This is Chapter 27 by Bertrand Byramjee and Andrew Weigl of the Handbook of Elliptic and Hyperelliptic Curve Cryptography, Henri Cohen, Christophe Doche, and Gerhard Frey, Editors, CRC Press 2006.

Chapter 27

Smart Cards

Bertrand Byramjee and Andrew Weigl

Contents in Brief

Although smart cards are now very common, this technology is still very new, with the first smart cards appearing in the 1970's. Since then, their evolution has been very rapid. Smart cards have advanced from simple memory cards to very efficient "microcomputers" with multiple applications.

Equipped with a microcontroller, these cards are able to store and protect information using cryptographic algorithms. They are also resistant to physical stresses such as twisting and bending. The physical structure of the smart card consist of a small plastic rectangle with a magnetic stripe, holograms, relief characters and an embedded chip. They are small, and easy to use and carry. The security and portability of smart cards provide a safe, reliable, convenient, and effective way to ensure secure transactions (banking, e-business, etc.), and to enable a broad range of applications. Thus, modern smart cards can actually be used in any system that needs security and authentication. They have been proven to be an ideal means of making high-level security available to everyone.

This chapter aims to present an overview of today's smart card technology and show the limitations that smart card manufacturers must take into account when implementing cryptographic algorithms, for example, elliptic or hyperelliptic curve algorithms, in a smart card environment.

27.1 History

In the beginning of the 1950's, the first plastic (PVC) cards appeared in the USA as a substitute for paper money. They were initially aimed at the rich and powerful, and were only accepted by prestigious hotels and restaurants. These cards were very simple with the owner's name printed in

relief, and sometimes the handwritten signature was added. These cards provided a more convenient payment system than paper money. With the involvement of VISA™ and MasterCard™ in plastic money, credit cards spread rapidly around the world. Later a magnetic stripe was added to reduce fraud and to increase security. Confidential digitized data was stored on this stripe, but this information was accessible to anyone possessing the appropriate card reader.

Between 1970 and 1973 there was a significant development in plastic cards with the addition of microcircuits to the card. Many patents were filed during this time; the best known inventors include: J. Dethleff, K. Arimura, and R. Moreno. The term "smart card" was proposed by R. Bright. It was not until 1984 that the smart card was first put into commercial use by the French PTT (postal and telecom services) with their first telephone cards (smart cards with memory chips). In 1986, millions of these smart cards were sold in France and other countries. After telephone cards, the next big application was their use as banking cards. This development was more difficult because they contained more complicated chips that were able to compute cryptographic calculations. The French banks were the first to introduce this technology in 1984. A number of ISO standards were created to encourage interoperability of smart cards. By 1997, bank cards were widely used in France and Germany. The microcontrollers continued to advance and became more powerful with larger memory capacity. This allowed for sophisticated cryptographic algorithms, providing higher levels of security.

Nowadays, smart cards are present all over the world, and their use is likely to spread even further.

27.2 Smart card properties

Smart cards are physically similar to the classic embossed plastic cards. The older model cards are used as the base design for the newer smart cards. There are two different categories of smart cards: memory only cards, which are the cheapest and the simplest, and the microprocessor cards, which are more expensive, but have more applications and security features. The structure of smart cards is standardized by ISO, principally: ISO 7816 [ISO 1999a, ISO 1999b, ISO 1999c, ISO 1999d], and ISO 7810 [ISO 1995].

The following sections look at the different aspects of the smart card properties.

27.2.1 Physical properties

The most widely used smart card format, ID-1, is part of the 1985 ISO 7810 standard [ISO 1995]. Most smart cards are made from PVC (polyvinyl chloride), which is also used for credit cards. Some are made from ABS (acrylonitrile-butadiene-styrol), but they cannot be embossed; an example application is a mobile phone card.

The body of the card includes the following components: magnetic stripe, signature stripe, embossing, imprinting of personal data (picture, text, fingerprint), hologram, security printing, invisible authentication features (fluorescence, UV), and a microprocessor chip.

27.2.1.a The chip module and its embedding

The chip module, also called the *micromodule*, is the thin gold contact on the left side of the smart card. This module needs to be firmly attached to the plastic of the card. Its purpose is to protect the card and the microprocessor chip. The contacts for contact-type smart cards can also be in the chip module.

Many embedding techniques have been tested and used with the aim to optimize overall card resilience to everyday physical and mechanical stresses (temperature abrasion, twisting, bending,

etc.) while keeping the production costs as low as possible.

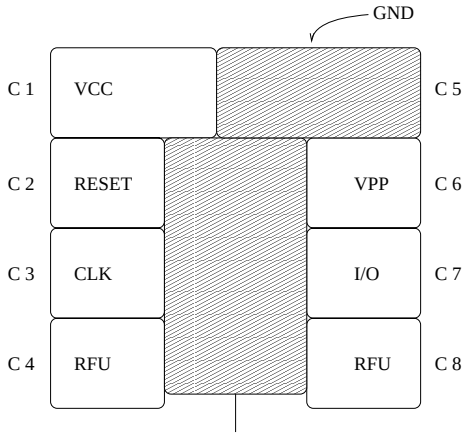
27.2.1.b Contact and contactless cards

There are two main ways a smart card can communicate with the card terminal: through physical contact or by using a contactless connection. The contact cards were the first types of smart cards on the market. However, with new advances in microcircuit technology, contactless cards have become physically feasible.

Contact cards

This is currently the most common type of card. It communicates via a card reader where the information passes through the contacts. There are metal contacts inside the card reader and on the chip module of the smart card. The position and dimensions of these contacts (power supply, data transfer, etc.) are set in the ISO 7816-2 standard [ISO 1999b]. Another standard, AFNOR, is still in use by some cards in France, but is likely to disappear in the near future.

Figure 27.1 Pin layout for contact smart cards.



There are 8 contact areas $C_1, \ldots, C_8$:

C_1: Supply voltage, VCC,
C_2: Reset,
C_3: Clock, CLK,
C_4: Not in use, reserved for future use,
C_5: Ground, GND,
C_6: External voltage programming,
C_7: Input/Output for serial communication,
C_8: Not in use, reserved for future use.

Contactless cards

These cards contain special circuits, which allow data transmission over short distances without mechanical contact and without a direct supply of power. This technology is not new but is difficult to apply to smart cards. At the moment it is possible to incorporate a battery into the card, but it increases the size and cost of the card. Research is ongoing to reduce this problem.

