

These standards are in turn based on previous ISO standards in the series 7810, 7811, 7812 and 7813, which define the characteristics of identification cards in the ID-1 format. These standards include embossed cards and cards with magnetic strips, which we all know in the form of credit cards.

In the standardization of Smart Cards (referred to in the ISO standard as 'integrated circuit(s) cards', or ICC), compatibility with these existing standards has been taken into account from the very start, in order to provide for a smooth transition from embossed and magnetic-strip cards to Smart Cards. This was achieved by allowing all functional elements, such as embossing, magnetic strips, contacts and interface components for contactless links, to be integrated into a single card. Of course, a consequence of this is that sensitive electronic components, which after all is what integrated circuits are, are exposed to high stress levels produced by the embossing process and the repeated impacts that occur when the embossed characters are printed onto paper. This makes heavy demands on the packaging of the integrated circuits and the manner in which they are embedded in the card.

A summary of the currently available standards, with brief descriptions of their contents, can be found in the Appendix.¹

In the last few years, an increasing number of specifications have been prepared and published by industrial organizations or other non-public groups, with no attempt being made to incorporate them into the standardization activities of ISO. The argument most commonly offered for this manner of working is that the way ISO works is too slow to keep pace with the short innovation cycles of the information and telecommunication industries. Since in many cases only a few firms are involved in the preparation of these 'industry standards', there is a great danger that the interests of smaller firms, and especially the interests of the general public, are not taken into account in the process. It is a great challenge to the future of ISO to come up with a working method that can safeguard general interests and still not hold back the pace of innovation.

¹ See Section 15.3, 'List of Standards with Comments'.

2

Types of Cards

As already mentioned in the Introduction, Smart Cards are the youngest member of the family of identification cards using the ID-1 format, which is defined in ISO standard 7810, 'Identification Cards – Physical Characteristics'. This standard specifies the physical properties of identification cards, such as flexibility, temperature resistance and the dimensions of three different card formats (ID-1, ID-2 and ID-3). The Smart Card standards (ISO 7816-1 ff) are based on the ID-1 card, millions of which are used nowadays for financial transactions.

This chapter provides an overview of various types of cards in the ID-1 format. In many applications, the combination of particular functions is of special interest, especially when the cards currently used in an existing system, such as magnetic-strip cards, are to be replaced by Smart Cards. In such cases, it is usually not possible to replace the existing infrastructure – in this case, magnetic-strip card terminals – with a new technology overnight.

The solution to this problem generally consists of issuing cards that have both magnetic strips and chips, which are to be used during a transition period. Such cards can be used in both the old and the new terminals. Naturally, new functions that are only possible with a chip cannot be utilized with magnetic-strip terminals.

2.1 Embossed Cards

Embossing is the oldest technique for applying machine-readable markings to identification cards. The embossed characters on the card can be transferred to paper using simple and inexpensive devices. Visual reading of the embossing is also straightforward. The nature and location of the embossing are specified in ISO standard 7811, 'Identification Cards – Recording Technique'. This standard, which is divided into five parts, deals with magnetic strips as well as embossing.

ISO 7811 Part 1 specifies the requirements for embossed characters, such as their form, size and embossing height. Part 3 defines the precise positioning of the characters on the card, and defines two separate regions, as shown in Figure 2.1. Region 1 is reserved for the card's identification number, which identifies the card issuer as well as cardholder. Region 2 is reserved for additional data relating to the cardholder, such as his or her name and address.

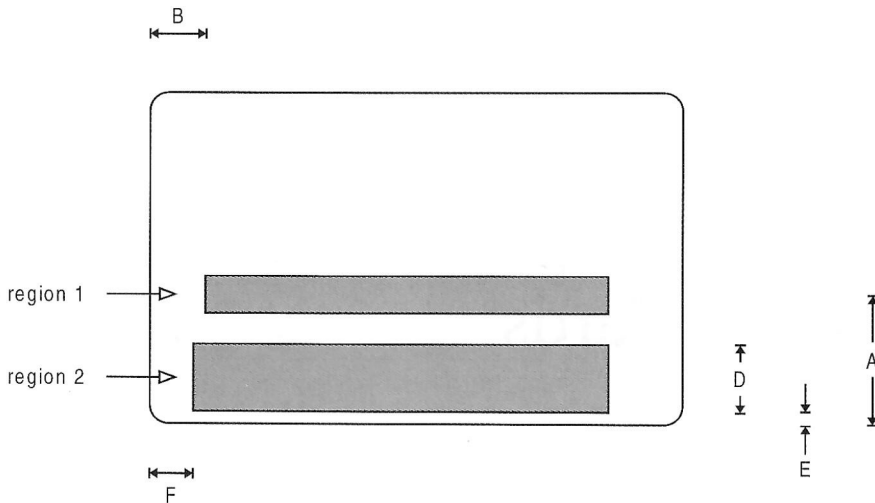


Figure 2.1 Embossing locations according to ISO 7811-3. Region 1 is reserved for the ID number (19 characters), and region 2 is reserved for the cardholder's name and address (4 · 27 characters). $A = 21.42 \pm 0.12$ mm, $B = 10.18 \pm 0.25$ mm, $D = 14.53$ mm, $E = 2.41$ to 3.30 mm, $F = 7.65 \pm 0.25$ mm.

At first glance, transferring information by printing from embossed characters may appear quite primitive. However, the simplicity of this technique has made worldwide proliferation of credit cards possible, even in developing countries. The exploitation of this technology requires neither electrical energy nor a connection to a telephone network.

2.2 Magnetic-strip Cards

The fundamental disadvantage of embossed cards is that their use creates a flood of paper receipts, which are expensive to process. One remedy for this problem is to digitally encode the card data on a magnetic strip located on the back of the card.

The magnetic strip is read by pulling it across a read head, either manually or automatically, whereby the data is read and stored electronically. Processing this does not require any paper.

Parts 2, 4 and 5 of ISO standard 7811 specify the properties of the magnetic strip, the coding technique and the locations of the magnetic tracks. The magnetic strip may contain up to three tracks. Tracks 1 and 2 are specified to be read-only tracks, while track 3 may also be written to.

Although the storage capacity of the magnetic strip is only about 1000 bits, which is not very much, it is nevertheless more than sufficient for storing the information contained in the embossing. Additional data can be read and written on track 3, such as the most recent transaction data in the case of a credit card.

The main drawback of the magnetic-strip technique is that the stored data can very easily be altered. Tampering with embossed marks requires at least a certain amount of dexterity and can be easily detected by the trained eye, but changing the data coded on the magnetic strip is relatively easy with a standard read/write device, and it is difficult to afterwards prove that the data have been altered. Furthermore, magnetic-strip cards are often used in automated equipment in which visual inspection is impossible, such as cash

dispensers. The potential criminal, having obtained valid card data, can easily use duplicated cards in such unattended machines without having to forge their visual security markings.

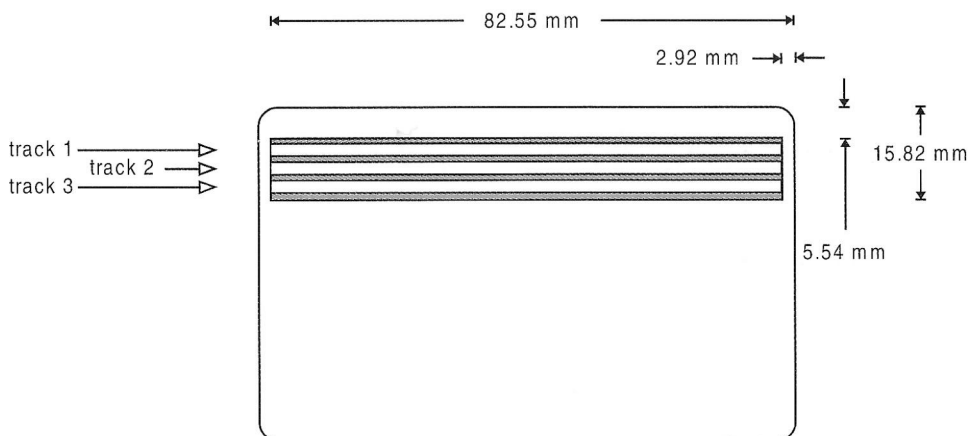


Figure 2.2 Location of the magnetic strip on an ID-1 card. The data region of the magnetic strip is intentionally not extended to the edges of the card, since the use of hand-operated card readers causes rapid wear at the ends of the strip.

