remote unmodified client on the Internet. This provides authentication, confidentiality, and data integrity for all network communication. Figure 5 shows some of the application frameworks that can be supported on top of socket layer and the SSL/TLS layer. Solid boxes represent components that have already been implemented and demonstrated. Other boxes with dotted outlines can easily be added.

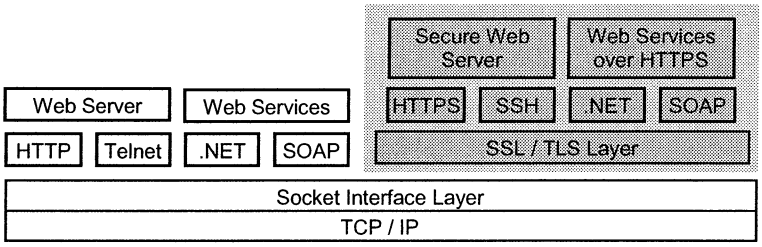

Figure 5. Application frameworks that can be supported in network smart card.

The current version of the network smart card supports a Telnet server and a secure web server that can serve static as well as dynamic HTML content. The static HTML content is served by reading the requested file from the smart card file system and returning its contents. The Dynamic content is supported by a more elaborate mechanism. The web server invokes the requested application, or library through a CGI interface and then redirects the resulting output to the browser. This allows the web sever to generate web pages on the fly, and implement various forms of access control for data stored on the smart card. No smart card specific application is needed to access the network smart card services. The user can simply open a standard Internet browser (IE, Netscape, Mozilla Firefox, etc.) and connect to the smart card in a secure manner over an HTTPS connection. From an administrator's point of view, programming the card can be as simple as creating a new HTML file and uploading it to the smart card using the same browser.

The network smart card also supports a Linux-like shell interface that can be accessed either through Telnet or through a secure HTTPS connection. This provides a very powerful way of interacting with the card, both as a user and as an administrator. New application frameworks such as .NET, SOAP, and Web Services can be easily added.

6. SECURE ONLINE TRANSACTIONS

Because the network smart card supports mainstream networking protocols, it can be seamlessly integrated into any network application. One example of such applications is the web-based online transaction. PC users have become accustomed to conducting online transactions as part of their

normal daily activity. However, a closer look at these transactions reveals that they are not as secure as people believe them to be. There are three players in any online transaction: the client application, e.g. a standard Internet browser running on local PC; the remote web server representing the merchant we do business with; and finally, the Internet infrastructure over which our data is carried back and forth between the client application and the remote web server.

Let us analyze these three players. The Internet infrastructure is considered open and any data traversing it can be viewed by anyone who cares to examine the packets. However, security protocols such as SSL/TLS that are used in online transactions encrypt the data in a way that data integrity and confidentiality are not compromised while the data traverses the Internet. The remote web server is under the control of the trusted merchant with which we are conducting our online transactions. We trust the merchant, otherwise, why would we conduct business with them? This leaves us with our local PC on which we run the client application, the Internet browser. PCs are extremely vulnerable to both hardware and software attacks, and therefore form the weak link in any online transaction. Any information that is entered through them can be compromised even before the SSL/TLS layer encrypts it. Similarly, an attacker can capture any data that may have originated on a different device but simply flows through the PC in the clear.

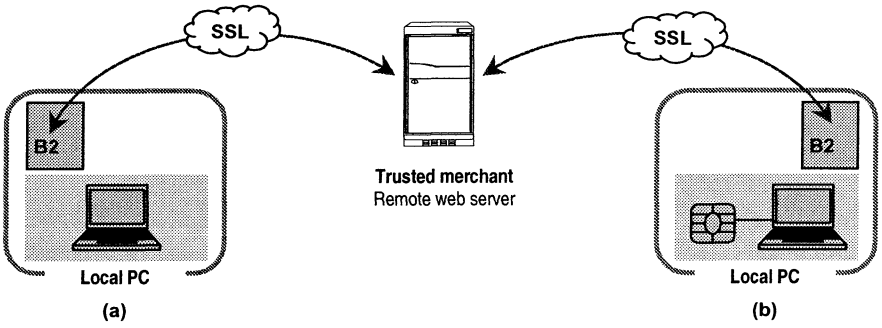

Figure 6. Online transactions: a) without a smart card, b) with a conventional smart card.