Not only is there a problem supplying power to the smart card circuits, but data and clock signals

also need to be transmitted between the card and the terminal. The technique of capacitive and inductive coupling, at this time, is the most suitable for smart cards and has been standardized in ISO/IEC 14443 [ISO 2000]. This standard presents a method for capacitive and inductive coupling where the card's conductive surfaces act as capacitor plates. One or several coupling loops are integrated into the card to receive energy from the terminal. A carrier frequency in the range 100-300 kHz is used, which allows very rapid transmission.

Dual interface or "combi-cards"

In the future it is likely that "combi-cards" will become more common. They combine the advantages of contact and contactless cards. In ISO/IEC 10536 the application is described as "slot or surface operation." Depending on the operation, the card must either be inserted in a slot to make contact or placed on a certain surface for contactless transaction. This type of card allows applications such as credit, debit, membership, and mass transit to be used on the same card.

27.2.2 Electrical properties

The electrical properties of a smart card depend on its embedded microcontroller, since this is the only component of the card with an electrical circuitry. The basic electrical requirements are defined by the ISO/IEC 7816-3 standard, Part 3: *Electronic signals and transmission protocols* [ISO 1999c]. Electrical characteristics and class indication for operating at 5 V, 3 V and 1.8 V are described within Amendment 1. Amendment 2, which describes an USB interface for smart cards, is currently under preparation. The GSM mobile telephone network (GSM 11.11) should be mentioned here, because it also contributes to the requirements in this area. Further modifications of the ISO/IEC 7816 standard are driven by the UMTS specification.

27.2.2.a Supply voltage

A smart card supply voltage is 5 V, with a maximum deviation of $\pm 10\%$. This voltage, which is the same as that used for conventional transistor-transistor-logic (TTL) circuits, is standard for all cards currently on the market. Since all modern cellular telephones are built with 1.8 V technology (GSM 11.18), modern smart cards are designed for a voltage range of 1.8-5 V $\pm 10\%$, which results in an effective voltage range of 1.6-5.5 V. They can be used in both, 1.8 V and 5 V terminals, to keep the advantage of simple and straightforward card usage.

27.2.2.b Supply current

The built-in microcontroller obtains its supply voltage via contact C1 (see Figure 27.1). According to the GSM 11.11 specification, the current may not exceed 10 mA, so the maximum power dissipation is 50 mW, with a supply voltage of 5 V and an assumed current consumption of 10 mA. Table 27.2 gives an overview of the actually defined maximum power consumption classes, specified by ISO 7816 and GSM.

The current consumption is directly proportional to the clock frequency used, so it is also possible to specify the current as a function of the clock frequency. State-of-the-art smart card microcontrollers use configurable internal clock frequencies for their processor and their arithmetic coprocessor. Hence, the current consumption is not only dependent on the external clock, but also on the given configuration of the microcontroller itself and the setting of the coprocessor. The coprocessor can be programmed to keep power consumption under a set value, for example, the GSM values.

Table 27.2 Smart card power consumption specified by ISO 7816 and the GSM specifications.

Specification	ISO 7816-3		GSM		
Notation	Class A	Class B	GSM 11.11	GSM 11.12	GSM 11.18
Supply voltage	5 V	3 V	5 V	3 V	1.8 V
Supply current	600 mA	50 mA	10 mA	6 mA	4 mA
Frequency	5 MHz	4 MHz	5 MHz	4 MHz	4 MHz
Power consumption	300 mW	150 mW	50 mW	18 mW	7.2 mW

27.2.3 Memory

Smart cards can be divided into two main components: the processor (including coprocessor) and memory. Memory can again be divided into volatile and non-volatile memory. Table 27.3 shows the different types of volatile and non-volatile memory. Since a smart card needs to be able to function as an independent unit, most cards will be found with a combination of RAM, ROM, and EEPROM.

Table 27.3 Types of memory found in smart cards.

Memory types found in smart cards	
Volatile memory	Non-volatile memory
RAM	ROM
	PROM
	EPROM
	EEPROM
	Flash EEPROM
	FRAM

27.2.3.a Read-Only Memory (ROM)

ROMs are non-volatile memory that can be randomly accessed during reading. There is no limit to the number of times the memory can be read, but it can only be written during production. This type of memory requires no voltage to hold the information, so when the power is disconnected, the data is still retained. This is excellent memory for storing vital programs that the smart card needs to run, like the operating system and the diagnostic functions. The data is imprinted onto the chip by using lithographic techniques. ROM cells require the least amount of area per cell compared to other available types of memory.

27.2.3.b Random Access Memory (RAM)

RAM is the work area for the smart card. It can quickly read and write data, and there is no limit to the number of writes a RAM cell can handle. However, since it is volatile memory, constant power needs to be supplied, or otherwise the contents will be lost. The method for accessing this memory is what gives it its name; *random access* means that the memory is selected and directly accessed

without having to sequentially traverse the memory block.

In smart cards, the most common form of RAM is static RAM (SRAM), which, unlike dynamic RAM (DRAM), does not need to be periodically refreshed. SRAM has flip-flops as the basic component while DRAM uses capacitors with refresh circuitry.

Smart card chip designers try to keep the amount of RAM to a minimum, since it requires a large area per cell. Indeed, RAM cells require seventeen times more area than a ROM cell.

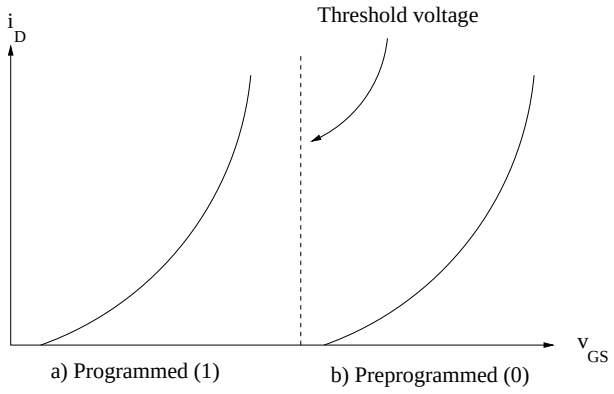
27.2.3.c Programmable read-only memory (PROM)

Programmable read-only memory is similar to ROM in that once it has been written it cannot be rewritten. The difference is that the code does not need to be written with lithographic techniques. PROM has a serious drawback; access needs to be granted to the address, data, and control buses for the writing process. This leaves a security hole in the smart card that a hacker could use to read the data stored on the chip. PROM is not used in smart cards because of this vulnerability.

