CHAPTER 12

The Smart Card
A Standardized Security Device Dedicated to Public Cryptology

LOUIS CLAUDE GUILLOU, MICHEL UGON,
AND JEAN-JACQUES QUISQUATER

The smart card will be an important tool in the hand of mankind. It will be a major usage for chip technology.

J. Svigals

At first glance, a smart card appears to be simply an improved traditional credit card. But a smart card is in reality a multipurpose, tamper-resistant security device. Some consider it to be either the ultimate incorruptible cell resisting virus attacks or a fourth level in the hierarchy after the host computer, the departmental computer, and the personal computer. As a matter of fact, these two concepts are not exclusive.

Smart cards are already in widespread public use. Through this user-friendly technology, cryptology is invading our everyday life. This invasion has a large influence on security in various fields of applications, not only in banking, but also in the areas of health, pay television, telephone, home computers, data processing, communication network, and more generally, information technology.

1 INTRODUCTION

Traditional financial cards rely on embossed characters and magnetic stripes for information storage. The relevant existing standards (International Standards Organization [ISO] 7811) specify the characters and stripes in so much detail that there is no additional degree of freedom for any further evolution.

Smart cards rely on VLSI chip technology not only for information storage, but for information processing as well. A microcircuit is embedded in the plastic base of existing smart cards. As illustrated in Fig. 1, the microcircuit consists of an electronic chip bonded to a circuit board and connected to electrical contacts on the board. The relevant existing standards (ISO/International Electrotechnical Commission [IEC] 7816) do not specify the size or the performance of the chip, but rather deal with the

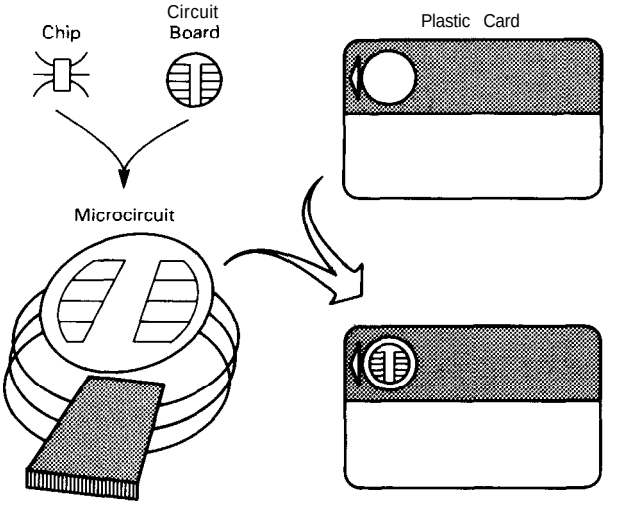

Figure 1 **Integrated circuit card with contacts.**

specification of the interface through which secure transactions are negotiated between the outside world and the embedded electronic circuits.

In the course of a transaction involving a smart card, the card delivers information (stored data, computation results) and/or modifies its contents (data storage, event memorization): The built-in electronic circuits both process data and store information in internal memory. Trade-offs between cost and performance of existing chips dedicated to smart cards are related to the state of the art in VLSI technology and to the current needs of the applications.

Advances in semiconductor technology modify the trade-offs between cost and performance. Smart cards improve in both memory size and processing power at the same rate as any other microprocessor while terminals remain unchanged owing to the standardized interface.

These technological trends definitely enhance both the physical and logical security of smart cards.

- Better integration (about a factor of ten every 5 years) enhances physical security by making it more difficult to physically probe and recover information from the VLSI chips dedicated to smart cards,
- Additions in processing power (central processing unit [CPU], random access memory [RAM]) and in operating systems (read only memory [ROM]) enhance logical security by allowing the implementation of more and more elaborate cryptographic algorithms and protocols in smart cards.

The more our society becomes computerized, the greater are the risks from banking fraud, economic sabotage, industrial spying, etc. The inescapable conclusion is that our computerized open systems require additional security.

Cryptography is a powerful security tool in the field of information technology. However, the expansion of public cryptologic knowledge is moderated by governmental

and political concerns aiming at controlling the spread of cryptologic technology and devices; expressed most often in the form of embargos or export controls.

The smart card, which stores, processes, and controls internal cryptographic algorithms [1,2], as we will see, suggests solutions that may satisfy both national regulations and commercial needs.

2 COMPREHENSIVE APPROACH

The smart card is a portable (or detachable) file system that plays an active role in a transaction, has large possibilities for giving or proving its identity, and incorporates many features ensuring physical and logical security [2].

In many applications [3,4,5,6], the smart card plays an active role in a security system, storing secrets, and providing an easy opportunity to change algorithms without changing the entire system.

2.1 What a Smart Card Is

The chip embedded in a smart card is a single-chip microcomputer (MCU). A MCU is a computer system in miniature integrated onto a single piece of silicon. The only computerlike resources it lacks are the external human or machine interface (I/O) devices such as keyboards, displays, disk drives, etc. The chip embedded in smart cards is in fact a "secure" MCU. But there are major differences between a secure MCU and a general-purpose MCU [7].

In a general-purpose MCU, different operating modes can be selected by the user. For example, during an operation in "expanded mode," the internal data and address buses are connected to the input/output (I/O) pins for accessing memories and resources outside the chip; and during an operation in "write mode," the control of the internal buses is taken over by the outside world for the purpose of modifying the contents of an internal nonvolatile memory; other special modes may be used for testability.

In a secure MCU, after the device has been tested and passed as fully functional by the semiconductor manufacturer, the only possible mode must be the "use-mode," under exclusive control of the user software in the on-board ROM. The internal buses must never be accessible through the I/O pins.

This is the main difference between secure MCUs and general-purpose MCUs: A secure MCU has the built-in capability to prevent, by various means, unauthorized access to the CPU, the memories, the buses, and any data being stored or processed within the device at any time.

Thus, the original title of this section "What a Smart Card Is" might better have been "What a Secure MCU Is."

2.2 What a Smart Card Does

The five basic operations of the smart cards are

1. Input data
2. Output data

3. Read data from nonvolatile memory (NVM)
4. Write or erase data in NVM
5. Compute a cryptographic function

Each of these five operations is related to the logical security of the card, but operations (2) and (4) are particularly sensitive. Operation (2) delivers data and results to the outside world, and operation (4) modifies the content of the NVM.

For example:

- Cryptographic secret keys are to be used by the microcomputer but are not to be output.
- Some data in the nonvolatile memory may give the right to access some resources, therefore precautions must be taken before writing or erasing.
- The result of a cryptographic computation may be a control word delivered to the outside (to descramble television signals, for example); therefore precautions must be taken before computing and outputting it.

The card must be sure that the right card holder is present during some operations, or that the received command has been formulated by the right card issuer. Various security mechanisms and techniques are used by the card for checking these facts, ranging from personal identification numbers (PINs) and message authentication codes (MACs) to sophisticated digital signatures and authentication schemes. These mechanisms are based on both cryptographic and noncryptographic techniques. The card may react upon detecting some types of fraud attempts. For example, the program may be such that three unsuccessive PIN presentations block the card, i.e., inhibit its further use. The increase of memory sizes and computation speeds together with the sophistication of the physical security features make possible the use of a set of more and more elaborate mechanisms for ensuring logical security of smart cards.

As illustrated by the specifications of the French bank cards [8], a well-structured use of these mechanisms and techniques during the basic operations makes it possible to organize several physical zones in the nonvolatile memory.

- An open zone, accessible without any control
- A protected zone, where a password is needed for writing, but where reading is free
- A confidential zone, where a password is needed for reading
- A secret zone, containing PINs, passwords, and cryptographic keys

In more recent masks [9], the nonvolatile memory is organized at the logical level rather than at the physical level. Therefore the zones are far less visible in these masks. The exact physical location of a given file in a NVM is immaterial and hence has no precise meaning in these masks.

3 STANDARDIZATION

Any discussion of standardization requires a good understanding of the International Organization for Standardization. For easy reference a comprehensive overview of the IS0 organization and procedures is given in Appendix A.

In 1980 the French standards institution (Association Française de Normalisation [Afnor]) proposed a new work item (NWI), Interface of Integrated Circuit Cards with Contacts. In October 1981, this NW1 was included by technical committee Information Systems (TC97) in the program of work of the subcommittee Identification Cards (SC17) which then created working group Integrated Circuit Cards with Contacts (WG4). The following participant members are very active: France, the United States, Japan, Germany, the United Kingdom, Canada, Italy, Denmark, the Netherlands, as well as the following liaison members: International Association for Microcircuit Cards (Intamic), Mastercard International, VISA International, Eurocheque, and International Air Transport Association (IATA). Other participant members of SC17 are Australia, Belgium, Czechoslovakia, Norway, South Africa, Sweden, Switzerland, Turkey, and the USSR.

Since 1987, the work in the field of information technology has been carried out through a joint technical committee established by IS0 together with the International Electrotechnical Commission (IEC): ISO/IEC JTCl Information Technology.

The smart card interface is now being standardized as a multipart standard [ISO/IEC 78161 prepared by ISO/IEC JTCl/SC17/WG4. As a result of this work, several parts of the standard are now available and future parts are in progress.

3.1 Standardization of Physical Characteristics

A draft proposal (DP) was registered in 1983. A draft international standard (DIS) was registered in 1985 and approved in 1986. The final international standard (IS) (IS0 7816/l) was published in 1987. The result will probably be merged in a more general standard (IS0 7810).

Smart cards must be pliable; the contacts must be sufficiently conductive; smart cards must also resist mechanical stresses like falls, torsion, and bending and be resistant to static electricity and to exposure to various types of radiation such as x-rays, ultraviolet (UV) light, and electromagnetic fields. These physical characteristics are very precisely specified in the existing standards.

3.2 Standardization of Contact location

In 1981, Afnor proposed a location and an assignment of six operational contacts plus two contacts reserved for future use. These are located on the front of the card, near the upper left corner, as shown in Fig. 2. This location corresponds to the minimum mechanical constraints for the microcircuit when the card is under torsion and bending stresses.

A DP was registered in 1984. A DIS was registered in 1985 and approved in 1986, but five votes were negative: Japan, Germany, United States, United Kingdom,

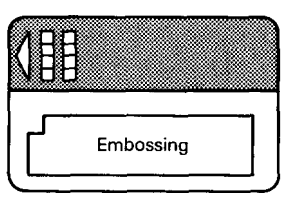

Figure 2 Upper location in front.

and Canada. Finally, an agreement was reached in 1987 and the final (IS0 7816/2) was published in 1988.

The unanimous agreement quickly reached on major points must be stressed:

- Type (surface contacts, and not edge contacts)
- Shape (minimum rectangular surface)
- Pattern (relative positions)
- Electrical functions and contact assignment

The exact location of the contacts was debated at length. While being the one mainly in use, the upper left-hand corner location proposed by Afnor and shown in Fig. 2 is now ***transitional*** in the ISO. The changeover to the new standard will occur sometime in the early 1990s. After that time a lower location has been adopted as shown in Figs. 3-5.

The standard refers to a corner, on any side of the card. Upper and lower locations form a regular pattern shown in Fig. 3. The two locations are deduced from each other by a rotation of the card in the plane. The same microcircuit, which consists of a chip connected to a contact board, may be used in any location. The standard preserves the existing chips. Hence, as long as all the contacts are in front, dual connectors are useful.

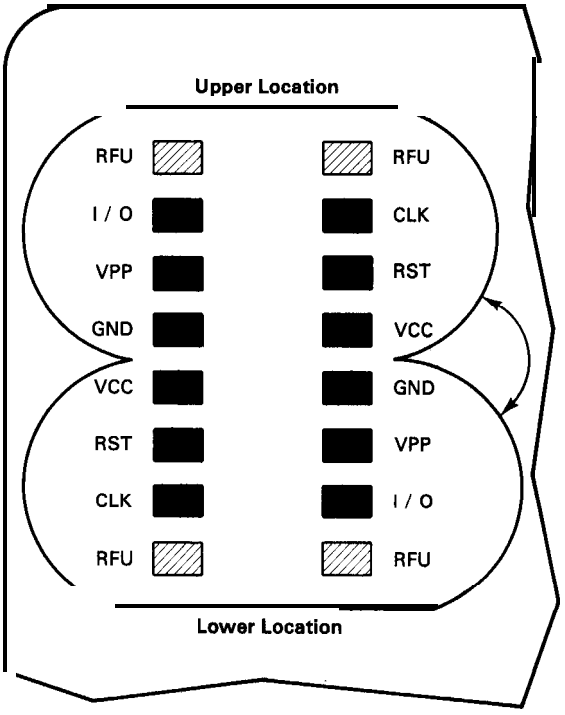

Figure 3 Contact assignment compatibility.

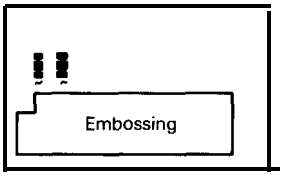

Figure 4 lower location on front.

But there are in fact two final lower locations: in front and on rear, as described in Figs. 4 and 5. The most probable ultimate location is now the lower one on rear, as shown in Fig. 5.

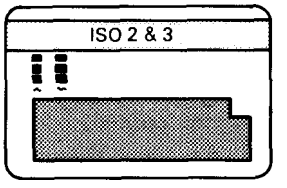

Figure 5 Lower location on back

From the beginning, Japan disagreed with the Afnor proposal because the proposed contact placement conflicted with their placement of magnetic stripes. Their magnetic stripes are currently on the front of the card, as shown in Fig. 6.

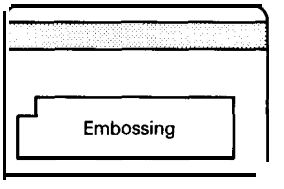

Figure 6 Front of a Japanese card.

Mastercard and Visa also objected to the Afnor proposal, pointing out that U.S. banks reserve the front of the card for identifying features: names, logos, and holograms. The technical aspects should be on the rear of the card: magnetic stripes, signature panel, as well as the electrical contacts of smart cards.

These marketing considerations raised a difficulty in France. Before IS02 and IS03 were standardized, **Transac** invented and designed the magnetic stripes T2 and T3 shown in Fig. 7. The French banks will surrender these stripes as soon as possible. The delay agreed on in 1990 for the transitional location means that these stripes will not exist on cards issued after that transition. The magnetic stripes T2 and T3 will have completely disappeared by the end of 1992.

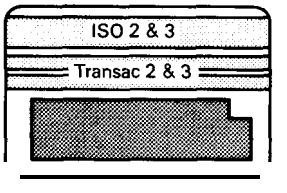

Figure 7 Back of a French card.

3.3 Standardization of Signals and Protocols

In October 1982, Afnor proposed a set of electrical characteristics, a reset procedure followed by an answer-to-reset from the card, and an exchange protocol for processing subsequent commands. Prepared by Task Force WG4/TF1 created in 1984, a first DP was registered in 1985. A first DIS was registered in 1986 and approved in 1988. The final IS (ISO/IEC 7816/3) has been unanimously agreed on and was published in 1989. The basic Afnor proposal has been considerably amended in its presentation, but its technical content is the basis of the ISO. Electrical characteristics of the contacts now include NMOS, CMOS, and HCMOS technologies.

Transactions between the outside world and the embedded microcomputer are conducted through six electrical contacts detailed in Fig. 3. With respect to contact GND (ground) used as reference voltage, signals must be correctly provided to four contacts: RST (reset), VCC (power supply), VPP (programming voltage), and CLK (clock), in order to exchange data in a half-duplex mode on contact I/O (input/output).

Each transaction with a card consists of the following successive steps:

1. Activation of the contacts by the device
2. Resetting of the card by the device
3. Answer-to-reset by the card
4. Optional selection of a protocol type
5. Processing of successive commands according to the scenario of the transaction
6. Deactivation of the contacts by the device

The notion of command has to be carefully explained: Through a command, the outside world instructs the card to carry out some elementary action. Security plays a major part during any transaction with a card. At each command, the card decides either to continue or to stop according to the results of internal computations and according to the internal context of the transaction.

