Smart-Card Devices and Applications

January 2001

Smart cards, small portable credit-card shaped devices, each have an internal integrated circuit (IC). The combination of the small size and IC make them valuable tools for security, data storage, and special applications.

Smart cards can provide different security-level options ranging from simple access control to complex data encryption. For example, employees can enter controlled buildings by passing their smart-card badges in front of a reader. Stored keys on smart cards can allow users to log on to networks or send encrypted e-mail messages.

Smart cards offer virtually unlimited application possibilities. Storage capacity is a maximum of 32 kilobytes (KB) per card, but this is more than adequate for storing:

- Personal information
- Electronic purse transactions
- Prepaid telephone transactions
- Personal authentication information
- Personal finance transactions
- Health-care data
- Loyalty program information

Some applications require more powerful smart cards containing memory and logic for handling larger tasks. Microprocessor smart cards run their own operating system (OS). Programmers can develop complex programs in common programming languages and a known application program interface (API). Special smart-card microprocessor applications can target specific tasks, such as launching a support website configured to a specific user's needs.

This article provides an overview of smart-card technology and explores some of the consumer applications currently in use.

Smart-Card Variations

Smart cards are composed of an IC, an interface between the IC and card reader, and a body. Smart cards are differentiated by the IC type, size, and the method of communication with the reader.

Integrated Circuits

Smart-card ICs provide the logic for specific card applications. The ICs are *memory chips* or *microprocessor chips* .

Memory Chips

Smart-card memory chips are used for data storage and identification applications. Data can consist of any information required for transmitting to a specific application. The main use for memory smart cards is to store *keys* and *certificates* for cryptography. Keys function as passwords to secure environments, and certificates verify the authenticity of keys.

Memory smart cards are built with erasable programmable read-only memory (EPROM) or electrically EPROM (EEPROM) chips. EPROM, which can only be changed once, is often used in prepaid service cards such as telephone calling cards that count off minutes used and then are discarded. EEPROM, which can be changed up to 100,000 times, includes built-in logic that can be used to update a counter in prepaid service cards.

A memory chip's architecture (and thus, cost) varies, depending on the application. However, the manufacturer identification (ID) and the application ID fields in the architecture are the same for all memory card chips. The smart-card reader uses these fields to communicate with the card. The application ID includes:

- Card issuer
- Card serial number
- Other user information (depending on the card application)

The serial number is unique for each card. Optional fields on memory chips include counter logic, data, and secret codes or keys. Application developers have options for several memory-card structures to meet design requirements.

Microprocessor Chips

Smart-card microprocessor chips are smaller, slower versions of the central processing units (CPUs) used in PCs (see Table 1).

Smart-Card Microprocessor	PC CPU
8-bit machine	32-bit machine
3.57-MHz speed	Up to 1-GHz speed

Table 1. Smart-Card Microprocessor/PC CPU Comparison

Their programming capability provides for many uses. Different applications can even be combined on a single card. Microprocessor smart cards are required for applications that manipulate or compare data, such as public key infrastructure (PKI) data encryption, Java applets, and electronic purses.

Every microprocessor smart card has an OS on the chip to operate the internal functions of the application. The OS loads off of the read-only memory (ROM), much like a basic input/output system (BIOS) on a modern PC. The primary function of the card OS is to enable memory access. The OS also manages the security functions that these cards typically perform. A microprocessor card, using an OS, has a predefined behavior that allows the card and application to communicate using predefined commands.

Smart-card microprocessors use either *open-OS* or *OS-like* programs. Open-OS applications are easier to write because software developers use programming interfaces that they already know. The development code is the same code used to write a program for an Intel or PowerPC machine; thus, the learning curve is eliminated. Three of the open-OS card standards include:

- Microsoft® Windows® for Smart Cards — uses common Microsoft Windows API calls
- Multi-Application Operating System (MULTOS) — developed by the MAOSCO[1] consortium for financial transactions with emphasis on security
- Sun Microsystem Java Technology — provides for dowloading and running Java applets

OS-like programs use proprietary software solutions for specific applications that are usually developed by the smart-card manufacturer. Because a developer must learn a proprietary code, the software is initially more difficult to write, but can provide additional security from hackers.