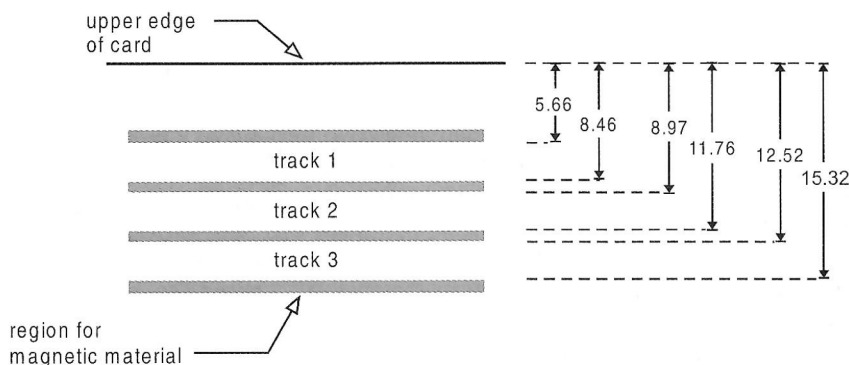


Figure 2.3 Locations of the individual tracks on an ID-1 card (all dimensions in mm)

Manufacturers of magnetic-strip cards have developed various means to protect the data recorded on the magnetic strip against forgery and duplication. For example, German Eurocheque cards contain an invisible and unalterable code in the body of the card, which makes it impossible to alter or duplicate the data on the magnetic strip. However, this and other techniques require a special sensor in the card terminal, which considerably increases costs. For this reason, none of these techniques has so far succeeded in becoming internationally established.

Table 2.1 Standard features of the three tracks on a magnetic-strip card, as specified in ISO 7811

Feature	Track 1	Track 2	Track 3
Amount of data	79 characters max	40 characters max	107 characters max
Data coding	6-bit alphanumeric	4-bit BCD	4-bit BCD
Data density	210 bpi (8.3 bit/mm)	75 bpi (3 bit/mm)	210 bpi (8.3 bit/mm)
Writing	not allowed	not allowed	allowed

2.3 Smart Cards

The Smart Card is the youngest and cleverest member of the family of identification cards in the ID-1 format. Its characteristic feature is an integrated circuit embedded in the card, which has components for the transmission, storage and processing of data. Data transfer can take place either via the contacts on the surface of the card or via electromagnetic fields, without the use of contacts.

The Smart Card offers a number of advantages when compared with magnetic-strip cards. For example, the maximum storage capacity of a Smart Card is many times greater than that of a magnetic-strip card. Chips with over 32 kB of memory are currently available, and this figure will multiply with each new chip generation. Only optical memory cards, which are described in the next chapter, have greater capacities.

However, one of the most important advantages of Smart Cards consists in the fact that their stored data can be protected against unauthorized access and tampering. As access to data takes place only via a serial interface that is controlled by an operating system and security logic, it is possible to write confidential data to the card so that it can never be read from the outside. The confidential data can be processed only internally by the chip's processing unit. In principle, the memory functions of writing, erasing and reading can be restricted and linked to specific conditions by both hardware and software. This allows the construction of numerous security mechanisms, which can also be tailored to the special demands of specific applications.

This together with the ability to compute cryptographic algorithms allows Smart Cards to be used to implement convenient security modules that can be carried about at all times, for example in a briefcase.

Further advantages of the Smart Card are its high degree of reliability and long life as compared to magnetic cards, whose service life is generally limited to one or two years.

The fundamental characteristics and functions of Smart Cards are laid down in the ISO 7816 series of standards. These standards are described in detail in the following chapters.

Smart Cards can be divided into two groups, according to differences in both functionality and price: *memory cards* and *microprocessor cards*.

2.3.1 Memory cards

Figures 2.5 and 2.6 show architectural block diagrams of memory cards.

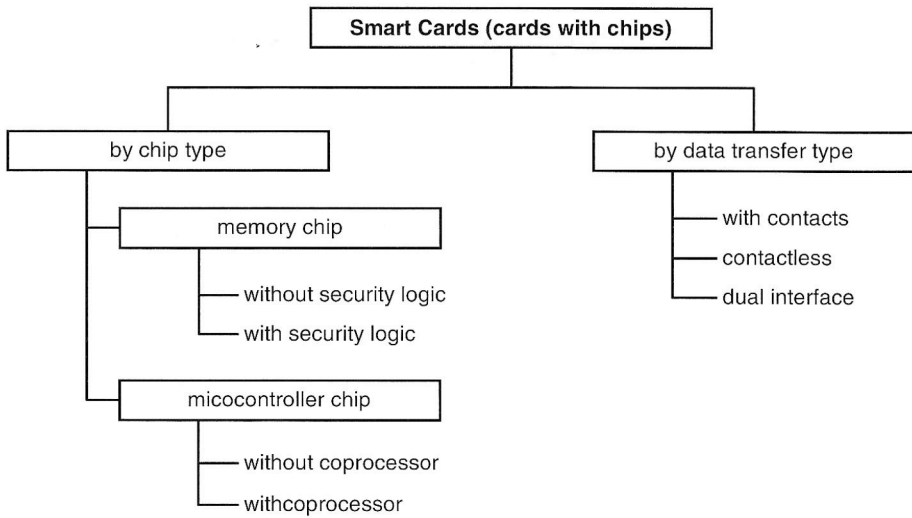


Figure 2.4 Classification chart for cards containing chips

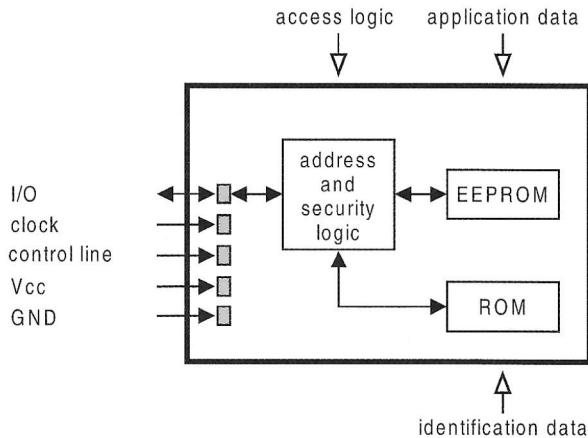


Figure 2.5 Typical architecture of a contact-type memory card with security logic. The figure shows only basic energy and data flows and is not a detailed schematic diagram.

The data required for the application are stored in the memory, which is usually EEPROM. Access to the memory is controlled by the security logic, which in the simplest case consists only of write or erase protection for the memory or some of its regions. However, there are also memory chips with more complex security logic, which can also carry out simple encryption. Data are transferred to and from the card via the I/O port. Part 3 of the ISO 7816 standard defines a special synchronous transfer protocol that allows a particularly simple and economical chip implementation. However, some Smart Cards use the I²C bus, which is commonly used with serial-access memories.