From the physical point of view, the card is a slave and the outside world is the master with control effected through the electrical contacts. But from the logical point of view, the card has autonomy of decision based on its processing power and its operating system.

During answer-to-reset, during subsequent option selection, as well as during processing of commands, data on I/O are organized in asynchronous characters transmitted in half-duplex mode. Each character consists of 10 consecutive bits: a start bit followed by 8 data bits completed by an even parity bit, as shown by Fig. 8. A minimum guardtime must be ensured before the next character to make it possible to resynchronize the receiver between subsequent characters.

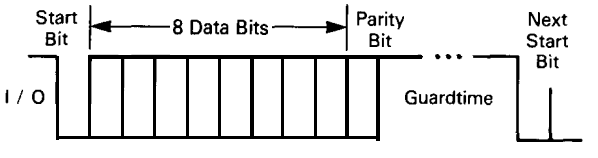

Figure 8 Character frame.

In those industrial fields in which smart cards promise to have the widest application, many struggles occur during standardization. Adopting some suggested modifications could eliminate existing cheap chips and/or create difficulties for existing cards. With strategic rather than technical underlying motivations, the following two struggles are characteristic of such competitions. The reference frequency has been strongly debated: This frequency is provided on CLK by the interface device to exchange data on I/O at a rate of 9600 bps. Some major silicon chip manufacturers, like Motorola, consider 4 MHz to be a lower threshold while Afnor proposed 3.579545 MHz. Basic in NTSC television sets, the frequency is the most used in the world at 10% under 4 MHz. The standard specifies that the bit duration is to be 372 clock cycles during answer-to-reset. This duration corresponds to the frequency proposed by Afnor for 9600 bps on I/O.

Japan and Germany proposed 4.9152 MHz (512 times 9600 Hz, the bit duration being then 512 clock cycles) as being a frequency more consistent with other telecommunications standards. The only argument with some technical basis is the use of standard universal asynchronous receiver transmitters (UART) on the I/O. This argument is not very compelling since, on the one hand, many existing UARTs use other frequencies such as 3.68, 4.02, and even 3.57 MHz. On the other hand, the timers suggested by Japan for accommodating UARTs show that these devices are not well suited for managing data in a half-duplex mode on one unique short line like I/O.

The exchange ***protocol*** has also been strongly debated. The standard specifies the asynchronous character protocol originally resulting from an agreement reached in France in 1981 between Honeywell Bull, Philips, and Schlumberger under the authority of the French PTT administration and with the technical expertise of CCETT. In this character protocol, an error signal is inserted by the receiver in the guardtime of any erroneous character. This error signal **asks** for an immediate repetition of the disputed character. Excluding the practical **use** of classic UARTs but well adapted to a local connection, this protocol is simple, efficient, and inexpensive: a few bytes of buffer and less than 200 bytes of handler. Japan argued, however, that the same end-to-end protocol should manage the exchanges between a host computer and a multiplicity of cards.

Some experts **suggested** that error detection and error recovery should be more sophisticated. In their opinion, vibrations in a car might disturb the exchanges between a card and a radiotelephone set. This argument is easily refuted by pointing out that other contacts like RST may also be influenced by such vibrations and that the consequences of spurious resets cannot be handled by an exchange protocol. This is a problem of connector design.

Japan and Germany have suggested that successive asynchronous characters in the same direction should be organized in blocks of characters with a redundancy checksum of one or two characters. Since 1987, all the experts have been working on a totally new block protocol inserted in 1991 in the standard. In a block protocol, error diagnoses are more complex and so the handler is more expensive: from 600 to 700 bytes depending on the services provided by the protocol. The card must store blocks in both directions: in reception for redundancy checking before processing and in transmission for a possible repetition. The block length in a card is limited by the RAM size. Also a block protocol is not efficient when the blocks are short. The recent publication of the standard has seriously reduced the interest in a block protocol: General use of the character protocol suggests that it is easily adapted to the large spectrum of present applications.

3.4 Additional Standardizations in WG4

A future fourth part of the standard (WD 7816/4) is in preparation. It will ensure inter-industrial interchange. Several points are under consideration.

- File architecture and related security
- Global access method to information
- Consistency in command coding
- Provisions in answer-to-reset for naming chip manufacturers, types, and masks, as well as card manufacturers and card issuers

The first point is essential. And the problems are now considered in the right order. Objects and entities inside the card must be clearly defined and characterized by their security attributes before naming and coding commands addressed to these objects and entities.

The second point is also important. Cards may initialize automatic processes in general-purpose devices. Examples of such processes are:

- Automatic dialing and automatic connection to a remote database
- Security instructions delivered to a terminal for confidentiality and integrity purposes
- Software loading from a card into a terminal

These specifications should accommodate the future evolution of devices, like pay television decoders using smart cards and card acceptor devices connected to electronic directory terminals (Lécam and Minitel in France, [10]).

A further step beyond part 4 should be the standardization of a smart card interpretive language (SCIL) for which interpreters should be implemented in the resident firmware of the terminals. Such a language should facilitate the writing of applications, for example in point-of-sale terminals using different processors.

3.5 Standardizations In IS0 Outside WG4

In 1985, the IS0 Banking technical committee adopted two new work items (NWI): Data Contents of Messages Exchanged with Integrated Circuit Cards, and Security Architecture of Banking Systems Using Integrated Circuit Cards.

In May 1986, this technical committee (TC68) felt the subject important enough for restructuring itself and creating the subcommittee Financial Transaction Cards, Related Media and Operations (SC6) with two working groups (WG5 and WG7) dealing with the two NWIs.

The work (IS0 9992) on Messages Exchanged with Integrated Circuit Cards (TC68/SC6/WG5) includes five parts:

- Concepts and structures
- Functions
- Messages (commands and responses)

- Common data for interchange
- Data elements

The work (IS0 10202) on Security Architecture of Banking Systems Using Integrated Circuit Cards (TC68/SC6/WG7) includes seven parts:

- Card life cycle
- Transaction process
- Cryptographic key relationships
- Secure application modules
- Use of algorithms
- Cardholder verification
- Key management

The first IS0 documents (DIS 9992/l and DIS 10202/l) are now two interim standards awaiting publication.

In October 1988, the joint technical committee (JTCl) adopted a NWI, Interface of Contactless Integrated Circuit Cards. This NW1 was assigned to a new working group. Contactless integrated circuit cards (SC171WG8). The work is just beginning. American Telephone and Telegraph (AT&T) (United States), GEC (United Kingdom), Valvo (Germany), and Dai Nippon (Japan) have all made proposals for contactless integrated circuit card standards. Several other proposals are presently under consideration as well.

3.6 Other Standardizations at the European level

In the European Broadcasting Union (EBU), a working group (V5) is looking for an agreement on a satellite pay television system. Such a system includes a security device, currently named conditional access subsystem (CASS). Two approaches are described: CASSs are either detachable or buried in receivers. Smart cards have been carefully considered as potential detachable CASSs, while secure MCUs are a basis for buried CASSs.

In addition, still at the European level, currently in Commission Européenne des Postes et Telecommunications (CEPT) and in European Telecommunications Standards Institute (ETSI), a working group is drafting the specifications of a subscriber identification module (SIM) for the cellular digital radiotelephone system. These SIMs have turned out to be either smart cards, or plug-in security devices. The same secure MCUs may be used in both cases.

3.7 Standardization of Security Techniques in IS0

The IS0 structures for standardizing cryptographic tools have been highly variable and very sensitive to the political context concerning cryptology and security [11] particularly to the U.S. context. The data encryption standard (DES) was published in 1977 [12] by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST). In 1981, the American National Standards Institute (ANSI)

adopted the DES as a U.S. commercial standard and published it as X3-92. This is a unique situation where a cryptographic algorithm has been submitted to public scrutiny and standardized. In 1980, IS0 technical committee Information Systems (TC97) created a special working group Data Encryption (WGl). The goal was the international standardization of the DES. In 1984, the technical committee replaced this special working group with the subcommittee, Data Cryptographic Techniques (SC20). This subcommittee in turn created three working groups:

- Secret Key Algorithms and Applications (WGl)
- Public Key Systems and Modes of Use (WG2)
- Use of Encipherment Techniques in Communication Architectures (WG3)

By January 1986, the DES algorithm had reached the status of accepted draft international standard, under reference DIS 8227, First Data Encipherment Algorithm DEAl, but in May 1986, the IS0 council decided to not publish the standard although publication had been imminent. The chief argument against publication was that the adoption of the DEAl as a standard might encourage overdependence on the DES which was already an attractive enough target for potential criminal cryptanalysts. The council decided, in fact, to stop standardizing cryptographic algorithms. Several experts tried to define what a cryptographic algorithm is with only partial success. Finally, the subcommittee adopted the principle of a register where both secret and published encipherment algorithms could be introduced and recorded. The algorithm register is now being itself standardized (DIS 9979).

At the same time, in SC20/WG2, Public Key Cryptosystems, the embargo on data encipherment algorithms was extended to include a draft proposal on the Rivest–Shamir-Adleman (RSA) algorithm and to a draft technical report surveying the state of the art in public key cryptography. The contents of this technical report have been disclosed [13]. In WG 2, the work is concentrating on solutions to integrity problems, while no more work is being done on solutions to confidentiality problems.

These decisions were in line with the political situation in the United States. In 1985, commercial as well as governmental cryptography was centralized under responsibility of the National Security Agency. However, in 1987 the situation changed in the United States and the responsibilities for these two areas were separated [14]. The governmental applications were assigned to NSA and the commercial applications to NBS.

In June 1989, the joint technical committee (JTCl) decided to disband SC20 and to create a subcommittee entitled Security Techniques (SC27). The first meeting of SC27 occurred in Stockholm in April 1990. This subcommittee in turn created three working groups:

- Security Services and Guidelines (WGl)
- Security Mechanisms (WG2)
- Evaluation Criteria (WG3)

A fundamental problem remains, however; the border between experts standardizing systems and protocols and experts standardizing security techniques, has still to be clarified in the new organization of JTC 1. Therefore, to coordinate its work on security, the

joint technical committee also created in June 1989 a special working group on security (SWG-S) which met in Rennes, France, in October 1989. This special working group is still in existence.

It appears that a majority of commercial needs can be met simply by protecting the integrity of information. Consequently in IS0 work, priority is given to integrity techniques: identification, authentication, and signature. In open systems, public key cryptosystems make possible standardizable solutions for ensuring integrity. The practical implementations of these schemes more often than not require the use of personal portable security devices, like smart cards.

4 TECHNOLOGY

During the past decade, enormous advancements have occurred in the semiconductor industry: greatly increased performance and memory sizes, and correspondingly great reductions of cost and power consumption. The number of transistors per chip was multiplied by 400 between 1970 and 1985 while the dimensions of a transistor were divided by 2 every four years as shown in Fig. 9. The evolution of random access memories and microprocessors results from advances in a number of areas, such as computer-aided design, photolithography, etching, ion implantation, process mastering, and scanning electron beam microscope. It can be confidently predicted that the next decade will follow the same trend.

In the 1980s, a breakthrough occurred with the development of CMOS technology which consumes much less power, and which also provides this capability at an acceptable cost. No doubt the next step in this development, high-speed CMOS (HCMOS) technology, will be an important part of the integrated circuit market before the end of this century, as indicated in Fig. 10.

Many factors influence the rapidity of the evolution of semiconductor technology; more complex circuits involve more sophisticated constraints and know-how in manufacturing, as well as huge investments and learning curves for mass production. As an

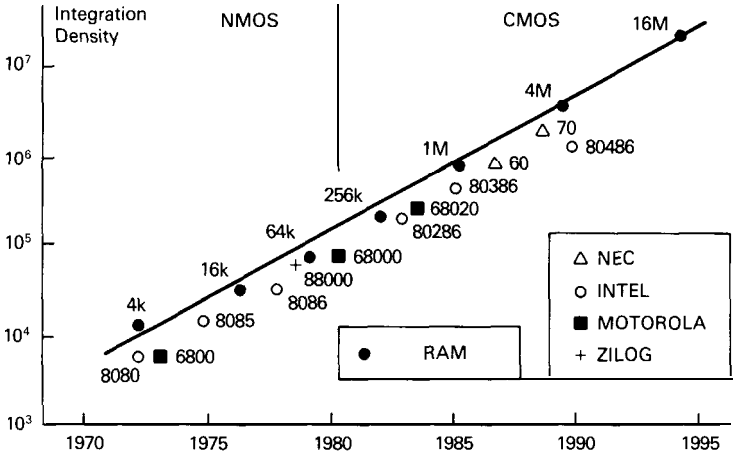

Figure 9 Memories and microprocessors.

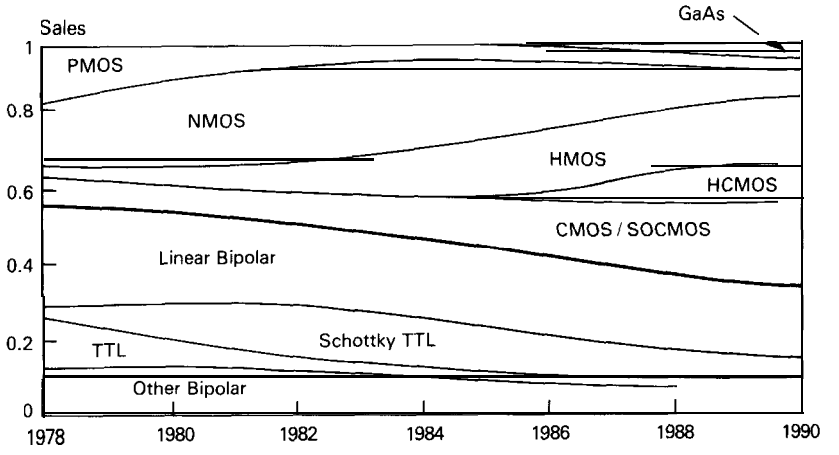

Figure 10 Market share (Bipolar vs. MOS).

example, research and development costs are multiplied by a factor of roughly 3 for each new technology. This explains the large size and the small number of the semiconductor companies that lead the market. As a matter of fact, it is agreed upon that it takes seven to ten years from research laboratory to large-scale production of integrated circuits. This phenomenon is the source of much confusion between future possibilities and the present reality. Figure 11 shows the delay from research phase to mass production of RAMs in MOS technology.

In the development of semiconductor technology, the simple structures were naturally developed before complex very large-scale integration (VLSI) circuits. Consequently, the best integration has been achieved in dynamic RAMs because of small size and low complexity of the elementary cell in the logic circuitry. However, because of cost and reliability considerations, complex chips involve compromises between different technologies for processors and for memories.

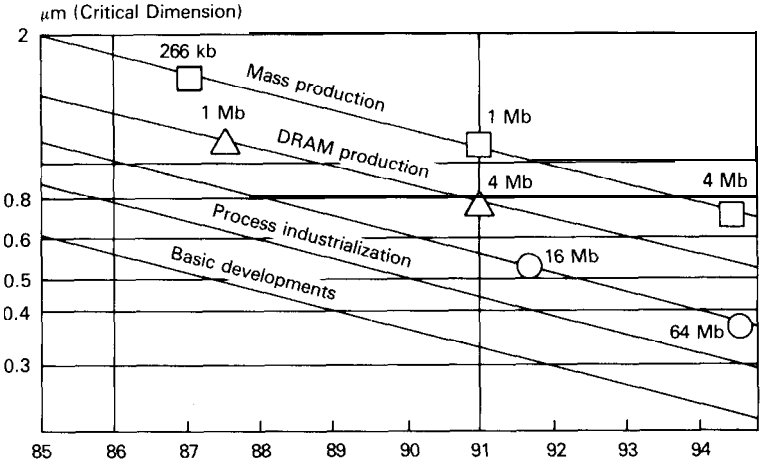

Figure 11 MOS technology maturation.