Smart-Card Dimensions

Two physical dimensions are specified for smart cards. The most popular form is approximately the size of a credit card. Small enough to be conveniently portable, the card is large enough to display graphics and advertising on its side.

The second, smaller smart-card size, specified by the European Telecommunications Standards Institute (ETSI), is used specifically for Global System for Mobile Communications (GSM) phones, the predominant cellular phone technology system in Europe.

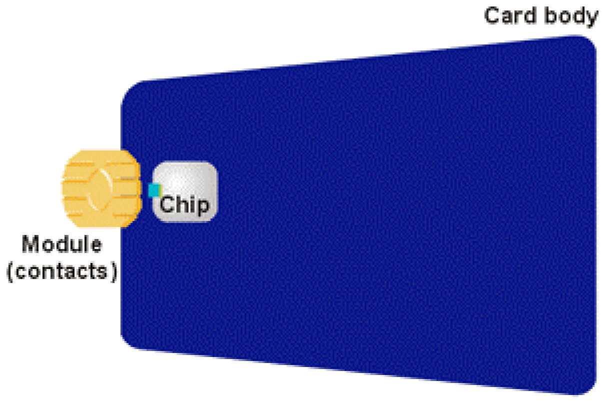
Communication with a Reader

A smart-card chip communicates with a reader by direct physical contact or by a radio frequency (RF) signal, depending on the system design. Three smart-card designs for chip-to-reader communications are:

- Contact cards
- Contactless cards
- Combination cards

Contact Smart Cards

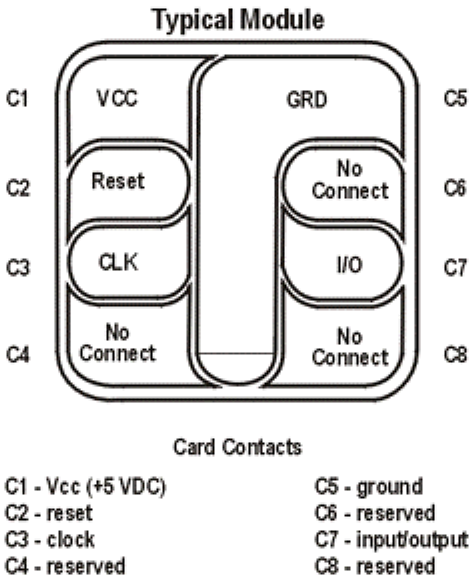
Contact smart cards are the most popular card-connection design, and are used for both card sizes and chip types. Figure 1 depicts the contact-type card.



Source: Gemplus - All About Smart Cards

Figure 1. Contact Smart Card

Contact cards use an eight-pin contact, micromodule to physically connect to the card reader. Five pins are defined as Vcc (+5 VDC), reset, clock, ground, and input/output (I/O). Figure 2 shows an example of a typical contact card module.

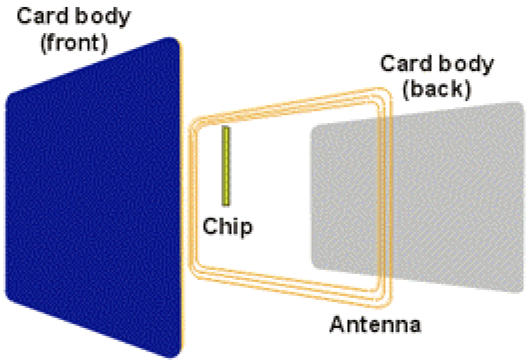


Source: CardLogix - Smart Card Basics

Figure 2. Eight-Pin Micromodule

Contactless Smart Cards

Contactless smart cards as shown in Figure 3 use an antenna with approximately a 10-centimeter (cm) range to communicate with the reader. These credit-card sized memory-chip devices derive their power from an RF field generated by the card reader. The RF field also transfers information to and from the card and card reader. Employee identification badges issued by large companies for building access are typically contactless smart cards.