Figure 6-a shows this typical scenario of conducting online transactions without any smart card support. Using an Internet browser, B2, the user establishes a secure SSL connection with a trusted merchant. Any confidential data entered by the user is secure while in transit, but is open to attack while it resides on the Local PC. This is because information can be

captured before the SSL layer can encrypt it. Current use of smart cards to augment the security of PCs is described in Figure 6-b. It is a step in the right direction but does not completely solve the security problem. The confidential data now resides on the smart card where it is secure. However, in the process of being SSL-encrypted for transmission to the trusted merchant, the data has to flow through the PC in the clear. Since PCs are open to attack, so is any information that flows through them in the clear. Due to their vulnerabilities, PCs are the weak links in any online transaction.

The use of the network smart card in an online transaction solves the problem introduced by the vulnerable nature of PCs. Instead of passing information through the PC, the network smart card establishes a secure connection directly with the remote server of the merchant. All confidential information is passed from smart card directly to the merchant. The PC is used only as an initial means of connecting the smart card to the Internet. Once that connection is established, the PC merely acts as a router that physically passes encrypted information back and forth between the network smart card and the remote merchant. Logically, the smart card has a direct connection with the remote merchant that completely bypasses the PC.

Figure 7 shows this use of network smart card in conducting a truly secure online transaction. The local PC, the trusted merchant, and the network smart card are three independent nodes on the Internet that are capable of establishing secure connections using SSL/TLS. Any two nodes can communicate with each other without disclosing the contents of the communication to the third node, or any other party that happens to be listening. The fact that network smart card is one of these nodes, is a critical distinction that elevates the security of online transactions to a level not

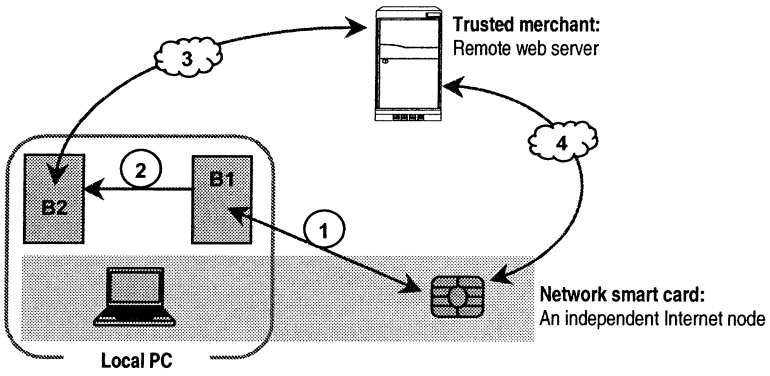

Figure 7. Network smart card in a secure on-line transaction.

possible with conventional smart cards.

Various interactions outlined in Figure 7 are described below. All the bi-directional arrows in this figure represent secure HTTPS connections using the SSL/TLS protocol. The steps listed here represent a simplified set of interactions when using network smart card in an online transaction. More details are covered in a separate paper [14] that talks about the use of network smart card in preventing online identity theft. The sequence of steps in chronological order is as follows:

1. The user opens a web browser on the local PC. This instance of the web browser is referred to as B1. From B1 the user connects to the secure web server running on the network smart card and authenticates himself through some form of card holder verification: a PIN, biometrics, etc. This connection is established over a secure HTTPS link. Once authenticated, he is presented with a list of trusted merchants. The user picks a trusted merchant and asks the network smart card to establish a secure connection with this service provider.
2. The user clicks on a link in B1 to start a new browser. This instance of the web browser is referred to as B2.
3. When browser B2 is launched, it automatically connects to the remote web server of the trusted merchant that was selected by user in step 1. Since the browser was launched from B1, the network smart card address can be passed to the trusted merchant. Alternatively, the smart card can initiate a connection to the trusted merchant and pass its address, as well as other login credentials to it.
4. Regardless of the method used for mutual discovery, the trusted merchant web server and the network smart card can now communicate directly without involving any third party – including the local PC. This direct SSL connection is represented by link 4 shown in figure 7.