Because a secure MCU is a complete system, including CPU, RAM, ROM, and nonvolatile programmable memory, one can understand why such VLSI circuits are a relatively recent development. Several functional blocks are gathered on the same substrate, including chip security features as well as all the resources needed by the application. Cost and reliability considerations induce trade-offs limiting ambitious designs and memory sizes. A generally agreed-upon limitation is that the chip size should not exceed 20 mm^2 if one is to obtain a reliable card at a reasonable price. While this is achievable with today's VLSI technology, smart card chips will undoubtably follow the same general trends as the rest of the semiconductor industry to achieve greater density and functions in the future.

4.1 Nonvolatile Programmable Memories

In smart cards, the built-in electronics include a nonvolatile programmable memory (NVM). Each cell of NVM is originally in logical state ONE; it may be turned to logical state ZERO by an electrical process under control of the built-in electronics itself. Data stored in the NVM vary from one card to another and changes during card life. Any NVM area may always be selectively erased by turning to ZERO all the bits in the area, but the possibility of returning back to original state ONE depends on the NVM technology used. At the beginning of the smart card story, 10 years ago, two technologies were considered for implementing the NVM: bipolar and MOS. Although it is quicker, bipolar technology is much more power consuming. It also uses more silicon to realize the same function. An elementary bipolar transistor is illustrated in Fig. 12. More importantly, the bipolar writing process physically destroys a metallic fuse in each NVM cell so that a cell once written into cannot be erased. Broken and intact fuses are visible in Fig. 13. Because the status of a fusable link can be read in a

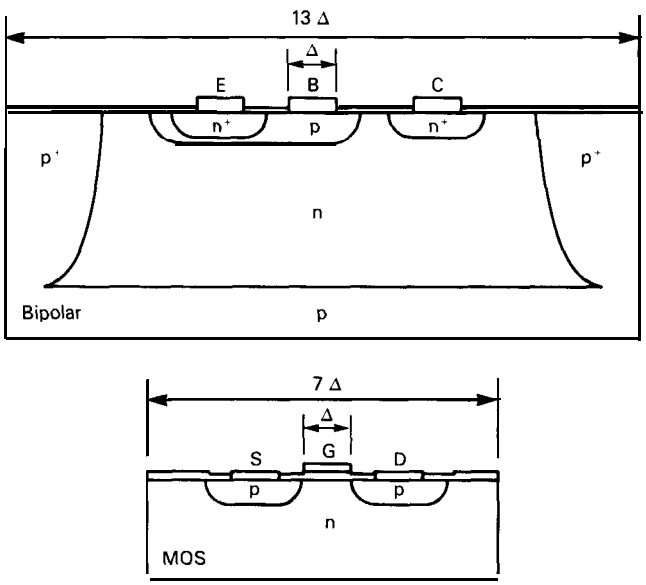

Figure 12 Comparison of transistor layouts.

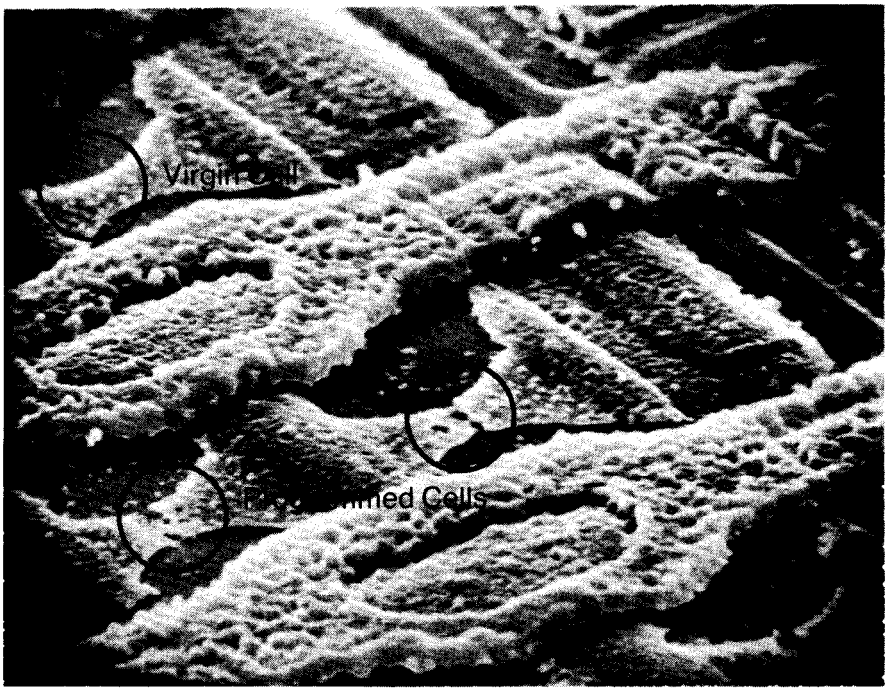

Figure 13 Fuses in a bipolar WM.

magnified image, a bipolar NVM cannot hide secrets from microscopic examination. This technology is therefore not appropriate for a secure smart card storing cryptographic keys and algorithms. Nevertheless, at the onset of smart cards, bipolar technology was considered: the reasons in favor of this technology were writing irreversibility and ability to support a logic array of gates on the same substrate as the memory.

As illustrated in Fig. 14, a MOS NVM does not have the same security problem as did the bipolar NVM: Since writing is reversible, a cell content cannot be read optically, but can only be determined by electrically accessing internal buses. Moreover, for large volumes and low costs, MOS technology is better suited than bipolar technology to the integration of a NVM together with a microprocessor on the same substrate.

In MOS technology, the NVM may be built either in electrically programmable read-only memory (EPROM) or in electrically erasable PROM (EEPROM). If in EPROM technology, the return from logical state ZERO to logical state ONE is not selective: Erasing radiations affects the whole NVM contents. But in smart cards, for obvious security reasons, such a global process would kill the card. As illustrated in Fig. 15, an EPROM cell uses the floating gate avalanche MOS (FAMOS) technology. To write a bit, high voltage is applied to the control gate and to the drain in order to cause high-energy electrons to avalanche through the drain junction. Quantum theory says there is a small probability for an electron to jump across a potential barrier higher than the energy of the electron, a process referred to as tunneling since the electron's energy is inadequate for it to cross the barrier. Therefore some of the electrons may tunnel through the insulating layer of silicon dioxide and be trapped inside the floating gate. This causes an increase in the threshold voltage yielding an off-state transistor.

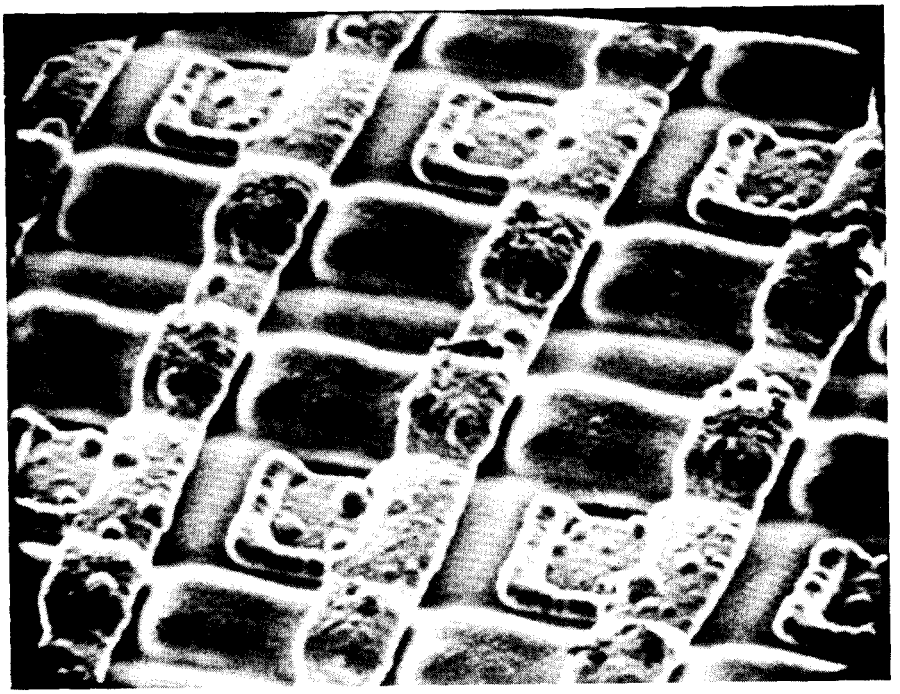

Figure 14 Invisible contents of a MOS NVM.

Erasing is done by ionizing the insulated layer with UV light during 20 min. to recover the virgin state by discharging the floating gate.

The programming voltage and the writing time are consequences of the thickness of the oxide which reflects the accuracy of the technology. At the early stage of EPROM technology, 25 V were needed. Subsequently this voltage was decreased to 21 V, and currently only 15-12 V are needed. During the same period, the writing time decreased from 50 ms to less than 10 ms. The writing time in EPROM is now of the same order of magnitude as the writing time in EEPROM. Despite this trend toward lower programming voltages, the energy needed for writing in EPROMs at the moment precludes the incorporation of a high-voltage generator on the chip as is common practice in EEPROM technology. In EEPROM Technology, the binary information stored in

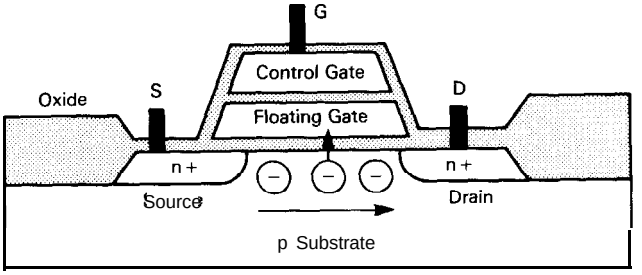

Figure 15 EPROM technology.

each cell (or block of cells) of NVM may be selectively inversed by an electrical process. The relevance of this technology to smart cards is that erasable reusable cards are very desirable.

An EEPROM cell uses the reversible Fowler-Hordheim effect to extract electrons by tunneling through a very thin oxide layer. This effect requires a high electric field, greater than 10^7 V/m, able to extract electrons from a doped semiconductor. As shown in Fig. 16, the structure of the most widespread EEPROM in use today is similar to the FAMOS structure. A cell includes a very thin layer of silicon dioxide under the polysilicon floating gate. The thickness of this insulating layer is difficult to master in mass production. This is the reason why this technology, which was born ten years ago, is only emerging now.

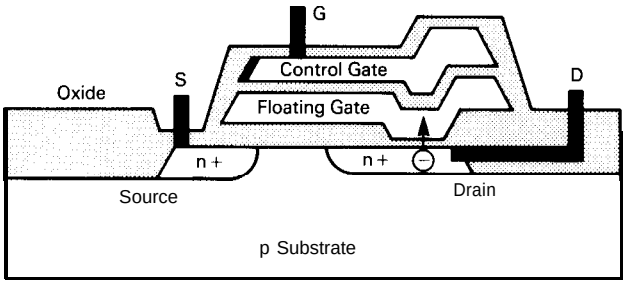

Figure 16 EEPROM technology.

To erase a cell means to force electrons from the floating gate and selectively discharge the gate. This is achieved by applying the programming voltage on the drain and keeping the gate at 0 V, thus allowing electrons to tunnel back to the drain. The write-erase cycles progressively destroy the thin layer of dioxide through which the electrons must tunnel: Every time, a few electrons are trapped within defects of the silicon dioxide. The number of cycles is limited to around 10^4. This phenomenon has to be carefully considered according to the smart card application requirements.

The programming pulse is 18-20 V with a duration of l-10 ms. The small amount of energy required allows use of a voltage converter on the chip itself. EEPROM cards do not use contact VPR consequently smart card terminals accepting exclusively such cards are much simpler.

If a smart card generates the programming voltage internally, then stored information may be modified without any control by the outside world, especially during power-on and power-off sequences. In such a card the chip designer must take this possibility into account to avoid undesirable perturbations in the NVM.

EEPROM technology requires two transistors for each cell of NVM whereas EPROM technology requires only one. This explains the 2:4 EPROM/EEPROM ratio in memory size for the same level of integration. Simpler than EEPROM technology, EPROM technology is also the most advanced in the semiconductor industry; the EPROM manufacturing process is improved permanently by the feedback of an important mass production. No doubt, EEPROM technology will follow the same trends in the near future.

4.2 Smart Cards in the Integrated Circuit Card Family

The NVM contents evolve during card life under control of the built-in electronics. Depending on an increasing complexity, three types of integrated circuit cards are illustrated in Fig. 17.

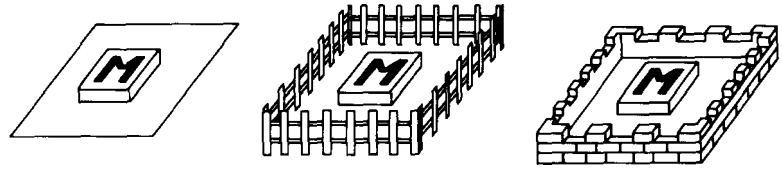

Figure 17 Integrated circuit card family.

The same system may support different types of cards. This is illustrated by European public telephones which accept all three types of cards. The number of cards in the following paragraphs were valid in 1990, but are increasing rapidly. For example, in France more than 5 million télécartes are presently manufactured each month; mid 1991.

In France, sixty million anonymous memory cards, named *télécartes*, have been produced for the seventy thousand French public phones in use. These public phones also accept the five million banking smart cards in circulation as well as the one million personal smart cards, named Cartes ***Pastel,*** delivered to phone subscribers by France-Telecom.

In Germany, more than three million anonymous memory cards, named *telekards*, have been produced for use with German public phones. The Bundespost is now manufacturing smart cards for its public telephone system.

The simplest cards are those specific to a single application. Economic considerations, though, dictate against making each card be totally unique. Instead, the manufacturing process capitalizes on the fact that smart cards are inherently multipurpose devices. The microcomputers are programmed by masks during the manufacturing process. Sharing chip production among several masks to satisfy different applications is easy; designing a new mask is not too complicated, and the same line produces smart card chips, irrespective of which mask is used. However sharing a very simple memory chip for several different applications can cause severe security problems.

The smart card operating system deals with different commands and with the general security of the whole system. Since chip design is not recurrent, software development cost may well exceed hardware cost. Similar to personal computers, the most important part of a smart card system lies in the software, as illustrated in Fig. 18. A poor software design can induce weak security, inefficient functions, erroneous data, deadlocks, and many other potential problems. On the other hand, a good software design provides the user with qualified operations and additional functions.

As a matter of fact, the user is not buying a hardware chip, but rather a complete smart card product providing functional solutions to his problems. The efficiency of a smart card operating system is not only related to ROM size, but also to the virtuosity of the software designer who finally specifies the technical configuration of the card.

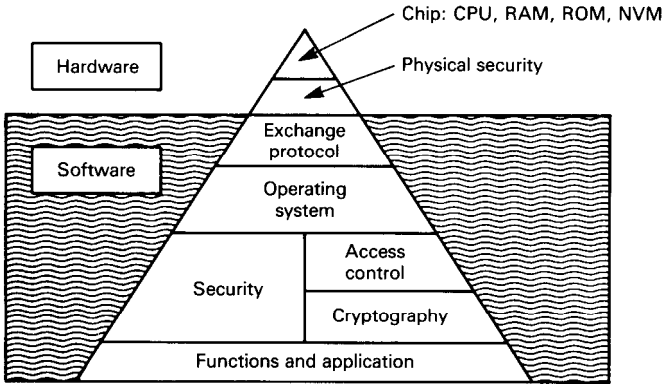

Figure 18 Smart card environment application.

4.3 Self-Programmable One-Chip Microcomputer

The first smart cards were produced in March 1979 as the result of a successful collaboration between CII Honeywell Bull and Motorola. These smart cards included two chips: a 2716 EPROM memory and a 3870 microprocessor originally designed by Fairchild. This dual-chip stage was essential to prove the feasibility of the concept and to convince potential users to start experiments. These dual-chip cards also played an important role in the initialization of applications and in the development of various other elements in systems using smart cards.