Source: Gemplus - All About Smart Cards

Figure 3. Contactless Smart Card

Combination Smart Cards

Multipurpose combination smart cards are a hybrid mix of the contact and contactless designs. They include the eight-pin contact for communication with a contact-type reader, and also include an antenna for communication with an RF-type reader.

Smart-Card Applications

Smart cards have their roots in security functions, so several smart-card applications are related to security issues. Typical applications include:

- Identification
- Network access
- Network security
- Electronic purses
- Special applications

Identification

Memory smart cards use a unique serial number to identify a user, and can be contact cards or contactless cards. They provide some security, but can be used by anyone in possession of the card. More complex identification schemes require the use of a microprocessor smart card, which can require the user to enter a Personal Identification Number (PIN) or other information to prove identity.

Network Access

Using a smart card to log on to a network is similar to the identification application. This can be a simple operation using the card serial number, or a complex operation requiring PINs and a network password. Microsoft Windows 2000 provides native support for this feature.

Network Security

Passing information over the Internet is a security risk because an unanticipated party can intercept the information. This can be critical if the information is confidential, such as a credit-card number or a financial report.

Microprocessor smart cards and PKI can be used to provide secured information transmissions. The cards store the algorithms, keys, and certificates required to encrypt the information. The information is extremely difficult to decode without the keys, and they never leave the card, which prevents them from being intercepted by a third party.

Electronic Purses

Electronic purses are memory or microprocessor smart cards used to perform monetary transactions. Memory smart cards are used to provide services, such as purchasing copies on a copy machine. When the counter is depleted, the card is thrown away. Electronic purses are also used for gaming endeavors where a user buys a card with gaming credits that can be used in a gaming machine. Credits are added or subtracted from the card according to the rules of the game. Electronic purses could also be used as cash so a user could move cash from one smart card to another for a transaction, or convert some or all of the card credit back to cash.

Special Applications

JavaCard and Personal Computer/Smart Card (PC/SC) allow programmers to write code for smart cards, much the same as the code written for PCs. The only limit for the code is the size of the EEPROM used in the smart card. Special applications that could be developed include:

- Launching a web page on insertion of the card
- Launching a support web page configured to the user's needs
- Storing personal information

The options for special smart-card applications are virtually unlimited depending on the creativity of the developer.

Smart-Card Readers

Card readers provide the physical link between the smart card and the host. Figure 4 shows a combination keyboard/card reader. The host can be a PC or a stand-alone device. The reader delivers power, initializes the card, and acts as the mediator between the smart card and the host. Power is delivered to the smart card through a contact on the micromodule of contact smart cards or by inducing current through the antenna of contactless designs. Initialization is a specified protocol that all cards must perform. All smart-card readers support the initialization of any smart card, but they may not support the card after it switches to its specific application.

Figure 4. Smart-Card Reader

Card Awareness

A reader's view of a smart card is called card awareness. Most card readers support both of the two view options:

- Aware — readers view the physical card structure, and act as a translator between the host and the card. Memory cards require the aware view because the reader must know the exact address for the data.
- Generic — readers know the logic structure of the card, and pass commands from the host straight to the card without changing the command. Microprocessor cards use the generic view because the cards have their own OS and logic to interpret the commands.

Reader Hardware Types

Smart-card reader devices are *transparent* - or *standalone* -type hardware. Transparent-reader hardware, sometimes simply called a reader, requires a host for all signaling functions, including initialization and application. This type of hardware has no internal logic, except for a line driver to condition the signal between the card and the host. A transparent reader is similar to a PC soft modem; a host drives the reader and the card. This requires more support from the software, which must understand the electrical design of the reader and how to communicate with the card. The driver must also have the logic to support the aware view for communication with memory cards.