27.2.3.d Erasable programmable read-only memory (EPROM)

An EPROM is essentially an n-channel MOSFET (metal-oxide-semiconductor field effect transistor) with an extra polysilicon gate called the *floating gate*. In Figure 27.4 the left curve has a relatively low V_t and is normally chosen as state "1." This state is also called the *"preprogrammed" state*.

Figure 27.4 Threshold voltage curves for programmed and preprogrammed state.

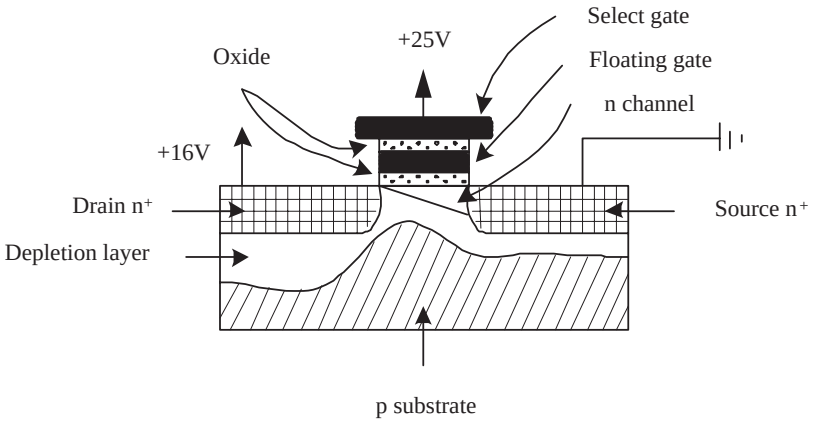


A voltage needs to be applied between the drain and source to program the EPROM to the "0" state (see Figure 27.5). On the select gate a voltage of 17 V to 25 V needs to be applied. Since smart card controllers use a supply voltage between 3 and 5 V, a cascaded voltage-multiplier circuit, or charge pump, needs to be used to generate the required voltage levels.

The device acts as a regular n-channel enhancement MOSFET when there is no charge present on the floating gate. With the voltages present, a tapered n-type inversion layer is formed at the surface of the substrate. The drain-to-source voltage accelerates the electrons through the channel. The electric field formed by the voltage on the selected gate attracts the hot electrons (the accelerated electrons) towards the floating gate. At the floating gate the electrons collect, causing the gate to

become negatively charged. This process continues until enough of a negative charge is formed on the floating gate to reduce the strength of the electric field to the point of not being able to accelerate any more hot electrons.

Figure 27.5 EPROM during programming.



The negatively charged floating gate repels electrons away from the surface of the substrate. To compensate for the loss of electrons in the region, a larger select gate voltage is required to form an n-channel. This will shift the $i_D - v_{GS}$ characteristic graph upwards, as can be seen in Figure 27.4 [SESM 1991].

For the microcontroller to read the state of the EPROM, the unit only needs to apply a test V_{GS} between the two $i_D - v_{GS}$ curves. If the current flows, the EPROM is in state "1" and if it does not flow then it is in state "0".

For smart cards, EPROM was used by the French PTT in their first telephone cards, since, at that time, it was the only ROM type memory available [RAEF 2000]. As with other ROM types, it does not require a supply voltage to retain the data. EPROM can be reprogrammed, but it first requires ultraviolet light to erase the old data. This method is not feasible for smart cards, so this technology has been abandoned for newer erasable ROMs.

27.2.3.e Electrically erasable programmable read-only memory (EEPROM)

As with regular computers, sometimes data needs to be read, altered and then stored with the possibility that the voltage supply is disconnected. Computers use hard drives to store the data for longer periods of time, but smart cards do not have this option. Instead they use a type of ROM that can handle multiple writes. EPROM can only be erased with ultraviolet light, which makes it unsuitable as a multi-write memory. The solution is found with another type of ROM that can be electrically erased, the electrically erasable programmable read-only memory (EEPROM), see Table 27.7.

EEPROM operates similarly to the method described in Section 27.2.3.d. There are two main differences between EPROM and EEPROM. The first difference is how the electrons travel from the substrate to the floating oxide layer. The method described in Section 27.2.3.d uses hot electron injection, while standard EEPROM uses the tunnel effect (Fowler–Nordheim effect). A high positive

voltage at the select gate causes electrons to migrate through the tunnel oxide to the floating gate, where they collect. Eventually, the floating gate becomes negatively charged.
The second difference between EPROM and EEPROM is how the data is erased. As stated earlier, EPROM requires ultraviolet light to reset its state. For EEPROM a negative voltage applied to the select gate forces the electrons from the floating gate back to the substrate. After this process, the EEPROM is classified again as discharged and the V_t is low.

Similar to RAM and other types of ROM, EEPROM can be read an unlimited number of times. However, there is a limit to the number of writes that can be performed. The life expectancy is limited by the quality, type, and thickness of the tunnel oxide layer, which is the oxide layer between the floating gate and the substrate (see Figure 27.5). During production, the tunnel oxide is one of the first layers to be produced. As the rest of the production continues, it undergoes large thermal stresses that cause minute faults in the oxide layer. This allows the tunnel oxide to absorb electrons during the programming cycle, which are not returned to the substrate when the data is erased. The trapped electrons then collect at the channel between the drain and source. This process continues until enough electrons collect that they influence the threshold voltage to a greater degree than the floating gate. The threshold voltage then stays in one state, regardless of whether the floating gate is charged or not; the EEPROM is then useless.

27.2.3.f Flash electrically erasable programmable read-only memory (flash EEPROM)

Flash EEPROM is a mixture of EEPROM and EPROM technology. It operates with hot electrons but uses an erase technique similar to EEPROM. Most manufacturers implement a combination of flash EEPROM and regular EEPROM onto their chips. Each memory type has its benefits when it comes to endurance and space requirements. A typical application for regular EEPROM is transient data or constants that may need to be occasionally changed. Flash EEPROM is better suited for program code, which may need to be upgraded or changed only a few times during the products, life cycle [BUR 1999].

EEPROM uses both, a read and a write transistor, while flash memory uses only one transistor. Also, each cell of the regular EEPROM requires signal routing and complex address-decoding logic, since each word has its own control signals. Flash memory, on the other hand, employs less complex word decoders that allow for more compact units. This leads to a trade-off in programming flexibility and write time.

With regular EEPROM any word can be altered by accessing that memory location, erasing the word, then writing in the new data. This is not possible with the word decoder of flash memory. Before any data can be changed, the whole array must be erased and then completely rewritten. This leads to a large difference in operating speed. EEPROM can be reprogrammed in a few milliseconds, whereas flash memory may take from a few microseconds to a second to reprogram.