Despite the fact that it is always possible to assemble several standard components in a plastic card, the natural solution is a one-chip one because of cost, security, and reliability of the final product. Indeed, the microcircuit manufacturing is simplified; the risk of failure is seriously reduced; and there are no wires from one chip to another that might allow access to internal buses for an attacker to exploit. Thus, the security is taken into account in the design of the chip.

A chip dedicated to smart cards must be able to execute an internal routine for writing to itself in its NVM. Such a chip is termed a self-programmable one-chip microcomputer (SPOM) .

Figure 19 describes the architecture invented by Honeywell Bull for managing registers on internal buses in such a way that the processor remains in control while holding the right address and the right content on the ports directed to the NVM.

The cooperation between Motorola and Honeywell Bull continued with the development of a SPOM. The first silicon SPOM was operational in 1981. Since 1982, Motorola has produced more than twenty million SPOMs in East Kilbride, Scotland. Since 1985, Thomson has also been producing SPOMs in Le Rousset, France. In all these SPOMs, successful trade-offs between cost and performance are a result of the know-how gathered from the first dual-chip cards. The Bull CP8 and Philips are currently manufacturing cards with these SPOMs which are about 18 mm^2 in size.

Recently, Honeywell Bull and Philips, the two major companies involved in smart card development from the beginning, decided to join their efforts and know-how. They created a common development team with a goal of designing and implementing a high-security, multiapplication card operating system called TBlOO [9].

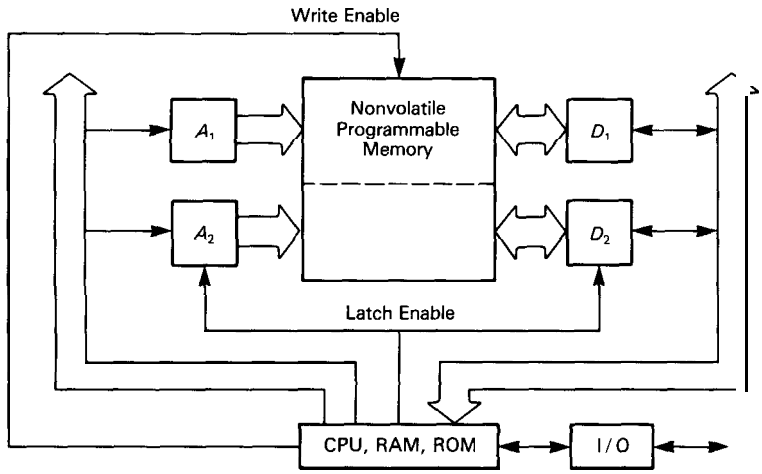

Figure 19 SPOM architecture.

The choice of the central processing unit (CPU) is an important decision in the design of such a chip. Currently Motorola is proposing a family of SPOMs based on a classic 8-bit CPU: the 6805. From the beginning, Motorola has used the same CPU. Now with a new SPOM family named ST16XYZ, Thomson is also moving toward the same CPU: the 6805. The consequences of these industrial decisions are important.

One advantage of choosing a classic CPU is the availability of very complete development tools which simplify software production. Another advantage is hardware evolution inside a large family of microprocessors. Mass production makes these step-ups easier for the transition from NMOS to HCMOS. In addition to NVM, a SPOM also includes two other types of memory: RAM and ROM. The RAM stores contexts and intermediate results during computations. The ROM stores the smart card operating system written by mask during chip manufacturing process at the factory. Memory cells differ not only in their function, but also in the amount of silicon "real estate" required for their realization. On the SPOM illustrated in Fig. 20, a cell of RAM is roughly 20 times larger than a cell of EPROM which in turn is roughly three times larger than a cell of ROM.

The four spare contacts on the left of the SPOM in Fig. 20 are additional I/O contacts available for connecting other devices inside the card and for using the chip in different environments.

Today, high-speed CMOS (HCMOS) technology is in production. In either HCMOS or CMOS technology, a switch consists of a pair of complementary NMOS and PMOS transistors. Such a switch is shown in Fig. 21. The gates of the two transistors are wired together and the input voltage is applied to both of them. The responses are complementary: A signal activating one transistor deactivates the other one and vice versa. The drain of the NMOS transistor is wired to the source of the PMOS transistor: Together they deliver the output signal. The source of the NMOS transistor is connected to a low-voltage line; and the drain of the PMOS transistor is connected to a high-voltage line.

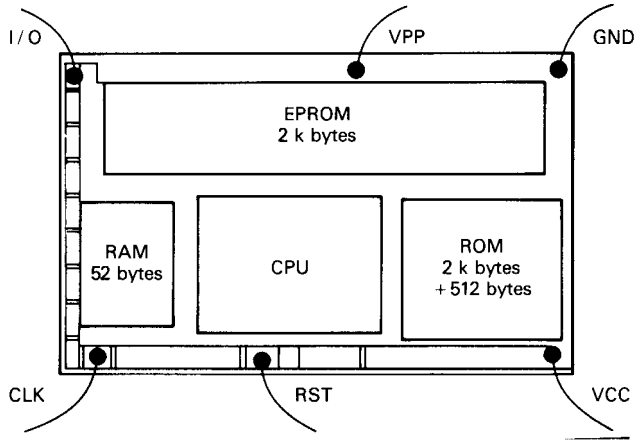

Figure 20 SPOM die (MC6805SC03).

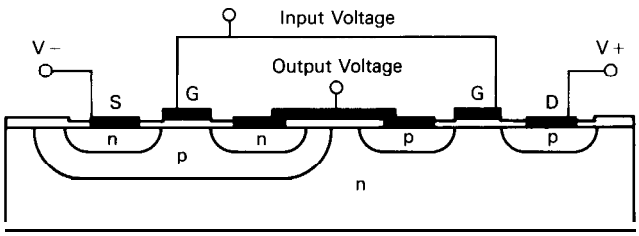

Figure 21 CMOS switch.

The two most important features of CMOS technology are the following ones [15]:

(i) CMOS devices consume less power than previous NMOS devices: No current passes between the two lines except during the short periods when the input signal is switched.

(ii) CMOS devices are also less susceptible to ambient electrical noise than NMOS devices: A spurious signal would have to be twice as great to force a CMOS device into an error setting as would be required to cause an error setting in an NMOS device.

With these new HCMOS designs, the range of SPOMs becomes broader. Memory sizes of the existing SPOMs are summarized in Table 1.

In the near future, some new SPOMs will include arithmetic operators running in parallel with the main processors. These operators are designed for multiplying and exponentiating large integers modulo large integers. Two such projects were publicly described in 1989 [16,171. New SPOMs of this type are presently under development by Philips, Honeywell Bull, and Siemens. Those types of SPOMs will be well adapted to processing public key algorithms and zero-knowledge schemes.

TABLE 1 SUMMARY OF EXISTING SPOMS (MEMORY SIZES IN BYTES)

SPOM Type	NVM		ROM	RAM
	EPROM	EEPROM		
Motorola				
68HC05SC01	1K		1.6K	36
68HC05SC03	2K		2K	52
68HC05SC11	8K		6K	128
68HC05SC21		3K	6K	128
68HC05SC23		512	3K	96
68HC05SC24		1K	3K	128
SGS-Thomson				
ST1002	1K		2K	44
ST1834	4K		3K	76
ST16402		2K	4K	256
ST16612		2K	6K	160
S9	8K		4K	256
Hitachi				
65901		2K	3K	128
6483108		8K	10K	256
Oki				
62720		2K	3K	128
62780		8K	6K	192

5 SECURITY

Absolute security does not exist, no more for the smart card than for any other computing device. However, security may be enhanced by a coherent set of physical and logical features. Several secret key algorithms are currently used in the numerous masks of existing smart cards. A reasonable question is: Why are there so many masks and so many algorithms? The answer is in part due to the widely differing card capabilities, and in part due to the variety of crypto algorithms available. There are two main families of masks: key-carrier cards KCO, KCl, KC2, and multipurpose cards M4, M9, MP, M64, B1, B2, Dl, D2, and TB100. There are also evolutionary stages in the algorithms: from the noninversible one-way function Telepass1 and the semireversible function TDF, up to the fully reversible functions Telepass2, Videopass, and DES.

We shall not describe in detail either the masks or the algorithms (only DES [12] has been made public). Our description is restricted to the functional evolution of masks and algorithms so as to give an overview of secret key cryptology in smart cards.

As an introduction to security, we will first describe some physical features of the chip itself and next some aspects of cardholder identification.

5.1 Chip Security Features and Card life Cycle

There is a hidden problem inherent to smart cards. The problem arises because a card must be considered in the course of its existence to have three phases: a birth, a life, and a death. This is unique in the data processing world, and one must resist the

temptation to consider the NVM as simply a conventional database without at the same time considering the creation process of smart cards and the associated rights it represents. This confusion comes from the broad use of mass memories such as disks, which have to be formatted by the user and then loaded with information. It seems relatively easy to securely handle data if only one application is able to be run in a card. If several applications may run in the same card, there is indeed a risk that other people may access or tamper with data belonging to a user.

In this section, we consider some basic security features in chip production and chip life. The file architecture of the NVM in a smart card is developed in Section 5.10.

Since the design of the very first chips for smart cards, two approaches have been considered to the problem of chip testing. These are:

(1) Each chip supports about twenty additional test contacts, and tests are conducted under control of the outside world; or

(2) each chip supports one or two test contacts, and tests are conducted by an internal self-testing program written in a small extra ROM (about a half-kilobyte).

During the development of the manufacturing process, both efficiency and flexibility of testing require the 20 additional test contacts. More information can be gained through measuring internal electrical signals in this way. Self-testing, though, is more economical. Therefore the number of test contacts is reduced when the manufacturing process is mature enough.

Before cutting wafers on SPOM03 production lines, a 512-byte internal routine is activated through two specific test contacts, visible in Fig. 20 near RST, under the CPU. The NVM of each validated component receives various information: locks, codes, erasure indicators, chip serial number, while nothing is written in rejected components. The two test contacts are then systematically destroyed by breaking fuse links buried in the silicon. An equivalent operation exists for any secure SPOM.

This operation, which eliminates non-user modes on valid chips, also positively disables invalid chips where nothing has been written.

As a matter of fact, only the self-testing routine may write these erasure indicators to be tested by the card before executing any command in user mode during any transaction. If such an erasure indicator is erased, either by accident or by violation, then the chip is definitely disabled. Such NVM cells are constructed so as to be the most sensible ones to erasing radiations. This is an example of the current reliability philosophy of using weak-link/strong-link designs to enhance reliability, since the weak-link is designed to disable the device before the operational strong-links can be subverted.

Valid chips are then inserted into cards during the process of card manufacture. A manufacturing code or key is used for protecting chips from the time of chip manufacturing to card issuing. Throughout the operational card life, several testers in the chip determine readiness: voltages, clock frequency, light, temperature are all measured. These indications may also be used by the operating system to increase security. The mapping of memory addresses should be controlled by the internal program itself, and not be accessible to outside control.

Whatever the physical security systems, system designers must carefully consider the potential consequences of chip violations. Secret keys must be as diversified as

possible, tied to user identification number and/or chip serial number. A successful violation then compromises only one user and does not endanger the whole system; thus reducing the risk of widespread fraud. These aspects of logical security are strongly related to cryptology.

5.2 Cardholder Identification

The identification of the cardholder has several aspects depending on the scenario of the transaction to be performed with the card.

In this section we will focus on the identification of the cardholder by the card, considering that the card itself is authenticated by other means. The problem of card authentication is developed in a subsequent chapter. We restrict our consideration to the simplest case where a cardholder is paying a retailer with a smart credit card, and where the smart card has to identify the person attempting to use it to be the authorized cardholder according to the security policy of the payment.

There are several ways for a card to identify the cardholder. The simplest one is to carry out a direct personal identification number (PIN) check inside the card. When a PIN is required, no operation can take place in the card without the presentation of the correct information. The card internally compares the PIN presented by the user with the reference PIN written in a secret area of its NVM. The card keeps the result of this comparison secret until after the results can be entered into its memory. If the result is incorrect, then this fact must be recorded in the NVM to total the number of successive erroneous attempts to use the card. When this number reaches a predetermined value (l-7, depending on the security policy), the card is blocked and cannot be used thereafter. If the result is correct, then the external behavior must be the same as above in order to not reveal the test result before it has been recorded.

By systematically recording the result, the card prevents a fraudulent (unauthorized) user from deriving any benefit from observing a difference in the card's actions, no matter how slight.

At least I bit must be written in the NVM whenever an access protected by a PIN is made.

In some masks such as MP and TBlOO, the PIN may be enciphered to foil attempts to eavesdrop on this confidential information. A security module is located in the **pinpad** of the point-of-sale terminal. The card produces a random number. The security module then computes a message from the random number and the identity claimed by the card using an internal master key. The card tests the message and reacts as above for memorizing the result. This method also provides an authentication of the terminal by the card.

In the same manner, an MP card is able to cooperate with an external biometric identification device which increases the authentication abilities in a system. Identification may be performed by fingerprint, retina pattern, dynamic signature, or any physical characteristic of the individual. For this purpose, the card must deliver a reference pattern to the external checking device and the dialogue between the card and the checking device has to be randomized and encrypted.

To close these considerations of cardholder identification, we give an example that costs only NVM memory and requires no processing power in the card. The identification of the cardholder by the retailer may also involve a device displaying digital pictures. For this purpose, the card must store a compressed digital photograph which

has been signed by the authority and tied to the chip serial number of the card. After having authenticated the card by other means, the retailer gets the signed reference picture, checks the associated signing appendix, and checks the picture visually to authenticate the cardholder.

5.3 Secret Key One-Way Function

Algorithm Telepassl was designed by Honeywell Bull in 1979. This unpublished algorithm is a one-way function coded by about 200 bytes in masks M4 and M9.

Mask M4 is a general-purpose mask, not dedicated to a particular application. The corresponding cards may be turned into banking cards under personalization BO, and into Pastel cards for public phone subscribers under personalization B03. Each M4 card holds only one cryptographic key. This unique secret key is generally computed by diversifying a secret master key by the chip serial number; such a computation is performed in security devices protecting the secret master key.

Telepassl computes a result ***R*** (64 bits) from four variables: an external argument ***E*** (48 bits), an address @ (16 bits) of any nonsecret word in the card, the content of this word (32 bits), and the unique internal secret key S (three words of 32 bits in BO) (Fig. 22). The access to a nonsecret word is either free or conditioned by the previous presentation of the personal identification number. Before involving a confidential word in a computation, the PIN must have been correctly presented.

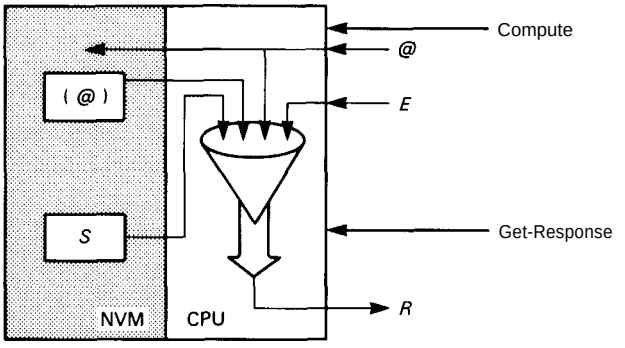

Figure 22 Telepassl algorithm.

Two types of security devices are currently involved in systems using M4 and M9 cards.

Mother cards are used for personalizing individual cards by computing one diversified key for each issued card. Mother cards transfer the diversified keys to the outside world.

Security modules internally recompute diversified keys for controlling results computed by cards. Security modules do not output such diversified keys which are systematically used in subsequent internal computations.

5.4 Dynamic Authentication by Security Modules

The Telepassl algorithm allows a dynamic authentication of cards by security modules, as illustrated in Fig. 23. At each authentication, the module picks at random a string of

48 bits and transmits it to the card. The security module checks that the response from the card corresponds to the internal result obtained by reconstructing the diversified key depending on the chip serial number of the card and then computing the response depending on the random challenge sent to the card. This result is internally compared with the response received from the card. Only 1 bit (yes or no) is transmitted to the outside world.