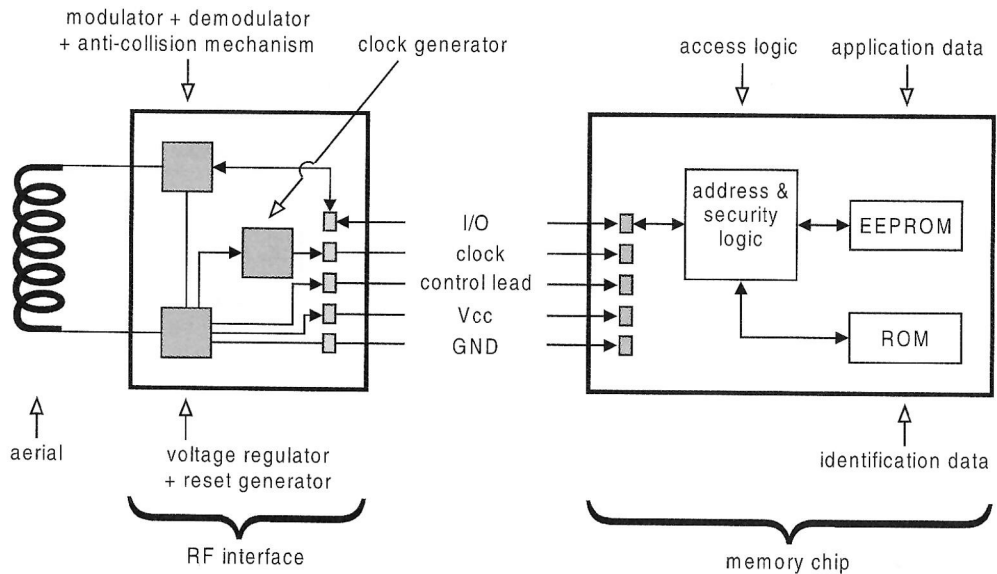


Figure 2.6 Typical architecture of a memory card with security logic and a contactless interface. The figure shows only basic energy and data flows, and is not a detailed schematic diagram.

Memory card functions are mostly optimized for a particular application. Although this severely restricts their flexibility, it makes them particularly inexpensive. Memory cards are typically used for pre-paid telephone cards and health insurance cards.

2.3.2 Microprocessor cards

The heart of the chip in a microprocessor card is – as the name implies – a processor, which as a rule is surrounded by four additional functional blocks: mask ROM, EEPROM, RAM and an I/O port. Figure 2.7 shows the architecture of a typical component of this type.

The mask ROM contains the chip's operating system, which is 'burned in' when the chip is manufactured. The content of the ROM is thus identical for all the chips of a production run, and it cannot be changed during the chip's lifetime.

The EEPROM is the chip's non-volatile memory. Data and program code can be written to and read from the EEPROM under the control of the operating system.

The RAM is the processor's working memory. This memory is volatile, and all the data stored in it are lost when the chip's power is switched off.

The serial I/O interface usually consists only of a single register, through which data are transferred bit by bit.

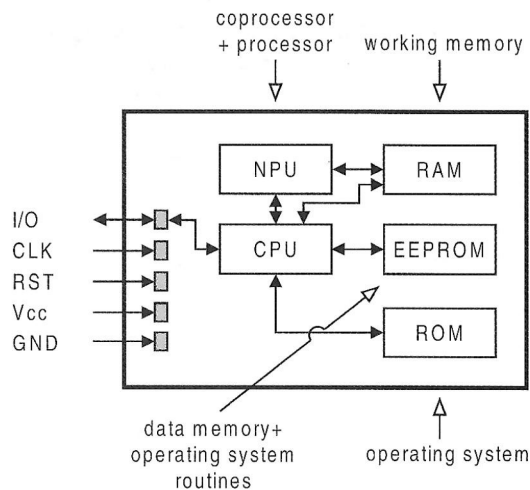


Figure 2.7 Typical architecture of a contact-type microprocessor card with a coprocessor. The figure shows only basic energy and data flows, and is not a detailed schematic diagram.

Microprocessor cards are very flexible in use. In the simplest case, they contain a program optimized for a single application, and thus can be used only for this application. However, modern Smart Card operating systems allow several different applications to be incorporated into a single card. In this case, the ROM contains only basic instructions of the operating system, while the application-specific part of the program is loaded into the EEPROM after the card has been manufactured. The most recent developments even allow application programs to be loaded into a card retroactively, after the card has already been personalized and issued to the cardholder. Special hardware and software measures ensure that the various security requirements of the individual applications are not violated by this capability. Special, optimized microprocessor chips with high processing capacities and large memory capacities are presently being developed, so we can expect to see suitable cards available in the near future.

2.3.3 Contactless Smart Cards

Smart Cards that employ contacts comply with the eight-contact layout specified in ISO standard 7816 Part 1. The reliability of contact-type Smart Cards has been steadily improved over the past years, due to increasing experience in the manufacturing of such cards. The failure rate of telephone cards over their expected lifetime of one year, for instance, is currently well under one in a thousand. Nevertheless, contacts remain one of the most frequent sources of failure in electromechanical systems. Disturbances can result from contamination or wear of the contact, for example. In mobile equipment, vibrations can produce brief intermittent contacts. Since the contacts located on the card's surface are connected directly to the inputs of the integrated circuit embedded in card, there is a risk of damage or destruction of the integrated circuit by electrostatic discharges – and charges of several thousand volts are by no means rare.

These technical problems are elegantly circumvented by the contactless Smart Card. In addition to its technical advantages, contactless-card technology also offers the card issuer and cardholder a range of interesting new potential applications. For example, contactless cards need not necessarily be inserted into a card reader, since there are systems that work at a distance of up to one meter. This is a great advantage in access-control systems in which a door or turnstile needs to be opened, since an individual's access authorization can be checked without requiring the card to be removed from a purse or pocket and inserted into a reader. One extensive application area for this technology is local public transport, in which a large number of people must be identified in the shortest possible time.

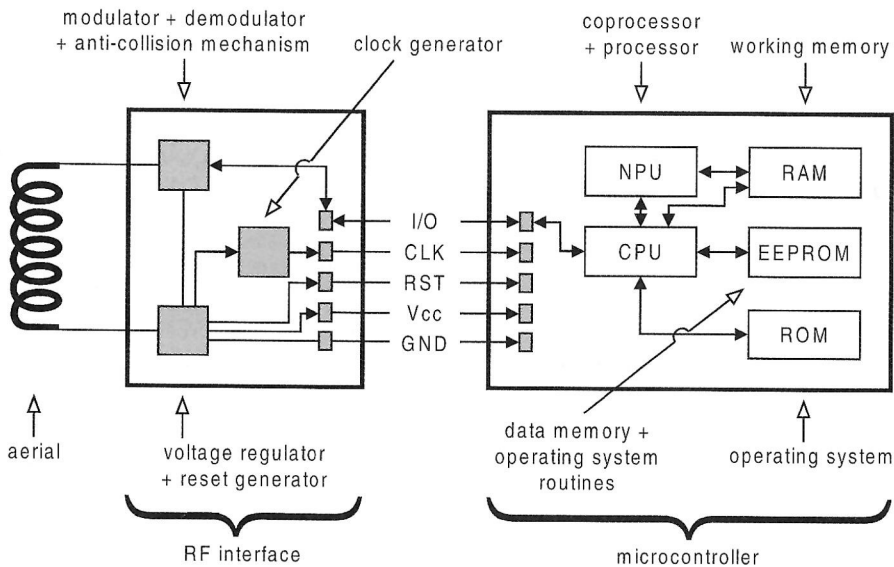


Figure 2.8 Typical architecture of a microprocessor card with a coprocessor and a contactless interface. The figure shows only basic energy and data flows, and is not a detailed schematic diagram.

However, contactless technology is also advantageous in systems that do require deliberate insertion of the card into a reader, since it does not matter how the card is inserted in the reader. This contrasts with magnetic cards or cards with contacts, which work only with a specific card orientation. Freedom from orientation restrictions simplifies operation and thus increases customer acceptance.

