New Directions for Integrated Circuit Cards Operating Systems

Pierre Paradinas[1&3] and Jean-Jacques Vandewalle[1&2]

[1] Rd2p, Recherche et Développement Dossier Portable, CHRU Calmette, 59 037 Lille Cédex, France,
tel. : (33) 20 44 60 44, fax : (33) 20 44 60 45, emails : *{pierre, jeanjac}@rd2p.lifl.fr*

[2] Université Laval, Département d'informatique, Québec, G1K 7P4, Canada
tel. : (1) 418 656 2580, fax : (1) 418 656 2324, email : *jeanjac@iad.ift.ulaval.ca*

[3] Gemplus Card International, Plaine de Jouques, B.P. 100, 13 881 Gémenos, France,
tel. : (33) 42 32 50 00, fax : (33) 42 32 50 90, email : *pierre@gemplus.fr*

Abstract. Integrated circuit cards or smart cards are now well-known. Applications such as electronic purses (cash units stored in cards), subscriber identification cards used in cellular telephone or access keys for pay-TV and information highways emerge in many places with millions of users. More services are required by applications providers and card holders. Mainly, new integrated circuit cards evolve towards non-predefined multi-purpose, open and multi-user applications. Today, operating systems implemented into integrated circuit cards cannot respond to these new trends. They have evolved from simple operating systems defining an hardware abstraction level up to file management systems or database management systems where the card behaviour was defined once at the manufacturing level or by the card issuer. The needs for open and flexible card life cycle enabling to accommodate executable code loaded by different service providers require a new generation of smart cards. Operating systems based on object-oriented technologies are key components for future integrated circuit cards applications.

Keywords. Integrated Circuit Card Operating System, Integrated Circuit Card Applications, Object-Oriented Technologies, Secured method execution.

1 Integrated Circuit Cards Features

Integrated circuit cards (ICC) are plastic cards made of the exact same plastic body as traditional magnetic-stripe banking cards. ICCs differ from plastics cards because they contain a very small microcontroller made up of a CPU, a program stored in ROM memory which defines the ICC behaviour, and a non-volatile memory, usually an EEPROM memory to store applicative data (see figure 1). The success of smart cards is mainly due to their security features. The physical security is assumed by all or parts of the following features :

- a silicon circuitry embedded into one single chip with none of the traditional address and data busses found in usual memory devices : multiple chips interconnected via busses would be too easy to spy upon
- some security logic within the single microchip controls the serial link and the associated address decoding function in order to restrict access from the external world to all or parts of the non-volatile memory
- the non-volatile memory is physically protected against attempts to get around the interface control logic by scrambling the memory patterns of addressing lines inside the chip
- address and data lines are also completely or partially buried at some deeper level of the silicon structure
- the ROM memory is protected against reverse-engineering by using ion-implanting rather than metal masking
- a number of silicon sensors are embedded into the chip to detect abnormal conditions such as high temperature, low voltage, low clock frequency, exposure to light, *etc.*

The logical security feature of smart cards is the ability of the operating system to control locally the access conditions to the data structures i.e. the verification of the security attributes attached to logical entities (memory zones, files, tables, *etc.*) and the maintenance of the security status obtained within a session. The definition of

ROM memory during the manufacturing phase implies that logical security cannot be modified. Moreover the card operating system can execute internally cryptographic algorithms for identification, authentication, integrity checking or confidentiality purposes [Sim92]. The following basic cryptographic services can be found in smart cards :

- DES encryption routine
- RSA and DSS digital signature generation and verification routines
- key-generation routines
- hash functions
- pseudo-random number generation