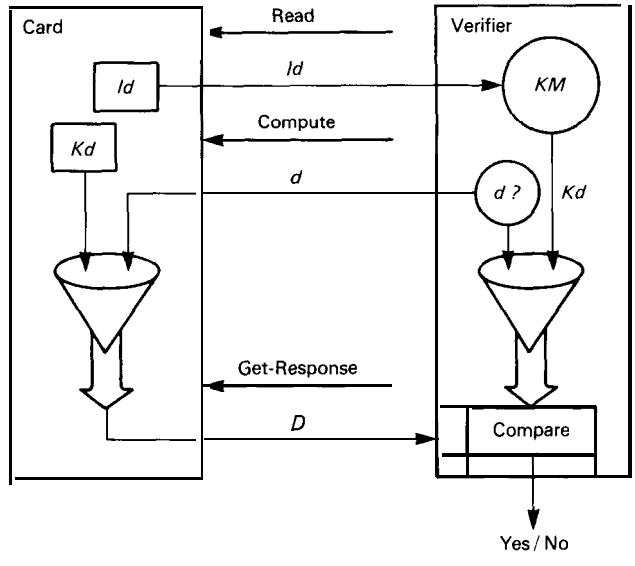

Figure 23 Authentication by a security module.

When the content of a confidential word in a card is involved, an authentication becomes an identification: The security module verifies that the right card has been activated with the right PIN. A similar authentication scheme may be implemented with any secret key algorithm.

Despite using a one-way function and only a single key per card, M4 cards provide several functions dealing with both confidentiality and integrity in both on-line and off-line operations. For confidentiality, Telepassl allows the management of secret keys between a central mother card and a set of distributed remote M4 cards. These secret keys may be used to protect both data and programs. For integrity, Telepass1 allows one to verify the content of a word in a card. This function is used in access control systems: The relevant word represents an access right or an authorization. This function is also used in management and payment systems for controlling the result of a write command. For example, certificates are stored by retailers during payment operations: Such certificates may be used later for resolving disputes.

A pseudosignature is obtained by appending to a message a certificate involving the content of a confidential word in the card and the hashing of a message. The reduced length of the variable (48 bits) may be compensated for by using the algorithm twice on a twice-repeated hashing to give twice as many hashed bits.

The violation of a card does not endanger the whole system. But the violation of a security module has major consequences: Any other card issued by the same authority may then be subverted.

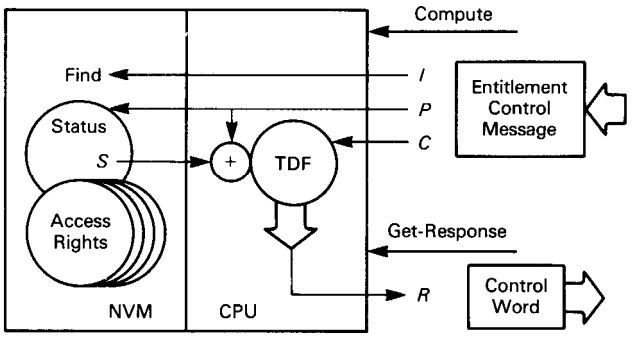

Figure 24 Control operation by TDF algorithm.

5.5 Semi-Inversible Secret Key Algorithm

The pair of unpublished algorithms TDF (for twisted double field) was designed by CCETT in 1980. The user algorithm is executed in KC0 cards. It is the left-inverse of the mother algorithm executed in security devices. KC0 cards are currently called **key-**carrier cards. The user algorithm is coded in about 300 bytes on mask. KC0 which was designed for controlled access to broadcast teletext Antiope [18,19]. At present, broadcast information on stock exchange rates is sold in France, where access is controlled on a monthly subscription basis using KC0 cards.

In each KC0 card, a hierarchy appears between a unique issuing key and up to 32 service keys. The unique issuing key is computed by diversifying a secret master key, as in the M4 card. The card issuer then uses this key to introduce new service keys into the cards it issues and for managing the status of existing service keys in the cards. The service keys are not diversified in a controlled access broadcast application. Each service key in a card is associated with a status limiting its use. Such a status is a set of conditions, such as periods for authorizations based on subscription, or credit amounts for authorizations based on a pay-per-view scheme. A service key and its status represent an authorization or an access right. In a broadcast environment, a service key and its status is generally referred to as an entitlement.

The user algorithm of TDF in KC0 cards computes a result ***R*** (61 bits) from three variables: an external cryptogram C (127 bits), an external parameter ***P*** (23 bits), and an internal secret key S (127 bits) selected by its name (3 bytes). A control operation is shown in Fig. 24. In the pay television technology, the three external variables (name, parameter, cryptogram) are generally referred to as a message (14 bytes), and more specifically, as an entitlement control message (ECM). The result is referred to as a control word (CW).

5.6 Invertible Secret Key Algorithms

The know-how gained from Telepass l and TDF is gathered in two unpublished algorithms named Telepass2 and Videopass, designed in 1984 by Bull CP8. Telepass2 is used in masks Bl and B2, which are the property of the French banks. Videopass, as well as Telepass2, computes a result ***R*** (64 bits) from three variables: an external cryptogram C (64 bits), a parameter ***P*** (32 bits), **and** an internal secret key S (128 bits)

selected by its name (3 bytes). The parameter is either an external variable provided to the card or an internal nonsecret word in the card. These two algorithms are invertible: We speak of a "user algorithm" in the one direction and of a "mother algorithm" in the opposite direction.

Mask Bl provides only user cards because it performs only the user direction of the algorithm. However, both directions are programmed in masks B2 and KC1 on about 250 bytes. In B2 and KC1 user cards, a lock restricts the algorithm to the user direction. This user lock is written during card personalization. A mother card may still execute the algorithm in both directions because the user lock is not written in its NVM.

Each user card holds a unique issuing key and several service keys. The hierarchy that appeared in mask KC0 has been significantly improved in KC1 . These masks, B1, B2, and KC1 , include all the functions previously developed for M4, M9, and KCO. In addition, two new techniques were introduced: cryptowriting and dissymetrization. Described below, these two techniques are based on a practical use of redundancy.

Algorithm DES was programmed in 1986 by Philips in mask Dl on less than 700 bytes. Before that practical proof by realization, it was thought that algorithm DES was too expensive in RAM and ROM for realistic implementations in smart cards. But the limited resources in the SPOMs have put pressure on cryptologist programmers to use this limited memory space very efficiently.

Algorithm DES is present in several multipurpose masks which separate keys for confidentiality and keys for integrity: Dl, D2 by Philips, M64 by Schlumberger, MP by Bull CP8, and TBlOO by a cooperation between Bull CP8 and Philips. In these masks, DES may be replaced very easily by any algorithm requiring resources equal to or less than those required by the DES. These five masks also use cryptowriting and dissymmetrization, discussed below.

5.7 Cryptowriting and Dissymmetrization

The mechanism called cryptowriting, or Secure writing, is a generalization of the subscription management mechanism invented in KC0 cards. After a computation involving its secret issuing key, the card checks the redundancy of the resulting 64 bits: if it is correct, then a secret word of 32 bits is recovered and written in the NVM.

The mechanism called ***dissymmetrization*** (literally-enforcing dissymmetry or asymmetry) was introduced in the masks B1, B2, and KC 1. A mother card must store two different copies of the same key to compute the algorithm in both directions with this key: In one storage, the key bytes are written in the opposite direction of the other storage. A key may thus be stored either in the user direction or in the mother direction. This mechanism is very efficient for controlling the mother cards and the user cards in the system.

Redundancy is used in both directions, either inserted by mother cards toward user cards for managing rights, or inserted by user cards toward mother cards for certification purposes. Even if the results are transmitted to the outside world, the insertion of redundancy prevents a mother card which has the key stored in the mother direction from simulating a user card in which the key is stored in the user direction.

This asymetric property is reminiscent of public keys. This is not surprising since complexity of computations are the basis for security in both cases: factoring large integers on the one hand and investigating NVM contents on the other hand.

5.8 Conditional Access to Audiovisual Services

The principles developed during the KC0 study [18,19], are the basis for standardizing a pay television system by the European Broadcasting Union (EBU) [20]. The conditional access system is only one element in a strategy leading to new European television standards which are the basis for generalizing direct broadcasting satellites and later, high-definition television (HDTV) pictures.

In the conditional access system illustrated in Fig. 25, all the receivers receive the same signal consisting of scrambled components and access control parameters.

The service components are scrambled before broadcast. The scrambling method depends on the component and its coding. Each scrambling operation is controlled by a control word (CW) typically randomly modified every 10 seconds. The cryptograms of the control words are multiplexed in the broadcast signal. Control word updating is anticipated by sending these cryptograms slightly in advance of when they become effective. The signal also transmits a clear synchronization for the descrambling process. The conditional access to service components is thus reduced to the conditional access to transient control words.

Access cards implement access rights, also called entitlements. The European Broadcasting Union (UER/EBU) has adopted a vocabulary [20] describing the various entities illustrated in Fig. 25.

The entitlement management messages (EMMs) are produced by management centers under the authority of a card issuer. Each EMM consists of a card number, a management parameter, a service name, and a management cryptogram.

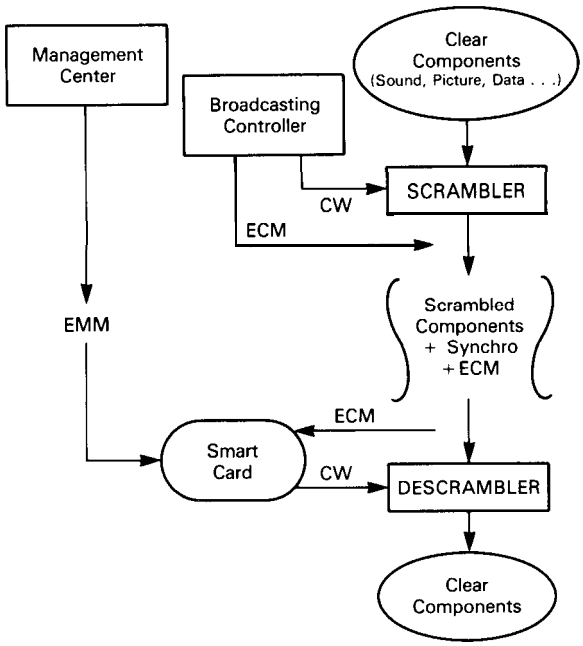

Figure **25 Conditional access system.**

The entitlement control messages (ECM) are produced by broadcast controllers under the authority of a service broadcaster. Each ECM consists of a service name, a control parameter, and the cryptogram of a control word.

The key hierarchy in KC0 cards results in two types of security devices for producing cryptograms using the mother algorithm of TDF.

- The management security devices are used by the card issuer for managing keys and rights.
- The control security devices are used by the service broadcaster for controlling access rights.

As used by a management center, management security devices first reconstruct diversified issuing keys from the secret master key and, second, compute the mother algorithm of TDF under control of a reconstructed diversified issuing key. With such a management security device, the card issuer computes personalized cryptograms where redundancy plays the main part. In a KC0 card, the computation involves the issuing key and the result is accepted as correct if the same bit string is generated twice. Such a result is not transmitted to the outside world. If the result is correct, then the card executes the command according to information given partly in the management parameter and partly in the resulting bit string. If the result is incorrect, then the card stops and waits for a reset by the interface. Owing to such secure mechanisms, the EMMs securely initiate special actions in KC0 cards, such as the canceling of a subscription.

Each card may thus securely authenticate its issuer, without any assumption regarding network and terminal security. Such personalized cryptograms may be transmitted on any network: mail, data network, telephone, television, etc. One speaks of over-the-air addressing when these management data are multiplexed in the television channel itself.

Owing to TDF semireversibility, access rights are remotely and securely managed in KC0 cards. As used by broadcast controllers, control security devices do not diversify keys; they compute only the mother algorithm of TDF. With such a control security device, a service broadcaster computes cryptograms of the control word in use for each service key in use and for the limiting conditions coded by the parameter. Each such cryptogram is associated with the name of its service key and its control parameter (a date or a cost), which constitutes an ECM. These ECMs are mandatorily multiplexed in the television channel with the scrambled service components.

If a card receives such a control message (key name, parameter, cryptogram), it first searches for the service key. It then verifies that the conditions indicated by the parameter are compatible with the conditions indicated by the stored status of the service key: for example, the broadcast date lies in a stored subscription period; or in another access mode, a new session is automatically opened by the card, thus reducing the internal amount of credit. If these conditions are satisfied, then the card reconstructs the control word from the cryptogram. Finally, the card delivers the control word upon a get-response command. Lasting less than 1 second, such a transaction with the card is performed once every 10 seconds, Owing to TDF semireversibility, for a given program, the same control word is enciphered in as many cryptograms as there are service keys in use at the same time.

At the present in Europe, in conditional access to audiovisual services on direct broadcasting satellites, several complete systems are in competition. The system in use

in France is named Eurocrypt; the corresponding specifications have been published under authority of the French government [21]. In conjunction with these developments, a new key-carrier card named KC2 has been designed. This card uses a family of unpublished secret key algorithms different from algorithm TDE

5.9 Control of Algorithm Execution in Mask KC2

Mask KC2 is the latest mask in the key-carrier family. It has been designed for conditional access to audiovisual services on direct broadcasting satellites [21]. There is no reason for publishing the cryptographic algorithms used in KC2 cards. Consequently, these algorithms are kept secret.

The execution of the algorithm in KC2 user cards is controlled systematically. A message (either ECM or EMM) consists of three successive fields. The first field of a message contains various fields of data indicating either parameters to be checked for entitlement control or actions to be taken for entitlement management. The second field of a message consists of a variable number (0, 1, or 2) of cryptograms enciphered in electronic codebook (ECB) mode [22] under a confidentiality key. The last field of a message is a redundancy block, also named message authentication code (MAC), and computed in cipher block chaining (CBC) mode [22] for authenticating the complete message under an integrity key.

If a user card receives a message, then it uses an integrity key for checking the message authentication code (MAC) before doing anything else. This mechanism generalizes the previous ***cryptowriting.*** If the MAC is incorrect, the card stops and waits for a reset from the outside world. If the MAC is correct, the card continues. Neither intermediate nor final results of a computation involving an integrity key are transmitted to the outside world.

The subsequent operation may be:

1. The computation of a pair of control words (the cryptograms of the current and next control words are present in the message),
2. The secure writing of a new secret key, such as storing a new service key (the corresponding cryptogram is in the message),
3. An update of rights associated with an existing service key, such as storing a new subscription period (in this case, no cryptogram is present in the message).

If a message picked at random is submitted to such a user card for decipherment, then the probability of getting a result is about 2^{-64}. The cryptographic decipherment by user cards is thus a function which is null almost everywhere.

User cards with such a property are similar to artillery shells without their fuses. User cards are unable to work together. User cards can only work under control of a security device (also called a mother card) for either managing or controlling rights.

Just as in the military, artillery rounds are secured by locking up their fuses, only the security devices have to be controlled in order to secure the whole card system.

5.10 logical Architecture of Card Operating Systems

The consecutive masks, from M4 to MP and TBlOO, on the one hand, and from KC0 to KC2, on the other hand, are more and more elaborate. Each M4 card holds only one key. In KC0 and KCl, in essence, the issuing key is different from the multiple service

keys. In MP, TBlOO, and KC2, keys for confidentiality are distinct from keys for integrity. In masks Dl and D2, the hierarchy of keys is more sophisticated than in Bl or KC1 cards, but the management of rights is less elaborate than in KC0 or KC1 cards. Masks MP and TBlOO aim at replacing M4 and M9 with substantial improvements to file management providing a total independence between data files.

In masks MP and TBlOO, the possibility of extending file structure by creating new files at any time during the life of the card provides a high degree of flexibility, allowing not only the implementation of applications not envisioned at the time the system was fielded, but also the introduction of new applications in already issued cards.