Another key factor that makes regular EEPROM able to better handle transient data is the endurance of its cells. Flash EEPROM can only handle about 1000 write cycles, which is too short for most application requirements. Since application data does not change that often, it is perfect for it to be programmed onto flash EEPROM. It can be upgraded after manufacture giving it an advantage over ROM or PROM. Most smart cards now incorporate some flash EEPROM in their system.

27.2.3.g Ferroelectric random access memory (FRAM)

Ferroelectric Random Access Memory (FRAM), a relatively new type of memory, has recently become available to the microprocessor manufacturers. The characteristics of FRAM are similar to those of both RAM and ROM. It is a non-volatile memory, but with a write speed considerably faster than flash or EEPROM memory.

Instead of using a floating oxide layer as the dielectric, FRAM employs a ferroelectric film. The composition of the film is usually either *PZT* $\big(\mathrm{Pb(Zr,Ti)O_3}\big)$, *PLZT* $\big((\mathrm{Pb,La})(\mathrm{Zr,Ti})\mathrm{O_3}\big)$, or *SBT*

$(\mathrm{Sr\,Bi_2Ta_2O_9})$. Figure 27.6 shows the crystalline structure of PZT. Polarization through electric fields has a dual advantage over the injecting of hot electrons or tunneling effect method, it is faster at writing and it requires less power (see Table 27.7 for the comparison between the different memory types).

Figure 27.6 View of the crystalline structure of ferroelectric material.

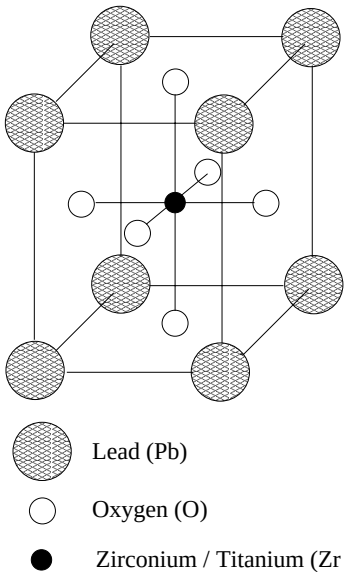


Table 27.7 Comparison of different memories [FUJITSU].

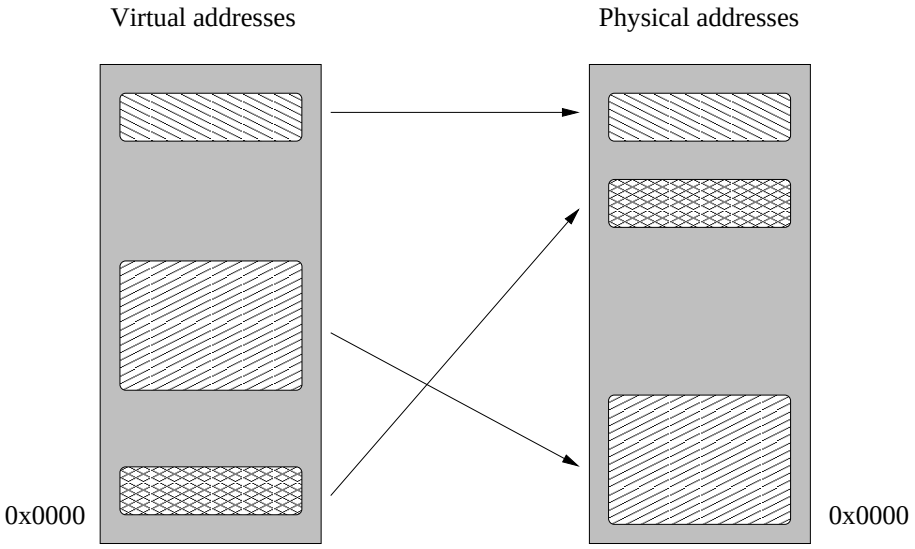
Memory	SRAM	DRAM	EEPROM	FLASH	FRAM
Type	Volatile	Volatile	Non-volatile	Non-volatile	Non-volatile
Read cycle	12 ns	70 ns	200 ns	70 ns	180 ns
Write cycle	12 ns	70 ns	5 ms	1 s	180 ns
Data write method	Overwrite	Overwrite	Erase & Write	Erase & Write	Overwrite
Write Endurance	∞	∞	10^4	10^3	10^{10}

27.2.3.h Memory management unit (MMU)

The physical memory is divided into memory pages with the assistance of the MMU. Inside the MMU a translation of the memory address occurs, and the visibility of the storage space is changed.

Each segment of the physical memory is placed into the logical memory. Figure 27.8 shows the behavior of the address translation.

Figure 27.8 Memory segmentation.



The size of one segment is programmable by the processor and can comprise the whole physical memory or only a small portion of it. It is only limited by the maximum number of allowed segmentations. The processor view of the memory may differ from the real memory alignment, since there is no direct connection between the processor and the memory.

In most cases, the MMU is not enabled after power-on or a global reset. At this point, the processor operates in "system" mode and the virtual addresses are equal to the physical addresses. All special function registers (SFR) of the MMU are accessible by the CPU; these registers perform the address translation.

When an application is executed from the code space the MMU is enabled and it is then in "user" mode. Direct access to the SFR is disabled to protect other applications. Thus, each application can only use its own memory region. This region contains the shared ROM for the program code, and a part of the RAM for the variable data (see Figure 27.9). Memory protection is distinctively important for the non-volatile data in the EEPROM, since that is where applications store sensitive data, and it is necessary to protect this information against spying or loss of power.

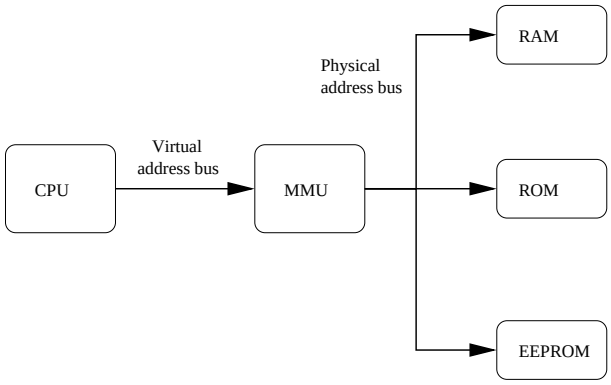
27.2.4 Environment and software

Smart card systems consist of two parts: the *host system*, residing in the terminal (or in a PC with a card reader) and the *card system*, inside the smart card. This section looks at the smart card software, including the system software and the user applications that run on the card system.