Stand-alone reader hardware, sometimes called a terminal, includes all of the logic required to initialize a card and to act as mediator between a memory card and the host. For example, the host may deliver a large packet of information to the reader to pass on to the memory card. The reader has to check the packet, and sometimes break it into two packets, before sending the information to the smart card. This means the host is only concerned with communication to the reader and not the smart card. Stand-alone hardware functions as a pass-through for microprocessor cards. The OS defines all of the commands that a microprocessor card understands, so the reader is not required to intervene.

Transparent readers require more drivers than stand-alone equipment, but are cheaper to manufacture and easier to change. Stand-alone readers, although more expensive than transparent devices, have generic driver sets that define the communication between a reader and a host. Memory-card drivers are built into the logic, making driver management easier, but the reader can become obsolete unless the drivers can be upgraded.

Smart-Card Standards

Smart-card standards define the operation of the technology, and promote product interoperability between smart-card manufacturers. Because there are a variety of smart-card applications requiring different solutions, several standards were needed to define smart-card technology. The International Organization for Standardization (ISO) 7816-X specification defines the complete characteristics of smart-card technology. Other specifications alter elements of the ISO 7816-X specification to meet specific application requirements. Smart-card specifications include:

- ISO 7816-X — the dominant standard for contact smart cards consisting of ten sections that detail the physical, electrical, mechanical, and application programming interface (see Table 2). All other smart-card specifications are variations of this standard.

Specification	Definition
ISO 7816-1	Physical characteristics
ISO 7816-2	Dimension and location of contacts
ISO 7816-3	Electronic signal and transmission protocol
ISO 7816-4	Interindustry commands and responses
ISO 7816-5	Registration system for application identifiers
ISO 7816-6	Data elements for interchange
ISO 7816-7	Smart Card Query Language commands
ISO 7816-8	Security architecture
ISO 7816-9	Interindustry enhanced commands
ISO 7816-10	Synchronous cards

Table 2. ISO 7816-X Specifications

- ISO/IEC 14443-1 — the ISO and International Electrotechnical Commission (IEC) specification for contactless cards that changes the contact description to an antenna, and defines the protocol for communication over the air.

- ETSI — the European Telecommunications Standards Institute specification that defines a smaller-sized smart card to fit into GSM phones.

- EMV — the integrated circuit card specification for payment systems, which is managed, maintained, and enhanced by Europay International, Master Card International, and Visa International (EMV). This standard defines the way smart cards interchange in a payment terminal by disallowing the reader to be transparent. This increases security by preventing reading of the card for low-level information. This condition conflicts with Microsoft's Windows Hardware Quality Labs (WHQL) specification, which requires full ISO 7816 compliance.

- PC/SC — this PC/SC Workgroup specification builds on existing EMV and ISO 7816-X specifications by defining the smart-card reader/writer abstraction layer. This is a complementary specification that defines low-level device interfaces, device-independent application APIs, and resource management, which allows multiple applications to share smart-card devices attached to a system.

- JavaCard — this specification defines the way the Java Virtual Machine is implemented so that an end user can run any Java applet on the smart card.The Java Card Forum drives this specification.

- WHQL — the Microsoft WHQL facility defines the guidelines for products that are compliant with Microsoft OSs. The focus is to ensure that devices work in the Windows environment, and are compatible with other devices. WHQL requirements for smart cards require full ISO 7816 compliance.

Standards Summary

Several standards define the operation of a smart card for various applications, so it is important to understand these standards when selecting a smart card and reader for a specific application. Dell plans to support WHQL-certified card readers; WHQL certification requires smart-card readers to be compliant with the ISO 7816-X standard.

Smart-Card Security

One major use for smart cards is to protect data as information is exchanged between the card and reader. Companies using smart-card systems often require security for:

- Confidentiality
- User authentication
- Application authentication
- Transaction authentication
- Nonrepudiation

Most of these security needs can be met by using PKI, which provides the policies and procedures required for establishing secured information exchange. PKI includes data encryption to ensure confidentiality, digital certificates to provide authentication, and digital signatures to prove the transaction was completed by the originator without intervention or error.