The independence between data files associated with flexible file management is the basis of any high-security multiapplicaton card operating system. This evolution clarifies the security architecture which is presently mature enough to be standardized. Security is fundamental in the logical architecture of smart cards. The security cannot be granted on an existing data file organization as in the existing operating system, disk operating system (DOS), UNIX, or FINDER. The difference takes place mainly in the security management which has to be taken into account in the model from the beginning of the design.

Because a file is fathered by another file, the essential creation process has to be protected. In other words, the right to create or to access a file has to be transmitted by heredity to enforce the independence between applications. This does not compel a son to have the same rights as its father, because it has the freedom to choose its way except for the creation procedure. The transmission of hereditary rights is managed by specific attributes that are transmitted to the son by the father, like chromosomes of a living creature.

The commands affect the objects and the entities specified by the security architecture. The set of commands should be defined afterward to permit compatibility and interchange between cards supporting different applications. TC68/SC6/WG5 (DIS 9992/l) introduced the notions of files, with a common data file (CDF) and application data files (ADF), according to the following set of definitions:

- File-organized set of data elements
- Common data file-unique mandatory file containing the common data elements stored in the card and used to describe the card, the card issuer and the cardholder
- Application data file-optional file supporting one or more services.

In this application-oriented structure, the CDF may clearly be interpreted either as a directory indicating the partition of NVM for applications or as the master in a security architecture. The role of the ADF is not clear and is in fact rather ambiguous. This structure is illustrated in Fig. 26.

JTCl/SCl7/WG4 (WD 7816/4) introduced the notions of master file (MF), dedicated file (DF), subdedicated file (SF), and elementary file (EF), with the following set of definitions.

- File-set of elements, having logical attributes related to security and access methods, and created under common rules
- Master file-unique and mandatory file containing control information and all the other files

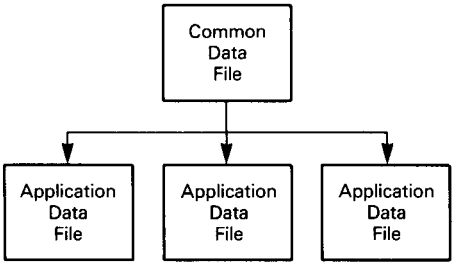

Figure 26 File structure seen by WG5.

- Dedicated file, subdedicated file-file containing control information and other fiies
- Elementary file-file having a security policy under which no other file may be created

Each file, except the MF, is the son of only one other file: either the MF or a (sub)dedicated file.

These definitions proposed in JTCl/SCl7/WG4 are not in contradiction with the previous ones proposed in TC68/SC6/WG5; but are more general and less related to the banking point of view. This revised structure can only be interpreted in terms of security. In fact, WG4 is standardizing the security architecture in the operating system of a smart card. This structure is illustrated in Fig. 27.

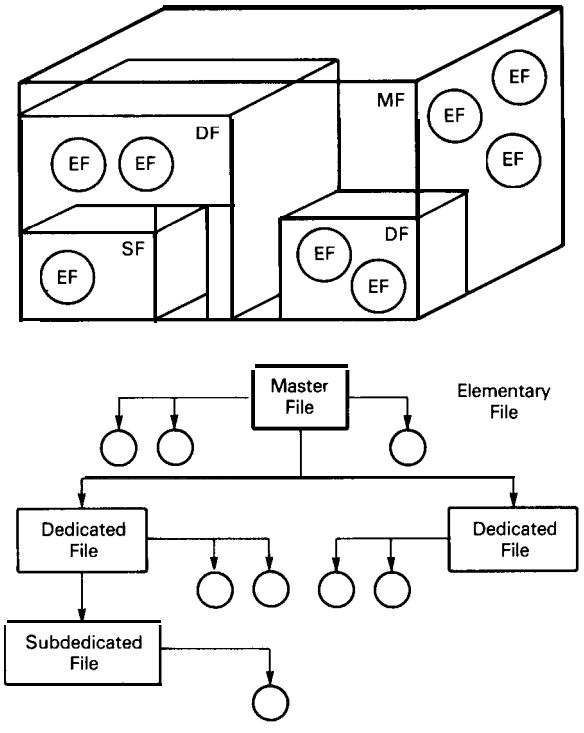

Figure 27 File structure seen by WG4.

These notions are illustrated by existing masks: KC2 (key-carrier), MP, and TBlOO (multipurpose). In a KC2 card, the card issuer, the service managers, and the service providers are clearly identified and associated with levels of files. In a broadcast environment, the card issuer controls the MF and the creation of DFs; each service manager controls the creation and the evolution of EFs in its own (sub)dedicated files; each program broadcaster either controls access rights or consumes credits in EFs. Therefore each KC2 card can support several independent "application" files. A new DF (alias ADF) can be created at any time under control of the MF (alias CDF). The MF and each DF contain a bunch of keys: up to eight management keys (for managing access rights and updating keys) and up to eight control keys (for deciphering control words). The first management key in a DF is mandatorily written under control of a management key of the ME The entitlements, along with various names and addresses, are stored in EF. In addition, in the MF, EFs may hold parameters for a general-purpose device. Such parameters are security information, software to be downloaded, or connection information to access a remote management center. For example, in the French videotex system called *Télétel*, a card reader called *Lécam* connected to a terminal called *Minitel* may thus automatically dial, connect, and access a remote application as long as the elementary data file created for this purpose respects the presentation of the data and one of the access methods recognized by the *Lécams* [6,10].

The problem of providing a means for global access to information in the cards has to be solved in the context of a security architecture. For example, how is it to be possible to access in any card an elementary file containing a phone number for automatic dialing? Each MP or TBlOO card supports several independent application files. A new DF (alias ADF) can be created at any time under control of the MF (alias CDF). The security policy of the MF and of each DF is based on a set of nine independent diversified cryptographic keys: one issuer key, four secondary keys used for authenticating the service provider and for securing operations, and four secondary keys for signature and authentication of the card. The MF, as well as each DF, also contains a set of other elementary files (EF) storing various data.

Mask TBlOO is a superset of MP mask. In each file, four erase keys are added for the erasement of the EEPROM memory, and three dedicated keys are added for digital signatures.

In both cards, the creation** of **the MF which is the birth** of **the card makes use** of **the test contacts before their destruction.

6 EVOLUTION OF CARD AUTHENTICATION

Two methods are currently used for authenticating existing banking cards and Pastel cards. The discussion of secret key one-way functions introduced one such method as was illustrated in Fig. 23. The corresponding dialogue is dynamic, but a secret master key is used in the security module. This method is mainly used for online authentication when a small number of security modules are easily protected in secure areas near central computers. Banking smart cards that use this approach are presently being mass issued in France and Norway. Each such card holds an authentication value which has been written during personalization. Similar certified identities are also used on Pastel cards issued by France Telecom for public phone subscribers.

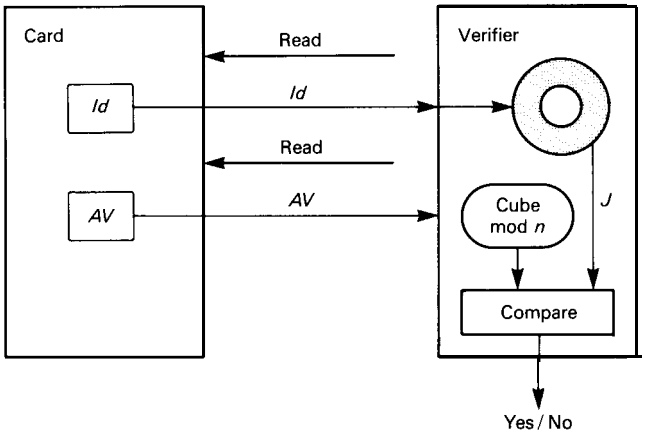

Figure **28 Static card authentication by RSA**

A second method for authenticating cards is illustrated in Fig. 28. Each card holds either its authentication value or its certified identity. A public key is used in the verifying devices. The corresponding dialogue is static. This method does not avoid the cloning of existing cards, and it is used only for assessing a local visual authentication.

Any new method should have the advantages (while eliminating the inconveniences) of both of these two existing methods. The verifying entities should use public keys, and the dialogue should be dynamic.

As shown later (Fig. 29), zero-knowledge techniques are one such method where each proving entity-implemented as a user card-privately uses a personal secret accreditation (analogous to certified identities and authentication values).

6.1 Present Use of Public Key Algorithms

During any financial transaction on point-of-sale terminals or on handheld certificators, a numerical value is provided by the card to the card-accepting-device; the verifying entity raises it to the cube modulo an integer (the public modulus) published by the card issuer. The value is accepted when the result repeats twice the same bit string consisting of various information such as chip serial number, bank account number, service code, and validity period. Figure 28 illustrates this process. If the result is inconsistent, then the point-of-sale terminal rejects the card. The result must be consistent for the transaction to be continued. The scheme is secure against forgery since to create a number whose cube is of the required form is equally difficult with the factoring of the modulus.

According to the RSA algorithm [23], the prime factors of the composite integer are involved in the computation of such authentication values. The prime factors are stored, protected, and used during card personalization by security devices named Camelias, while the composite integer published by the issuer is known and used by any verifying device. The composite integers presently in use are 320 bits long (297 decimal digits).

As a result of advances in factorization techniques over the past few years, a 320-bit composite integer is no longer secure against a network of workstations (DEC or SUN) with a few day's computation. The fact still remains that the scheme is part of the specifications written in 1983 [8]. At the very first revision of these banking specifications, larger composite integers should be introduced.

ISO/IEC JTCl/SC27/WG2, Security Techniques, Security Mechanisms, is preparing a standard on a Digital Signature Scheme Giving Message Recovery (DIS 9796). This standard may be used for specifying an accreditation that generalizes the authentication value of a banking card and the certified identity of a ***Pastel*** card. Redundancy rules are more elaborate; odd and even exponents are specified, thus extending the RSA algorithm and known generic attacks against the RSA algorithm are eliminated.

6.2 Zero-Knowledge Techniques

The first practical zero-knowledge scheme was proposed by Fiat and Shamir [24] in 1986. In their scheme, computations were reduced to a near minimum. The security level grows exponentially with the product of the number of interactions (challenge/response pairs) by the number of accreditations (accreditation is a better name for authentication value and certified identity). Any desired level of security may thus be achieved as a compromise between the number of accreditations stored in the card and the number of successive successful interactions required for an acceptance. However, in the design of smart cards one must be concerned with both exchanges and storage: The exchanges with the outside world are time-consuming, while the NVM is an expensive resource. Therefore, minimization of computations alone does not seem to be the best optimization.

A second solution suited to smart cards was published in 1988 by Guillou and Quisquater [25,26]. In this protocol, storage and exchange are reduced to an absolute minimum: only one accreditation and only one interaction with the outside world. The computations required in the Guillou-Quisquater scheme are greater than is required by the Fiat-Shamir scheme for the same level of security, but only by a factor of approximately three. We give a sketch of the Guillou-Quisquater scheme. Each card is characterized by its own set of credentials (a better name for what we have been calling the card's identity). A set of credentials consists of data specified at the application level, such as bank account number, chip serial number, validity period, and service code. More generally, the set of credentials of a proven entity includes at least a validity period and a distinguished name. The set of credentials ***Cr*** is transformed into a longer integer, termed representative ***J*** of the same size as modulus n. The transformation from ***Cr*** into ***J*** is specified by publicly known redundancy rules. Such public rules are being standardized in ISO/IEC 9796 (DIS 9796). The public key of the accrediting entity consists of a public exponent v and a composite integer n. Therefore, any accreditation, denoted by ***B,*** is the secret solution to a public equation. DIS 9796 specifies such an accreditation.

$$JB^v \equiv 1 \pmod{n}$$

Today, a good size for the composite integer is 512 bits, and then the set of credentials of a card may be as long as 256 bits.

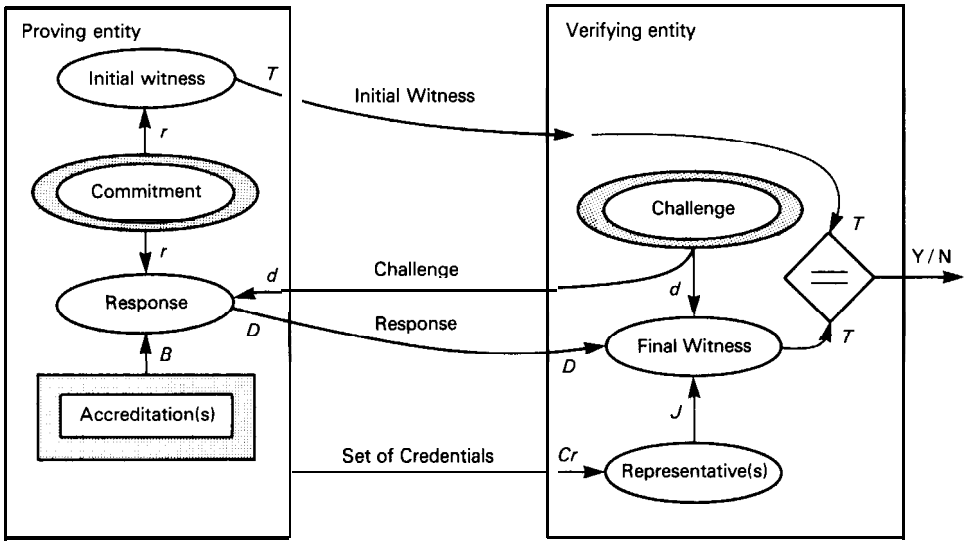

Figure 29 General zero-knowledge interactive authentication mechanism.

Figures 29 and 30 show an authentication in three moves with a card claiming the set of credentials ***Cr.***

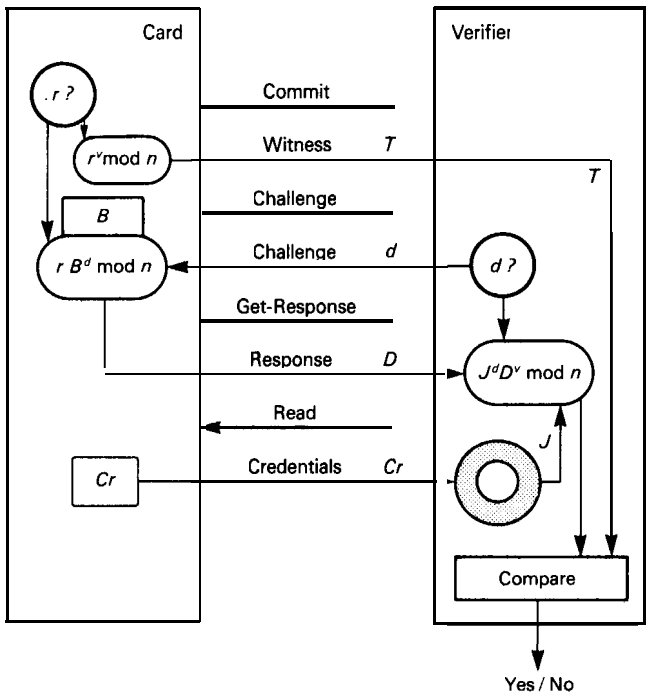

Figure 30 Card authentication by the Guillou–Quisquater scheme.

The sequence of actions by the card and by a verifying entity in proving the identity of the card are:

1. For each transaction, the card privately and randomly selects a new integer, ***r,*** in the ring of integers mod ***n.*** This integer is termed the ***commitment r.*** The card then privately computes the v-th power modulo ***n*** of commitment ***r.*** The result is called the ***initial witness T. As a first move, the card transmits this initial witness to the verifier.***

$$T \equiv r^v \pmod{n}$$

2. Then, and not before, the verifier randomly selects an integer, ***d,*** between zero and v − 1. This integer is called the ***challenge d. As a second move, the verifier transmits this challenge to the prover.***
3. The card then computes the product mod ***n*** of commitment ***r*** by the ***d-th power*** *of* ***accreditation B. The result is the response D. As a last move, the card transmits this response to the verifier.***

$$D \equiv rB^d \pmod{n}$$

4. Finally, the verifying entity computes the product mod ***n*** of the v-th power of response ***D*** by the d-th power of representative J. The result is called the ***final witness T'. The authentication succeeds*** *if* ***and only*** *if* ***initial and final witnesses are equal modulo n.***

$$T' = D^v J^d \pmod{n}$$

If all steps have been properly executed,

$$T' = D^v J^d = (rB^d)^v J^d = r^v B^{dv} J^d = r^v (JB^v)^d \equiv r^v = T \pmod{n} \qquad (1)$$

since B was constructed (secretly) to satisfy the equation

$$JB^v \equiv 1 \pmod{n}$$

The following three statements are crucial to understanding the conditions imposed in the protocol and why the verifying entity learns nothing about the underlying secret accreditation in the process.