Developing a smart card application traditionally was a long and difficult process. Most smart card development tools were built by the smart card manufacturers using generic assembly lan-

guage tools and dedicated hardware emulators obtained from the silicon chip vendors. Therefore, developing smart card applications was limited to a group of highly skilled and specialized programmers who had intimate knowledge of the specific smart card hardware and software. They had to deal with very low-level communications protocols, memory management and other minute details, dictated by the specific hardware of the smart card. Upgrading software or moving applications to a different platform was particularly difficult or even impossible. Furthermore, because smart card applications were developed to run on proprietary platforms, applications from different service providers could not coexist and run on a single card. Lack of interoperability and limited card functions prevented a broader deployment of smart card applications. The development of the Java Card technology changed this situation. This language, designed by SUN technology, offers new possibilities for smart cards, as will be seen in the following sections.

Figure 27.9 Example for the processor view.



27.2.4.a Operating System (OS)

Each operating system depends on the manufacturer's philosophy. The conventional operating system is the one based on standard instructions, which is designed such that it can be implemented for any application on the component. Smart card operating systems support a collection of instructions that the user's applications can access.

Many operating systems have been designed for smart cards: Windows ® for smart cards, which is no longer used, Multos, Java Card, etc. There is no real standard for a smart card operating system. The OS is strongly linked to the manufacturer, and standardizing it would affect the intellectual property of the manufacturer. The group SCOPE, directed by GlobalPlatform, tried to design a document specifying the main functionality that should be supported by a smart card OS. The ISO 7816-4 [ISO 1999d] standardizes a wide range of instructions in the format of the application protocol data unit, APDU. A smart card operating system may support some or all of these APDUs as well as the manufacturer's additions and extensions.

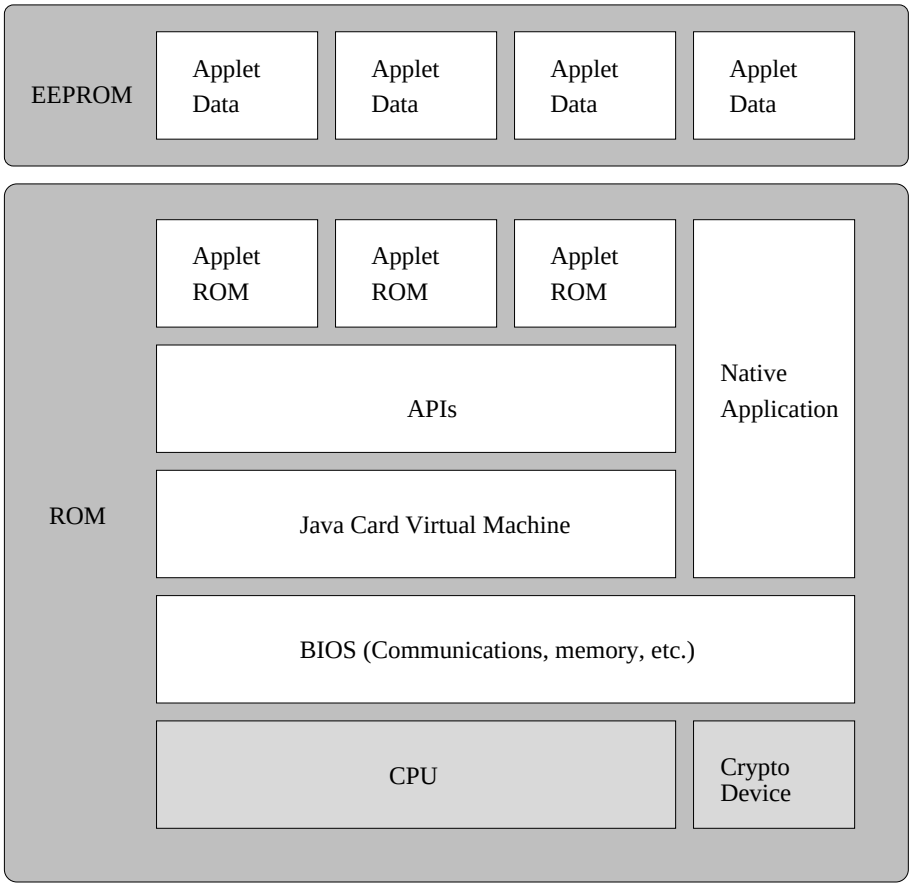
The operating system on a smart card has fewer operations to deal with than those present in standard personal computers. Its purpose is essentially to regulate the input and output (I/O), timers, exceptions, communications between the server and the terminal, memory management, and to

securely load and run specific programs. In Figure 27.10, the OS is composed of the following elements:

- the BIOS: for the protocol communication T (0, 1, both, or others), the timers, the exceptions, the EEPROM management,
- the Java Card Virtual Machine (VM),
- some native applications used for access rights on memory.

The majority of applications are applets, native applications and the link with the Java Card VM. No external instruction is allowed to interfere with the operation of the card. System crashes or uncontrolled reactions due to a faulty instruction or as result of failed EEPROM sections must not occur under any circumstances.

Figure 27.10 Architecture of the smart card.



27.2.4.b Java Card

Java Card™ offers a way to overcome obstacles hindering smart card acceptance. It allows smart cards and other memory-constrained devices to run applications (called applets) written in the Java programming language. Essentially, Java Card technology defines a secure, portable, and multi-application smart card platform that incorporates many of the main advantages of the Java language. This type of platform has found wide acceptance and is currently being shipped in high volume. A Java Card platform-based smart card runs Java based applications in the form of byte-code. These are loaded into the memory of the smart card's microprocessor where they are run by the virtual machine (typically in the EEPROM). The advantages provided are:

- Multiple application management: multiple Java applications (electronic purse, authentication...) can reside on a single card. These applications can be upgraded with new or updated applets without the need of issuing a new or a different card.
- Security: the different applets present on the card are each separated by the applet firewall. It ensures their integrity and eliminates program tampering; the level of access to all methods and variables is strictly controlled.
- Hardware independence: Java Card technology is independent of the type of component used. It can run on any smart card (8, 16, or 32 bits) because the applets are written on top of the Java card platform.
- Compatibility: any card can run any application. Java Card technology is based on the Java Card international standard ISO 7816, applets can interoperate not only with all Java smart cards but also with existing card acceptance devices.
- High level development: applet developers do not have to deal with the details of microcontroller, with the exception of cryptographic algorithms.