Once the mutual discovery and authentication of the network smart card and the trusted merchant is complete, the user can use browser B2 to interact with the web server of the trusted merchant. During this interaction, all confidential data interchange can be deferred to the direct connection between the trusted merchant and the network smart card. Such confidential data can be sent from the network smart card directly to the trusted merchant. The data is as safe during transit as it was when stored on the smart card.

Some scenarios of secure online transactions that have been prototyped with the network smart card are listed in the following sections. These

applications have been demonstrated at various smart card conferences [15,16,17]. See Figure 8 for a sample screen-shot from these demonstrations.

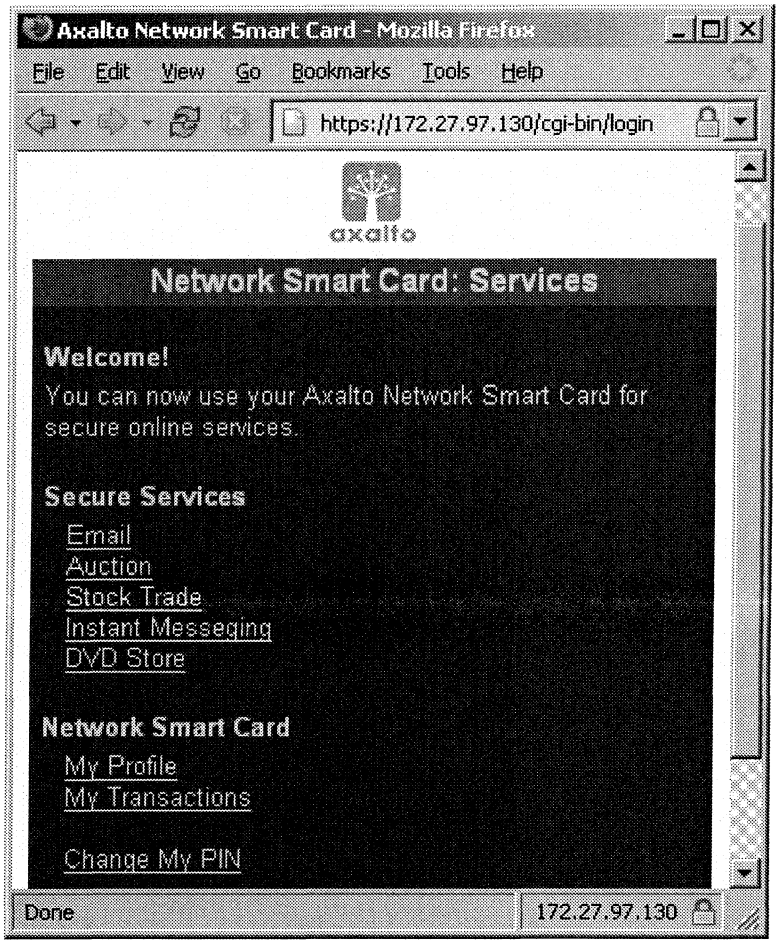

Figure 8. Screen-shot of secure web server running on network smart card.

Figure 8 shows the list of secure services (trusted merchants) that is presented to the user once he has logged on to the network smart card. All information displayed in this browser window is coming directly from the web server running on the network smart card using an HTTPS connection. As evident from the URL, the network smart card has been assigned its own IP address: in this case 172.27.97.130. Standard unmodified Internet client applications can communicate with the smart card using this IP address. The

Mozilla Firefox browser shown in Figure 8 corresponds to the browser B1 in the scenario outlined in Figure 7.

6.1 Online Stock Trade

In this scenario, the trusted merchant is a stock brokerage house where the user wants to sell some stock from his portfolio. As shown in figure 7, the user uses browser B2 from the local PC and selects the stock that need to be sold. As the final submit button is pressed, the brokerage house will not automatically complete the transaction. Instead it would use link 4 to contact the network smart card directly. This is done for two reasons: Firstly, the remote merchant wants to make sure that the network smart card, which represents the user in this online transaction, is still connected to the network. Secondly, it allows the user to manually approve any information that is disclosed by the network smart card. This secure close-back loop is a critical step that is missing in all current on-line transaction scenarios that use conventional smart cards, or are conducted without any smart card support.