A further interesting variation on the use of contactless cards relates to using a surface terminal. Here the card is not inserted in a slot, but simply placed on a marked location on the surface of the card reader. In addition to simplicity of use, this solution is attractive because it significantly reduces the risk of vandalism (for example, by forcing chewing gum or superglue into the card slot).

As far as marketing is concerned, contactless technology offers the benefit that no technical elements are visible on the card's surface, so that the card's visual design is not constrained by magnetic strips or contact surfaces.

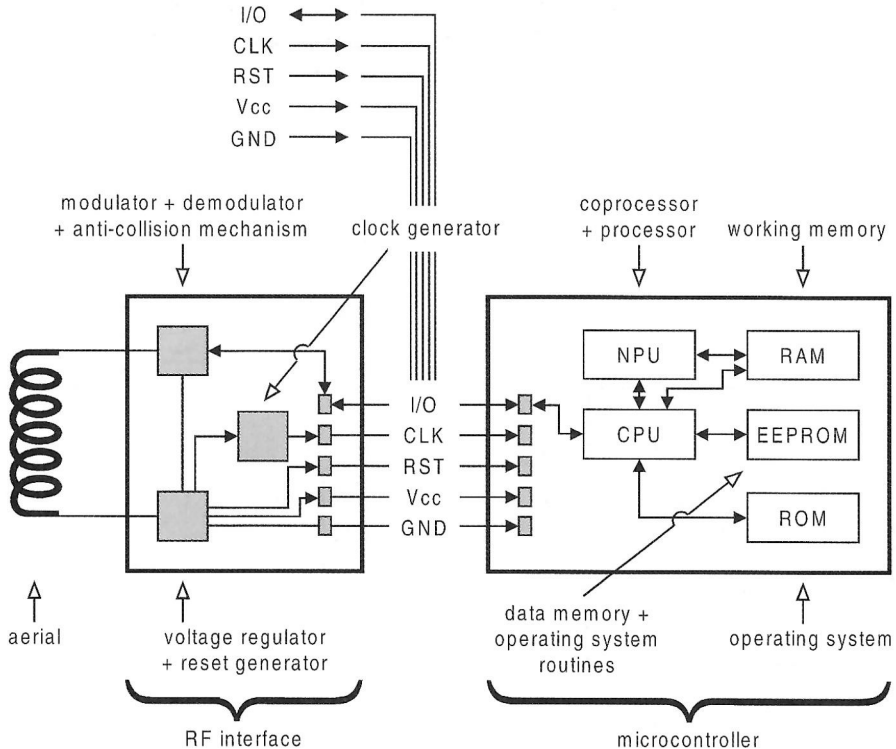


Figure 2.9 Typical architecture of a microprocessor card with a coprocessor and both contactless and contact interfaces. The figure shows only basic energy and data flows, and is not a detailed schematic diagram.

Manufacturing technology for the mass production of contactless cards has matured to the point that high-quality products are available at prices that do not significantly differ from those of comparable contact-type products. Up to now, contactless cards have been used predominantly for local public transportation systems, in which they act as electronic tickets in modern electronic fare systems. The systems presently in use mainly employ single-function cards, for which inexpensive chips with hard-wired logic have been developed.

However, there are increasing signs of a demand for adding supplementary functions to the electronic ticket. For this reason, the use of multifunction cards with integrated microprocessors will increase in the near future. With these cards, payment functions will primarily be performed using the conventional contact-based technique in order to utilize existing infrastructures, such as electronic purse systems. These new multifunction cards have both contacts and contactless coupling elements, and are referred to as dual-interface cards or combicards.

The technology and operating principles of contactless Smart Cards are described in detail in Section 3.6.

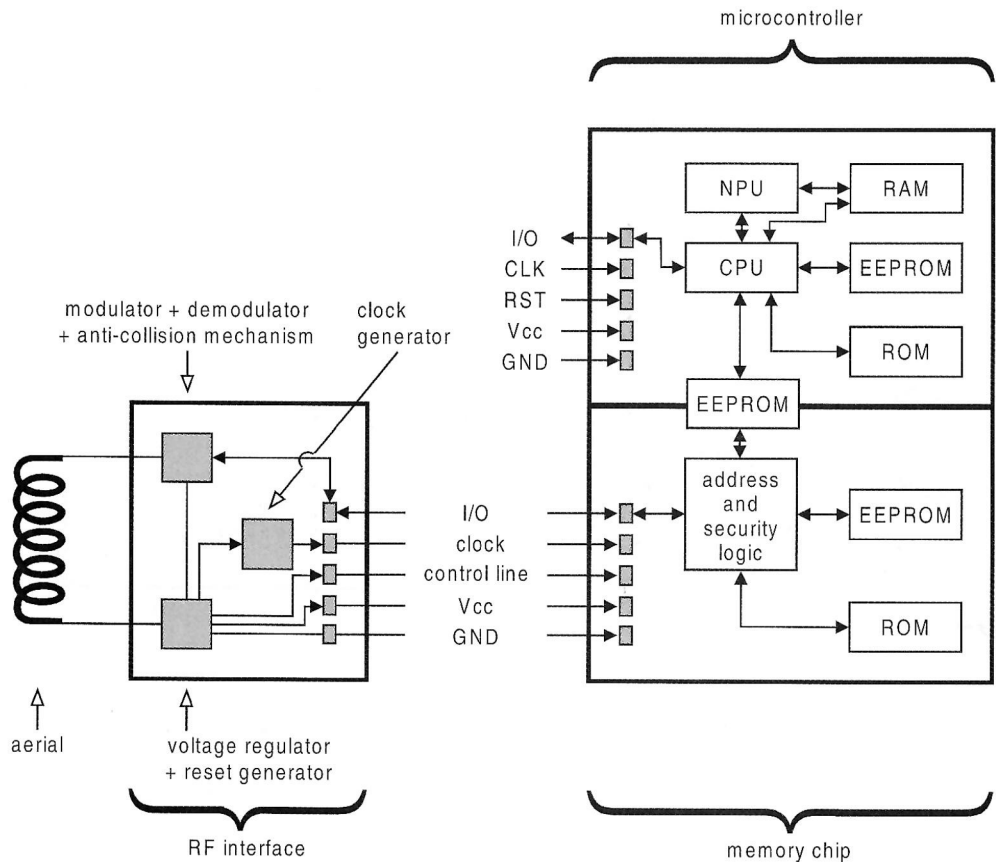


Figure 2.10 Typical architecture of a dual-interface card, which is a combination of a contactless memory card and a contact-type microprocessor card. The figure shows only basic energy and data flows, and is not a detailed schematic diagram.

2.4 Optical Memory Cards

In applications where the storage capacity of Smart Cards is insufficient, it is possible to use optical cards that can store several megabytes of data. However, with currently available technology these cards can be written only once and cannot be erased.

The ISO/IEC 11 693 and 11 694 standards define the physical characteristics of optical memory cards and the linear data-recording technique.

Combining the high storage capacity of optical cards with the intelligence of Smart Cards results in interesting new features. For example, data can be written in encrypted form to the optical memory, while the key is securely stored in the chip's private memory. The optically stored data is thus protected against unauthorized access. Figure 2.11 shows the typical layout of an optical Smart Card with contacts and a magnetic strip. It can be seen that the available optical storage area is restricted by the chip's contacts, which of course reduces the total storage capacity. The magnetic strip is located on the card's reverse side.

Equipment for reading and writing optical storage cards is currently still very expensive, which so far has seriously restricted the use of this type of card.

Optical cards are used for example in the medical sector for recording patient data, where their large memory capacity allows even X-ray images to be stored on the card.