The main elements in Java Card are:

- The Java Virtual Machine: necessary to compile the byte codes of the applet and to run the applications.
- The different applets that can be loaded in EEPROM or stored on ROM for some special applications.
- The APIs: these are the classic APIs of the functions used in the card, for example the cryptographic functions. These are designed and standardized by Sun to ensure that applets are portable.

For more detailed information, the Java Card environment is described in the book *Java Card Technology for Smart Cards: Architecture and Programmers Guide* [CHE 2000].

27.3 Smart card interfaces

27.3.1 Transmission protocols

This section explains the structure of transmission and specific control in half-duplex transmission protocols. Three types of protocols are defined in the ISO/IEC standards 7816 [ISO 1999a, ISO 1999b, ISO 1999c, ISO 1999d] and 14443 [ISO 2000, ISO 2000a, ISO 2000b, ISO 2000c, ISO 2000d]: one character protocol (T = 0), and two block protocols (T = 1 and T = CL). The designation T = CL is currently in the final stage of specification, which is why it does not appear in literature. Every smart card must support at least one of these protocol types, while terminals need to support all of them.

Table 27.11 Transmission protocols [RAEF 2000].

Protocol	Explanation	Specification
T = 0	Asynchronous, half-duplex, byte-oriented	ISO/IEC 7816-3
T = 1	Asynchronous, half-duplex, byte-oriented	ISO/IEC 7816-3
T = 2	Asynchronous, half-duplex, byte-oriented	ISO/IEC 10536-4
T = 3	Full-duplex	–
T = 4	Asynchronous, half-duplex, byte-oriented, extension of the T = 0 protocol	–
T = 5-13	Reserved for future use	–
T = 14	Reserved for national use (in Germany: Deutsche Telekom)	Proprietary standards
T = 15	Reserved for future use and extensions	–

Altogether, the ISO/IEC standards comprise of fifteen types of transmission protocols. Up to now, only three of them are fully specified while the rest is either still under definition, reserved for future use, or reserved for national use (see Table 27.11).

The half-duplex block transmission protocol T = CL addresses the special needs of contactless card environments and is not listed in Table 27.11. It sets out the requirements for contactless smart cards to operate in the vicinity of other contactless cards that conform to the ISO/IEC 10536 and ISO/IEC 15693 standards.

Which transmission protocol is to be used for subsequent communication between terminal and smart card is indicated within the answer-to-reset (ATR) interface bytes and shall always be T = 0 or T = 1 for cards with contacts. If the interface bytes are absent, T = 0 is assumed by default. For contactless cards only T = CL is designated. In Europe only Germany uses the T = 1 protocol, whereas the rest of Europe uses T = 0 protocol.

27.3.1.a Protocol T = 0

The half-duplex asynchronous transmission protocol T = 0 (cf. standards [ISO 1999a, ISO 1999b, ISO 1999c, ISO 1999d]) is the basic protocol for cards with contacts. It is the oldest transmission protocol and is designed for a minimum of technical requirements. For this reason, it is used in the present GSM technology for mobile phones. The protocol T = 0 is completely byte-oriented and it uses the following character frame format: a *character* consists of 10 consecutive bits:

- 1 start bit in state "Low,"
- 8 bits, which comprise the data byte,
- 1 even parity checking bit.

A character is the smallest data unit that can be exchanged. The interval between two consecutive character's leading edge start bits is comprised of the duration of the data plus a *guard time*. The minimum value of this guard time is 2 Elementary Time Units (ETU). Thus, for T = 0 the minimum duration between two consecutive start bits is 12 ETU. With the ATR parameter N, the guard time value can be changed. Its value represents the number of ETU to be added.

During the guard time, both communication partners are in receive mode (I/O line in state "High"). The transmitter checks the state of the I/O line by sampling the I/O port. In the event of an error-free transmission, the I/O line remains in the "High" state and the next character is

expected after the guard time has expired.

If the card or the terminal detects a parity error in the received character, it sets the I/O line to the "Low" state for one or two ETU to indicate an error. Since the transmitter samples the I/O port during the guard time, it detects the "Low" state and identifies it as a parity error. After detection of the error signal, the sender repeats the character instead of transmitting the next one.

The error signal and character repetition procedure is mandatory for all cards offering the T = 0 protocol.

27.3.1.b Protocol T = 1

The half-duplex asynchronous transmission protocol T = 1 (cf. standards [ISO 1999a, ISO 1999b, ISO 1999c, ISO 1999d]) consists of block frames exchanged between the two communication partners. They convey either application data transparent to the protocol or transmission control data including transmission error handling. The protocol T = 1 is designed using the layering technique found in the Open System Interface (OSI) model [RAEF 2000], with particular attention paid to the minimization of interactions across boundaries.

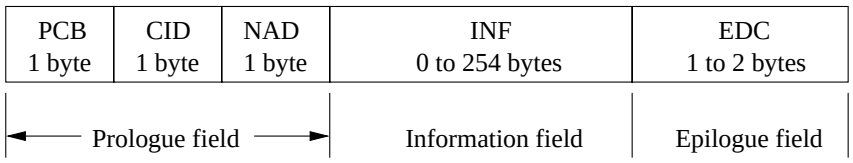
A block is the smallest unit exchanged. It is defined as a sequence of bytes, whereby each byte is conveyed in a single character, see Section 27.3.1.a for the definition of a character. The protocol always starts with a block sent by the terminal and continues with alternating the right to send a block. The block structure allows us to check the received data before processing the conveyed data.

Within a block, the standard duration between the start bit leading edges of two consecutive characters is 12 ETU. The error signal and character repetition procedure, as defined for T = 0, does not take place in the T = 1 block protocol. Thus, the minimum delay between the leading edge of the start bits of two consecutive characters may be reduced from 12 ETU to 11 ETU. If the corresponding ATR interface parameter indicates N = 'FF' (N = 255), this leads to a guard time of only 1 ETU.

A block consists of the following three fields where the items in brackets indicate optional requirements (see Figure 27.12):

- The "Prologue" field consists of a "Node Address" byte (NAD), a "Protocol Control Byte" (PCB) and a "Length" byte (LEN).
- The "Information" field is optional. If present, it is comprised of 0 to 254 bytes (INF).
- The "Epilogue" field consists of one or two mandatory bytes for the "Error Detection Code" (EDC).