Data Encryption

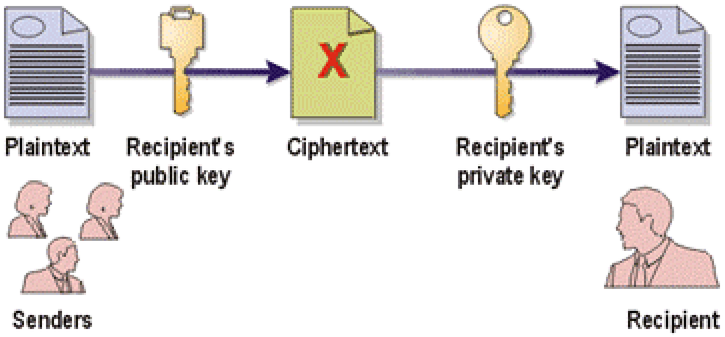
Four components of data encryption include:

- Algorithms — mathematical functions used to encrypt data
- Keys — parameters, or numbers, used in algorithms
- Encryption — the process of using the algorithm and key to code, or encrypt, the data
- Decryption — the process of converting the encrypted data back into its original form

Algorithms

Algorithms are *symmetric* or *asymmetric* mathematical functions used to scramble and unscramble data. Symmetric functions use a secret key to encrypt and decrypt the data, but sending the secret key to the intended recipient presents a security risk. Anyone in possession of the secret key can potentially read the encrypted information. Symmetric functions may be appropriate for one-to-one transactions, but do not provide adequate security when data is transferred on a network because secret keys could be intercepted by a third party.

Asymmetric functions solve this security issue by using a public key cryptography algorithm that includes a public key and a private key, as shown in Figure 5.



Source: Web Security - A Step-By-Step Reference Guide

Figure 5. Public Key Cryptography

To transfer secured data using public key cryptography, the sender:

- Encrypts the data using the receiver's public key
- Sends the data to the target recipient

To recover secured data using public key cryptography, the recipient:

- Receives the encrypted data from the sender
- Decrypts the data using their private key

Unlike a symmetric function that requires a sender to transfer a secret key, the private key used for an asymmetric function is never transferred. Because the private key is secure, data sent to a user on an unsecured network is secure. Public key cryptography's limitation is a lack of speed; the fastest implementation of an asymmetric function is much slower than a typical symmetric function.

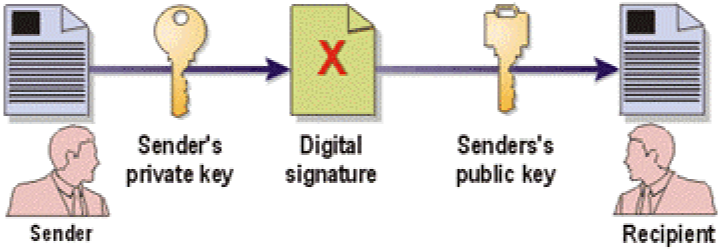
There are several algorithms in use for public key cryptography, depending on the security level required. Rivest-Shamir-Adleman (RSA), named for its inventors, and Elliptic Curve Cryptography (ECC) are examples of two widely used algorithms that meet most requirements for public and private keys.

Digital Certificates

The algorithm and keys take care of the data encryption and decryption process, but do not address the authentication security requirement. Anyone can generate a public and private key. Digital certificates resolve this issue by validating that the public key used by the sender is the actual public key of the recipient. The certificate also contains identifying information about the key owner. A trusted third party, or certifying authority, issues the public key certificate (PKC) to users requesting the certified public key of a target recipient so that a secured communication can be established. The PKC also contains a time stamp that expires at a predetermined time so that PKCs must be reissued periodically.