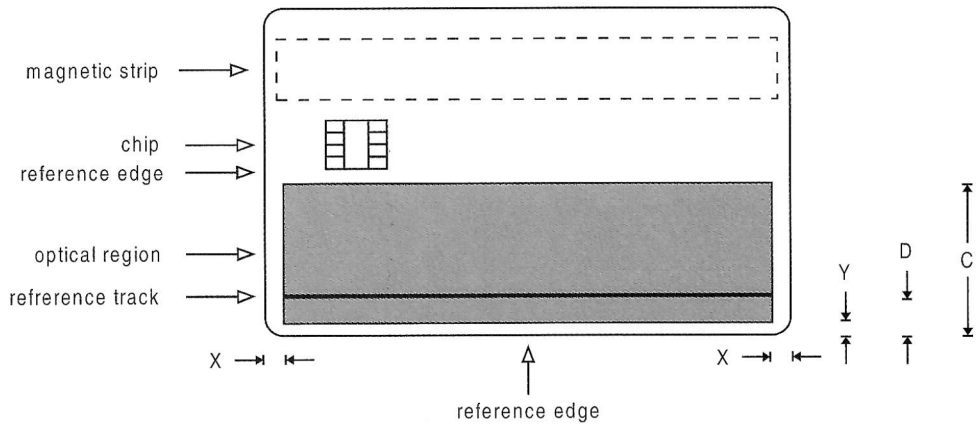


Figure 2.11 Location of the optical storage area on an ID-1 card according to ISO/IEC 11 694-2. $C = 9.5$ to 49.2 mm, $D = 5.8$ mm ± 0.7 mm; $X = 3$ mm max with PWM, 1 mm max with PPM; $Y = 1$ mm min with PWM ($Y < D$), 4.5 mm max with PPM. (PWM = pulse-width modulation, PPM = pulse-position modulation)

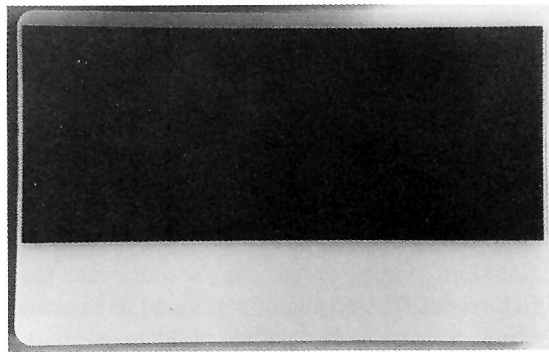


Figure 2.12 Photograph of a typical optical memory card with a net storage capacity (with error correction) of around 4 MB. The raw capacity (without error correction) is around 6 MB.

Operating system APIs

The original Smart Card operating systems did not allow third-party software to be loaded into the card, where it could be executed as necessary. Consequently, conventional operating systems do not have any published programming interfaces that can be used by third parties to call operating system functions. New developments in Smart Card operating systems, such as MULTOS and operating systems that support Java, have made it possible for other parties to load their own program code into Smart Cards. These operating systems include well thought out APIs (application programming interfaces) that allow access to the most important functions of the operating system. This means that third-party programs do not have to contain code for routines that are already present in the operating system. Naturally, practically all operating systems have their own internal APIs, but these are not designed for external use and are usually confidential.

As an exception to the usual situation in the Smart Card world, there are no general standards relating to APIs for Smart Card operating systems. Instead, there are primarily two industry standards that have become established up to now. One of these is the Java Card API and the other is the Multos API. The APIs described in the associated specifications allow file access via the File Manager, calls to the available cryptographic functions and naturally also the transmission and reception of data. A Smart Card operating system with a complete operating system API, and which also allows program code to be downloaded, actually differs from the well-known PC operating systems only in the amount of memory that it has available.

5.5 Memory Organization

The three different types of memory in Smart Card microcontrollers have completely different properties. ROM can only be programmed by means of a mask during manufacturing, and it is programmed all at once. It is static for the life of the chip. Given its construction, the probability of an undesired change in the content of the ROM is practically zero.

RAM is the opposite of ROM. It retains its content only as long as power is applied to the Smart Card. A power failure causes the total loss of all data stored in the RAM. However, data can be written to the RAM at the full working speed of the processor, and the RAM can be erased an unlimited number of times. EEPROM, on the other hand, can retain data without external power. However, it has three disadvantages, which are its limited lifetime, long writing and erase times (around 1 ms/byte) and page structure.

The program code for the operating system that is stored in the ROM does not have to comply with any particular organization, except for the interrupt vectors which have fixed formats prescribed by the microcontroller. The individual program segments can thus be connected to each other in any desired sequence, although an attempt is made to limit the length of software jumps in order to save memory space. What is important is that the ROM is protected by error detection codes (EDCs), since it is certainly possible for errors to occasionally occur in the ROM. A scratch in the ROM area of the microcontroller, for example, or a fracture that occurs during the wire bonding process, can result in erroneous data in the ROM. Interestingly enough, this does not necessarily mean that the operating system can no longer function; instead, it is certainly possible that only specific routines will run incorrectly. In order to eliminate such cases from the very start, the ROM is always tested when the operating system starts up to see if it is error-free.

Figure 5.5 shows how a 256-byte RAM is usually partitioned. It is divided into the registers, the stack, general variables, workspace for cryptographic algorithms and the I/O buffer. If an I/O buffer of 256 bytes is required, for example, or if additional variables must be stored in the RAM, the limits of the available memory can be reached very quickly. This problem is solved by having workspace in the EEPROM, which is thus used like RAM. The disadvantage of this is that the writing time is around 10,000 times longer than for RAM addresses. An additional disadvantage is the limited lifetime of EEPROM cells, which, unlike RAM cells, cannot be written an unlimited number of times. However, moving the content of the RAM to the EEPROM is often the only solution—for example, if an I/O buffer is required that is larger than the entire amount of available RAM.

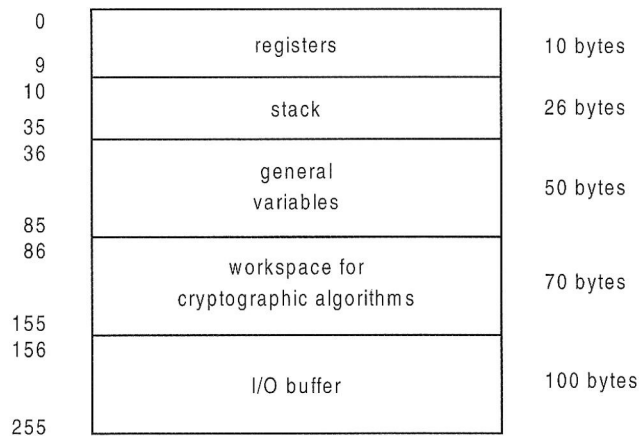


Figure 5.5 Example partitioning of a 256-byte RAM

The partitioning of the EEPROM is much more complicated and intricate than that of either of the other two types of memory. With modern operating systems, the EEPROM is divided up essentially as illustrated in Figure 5.6. In many microcontrollers, there is a region at the beginning of the EEPROM that has special hardware protection. This can be used to store special finishing information, such as a number that is only used once and is thus unique to the chip. Many semiconductor manufacturers also mark the chip type and amount of EEPROM that is available to the operating system in this region. This region is usually designed for write once, read multiple (WORM) access, which means that it can be written only once and after that can only be read. Technically, this is usually achieved by using normal EEPROM cells that however cannot be electrically erased.

On top of this region, which is often 16 to 32 bytes in size, come the tables and the operating system pointers. These are loaded into the EEPROM when the card is completed. The tables and pointers combine with the ROM programs to yield the complete Smart Card operating system, which is fully present in the card only after they have been loaded. This region is protected by an error detection code (EDC) that is recalculated before the first access, or in many cases before every access. This ensures that the operating system can run in a secure and stable environment. If a memory error is detected when the EDC is verified, the affected portion of the EEPROM must subsequently not be used, since an error means that the correct functioning of the operating system can no longer be assured.