1. If a cheater could guess the challenge, ***d,*** then he would have a winning strategy. If he knew what the challenge from the verifier would be before he had to commit himself to the initial witness he could construct a ***T*** to satisfy the test in (4) without having to know ***B.***
2. A judge cannot distinguish enrolled data corresponding to successful verifications from enrolled masquerades where the challenges have been asked before fixing the witnesses.
3. The knowledge of two responses ***D,*** and D_2 to two distinct challenges d_1 and d_2 for the same witness ***T*** is equivalent to the knowledge of the k-th power of accreditation ***B*** where ***k*** is the greatest common divisor of v and $d_2 - d$,.

(1) and (2) require that guessing the challenges should be impossible (or at least extremely improbable) because a successful guess would allow cheating. Similarly, guessing the commitments should also be impossible because that would compromise the secrecy of the accreditation. It is a little more difficult to explain why the proving entity must use a new and randomly chosen initial witness for each transaction and respond only to a single challenge to each witness.

Let the public key of the accrediting entity be a modulus, II, and a public verification exponent, v. The modulus is the product of two distinct primes p and q large enough to insure that n will be infeasible to factor. The public verification exponent v is chosen to be a prime number that does not divide $p - 1$ or $q - 1$; that is, such that

$$(v, (n)) = 1$$

We will show how the verifying entity could determine the proving entity's secret accreditation B if he were to respond to two challenges using the same witness, T. Let d_1 and d_2 be the two challenges (integers) such that

$$0 \leq d_1 < d_2 < v$$

and let D, and D_2 be two responses to the challenges d, and d_2 respectively for the same witness T. Then

$$D_1^y J^{d_1} \equiv D_2^y J^{d_2} \equiv T \pmod{n}$$

or

$$\left(\frac{D_2}{D_1}\right)^v J^{d_2 - d_1} \equiv 1 \pmod{n} \tag{2}$$

Given any pair of positive integers x and y, the congruence

$$ax - by = \pm (x, y) \tag{3}$$

always has a solution, where a is a reduced residue modulo y; that is, $0 < a < y$, and b is a reduced residue modulo x. (x, y) denotes the greatest common divisor of x and y. Equation (3) is a specialized form of Bezout's identity* (after Etienne Bezout) and the unknowns a and b are known as the Bezout coefficients. It is easy to calculate a and b using the extended Euclidean algorithm.

Replacing x by $d_2 - d_1$ and y by v in Eq. (3), we get

$$a(d_2 - d_1) - bv = \pm 1 \tag{4}$$

since $(v, d_2 - d_1) = 1$. Equation (4) can be solved to find the reduced residues a and b.

If Eq. (2) is raised to the exponent a, we get

$$\left(\frac{D_2}{D_1}\right)^{av} J^{a(d_2 - d_1)} = \left(\frac{D_2}{D_1}\right)^{av} J^{\pm 1 + bv} \equiv 1 \pmod{n}$$

---

*According to Gauss, Lagrange, and Legendre, the Bezout identity was discovered earlier by Bachet de **Méziriac** but hears Bezout's name. Such is the history of science.

or

$$J^{\pm 1}\left[\left(\frac{D_2}{D_1}\right)^a J^b\right]^v \equiv 1 \pmod{n}$$

But since the secret accreditation B is known to be the (supposedly secret) solution to the public equation

$$JB^v \equiv 1 \pmod{n}$$

we have

$$B^{\pm 1} \equiv \left(\frac{D_2}{D_1}\right)^a J^b \pmod{n}$$

Therefore, if the proving entity were to respond to two challenges, d, and d_2, chosen as described above, an opponent could then solve for the proving entity's secret accreditation B. Consequently the proving entity must construct a new and random initial witness T for each transaction if B is to be kept secret.

The size of the public exponent is determined by the security requirements of the particular application: A cheater has at most one chance out of v to deceive a verifier, and a verifier has at least v − 1 chances out of v to detect a cheater.

In a local verification where the verifying entity retains the card if it fails an authentication exchange, the public exponent may (need to) be as large as $2^{17} + 1$. But if the cardholder itself is retained by the verifying entity, then the public verification exponent may even be reduced to be as small as $2^8 + 1$ or $2^4 + 1$.

Cards and verifiers perform similar operations, with the same complexity of computation. If a card can be authenticated, it can also authenticate other cards. With such schemes, banking security modules should be personalized as retailer cards. All these cards, user cards as well as retailer cards, should authenticate each other in a very symmetrical way.

6.3 New Signatures

By using general principles first suggested by Fiat and Shamir [2,3] the previous method can be adapted in a natural way to a digital signature scheme by using a hash function. A good hash function must be one-way and collision-resistant, in the sense that finding a collision is very difficult; that is, practically impossible. A collision of the function "hash" is a pair of distinct arguments x' and x" such that

$$\text{hash}(x') = \text{hash}(x'')$$

ISO/IEC JTC1 SC27/WG2 is now standardizing hash functions for digital signatures (DP10118).

In several signature schemes, the hashing of the message is the input to the inverse of a trapdoor permutation such as the RSA algorithm. Such schemes are susceptible to the *birthday attack* where a cheater looks in advance for a collision of two messages: one favorable to the signer and the other one favorable to himself. When the cheater obtains the signature of the message favorable to the signer, it can reveal the

second message favorable to himself. In these schemes, the hashing of the message must be long enough to avoid a search for collisions using a birthday attack. Hash results of 64 bits are too short. In such signature schemes, the hashing of the message must be longer: The suggested length is 128 bits.

But in signature schemes that are derived from zero-knowledge proof techniques, the random challenge selected by the verifying entity is replaced by a pseudorandom challenge computed by hashing together the message and the initial witness. When message and witness are hashed together into a 64-bit challenge, the same pseudorandom variable (challenge ***d)*** is involved at both the beginning and the end of the verification process. In this case the birthday attack seems to be irrelevant. Therefore a 64-bit hashing is secure in this case.

The size of the exponent v is now around 64 bits (instead of 17 or fewer as before), which corresponds to the entropy of the challenge ***d.*** This is the price one must pay for giving up an interactive protocol.

Figure 31 illustrates the G-Q signature process: Message ***M*** is signed by an appendix consisting of initial challenge ***d,*** response ***D,*** and credentials ***Cr.*** In such a scheme, challenge ***d*** represents redundancy while response D represents randomness.

Figure 32 illustrates the G-Q verification process which begins by recomputing representative ***J*** from the set of credentials ***Cr*** according to public redundancy rules. Then final witness ***T'*** is obtained by computing mod ***n*** the product of the v-th power of response ***D*** by the d-th power of representative ***J.*** Then final challenge d' is obtained by hashing message ***M*** and final witness ***T'.*** A signature is valid if and only if initial and final challenges are equal.

Let the public exponent be $2^{64} + 1$ (a product of two primes). Let the Hamming weight of challenge ***d*** be limited to 32. For example, by choosing as challenge ***d*** the complementary value of hash(T, M) when the result contains more than 32 1's. Then the final witness T' may be computed by squaring response ***D*** 64 times, interleaved with at most 32 multiplications by representative J, plus a last multiplication for adjusting with response ***D.*** The computational complexity of a verification is therefore less

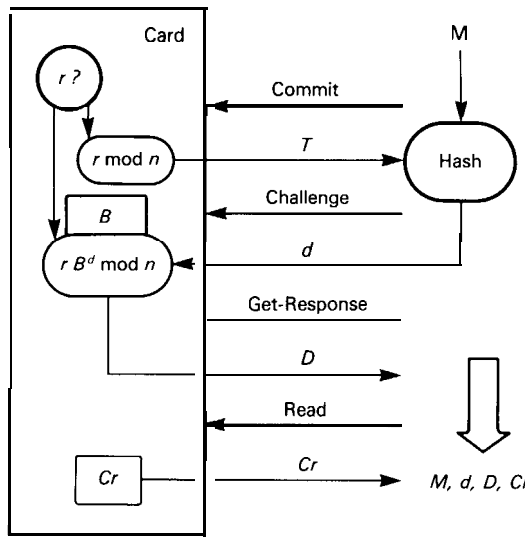

Figure 31 Guillou–Quisquater signature scheme.

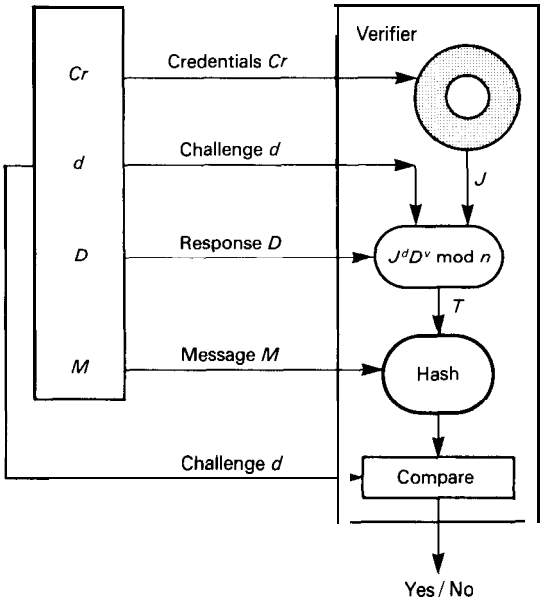

Figure 32 Verification of a Guillou–Quisquater signature.

than 100 multiplications mod n. Signature and verification processes have the same level of complexity. Therefore, if a card can sign, it can also verify. These authentication and signature schemes are based on identity. It is an evolution of methods published separately in 1984 by Shamir [23] and by a French banking card organization [8].

The cards in the Guillou-Quisquater scheme hold secrets that are not cryptographic keys. An accreditation does not provide immediate confidentiality services. The method cannot be directly derived from its basic goal-integrity. The RSA scheme does not have this property. In the RSA scheme, the secret prime factors can be used for both integrity and confidentiality purposes. This property that the secret is exclusively usable for integrity purposes is in tune with political and governmental requirements. Indeed, a fair integrity (signature and authentication) scheme should not make any assumption as to the integrity (morality and good citizenship) of its potential users.

6.4 Multisignatures

In multisignature schemes, several signing entities collaborate in signing the same message. In a first-trivial-solution, each signing entity signs separately; but that is not cooperation. In a second solution, the signing entities sign successively, one after another, on a progressive basis: The present signing entity signs the result produced by the previous ones. But this method introduces between entities an order that is unnatural in some applications. Intermediate significant results are a potential cause of dispute. Later on, an additional signing entity may always sign as the new last one. A progressive process intrinsically excludes the notion of simultaneity which looks very attractive for avoiding several problems resulting from signature repudiation.

In a satisfactory and natural multisignature scheme, the intermediate results should be meaningless so long as the last result has not yet been obtained. Each signing entity should know precisely the other participants involved in the multisignature process. Also, there should be no undetectable way to introduce at a later stage an additional ultimate signer. The notion of progressivity is thus replaced by a notion of simultaneity. This attractive property is offered by signature schemes derived from zero-knowledge.

Let us first consider the general solution derived from zero-knowledge techniques. Several signing entities collaborate in a global process. These signing entities may depend on different authorities that are members of a directory system as described in CCITT in X509 and now under going standardization by IS0 (ISO9594/8). Each signing entity proposes an initial witness. Then a global initial challenge is produced by hashing all the initial witnesses together with the common message to be signed. All the initial witnesses and the global initial challenge are then sent to each signing entity for verification of the initial challenge and construction of individual responses. Finally, the message is signed by an appendix consisting of the global initial challenge, all the individual responses, and all the sets of credentials.

The verification begins with the computation of each final witness from each response, each set of credentials, and the global initial challenge. Then all the final witnesses are hashed together with the message so as to obtain the final challenge. The signature is accepted if and only if the initial and final challenges are equal.

Let us now consider two smart cards issued by the same accrediting entity: Each card stores its unique accreditation related to its own set of credentials. The accrediting entity has published a public key consisting of two integers n and v. Both cards, with sets of credentials Cr, and Cr_2, cooperate on the same personal computer to produce a global signature of message M.

The signature protocol consists of the following steps:

1. Global initial witness T is the product mod n of both individual initial witnesses, T, and T_2.
2. Global initial challenge d is the hashing of global initial witness T and message M.
3. Global response D is the product mod n of both individual responses D, and D,.

Let us write the corresponding equations:

$$T = T_1T_2 \bmod n;\quad d = \text{hash}(T, M);\quad D = D_1D_2 \bmod n$$

Message M is signed by the appendix, consisting of global initial challenge d, global response D, and two sets of credentials Cr, and Cr_2.

This method may be extended to any number of participants. The signing appendix then consists of one global initial challenge, one global response, and all the sets of credentials.

The verification is performed in the usual way. The global representative is the product mod n of all the representatives. Global final witness T' is computed as the product mod n of the v-th power of global response D by the d-th power of the global representative. Finally, global final challenge d is computed as the hashing of global final witness T' and message M. The signature is accepted if and only if initial and final challenges are equal.

As a matter of fact,

$$T' = D^v(J_1J_2)^d = (r_1B_1^d r_2B_2^d)^v J_1^d J_2^d \mod n$$
$$= (J_1B_1^v)^d\,(J_2B_2^v)^d\,(r_1r_2)^v = T_1T_2 = T \mod n$$

A cosignature is attractive for retail banking applications. The financial message may consist of a date and an amount, plus a serial number for the buyer and a serial number for the seller. This message is signed simultaneously by the retailer and by the customer. The result is an electronic check which may be verified by any other card, including both the retailer card and the customer card.

6.5 Probable Future Evolution of Smart Cards

Examining the development of smart card systems, we see a strong clue to the future evolution of smart cards: ***Curds will interact without sharing secrets!***

Cryptology in the cards will soon include new authentication and signature methods derived from zero-knowledge techniques. New access methods which are standardizable are now being developed. In addition, the IS0 working group on public keys is preparing a first working draft on zero knowledge techniques. In open systems, the standardization of authentication and signature tools is an important issue that does not seem to conflict with political and governmental considerations. This evolution has a good influence on banking card systems by making more nearly symmetric the personalization of user cards and security devices; anonymous secrets can be replaced by personalized secrets. The security devices may thus be considered as retailer cards.

Let us give an illustrative practical example of the practical use of this cryptologic development in another field of application. Health cards record a lot of confidential data about cardholders. These data should always be easily accessible when the PIN is presented correctly. However, the cardholder may select fields of data to be accessible by other methods, even though he cannot present his PIN. A health card should authenticate accredited physician cards and the cards of other rescue personnel. To provide access to information that has been previously selected by the cardholder in a proper fashion, the card should only recognize the public keys made public by the relevant emergency medical personnel.

Continuing this line of reasoning, the general public will probably buy smart cards in the future as it does calculators today. Having purchased such a card, the consumer would visit his service provider and after filling out a form and signing a contract, the service provider would write an accreditation in the personal card of the user. The same card would hold several accreditations from different authorities for different purposes. Using the appropriate accreditation, the consumer would later be able to access the corresponding service either by interactively authenticating himself or by signing. This scenario describes what we believe to be both a possible and probable evolution of smart card systems toward open systems where multiapplication cards will be the property of their users. We don't see why bankers should have to deal with all the side effects due to a card expiring on a certain date, such as refunding taxable units of parking or telephone messages. If a card is the personal property of its holder, the problem of refunding small amounts is to be solved directly between the cardholder and the service provider. Such situations modify considerably the economy of the systems. The economy of smart card systems is related in a very sensitive way to the cost of the

cards and to the security architecture of the global systems. Up to the present, side effects due to shared security such as rigid expiration dates have slowed down the synergy between applications of smart cards.