Digital Signatures

Once the data is encrypted and the receiver is verified, a digital signature is used to ensure that the data was not tampered with during transmission. A digital signature, shown in Figure 6, is an encrypted digest of the data that is generated using the sender's private key. The digest of the data produces a hash value. A hash value, similar to a checksum, is generated from a hash function algorithm that converts text into a fixed-sized output.



Source: Web Security - A Step-By-Step Reference Guide

Figure 6. Digital Signature

To transfer encrypted data using a digital signature, the sender:

- Runs the data through a digest function to obtain a hash value

- Encodes the hash value with their private key to create a digital signature
- Obtains the recipient's public key from a certifying authority
- Attaches the digital signature to the encrypted data
- Encrypts the combined digital signature/encrypted data using the recipient's public key
- Sends the package to the recipient

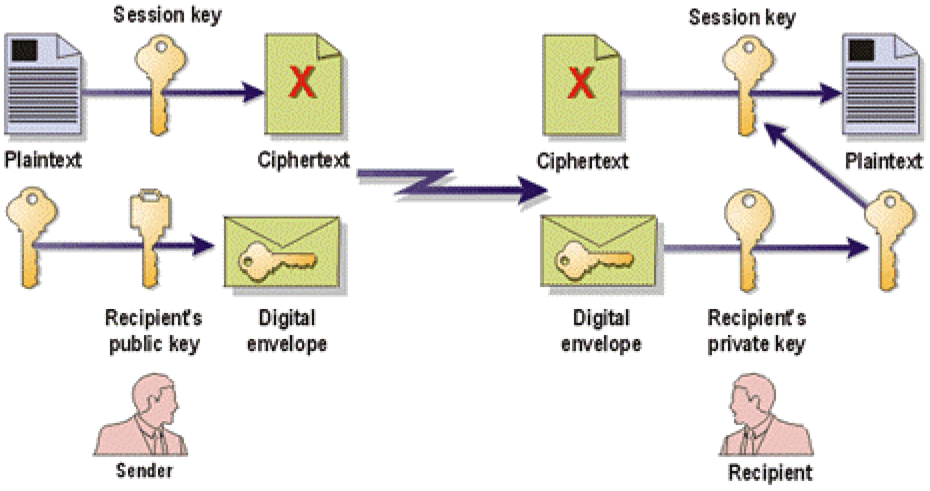
To recover the encrypted data and digital signature from the package, the recipient:

- Decrypts the combination digital signature/encrypted data using their private key
- Decrypts the encrypted data using their private key
- Runs the data through a digest function to generate a hash value
- Obtains the sender's public key from a certifying authority
- Decrypts the digital signature using the sender's public key to get the sender's generated hash value
- Compares the recipient's generated hash value with the sender's generated hash value to verify the integrity of the data

Digital signatures are a reversal of public-key cryptography—data encrypted using a sender's private key can only be decrypted using the sender's public key. By obtaining the sender's public key to decrypt the digital signature, the recipient ensures that the digital signature was generated by the sender's private key. Anyone with access to the sender's public key can verify the digital signature. By comparing the hash values generated from the data by the sender and the recipient, the recipient ensures that the data did not change during the transfer.

Digital Envelopes

Asymmetric systems, while ideal for secure data transmission, are not well suited for large documents because they are much slower than symmetric systems. The solution, a digital envelope shown in Figure 7, combines symmetric and asymmetric functions.



Source: Web Security - A Step-By-Step Reference Guide

Figure 7. Digital Envelope

To transfer a document using a digital envelope, the sender:

- Generates a secret key at random. This secret key is usually called a session key because it is discarded after the communications session is completed.
- Encrypts the document using the session key and a symmetric algorithm.
- Encrypts the session key using the recipient's public key (thus, uses an asymmetric function to create the digital envelope).
- Sends the encrypted document and the digital envelope to the recipient.

To recover a document using a digital envelope, the recipient:

- Uses their private key to decrypt the digital envelope and recover the session key (an asymmetric function).
- Uses the session key to decrypt the document (a symmetric function).