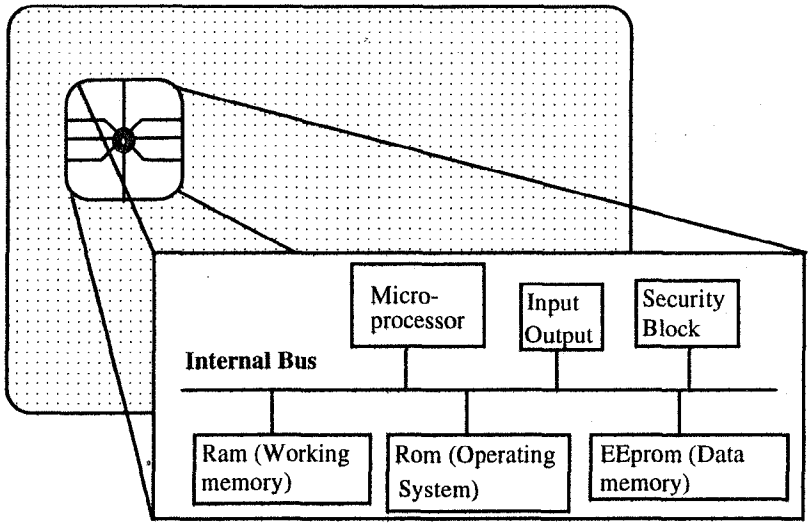

Figure 1 : Integrated circuit card architecture

The most sophisticated ICCs embed operating systems that are about like the operating systems for microcomputers. An ICC operating system is an executable code that resides in the ROM of the ICC microcontroller. Its main functions are to control access to the applicative data, provide communication with interface devices (IFDs), and maintain privacy and integrity of the applicative data. These features have been adequately used to widely spread the diffusion of ICCs issued from an authority such as a bank, a government department, or a corporation. ICC operating systems provide functions for managing data stored in structures such as files or SQL tables, and performing general purpose access to these structures, and sometimes specific applicative functions [PV94].

In fact, the term operating system is quite misused because the usual distinction between operating system software and application programs is very blurred. Today, the ability for an ICC operating system to load executable application programs is strictly limited. That operation can be done only before the issue of the card and no longer after. Thus, application programs are part of the operating system and cannot be updated or removed during the card life cycle. Nevertheless, the term operating system is widely used in the smart card area to refer to the ROM code, and it is usually called *mask*. In this paper, we propose a new mask — or operating system — for smart cards that we wish to be more closed to the traditional understanding of that term.

2 Card Applications Requirements

Service providers and end users increasingly request for new functions to be performed by ICCs. Loading several

and non-related applications into ICCs is the next challenge. The ICC should become a mobile computer belonging to the bearer. The bearer will be able to purchase services from many providers, and load specific provider data and functions into his ICC. Data should be issued from many applications and should be used in different ways depending on services. Data could have different structures, shared by several providers, and processed by specific provider functions. An ICC application should be a set of data and functions. Functions provide executable code to create and manage data and to define security architecture linked to data structures. As the cards are widely distributed, the card data have to be protected from an user to write or update data in place of another user. That can be done by cryptographic features performing user authentication and signature functions [GQU92] [Sim92].

The main evolution concerns new supplied features that imply new card life cycle. The figure 2 resumes the card life cycle evolution and the new supplied features appear in italic.

Intervening Parties	**Current ICCs**	**Future ICCs**
MANUFACTURER (once)	- Define ICC mask : HAL + data structures and users management commands	- Define ICC operating system : HAL + users management commands + *execution support environment*
CARD ISSUER (once)	- Add application specific functions - Create data structures - Create user certificates and access rights	- *Create service provider certificates*
SERVICE PROVIDERS (many times)	- Not defined	- *Are authenticated by the card* - *Create structure and data stored into the card* - *Create functions performing on those data and distribute them to card users with signatures* - *Provide keys to the card to control of the function signatures* - *Create user certificates and access rights*
CARD USERS (many times)	- Are authenticated by the card - Request the execution of commands according to their access rights - Get a response	- Are authenticated by the card - Request the execution of a service provider functions according to their access rights - *Provide the function code and its signature to the card* - *The card control the signature and execute the provided function* - Get a response

Figure 2 : Card life cycle evolution

These new trends point out requirements concerning operating system features. ICC hardware based on 8-bits microcontrollers has rapidly evolved during the past decade. New generations of ICC microcontrollers are being developed including a RISC-Based circuit [Pey94]. It will enable the design of more powerful operating system supporting a large variety of ICC applications. Nevertheless, current smart cards (8 bits ones) can be used to implement such sophisticated operating system that will be more efficient with the next generation of ICC microcontrollers.