7 CONCLUSIONS

The development of smart card systems raises a major question related to cryptography. Smart cards to be used by the general public are very efficient security tools because their logical security is based on cryptographic techniques. The manufacture, use, and export of cryptographic materials are subject to national regulations and export controls because cryptography has national security ramifications.

A crucial question therefore (to the development and application of smart cards) is how are the commercial needs to be accommodated while at the same time satisfying the governmental concerns?

A card is an element of solution to a problem, and the whole solution has to be considered. The following schematic example shows three generic levels; authentication, keying, and (de)scrambling. If confidentiality is required on a communication network, then a session key may well be created by exponentiating in finite fields, as suggested by Diffie and Hellman [28]. An intruder may well be active on the communication path. But such an intruder is immediately detected if, before doing anything else, all the keying data elements are authenticated (e.g., by a zero-knowledge protocol). The session key is subsequently used for scrambling/descrambling the transmitted signal.

(De)scrambling computations are to be performed by dedicated devices such as radiotelephones and decoders. For example, the picture and sound signals of a pay television program are scrambled according to their respective natures and the codings fixed by the television standards. Solutions to confidentiality problems appear to be essentially related to the network and to the service provided on the network. Scrambling mechanisms are standardized according to the application. National regulations may influence the solutions.

Authenticating computations are performed by detachable security devices **such as** smart cards. Authentication is always performed in reference to an authority such as a card issuer. Essentially related to the individuals, solutions to integrity problems must be international in scope. Even if the technology evolves apace, a European payment card must continue to be usable in the United States and vice-versa. Therefore the standardization of authentication mechanisms is required.

Keying computations may be performed either by the dedicated devices or by the personal cards. Keying and authentication should not be confused because the commercial cryptographic techniques separate as much as possible integrity (certification, authentication, identification, signature) and confidentiality (secrecy, discretion).

In this context, what is the role of smart cards? There are two main approaches depending on whether secret key or public key techniques, respectively, are used. In making such a decision, system designers must be guided by practical trade-offs between cost and performance. The smart card controls the use of any internal cryptographic algorithm (i.e., one able to provide confidentiality). Cards KC2 provide a good illustration: Any cryptogram submitted for decipherment to a card KC2 must be included in a message terminated by a message authentication code (MAC). Deciphering

occurs if and only if the associated MAC is correct. Hence, KC2 user cards cannot interact for keying a communication. Cards KC2 very well fit access control problems such as pay television. Such smart cards are the domain of customized secret algorithms that ensure independence between applications. But the corresponding mother cards still have to be controlled. The management of security in this approach cannot be generalized to an open context.

There is a strong need for standards based on public key and zero-knowledge techniques. If these standards are restricted to integrity, and cannot be derived for confidentiality purposes, then they should not conflict with governmental and political policy. For example, some authentication schemes are by nature restricted to integrity: An accreditation is a secret but not a key. However, the corresponding implementations should also avoid any potential misuse, even though the basic operations (multiplying and exponentiating large integers modulo large integers) are also the basic operations of various confidentiality schemes. If such an authentication scheme is efficiently performed in an "open" operating system, then the corresponding efficient arithmetic operator (a dedicated chip, for example) may be misused for keying any pair of users on an open communication network by exponentiating in finite fields. Smart cards, however, must be tamper-resistant. If the mask is designed correctly, then the card's computational power, including the built-in arithmetic operators, cannot be misused for other purposes. Therefore the integrity standards (authentication, signature, and key management) based on public key and zero-knowledge techniques should be completed by national agreements authorizing the corresponding implementations. Some masks for SPOMs should obtain such an agreement.

The approach just described makes it possible to predict the concept of "basic common cards" to be issued by a public authority (like a bank of issue) and to be freely sold to the general public for general purposes. On such an "agreed-upon" card, the user will ask for the introduction of several various applications, such as payment and credit by a bank, public transportation by an authority, access by a service provider, telephone privileges granted by an operator, personal files, etc. The study, specification, experimental trials, and standardization of these basic common cards will almost certainly become a major international development over the next few years.

APPENDIX A: IS0 PRESENTATION

IS0 has three official languages (don't try to spell IS0 in any of them!):

- English: International Organization for Standardization
- French: Organisation Internationale de Normalisation
- Russian: Международная Организация no Стандартизации

A. 1 IS0 STRUCTURE

The IS0 is the specialized international agency for standardization, comprising the national member bodies of about ninety countries.

A national member body is the most representative organization for standardization in its country. The United States is represented by ANSI (American National

Standards Institute), France by Afnor (Association Française de Normalisation), the United Kingdom by BSI (British Standards Institution), and the USSR by GOST.

IS0 is administered by a council consisting of a president, a vice-president, a treasurer, and eighteen member bodies. The council creates technical committees (TC) in main fields of interest. Each TC then creates subcommittees (SC). Each SC creates working groups (WG). And finally, each WG may create task forces (TF). A central secretariat in Geneva prepares the ballots and edits and prints the standards. This pyramidal structure is illustrated in Fig. A. 1.

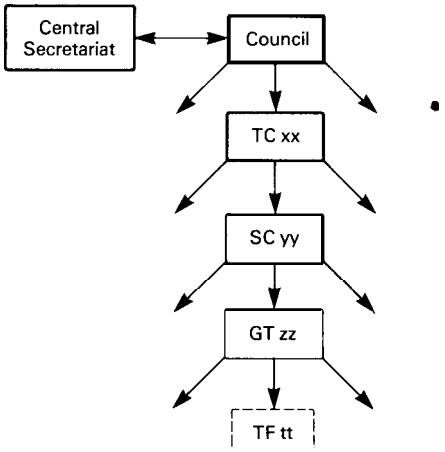

Figure A.1 ISO Structure.

At TC and SC levels, member bodies interested in the program of work are either participants (P-members) or observers (O-members).

Decisions are taken in association with liaisons (L-members) like IEC (the International Electrotechnical Commission), CCITT (le Comité Consultatif International Télégraphique et Téléphonique), and more generally, any international organization interested in the program of work.

The votes are organized at SC and TC levels, with one ballot per country. L-members do not vote, but their technical comments are considered.

At WG and TF levels, there is no ballot, only experts who are nominated by P-, 0-, and L-members and who draft all the documents.

A.2 IS0 PROCEDURE (Figure A.2)

0	**WI**	New Work Item
1	**WD**	Working Draft
2	**DP**	Draft Proposal
3 to 5	**DIS**	Draft International Standard
6 to 8	**IS**	International Standard

Figure A.2. ISO procedure (stage numbers).

Any P-, 0-, or L-member may propose a new work item (WI) together with estimated resources, dates, and targets. After approval by a ballot at TC level, the WI is included in the program of work of the relevant SC which assigns it to a WG.

The WG produces a document as soon as possible. After reaching a consensus, the preliminary document is transmitted to the SC as a working draft (WD). Through a resolution at a plenary meeting, the SC approves the document for registration as a draft proposal (DP).

The DP number is that of the future International Standard. A DP is circulated at SC level for a three-month ballot. Results and comments are sent to experts to consider the positive comments and to resolve the negative ballots. The goal is to reach unanimity or at the very least a substantial consensus.

After approval, a DP becomes a draft international standard (DIS): It is translated, edited, and printed under central secretariat control, and then balloted for 6 months at TC level. A DIS must be approved by at least 75% of the members voting, including a majority of P-members.

After approval, a DIS is transmitted to council with a comprehensive report for final decision. After council acceptance, the International Standard (IS) is published.

Every 5 years after publication, each IS is subject to revisions to confirm, amend, or withdraw the IS.

Several successive DP or DIS versions may be balloted; amendments due to negative comments of a member may offset a positive vote of another member; industrial and commercial ventures may modify the position of a member; compromises obtained during a meeting may be denied at the next meeting. Thus technical experts often become involved in strategic games without being prepared.

The elaboration of an IS is a very long story (5–10 years): Any efficient standard is largely supported by the members.

GLOSSARY

CLK. Clock line, one of the six contacts standardized by ISO.
CPU. Central process in 1 unit.
Die, pl. dice. Individual device (microcomputer or other) on silicon.
EEPROM. NVM-may be erased by applying special voltage.
EPROM. NVM-may be erased by exposure to UV light.
GND. Ground, reference voltage, one of the six contacts standardized by ISO.
I/O. Input/output communication line, one of the six contacts standardized by ISO.
MAC. Message authentication code; artificial redundancy used to check message authenticity.
Mask. Medium used to convert customers' application software (ROM code) to a pattern on silicon and, by extension, the application software itself.
MCU. Single-chip microcomputer unit.
Microcomputer. A system containing a microprocessor, various memories, and other peripheral devices.
NVM. Nonvolatile memory.
PIN. Personal identification number.
RAM. Random access memory.
ROM. Read-only memory.
RST. Reset line, one of the six contacts standardized by ISO.
Smart card. A card, which looks like a credit card, but which contains a microcomputer.
SPOM. Self-programmable one-chip microcomputer, a type of secure MCU.

Test mode. Special operating mode for an MCU for testing by the manufacturer prior to shipping to customer.

User mode. Normal operating mode for MCUs, the only mode used in a smart card.

VCC. Power supply line, one of the six contacts standardized by ISO.

VPR Programming voltage line, one of the six contacts standardized by ISO.

Wafer. Slice of silicon which, after processing, contains typically hundreds of individual dice.

REFERENCES

[l] L. C. Guillou and M. Ugon, "Smart card: A highly reliable and portable security device," in ***Lecture Notes in Computer Science 263; Advances in Cryptology: Proc. Crypto'86,*** A. M. Odlyzko, Ed., Santa Barbara, CA, Aug. 11-15, 1986, pp. 464-479. Berlin: Springer-Verlag, 1987.

[2] M. E. Haykin and R. B. Warnar, ***Smart Card Technology, New Methods for Computer Access Control,*** NIST 500-175. Gaithersburg, MD: National Institute of Standards and Technology, Sept. 1988.

[3] D. W. Davies, "Smart cards, digital signatures, and negotiable documents," in ***Proceedings of the International Conference on Secure Communications Systems,*** London, UK, February 22-23, 1984, pp. l-4. London: Institution of Electrical Engineers, 1984.

[4] A. G. Mason, "Conditional access for broadcasting," in ***Proceedings of IBC'88, International Broadcasting Convention (Conf. Publ.*** No. 293), Brighton, UK, Sept. 23-27, 1988, pp. 328-332. London: Institution of Electrical Engineers, 1988.

[5] J. Svigals, "Improved security with integrated circuit cards," ***J. Inform. Syst. Management, vol. 5,*** no. 2, pp. 32-38, 1988.

[6] A. Turbat, "The smart card, an ace in France's telecom system," ***Telephony,*** vol. 204, no. 26, pp. 78, 80, 82, 86, June 27, 1983.

[7] M. Paterson, ***Secure Single Chip Microcomputer Manufacture,*** Eng. Bull. EB4OO/D. Phoenix, AZ: Motorola Semiconductor, 1990.

[8] Specifications et normes de la carte à mémoire bancaire. Paris: Groupement des Cartes Bancaires, Jan. 1984.

[9] M. Ugon and P. Schnabel, ***TB100, The Highly Secure Multipurpose Smart Card Family,*** Bull. CP8. Trappes, France, Jan. 1990.

[10] J. F. Briend and J. J. Plancke, "French PTT Minitel and L&am programme," ***Philips Telecommun. Data Systems Rev.,*** vol. 45, no. 2, pp. 10-26, June 1987.

[ll] W. L. Price, "Standards for data security: A change of direction," in ***Lecture Notes in Computer Science 293; Advances in Cryptology: Proc. Crypto'87,*** C. Pomerance, Ed., Santa Barbara, CA, Aug. 16-20, 1987, pp. 3-8. Berlin: Springer-Verlag, 1988.

[12] ***Data Encryption Standard,*** FIPS PUB 46. Gaithersburg, MD: National Institute of Standards and Technology, April 198′.

[13] L. C. Guillou, M. Davio, and J.-J. Quisquater, "Public-key techniques: Randomness and redundancy," ***Cryptologia,*** vol. 23, no. 2, pp. 167-189, April 1989.

[14] ***Defending Secrets, Sharing Data; New Locks and Keys*** *for* ***Electronic Information,*** OTA-CIT-310. Washington, D.C.: U.S. Congress, Office of Technology Assessment, Oct. 1987.

[15] R. McIvor, "Smart cards," ***Scientific American,*** vol. 253, no. 5, pp. 130-137, Nov. 1985.

[16] J. K. Omura, "A smart card to create electronic signatures," in ***Proceedings of BOSTONICCI89, IEEE International Conference on Communications, vol. 3,*** Boston, MA, June 11-14, 1989, pp. 1160-1164. New York: IEEE, 1989.

[17] J.-J. Quisquater, D. de Waleffe, and J.-P Bournas, "Corsair, a chip card with fast RSA capability," in ***SMART CARD 2000: The Future of IC Cards; Proc. IFIP WG II.6 Internat. Conf.,*** D. Chaum and I. Schaumuller-Bichl, Eds., Laxenburg, Austria, Oct. 19-20, 1987. Amsterdam: North Holland, 1989.

[18] L. C. Guillou, "Radiodiffusion à péage pour application au télétexte ANTIOPE," in ***Actes du congrès de Liege,*** Belgique, Nov. 24, 1980.

[19] L. C. Guillou, "Smart cards and conditional access," in ***Lecture Notes in Computer Science 209; Advances in Cryptology: Proc. Eurocrypt'84.*** T. Beth, N. Cot, and I. Ingemarsson, Eds., Paris, France, April 9-11, 1984, pp. 480-489. Berlin: Springer-Verlag, 1985.

[20] ***Specification des systèmes de la famille MAC/paquets,*** Document Technique 3258. Bruxelles: Centre technique de l'UER/EBU, Oct. 1986.

[21] ***Systèmes d'accès conditionnel pour la famille MAC/paquet,*** EUROCRYPT. République Française: Ministere des PTT, Ministere de l'Industrie, Ministbre de la Culture, March 1989. (Available on request at CCITT, Rennes.)

[22] ***DES Modes of Operation,*** FIPS PUB 81. Gaithersburg, MD: National Institute of Standards and Technology, Dec. 1980.

[23] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," ***Commun. ACM,*** vol. 21, no. 2, pp. 120-126, Feb. 1978.

[24] A. Fiat and A. Shamir, "Unforgeable proofs of identity," in ***Proceedings, SECURICOM'87: 5th Worldwide Congress on Computer and Communications Security and Protection,*** Paris, France, March 4-6, 1987, pp. 147-153. Paris: Societe d'edition et d'organisation d'expositions professionnelles, 1987.

[25] L. C. Guillou and J.-J. Quisquater, "A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory," in ***Lecture Notes in Computer Science 330; Advances in Cryptology: Proc. Eurocrypt'88,*** C. G. Gunther, Ed., Davos, Switzerland, May 25-27, 1988, pp. 123-128. Berlin: Springer-Verlag, 1988.

[26] J.-J. Quisquater and L. C. Guillou, "Des procédés d'authentification bases sur une publication de problèmes complexes et personnalisés dont les solutions maintenues secretes constituent autant d'accreditations," in ***Proceedings of SECURICOM'89: 7th Worldwide Congress on Computer and Communications Security and Protection,*** Paris, France, March l-3, 1989, pp. 149-158. Paris: Société d'édition et d'organisation d'expositions professionnelles, 1989.

[27] A. Shamir, "Identity-based cryptosystems and signature schemes," in ***Lecture Notes in Computer Science 196; Advances in Cryptology: Proc. Crypto'84, G.*** R. Blakley and D. Chaum, Eds., Santa Barbara, CA, Aug. 19-22, 1984, pp. 47-53. Berlin: Springer-Verlag, 1985.

[28] W. Diffie and M. Hellman, "New directions in cryptography," ***IEEE Trans. Inform. Theory, vol. 22,*** pp. 644-654, 1976.