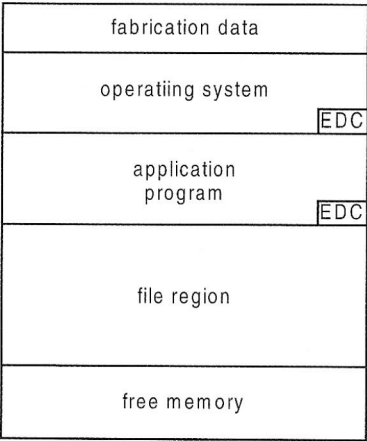


Figure 5.6 An example of how the EEPROM is partitioned by a Smart Card operating system

Table 5.4 Examples of manufacturer’s data that are written by some semiconductor manufacturers to a WORM region of the EEPROM when the chip is finished. The first five manufacturer data items, taken together, yield a unique 8-bit chip number. The major advantage of a chip number generated in this manner is that it does not require highly accurate time information, synchronized over several production locations, but only data that are available to all production machines in every manufacturing facility.

Data element	Size
Semiconductor manufacturer	1 byte
Production facility	1 byte
Semiconductor processing batch number	2 bytes
X coordinate on the wafer	2 bytes
Y coordinate on the wafer	2 bytes
Chip type	4 bytes

Above the protected portion of the operating system, there is a region that contains additional application program code. If need be, this region may also be protected against alterations by a checksum. Application-specific commands or algorithms that should not be located in the ROM or that are too large to fit in the ROM are placed in this region.

The following region contains all of the file structures, or in other words the entire externally visible file tree. This region is not protected as a whole by a checksum, but instead usually has strong file-oriented protection. The internal structure of this region is shown in more detail in Figure 5.7.

At the top of the EEPROM, there is an optional free memory region that has its own memory manager. However, the free memory is often assigned to individual applications in the file region, where it can be used within the applications for the creation of new files. Otherwise it belongs to the general file region and is available for all new applications that are loaded in their entirety.

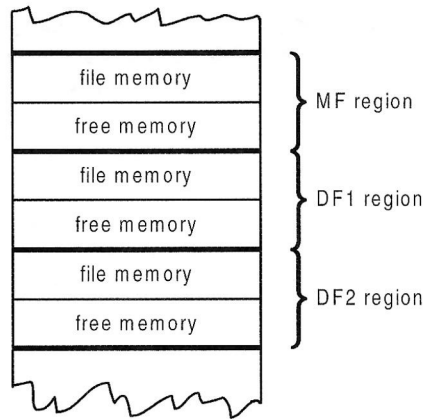


Figure 5.7 Example file region partitioning for a Smart Card operating system that supports multiple applications that are separate and independent

If a file is deleted, the free memory region should inherit the memory that becomes free. If the memory is divided into file memory and free memory regions as described here, only the memory occupied by the final file can actually be assigned to the free memory region if this file is deleted. This severely limits the options of the memory manager, but the limited amount of space for program code in Smart Cards does not allow any other solution.

An ideal solution would be to have true memory management with doubly linked lists for occupied memory and free memory respectively. This would make it easy to create new files and delete files in a manner similar to DOS or Unix file management. However, the cost of the file management program and the overhead for the file descriptors would not represent a reasonable design for a Smart Card operating system. The restrictions are actually not all that painful, since it rarely happens that files must be created or deleted within an application.

5.6 Smart Card Files

In addition to containing mechanisms for identification and authentication, Smart Cards are above all data stores. Here they have a decisive advantage with respect to other storage media, such as diskettes, in that access to the data can be coupled to certain conditions.

The first Smart Cards had only more or less directly addressable memory regions, which could be used for writing or reading data. The data were accessed by specifying physical memory addresses. Nowadays, all newer types of Smart Cards have complete, hierarchical file management systems with symbolic, hardware-independent addressing.

Of course, these file management systems have certain features that are specific to Smart Cards. The most noticeable of these is that there is no man-machine interface. All files are addressed by hexadecimal codes, and all commands are strictly based on this addressing, since the communication takes place here only between two computers. Equally typical of these file management systems is that they are designed to use a small amount of memory. Every redundant byte is avoided if possible. Since the 'user' in the terminal is a computer, this does not present any disadvantages.

There is usually no elaborate memory management, which also helps to keep memory usage as small as possible. If a file is deleted (and only a few operating systems have this capability), the space that comes free does not necessarily become available for use by a newly created file. Normally, all files are created and loaded into the Smart Card when it is initialized or personalized. After this, changes to the file contents are limited.

Naturally, the characteristics of the memory that is used also affect the way that the memory management works. The memory pages in the EEPROM simply cannot be written or erased an unlimited number of times, as can the hard disk of a PC. This means that there are special file attributes that allow information to be stored redundantly, so that it may even be possible to correct any errors that occur.

Internal file structures

Recent file management systems for Smart Cards have an object-oriented construction. This means that all information about a file is stored in the file itself. An additional consequence of this principle is that a file must always be selected before any action can be performed. Files in these object-oriented systems are thus always divided into two parts. The first part, which is called the file header, contains information about the layout and structure of the file, as well as the access conditions. The variable data are stored in the second part (the file body), which is linked to the header by a pointer.

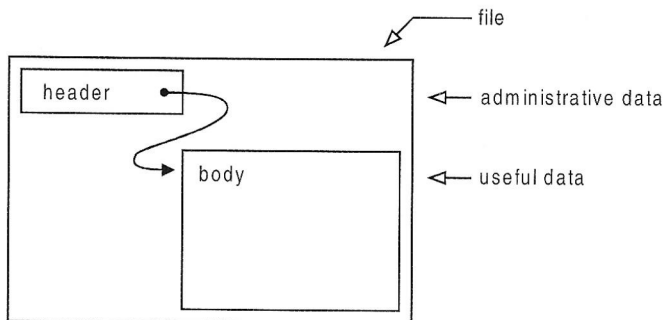


Figure 5.8 The internal structure of a file for a Smart Card file management system

In addition to providing better data structuring, this arrangement also has the advantage of better physical security for the data items. The page-oriented EEPROM that holds all the files allows only a limited number of write/erase cycles. The file header and the file body are always located on separate memory pages. The header, which is normally never altered, stores all the access conditions. A write or erase error related to the file body thus cannot affect these conditions. If the header and the file body were located in the same page of memory, it would be possible to utilize deliberately produced write errors to alter the access conditions such that confidential information could be read from the file body.

5.6.1 File types

The structure of Smart Card file systems, which is specified in the ISO/IEC 7816-4 standard, is similar to the file structures of DOS or Unix systems. The major difference is

that Smart Cards contain no application-specific files, such as special files for a particular word processor. With Smart Cards, only the standardized file structures may be used.

There are basically two categories of files for file Smart Cards. The first category is directory files, which are called 'Dedicated Files' (DFs). The second category consists of the files that hold the actual useful data, which are called 'Elementary Files' (EFs). The DFs act as a sort of file cabinet that contains other, lower-level DFs or EFs that logically belong together. The EFs can be classified into those for the external world (*working EFs*) and those for the operating system (*internal EFs*). The various file types are described below, and their relationships are illustrated in Figure 5.10 (overleaf).