Current mask are insufficient to provide services required by a non-predefined open multi-purpose ICC. They are always dedicated to a single type of data structures or to a single type of application. They cannot evolve in accordance with new services. Characteristically, operating systems should define an Hardware Abstraction Layer (HAL) to hide the implementation details of the data memory management and the communication protocol with IFDs. Mainly, an ICC operating system should offer a set of rules for supporting the implementation of an application as the relationships among its three components : data structures, functions performing on data and security architecture related to data structures. Moreover, it should provide mechanisms to enable sharing data

among applications. It should also enable the evolution of services all along the ICC life cycle. There are the key features of a multi-purpose and open operating system for ICCs.

3 New Directions for ICC Operating Systems

Mainly, it is necessary to extend the data stored into the ICC microcontroller to executable code that can be loaded all along the card life cycle. It is a limitation of current ICCs. The code is predefined by the card issuer to act as an operating system offering an HAL to users and service providers. This code is defined by the issuer and cannot enable ICCs evolution. Sometimes, it can include some specific functions dedicated to an application area, that implies a more restrictive use of the ICC. Service provider functions cannot reside permanently into the ICC for reason of memory space and evolutivity. Thus a solution is to provide to the ICC a function only at the time it is needed. Data should be stored in the card to ensure security. The functions use these data without disclosing any unnecessary data to the outside world. It is the key feature of ICC : the outside world obtains from the card only what it can get and no more, it cannot directly access to the ICC data.

Functions should not run onto the ICC operating system, they should be part of the operating system, because they must be strongly controlled. A secured *microkernel* operating system is the key concept to realize such an implementation, in that the ICC microkernel operating system is the central component of the system offering services well suited to perform the main tasks of the ICC[1]. It should provide the following functions :

- to load and unload functions performing on card data
- to secure this operation by providing cryptographic services to assume privacy, integrity and authenticity of functions [GQU92] by means of signature verification according to keys provided by the service provider
- to control access rights to verify the ability of a user requiring the execution of functions and to authenticate the user to be sure he is an authorized user
- to provide a secured operating environment to control each instruction of the function loaded into the ICC. A highly secured interpreter could be integrated to the ICC operating system to serve as an execution support, it should verify the address limitations for each instruction to be processed

With such operating system, the ICC should store data and cryptographic keys from different service providers. Service providers should deliver functions to all partners for whom they agree to use the ICC data. These functions should not be able to be forged and should be provided with information (or keys) to enable partners to be authenticated by the ICC operating system. By storing the functions outside ICCs, cards become more flexible and can evolve according to the outside world evolution.

The means of implementing such ICC operating system should be based on object-oriented technologies (see figure 3). The interpreter could be implemented as an object secure interpreter for which instance variables of objects are stored into the ICC non-volatile memory and methods are provided by the outside world at the execution time. It is necessary to define that intermediate level of method call, because execution of native code could directly access to data without control. A first version of the interpreter is developed by the second author as part of his PhD in computer science. That work can be related with implementation described in [Ben87] [Mir87]. Others future planned works are to define security protocols to control the distribution of the service provider functions, and to manage them like as other network objects by means of services such as naming, replication, life cycle, transaction consistency or licensing [OMG91].

---

[1] The term microkernel is assumed from distributed operating system researches, but we do not deal with the problems of distribution. In this paper, microkernel operating system refers to a small operating system (that can take place in the ICC ROM memory) performing the minimum set of operations to securely load, execute and download service providers' functions.