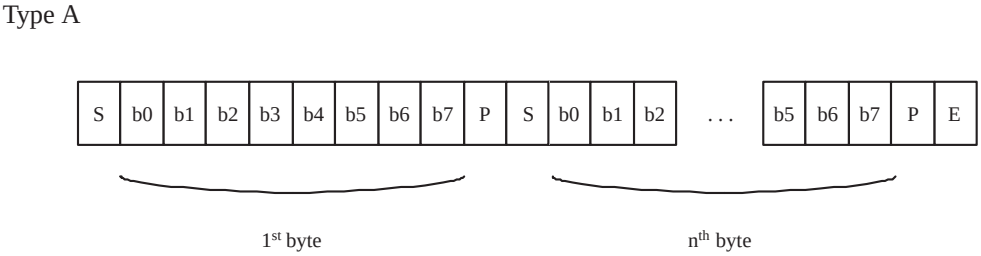
Figure 27.12 Structure of a T = 1 block frame.



27.3.1.c Protocol T = CL

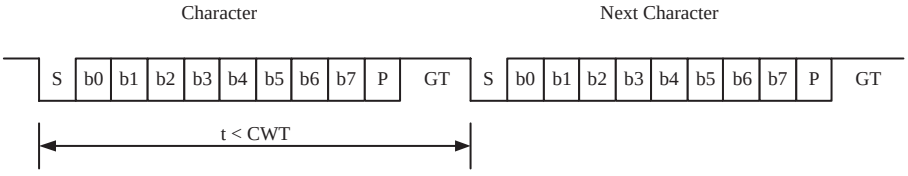
The half-duplex block transmission protocol T = CL controls the special needs of contactless smart card environments [ISO 2000, ISO 2000a, ISO 2000b, ISO 2000c, ISO 2000d] and supports both type A and type B frame formats (see Figure 27.13). An *extra guardtime* is inserted between the characters in type B frame, which is used to separate the single characters. As already annotated, this protocol is currently in the final stage of specification with the designation T = CL, which is the reason why it does not appear in current literature. This protocol is designed according to the principle of layering in the OSI reference model, with particular attention to the minimization of interactions across boundaries, which is similar to T = 1 (cf. Section 27.3.1.b).

Figure 27.13 Type A and type B frame formats.



Another similarity with T = 1 is the method of transferring the character frame formats. The protocol T = CL also exchanges block frames between terminal and smart card. A block is the smallest data unit that can be exchanged, and it may be used to convey application data or transmission control data, including transmission error handling. This is one reason why in T = CL the error signal and character repetition procedure, as specified for the T = 0 protocol, does not take place. Another reason is that T = CL uses different frame formats than the T = 0 or the T = 1 protocols.

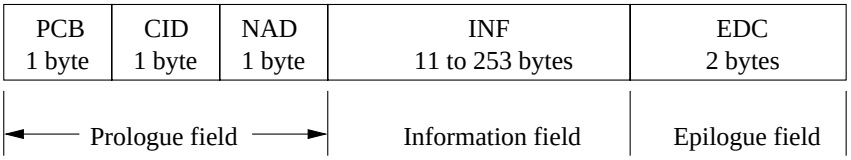
Figure 27.14 Character and next character transmission.



The protocol starts with a first block sent by the terminal and continues with alternating the right to send a block. As shown in Figure 27.14, a block consists of the following three fields, where the items in brackets indicate optional requirements:

- The “Prologue” field consists of up to three bytes, where the first byte (PCB) is mandatory and the two other bytes (CID and NAD) are optional.
- The “Information” field is optional. If present, it comprises 11 to 253 bytes (INF).
- The “Epilogue” field consists of two mandatory bytes for the “Error Detection Code” (EDC).

Figure 27.15 Structure of a T = CL block frame.



27.3.2 Physical interfaces

There exist different interfaces in the smart card environment. The ‘universal asynchronous receiver transmitter’ (UART) is the most used interface for data exchange. There also exist the ‘universal serial bus’ (USB) and, for special cases, radio frequency waves. The UART and the USB are investigated in the following sections.

27.3.2.a UART

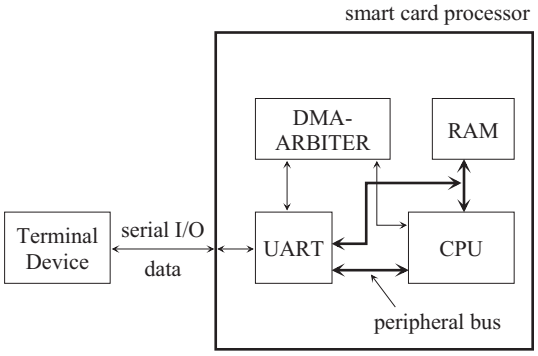
The UART enables hardware supported data transfer at the serial I/O line. In former smart card architectures, the software, in conjunction with the OS, exclusively handled the process of transmitting and receiving data. This was sufficient as long as the desired baud rate did not exceed approximately 111 Kb/s. However, with higher rates, the received bits could not always be identified as valid, since the verification time was too short. An error-free communication was then no longer possible.

Today’s smart card applications are becoming more complex. Apart from the increasing baud rate mentioned above, the security requirements, like cryptographic algorithms, need CPU time for their computation. Another factor is the possibility of several applications running simultaneously during a card session. These developments result in an increasing effort by the CPU, OS, and software to manage all requests in a satisfactory manner.

Using a UART is an efficient solution to increase the speed of data exchange without loading the CPU (see Figure 27.16). The hardware and software, for the UART implementation and control, are of a manageable size. The UART decouples the CPU from direct I/O data handling during communication. In this way, high baud rate can be achieved without decreasing the speed of the running applications. The UART autonomously organizes the transfer via the I/O lines, and, when

necessary, prepares the data for the CPU. Also, there is a UART extension for direct memory access (DMA) operation. The function allows the UART to exchange the data with the memory without assistance of the microprocessor.

Figure 27.16 Example for an UART environment.



27.3.2.b USB

The universal serial bus (USB) specification [USB] is a standardized peripheral connection developed in 1995 by leading companies in the PC industry. The major goal of USB is to define an external expansion bus, which makes adding of peripherals to a PC as easy as possible. In the smart card world, the USB interface serves as a high speed connection between smart card chip and external peripheral components.

With an implemented USB interface, the same smart card chip can be used for communication (card and reader).