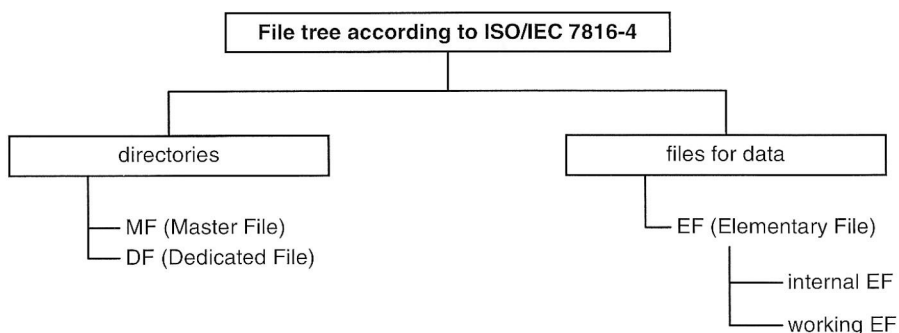


Figure 5.9 Classification of Smart Card file structures according to ISO/IEC 7816-4

MF

The root directory is referred to as the MF (master file). It is implicitly selected after the Smart Card is reset. The MF contains all other directories and all files. It is a special type of DF, and it represents the entire extent of the Smart Card memory that is available for the file region. A master file must be present in every Smart Card.

DF

Dedicated files (DFs) may exist underneath the MF as needed. The name 'Directory Files' is frequently used for these files, although this is contrary to the official definition of the abbreviation 'DF' in the ISO/IEC 7816-4 standard. A DF is a directory in which other files (DFs and EFs) can be grouped together. A DF may contain other DFs. In principle, there is no limit on the number of levels of DFs. However, it is rare for there to be more than two levels of DFs under the MF, due to the very limited amount of memory in Smart Cards.

EF

The useful data that are needed for an application are located in the EFs. 'EF' is the abbreviation of 'elementary file'. EFs may be placed directly under the MF or under a DF. To allow data to be stored in a minimum amount of memory and to logically optimize the data structures, an EF always has in internal file structure. This is the main difference between EFs and files in a PC, whose internal data structures are determined by applications (such as word processors) rather than by the operating system. EFs are divided into working EFs and internal EFs.

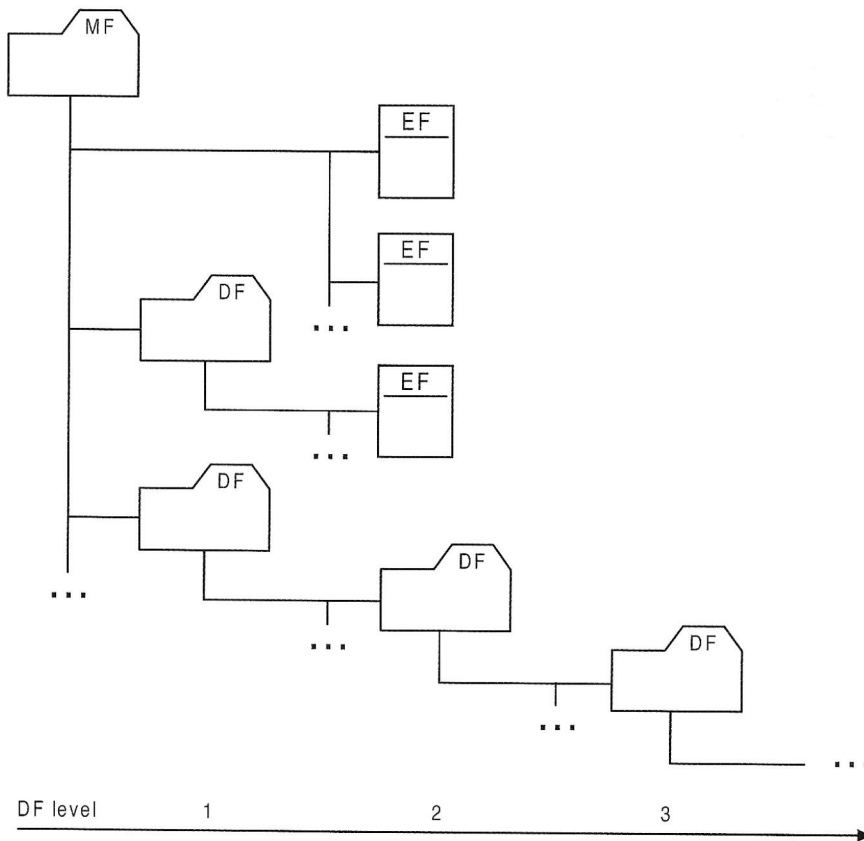


Figure 5.10 The various types of files used in a Smart Card directory structure.

Working EFs All application data that must be read by or written from the terminal, or in other words, all data that are intended for the external world (as seen by the Smart Card), are located in working EFs. The data contained in such files are not used by the operating system.

Internal EFs In addition to the EFs for applications, there are also internal system files that store data for the operating system itself, data for the execution of an application, secret keys or program code. Access to these data is specially protected by the operating system. These system files can be integrated into the file management system in two different ways. In the method specified in the ISO/IEC standard, these files are located in the relevant application DFs and are invisible, so they cannot be selected. They are managed fully transparently by the Smart Card operating system, in a manner that is similar to the management of resource files in the MacOS. In the model specified by the ETSI EN 726 standard, on the other hand, these system files receive regular file names (in other words, FIDs) and can be selected using these names. This is essentially the same principle as is used for file management under DOS. Each of these two approaches has a number of advantages and disadvantages, but they both provide the same functions in somewhat different manners.

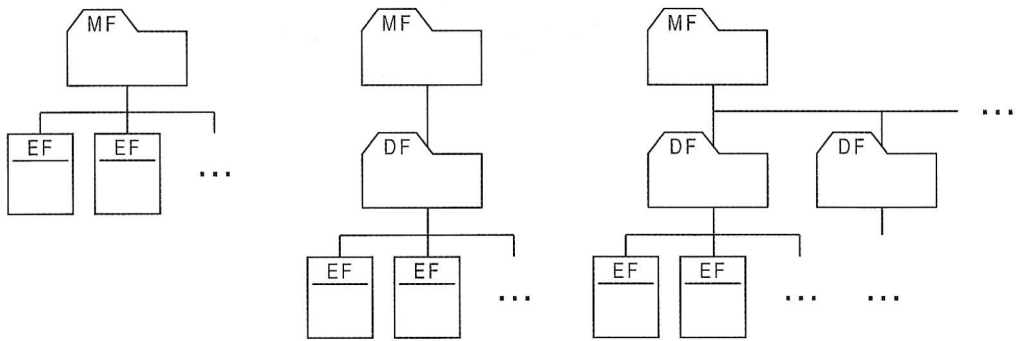


Figure 5.11 The differences between file structures for a Smart Card with only application and one with several applications. The two practical arrangements for cards with only one application are shown on the left and in the middle. The arrangement for a Smart Card with several applications is shown on the right.

Application files

According to convention, all files containing useful data for a particular application (the EFs for that application) are always grouped together in a single DF. This produces a clear and easily understood structure, and it makes it easy to enter a new application into a Smart Card by creating the appropriate DF.

Since the MF is a special sort of DF, it goes without saying that in a single-application Smart Card, all the application files can be placed directly under the MF. In a typical single-application Smart Card, therefore, all the EFs can be located either directly under the MF or in a solitary DF. Smart Cards with several applications have a corresponding number of DFs, in which the EFs belonging to the applications are located.

Additional DFs can be placed within such application DFs. For example, a DF placed directly under the MF could be dedicated to the 'Traffic Control' application. An additional level of DFs within the application DF could contain the files for the languages supported, such as 'English' and 'Deutsch'.

This file structure has an additional advantage with regard to security. Each time a memory access to the application occurs, the operating system can check whether it lies outside the boundaries of the memory allocated to the application DF in question. If so, it may prohibit the access.