A digital envelope provides fast, secure data transfer because only the session key is encrypted with an asymmetric function, and only the intended recipient possesses the private key required to decrypt the session key. The document is encrypted and decrypted with a faster symmetric function using this session key. When the data transfer is completed, the session key is discarded.

PKI Example

Algorithms, public and private keys, digital certificates, digital signatures, and digital envelopes all work together to provide PKI security. The following example demonstrates how the process works when Harry sends a company report to Sally.

Encryption

Harry wants to e-mail a long, confidential company report to Sally. To provide a fast, secure transfer, guaranteed data integrity, and assurance that the report came from him, Harry:

- Runs the report through a digest function to obtain a hash value
- Encrypts the hash value using his private key to create a digital signature
- Generates a session key
- Encrypts the company report using the session key and a symmetric algorithm
- Obtains Sally's public key from a certifying authority
- Attaches the digital signature to the encrypted company report
- Encrypts the combined digital signature/encrypted company report using Sally's public key
- Encrypts the session key using Sally's public key, producing a digital envelope
- Transfers the encrypted digital signature/encrypted company report package to Sally
- Transfers the digital envelope to Sally

Decryption

Sally receives the digital signature/encrypted company report package and the digital envelope, so she:

- Decrypts the digital signature/encrypted company report package using her private key
- Decrypts the digital envelope using her private key to recover the session key
- Decrypts the encrypted company report using the session key
- Runs the report through a digest function to obtain a hash value

- Obtains Harry's public key from a certifying authority
- Decrypts the digital signature using Harry's public key to get Harry's generated hash value
- Compares Harry's generated hash value to her own generated hash value

If Sally successfully decrypts the digital signature using Harry's public key, she can be confident that the confidential company report came from Harry. If the two hash values generated from the report also match, Sally has a guarantee that the report is intact.

Smart Cards and PKI

Smart cards are used to store the public and private keys, the algorithm, and the digital certificates. The keys never leave the card, and the algorithm is used on the card to decrypt the message. This means that no third party can "listen" to the communication between the card and the reader to intercept the private key.

PINs also provide protection for keys stored on a smart card. If the card is stolen and the thief attempts to guess the PIN to access the keys, the system can lock out the card after a few wrong guesses, thus preventing any further use of the card.

Smart-Card Support

Windows 2000 Professional is the first Microsoft OS with native support for smart cards. Other OSs require adding drivers supplied by the device manufacturer. Smart-card native support provided with Windows 2000 Professional includes:

- Reader drivers — supports several smart-card readers, and only requires that the reader is connected to the system, which allows Plug and Play to detect and configure the reader
- PC/SC — supports the PC/SC card
- Smart card ready — supports network PKI login

Although Windows 2000 Professional supports smart cards used for network PKI logins, a user needs a certificate server and a smart card configured with a certificate before this function can be used. The certificate server acts as the certifying authority by issuing and verifying certificates in use on the network. After the network is set up and configured, the user is required to use the smart card to log on to the network.

Microsoft Outlook and Internet Explorer also provide smart-card support. Smart cards can be used with Outlook to digitally sign e-mail, and to send PKI secure transmissions. They can be used with Internet Explorer to control access to secured websites.

Dell's Plans

Dell currently offers a wide range of security devices, including smart-card readers, through Dell Software & Peripherals. Combined with the native support for smart cards provided in Microsoft Windows 2000 Professional, these readers support enhanced client security without the need for additional software.

Dell plans to introduce a combination keyboard/smart-card reader in the spring of 2001. This new smart-card device will support convenient identification, network access, and network security options at an affordable price for Dell™ customers.

Conclusion

Smart cards are becoming important and useful security tools for use in the PC environment. These simple, yet powerful devices can be programmed to perform a number of security-related tasks, ranging from user identification to secured network transmissions. New Java and PC/SC smart cards provide almost unlimited potential for PC applications.

With Microsoft Windows 2000 Professional providing native support, and other OSs expected to follow, smart-card technology is positioned to become a mainstream computer industry tool.