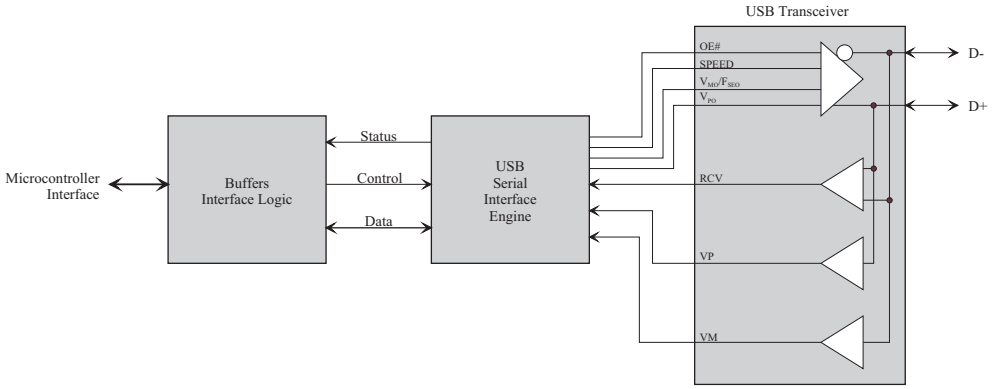
The employment of a standard smart card chip on the reader side leads to a cheaper system, since developing one chip for both sides reduces the costs. This results in a shorter development period. A standard USB interface consists of three components: the Interface Logic, the Serial Interface Engine, and the USB Transceiver. The transceiver (see Figure 27.17) transforms the differential signal at ports D+ and D- into traditional digital logic.

27.4 Types of smart cards

27.4.1 Memory only cards (synchronous cards)

This is the first type of card to be widely used. The prepaid telephone cards mentioned in the introduction are an example of this type of card. The data required for the applications are stored in the EEPROM memory (EPROM for the first cards). In the simplest case the cards use memory that can only be written once, and after use, the memory is deleted and made inoperable (the Thomson ST1200 SGS, introduced in 1983 for the French telephone card, worked in this way). The addition of a security logic device allows more control over the memory access. There now exist more complex memory cards, which can perform simple encryption.

Figure 27.17 Example for an USB interface.



These types of cards are easy to use, the electronics are simple, the chip is small, and the price is low. However, memory space and flexibility are limited, and they are not adapted to security applications.

27.4.2 Microprocessor cards (asynchronous cards)

These cards are equipped with an "intelligent circuit": a processor connected to memory blocks capable of carrying out complex calculations. The added functionality of the microprocessor allows for higher security and application choices. However, as a result, these cards are larger and more complex. It is possible to connect other devices to the microprocessor for communication, special operations or security. Figure 27.18 shows many of the possible components that can be added to the microprocessor card. In smart cards, there are many different types of microprocessors. All of them function as a secured unit, protected from unauthorized access.

Figure 27.18 Components of the microprocessor.

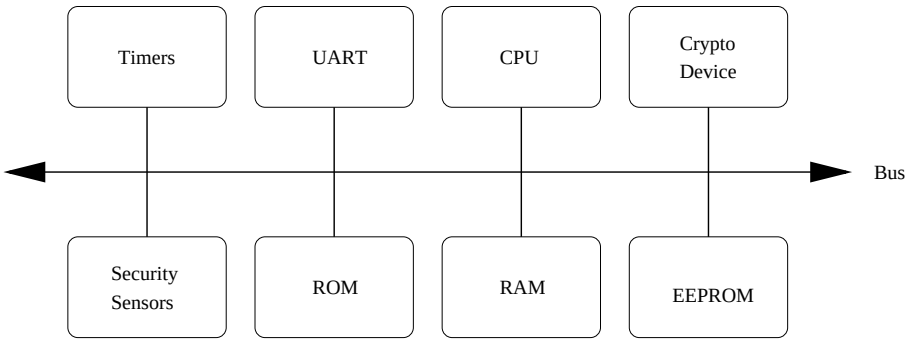


Table 27.19 Characteristics of CISC and RISC based processors.

CISC	RISC
Extensive instruction set	Small instruction set
Complex and efficient machine instructions	Simple instructions
Advanced instructions microencoded	Hardwired machine instructions
Extensive addressing capabilities for memory operations	Few addressing modes
Few registers	Many registers

All microprocessors (and most computers) employ the principle of *stored program digital computer*. This means that data and instructions, which are stored in a memory area, must first be loaded into registers. Then the central processing unit, CPU, operates on these registers and places the results back into the memory areas.

The CPUs used in smart cards are usually built around proven modules from other applications. Many CPUs are based on the CISC (complex instruction set computer) architecture, which requires several clock cycles per instruction. However, CPUs based on the RISC (reduced instruction set computer) architecture are becoming more common. Table 27.19 shows the different characteristics between the CISC and RISC type processors. Many current CISC type processors are based on either one of the two main families: the Intel 8051 or the Motorola 6805 family. Manufacturers, such as Philips, Infineon, and ARM, take the base design of either a CISC or RISC processor and add their own functionality as needed.

The processing speed of the smart card is controlled by a clock circuit normally set to 5 MHz. Modern smart card processors use clock multipliers (by two or four) to increase this operating clock speed for internal calculations. Using clock multipliers, smart cards are able to operate at speeds between 20 and 48 MHz.

The area occupied by the microprocessor on the chip has a big influence on its manufacturing costs and its resistance to bending and shearing forces. Therefore, effort is made to reduce the chip's size as much as possible. The chip's surface area must be less than $25\,\text{mm}^2$. Using current chip technology, 0.25 or 0.30 μm, this means that the microprocessor contains between $150,000$ and $200,000$ transistors. Future microprocessors will be produced using newer 0.18 μm technology.

To provide additional functionality to the smart card, manufacturers add specialized processors and coprocessors to perform only specific tasks. The next section takes a closer look at coprocessors in smart cards.

27.4.2.a Coprocessors

Coprocessors are used on the majority of current chips for special operations and to optimize standard operations. Among those used for cryptography are:

- a coprocessor for DES encryption/decryption,
- a random number generator coprocessor: allows the use of random values in algorithms,
- an arithmetic coprocessor: dedicated to arithmetic operations (modular operations) on long integers; for example, 160-bit or longer integers.

An arithmetic coprocessor element is essential for asymmetric cryptography algorithms such as DSA, ECDSA, etc. Now there are more coprocessors that are not only optimized for RSA operations but also for elliptic and hyperelliptic curves operations. The first ECC cards started to appear around 2001 and 2004 the first smart card using HEC was produced.

Adding such coprocessors has a significant impact on the cost of the chip, increasing it by as much as a factor of ten. This being the case, one may wonder why with increasingly powerful processors it continues to be necessary to add coprocessors. But at the same time, cryptographic algorithms require longer keys to keep them secure, so coprocessors are likely to remain necessary for high performance cards.