5.6.2 File names

In modern Smart Card operating systems, the files are without exception addressed by logical names, rather than via direct physical addresses. The latter approach was perfectly normal in the Smart Card world earlier on, and there are a few Smart Card microcontrollers that still use it. With simple applications that occupy precisely defined memory regions, direct physical access can save a lot of memory space. It also does not result in any loss of user-friendliness, since all files are accessed by the computer in the terminal. However, direct physical access does not in any way fit with the criteria of modern software design, and it also creates major problems for software enhancements and for Smart Card microcontrollers with differing address spaces.

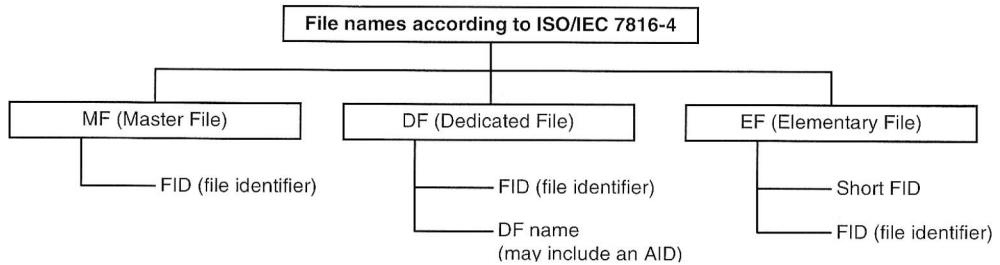


Figure 5.12 Classification chart of Smart Card operating system file names, per ISO/IEC 7816-4

All concepts that use logical file names are significantly better, and above all much easier to extend. Without question, it can be assumed that in a few years file addressing using logical names will be the only type of file addressing used in Smart Cards with microcontrollers. With memory cards, on the other hand, physical file addressing will continue to be used for the foreseeable future.

The File Identifier (FID)

The system described here is based on the ISO/IEC 7816-4 standard, and it is in principle reflected in all other international Smart Card standards. Every file, including directory files, has a 2-byte File Identifier (FID), which can be used to select the file.

For historical reasons, the FID of the MF is '3F00'. This FID is reserved for the MF within the entire logical address space. The logical file name 'FFFF' is reserved for future applications, and may not be used. There are also other FIDs that are reserved by the ISO standard and by other standards. These are listed in Table 5.5.

The GSM application is a typical example of the fact that various groups of FIDs cannot be freely used. In the GSM 11.11 specification, the more significant byte is fixed by the location of the file within the directory structure (file tree). The coding itself has developed historically and originates from the first French Smart Cards. GSM DFs have a value of '7F' for the first (more significant) byte. The FIDs of EFs located directly below the MF have '2F' as the first byte, and the FIDs of EFs located underneath a DF start with '6F'. The less significant bytes are numbered sequentially. This specification applies only to the GSM application and is not a general standard. In other cases, the full 2-byte address region of the FIC can be fully exploited and is not subject to any restrictions.

The FIDs in the directory structure must be chosen such that the files can be unambiguously selected. It is thus prohibited to use the same FID for two different files within the same DF. A DF may also not have the same FID as an EF located directly underneath it, since this would mean that the operating system would have to decide whether to select the DF first or the EF.

The following rules apply to the selection of unambiguous FIDs:

- Rule 1: All EFs within a single directory must have different FIDs.
- Rule 2: Nested directories (DFs) may not have the same FIDs.
- Rule 3: An EF within a directory (MF or DF) may not have the same FID as the next higher or next lower directory.

8

Security Techniques

One of the main advantages of Smart Cards in comparison with other media, such as magnetic-stripe cards or diskettes, is that they can store data in a protected or confidential manner. Consequently, chip hardware that is tailored and optimized for this purpose, with suitable cryptographic procedures for protecting confidential data, is an unavoidable requirement. However, security depends on more than just the special microcontroller hardware and the algorithms implemented in the operating system software. The security of the Smart Card application and the design principles used by its developers are also of fundamental importance. This chapter is a compendium of the essential principles, procedures and strategies that lead to secure Smart Cards and Smart Card applications.

8.1 User Identification

Since ancient times, a number of different methods have been used for the unambiguous identification of persons. The simplest form of identification is an identity card with a photograph or a signature written in the presence of witnesses. The photograph attached to an identity card can be compared with the actual person, and the result is a judgement regarding the genuineness of the person's identity.

In the field of information technology, this comparison is not so easy, since it must be performed by a computer instead of another person. Since computers, in spite of all their success in the execution of mindless activities, still have tremendous problems performing intelligent tasks, entering passwords via a keypad has largely become the preferred technique. The effort needed for the comparison is minimal, since in principle all the computer has to do is to compare the entered password with a stored reference value and make a pure yes/no decision. The password comparison effectively makes a decision regarding the genuineness of the identity of the person being tested.

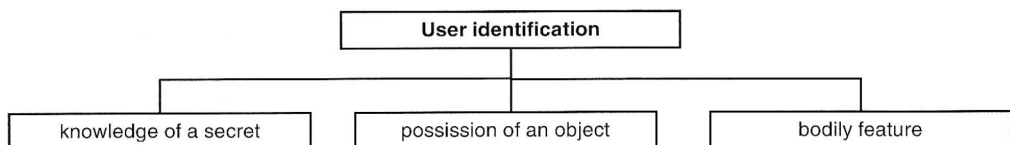


Figure 8.1 Classification of methods for identifying a person

Basically, three different techniques can be used to identify a person. If a password is used, what is tested is whether the person knows a particular secret. If he or she does, the conclusion is that that person is who he or she claims to be. This is similar to testing whether a person possesses a particular object. The third possibility is to test specific, unique bodily features of the person.

Techniques that rely on the knowledge of a secret or possession of a particular item have a significant disadvantage, which is that the person to be identified must either memorize something or carry some item on his or her person. Depending on the situation, the fact that the secret or object can be passed to another person can be considered to be an advantage or a disadvantage. In any case, it is not possible to unambiguously ascertain that the person holding the secret or the object is truly its legitimate owner, rather than someone else who may have illegitimately acquired the secret or item.

The third method of identification eliminates this transferability, since it is based on using specific features the human body for purposes of identification. Of course, the measurements are in most cases technically difficult, since for obvious reasons biological features that are easily measured, such as weight or height, cannot be used.

It is easier to understand these three possible identification methods if you consider the following example. Suppose you have to meet an unknown person at the train station. As soon as you see a possible candidate at the station, you have the problem of deciding whether he is really the person you are looking for. However, if the unknown person shows up at the right place and at the right time, this actually amounts to an implicit test of a secret, since you can at least hope that the place and time of your meeting are not generally known. An explicit test of a secret would occur if the unknown person were to utter a password that is known only to you and him. Alternatively, he could also identify himself by means of an item that he possesses, for example by holding a newspaper printed on a specific day under his arm. Certainly, the most secure method would be to check the person for a specific bodily feature. Perhaps he has a nose like Pinocchio's, which becomes very long when he lies.

This train station scenario clearly shows that the identification of an unknown person can be considered to be a nearly classic problem. It occurs not only in all spy novels but also in everyday life, and it is not limited to computers or Smart Cards.

Entering a PIN into many types of automated equipment and computers has by now become commonplace. The resulting marked increase in the number of PINs used for various purposes makes it very difficult for the ordinary person to keep track of all of his or her PIN codes. After all, who can remember twenty or more different PIN codes? The security and good name of a system are naturally not improved if every user jots down his or her PIN on the card, since the number of cases of fraud will be excessive. For this reason, the desire for other identification methods has arisen in recent years, to replace the use of PIN codes. Biometric features that allow a particular person to be unambiguously identified by a machine are ideal for this purpose.

8.1.1 Testing a secret number

The most commonly used method of user identification is the entry of a secret number, which is generally referred to by the acronym PIN (for personal identification number), or sometimes CHV (cardholder verification).