6.2 Online Shopping

In the online shopping scenario the network smart card provides a secure way of transferring one's credit card information without having to type it manually. Such confidential details like credit card number and expiration date are stored on the network smart card. Storing this information on a smart card provides a greater level of security than storing it either on the remote merchant server, or on the local PC. Once the user is ready to pay for online shopping, the user instructs the network smart card to send the credit card details to the selected merchant. This secure transfer of credit card information between the smart card and the merchant using an end-to-end SSL connection solves one of the most critical issues with online shopping today. The path of confidential data transfer is shown as link 4 in figure 7.

6.3 Online Auction

The online auction scenario is quite similar to the online stock trade scenario. Instead of trading stocks, the user participates in an online auction by placing bids. However, before the merchant accepts any bids, the user's identity is confirmed by a secure direct connection between the merchant web server and the network smart card. In conventional online auctions, attackers can impersonate the user by compromising the user's username and password. The use of network smart card eliminates such fraudulent bids.

7. CONCLUSION

The network smart card presented in this paper enables smart cards to participate as autonomous nodes on the Internet with no host or remote application changes required. Smart cards may, finally, be accommodated within the existing mainstream computing infrastructure. This innovative technology combines the best aspects of two worlds of computing: the world of PCs, and the world of smart cards. In the world of PCs we have ubiquitous infrastructure access and very strong network security protocols for remote access. However, a PC is a very weak hardware device that is vulnerable to security attacks. Smart cards on the other hand are very secure hardware tokens, but suffer from lack of seamless access to the mainstream communication networks. The network smart card combines the hardware security of smart cards with the ubiquitous access and network security of PC. The result is a new paradigm of portability, enhanced security, and tamper resistance. We foresee this new paradigm triggering an unprecedented growth in the deployment of smart card applications for the Internet and other network environments. The network smart card can bring a level of security to online transactions that has so far not been possible.

REFERENCES

1. Simpson, W. "The Point-to-Point Protocol (PPP)", RFC 1661, July 1994.
2. Postel, J. "Internet Protocol," RFC 791, September 1981.
3. Postel, J. "Transmission Control Protocol," RFC 793, September 1981.
4. Socolofsky, T. "A TCP/IP Tutorial," RFC 1180, January 1991.
5. Almquist, P. "Type of Service in the Internet Protocol Suite," RFC 1349, July 1992.
6. Freier, Alan O., et al. "The SSL Protocol, Version 3.0," Internet Draft, November 18, 1996. Also see the following Netscape URL: http://wp.netscape.com/eng/ssl3/.
7. Dierks, T., Allen, C., "The TLS Protocol, Version 1.0," IETF Network Working Group. RFC 2246. See http://www.ietf.org/rfc/rfc2246.txt.
8. Elgamal, et al. August 12, 1997, "Secure socket layer application program apparatus and method." United States Patent 5,657,390.
9. Rees, J., and Honeyman, P. "Webcard: a Java Card web server," Proc. IFIP CARDIS 2000, Bristol, UK, September 2000.
10. Urien, P. "Internet Card, a smart card as a true Internet node," Computer Communication, volume 23, issue 17, October 2000.
11. Guthery, S., Kehr, R., and Posegga, J. "How to turn a GSM SIM into a web server," Proc. IFIP CARDIS 2000, Bristol, UK, September 2000.
12. Muller, C. and Deschamps, E. "Smart cards as first-class network citizens," 4th Gemplus Developer Conference, Singapore, November 2002.
13. Montgomery, M., Ali, A., and Lu, K. "Implementation of a Standard Network Stack in a Smart Card", CARDIS 2004, Toulouse, France, August 2004.

14. Lu, K., and Ali, A. "Prevent Online Identity Theft - Using Network Smart Cards for Secure Online Transactions," 7th Information Security Conference, Palo Alto, CA, September 2004.
15. Montgomery, M., et al., "Web Identity Card", Axalto booth, CARTES & IT Security 2003, Paris, France, November 2003.
16. Ali, A., et al., "Web Identity Card", Axalto booth, CTST 2004, 14th Annual Conference and Exhibition, Washington D.C., April 2004.
17. Montgomery, M., et al., "Web Identity Card", Axalto booth, CARTES & IT Security 2004, Paris, France, November 2004.