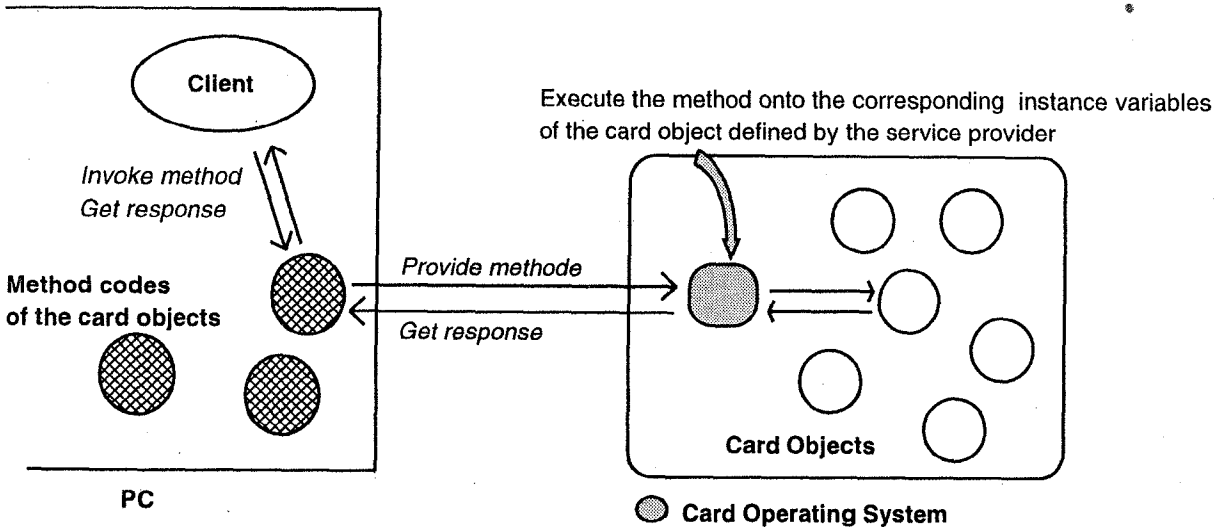

Figure 3 : Method invocation onto the smart card

4 Conclusion

ICCs could take many advantages of the object-oriented technologies for operating systems. Technical solutions proposed in this paper are timely defined for the next generation of ICC microcontroller and the new applications considered. Nevertheless, some basic assumptions of this approach could be implemented for current ICCs and enhance their functionalities. ICCs should become such as Personal Digital Assistants (PDA's). They should become tools for many people to purchase services, use these services all over the world, obtain connections to the growing number of data sources, and be identified as someone, or be represented, into electronic data highways and networks. This vision of the future of smart ICCs is related to the future of world-wide computing architectures [BRO93]. ICCs should integrate the next computers infrastructure where operating systems will be scattered among several machines and interoperable, and where object-oriented technologies will be key concepts for designing, distributing, accessing and processing data all over the world [OMG91].

Acknowledgements

We acknowledge the cooperation of Edouard Gordons, Thierry Peltier and Patrice Peyret for providing new ideas, constructive criticisms and information during many discussions. Also to Patrick George from which we receive continuous support.

References

[Ben87] J. K. Bennett. *The Design and Implementation of Distributed Smalltalk, in* proceedings of OOPSLA'87, special issue of SIGPLAN notices, vol. 22, n° 12, pp. 318-330, ACM Press, December 1987.

[Bro93] M.L. Brodie. *The promise of distributed computing and the challenges of legacy information systems, in* IFIP Transactions A-25, Interoperable Database Systems (DS-5), Elsevier Science Publishers B.V., North-Holland, pp. 1-31, 1993.

[GQU92] L. C. Guillou, J-J. Quisquater, and M. Ugon. *The Smart Card : A standardised Security Device Dedicated to Public Cryptology, in* Contemporary Cryptology, The science of Information Integrity, ed. G. Simmons, pp. 561-614, IEEE-Press, New York, U.S.A., 1992.

[Mir87] E. Miranda. *BrouHaHa - A Portable Smalltalk Interpreter, in* proceedings of OOPSLA'87, special issue of SIGPLAN notices, vol. 22, n° 12, pp. 354-365, ACM Press, December 1987.

[OMG91] Object Management Group. *OMG Common Object Request Broker Architecture: Architecture and Specification (CORBA)*, Revision 1.1, OMG Document Number 91.12.1, December 1991.

[Pey94] P. Peyret. *RISC-Based, Next-Generation Smart Card Microcontroller Chips, in* proceedings of CardTech'94, pp. 9-36, Washington D.C., U.S.A., April 1994.

[PV94] P. Paradinas and J.-J. Vandewalle. *A Personal and Portable Database Server : the CQL Card, in* proceedings of Application of Databases, ADB'94, Vadstena, Sweden, eds. W. Litwin and T. Risch, Lecture Notes in Computer Science, n° 819, pp. 444-457, Springer-Verlag, Berlin, Germany, June 1994.

[Sim92] *Contemporary Cryptology, The science of Information Integrity*, ed. G. Simmons, IEEE Press, New York, U.S.A., 1992.