Extending the Processing and Memory Capabilities of a Java Card

Susana Maria Ramirez Brey, Ms.C[1], Adonis Cesar Legón Campos, Ms.C[1]
[1]Universidad de las Ciencias Informáticas, La Habana, *smramirez@uci.cu*, *alegon@gmail.com*

Abstract– ***The smartcard chips tend to be very small, so these can be used as portable documents and meet some requirements related to cost, flexibility and energy. The processing and memory limitations that are associated with the reduced chip size, impact in the extensive use and applicability of smartcards, fundamentally for some application areas critical for the resources they need. To solve this problem it was designed and implemented a software platform that allows to exploit the computational and storage capacity of computers, in conjunction with smartcards. The platform allows the implementation and safely execution, of extension operations for applications that require resources that the card is not able to provide. Was performed implementations of some specific extension operations in different scenarios, that allowed demonstrate the effectiveness of the platform and also provide guidance for the development of other applications.***

Keywords-- ***extension, limitations, memory, platform, processing, smartcards***

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2015.1.1.008
ISBN: 13 978-0-9822896-8-6
ISSN: 2414-6668

Extendiendo las capacidades de procesamiento y memoria de una tarjeta Java Card

Susana Maria Ramirez Brey, Ms.C, Adonis Cesar Legón Campos, Ms.C
Universidad de las Ciencias Informáticas, La Habana, smramirez@uci.cu, alegon@gmail.com

Resumen – Los chips de las tarjetas inteligentes tienden a ser muy pequeños, para que estas pueden ser utilizadas como documentos portables y cumplan ciertos requerimientos relacionados con el coste, la flexibilidad y la energía. Las limitaciones de procesamiento y memoria que están asociadas al tamaño reducido del chip, impactan en el uso extensivo y aplicabilidad de las tarjetas inteligentes, fundamentalmente para algunas áreas de aplicación críticas por los recursos que necesitan. Para dar solución a esta problemática se diseñó e implementó una plataforma de software que permite explotar la capacidad computacional y de almacenamiento de los ordenadores, de conjunto con el de las tarjetas inteligentes. La plataforma permite la implementación y ejecución de manera segura, de operaciones de extensión para aplicaciones que requieran recursos que la tarjeta no es capaz de proveer. Se realizaron implementaciones de algunas operaciones de extensión específicas en diferentes escenarios, que permitieron demostrar la efectividad de la plataforma y además sirven de guía para el desarrollo de otras aplicaciones.

Palabras clave - extensión, limitaciones, memoria, plataforma, procesamiento, tarjetas inteligentes.

Abstract – The smartcard chips tend to be very small, so these can be used as portable documents and meet some requirements related to cost, flexibility and energy. The processing and memory limitations that are associated with the reduced chip size, impact in the extensive use and applicability of smartcards, fundamentally for some application areas critical for the resources they need. To solve this problem it was designed and implemented a software platform that allows to exploit the computational and storage capacity of computers, in conjunction with smartcards. The platform allows the implementation and safely execution, of extension operations for applications that require resources that the card is not able to provide. Was performed implementations of some specific extension operations in different scenarios, that allowed demonstrate the effectiveness of the platform and also provide guidance for the development of other applications.

Keyword- extension, limitations, memory, platform, processing, smartcards

I. INTRODUCCIÓN

Las tarjetas inteligentes han alcanzado un gran auge en los últimos años, esto se debe en gran medida a la seguridad tanto física como lógica que brindan estas tecnologías. Debido fundamentalmente a esta razón, ha ido en aumento el área de aplicación de las tarjetas inteligentes y son aplicadas ya no solo en el campo de la telefonía celular o en el comercio electrónico como en sus inicios, sino en áreas muy diversas como: transacciones seguras, identificación, firma digital, control de acceso, entre otras. Como una tendencia se ha observado también que varios países han comenzado a modernizar su documento de identificación nacional [1] hasta convertirlo en un documento de identificación electrónico (eID) con el uso de esta tecnología.

Los chips de las tarjetas inteligentes tienden a ser muy pequeños y por esto siempre existe un límite en las funcionalidades y los recursos que estos pueden manejar. Existen varias razones para las limitaciones del tamaño del chip, entre ellas se encuentran: el costo del chip que es proporcional al área de silicio usado y aunque de forma individual puede no ser mucho la producción a gran escala las hace sensible al coste; los requerimientos de las tarjetas inteligentes en cuanto a estrés y flexibilidad, que no son posibles alcanzar con chips muy grandes; y por último, que cuando el chip se hace mayor y más complejo, los requisitos de energía para su alimentación también crecen [2].

Dado el tamaño reducido del chip existen limitaciones del *hardware*, relacionadas fundamentalmente con las capacidades de memoria y de procesamiento de estos dispositivos. Son varios los trabajos que hacen referencias a estas limitaciones en los recursos de las tarjetas inteligentes [2–6]. Estas limitaciones del *hardware* de la tecnología impactan en gran medida en el uso extensivo y aplicabilidad de las tarjetas inteligentes, fundamentalmente para algunas áreas de aplicación [7].

La industria ha estado revisando temas de diseño y se han ido desarrollando algunos dispositivos con una configuración de *hardware* superior. A pesar de los avances que significa esta nueva generación de tarjetas inteligentes, el mercado aún es dominado por las generaciones anteriores, debido fundamentalmente a los costos de producción que implican los dispositivos con estas características

Se considera que no ha sido lo suficientemente explotada por la industria o la academia la idea de proveer soluciones de software a estas limitaciones [7]. Aunque en la revisión de la literatura se encuentran algunas soluciones de software que tienen como propósito ampliar las capacidades de memoria o procesamiento de las tarjetas inteligentes mediante el uso de ordenadores [4], [8], [9], ninguna se encuentra estandarizada ni es de propósito general, sino para áreas de aplicación específicas. Esto se debe en gran medida a que esta tecnología es controlada y desarrollada fundamentalmente por empresas productoras de tarjetas inteligentes.

II. BASES DE LA PLATAFORMA DE EXTENSIÓN

Tomando como base el modelo tradicional de desarrollo para tarjetas inteligentes *Java Card*, así como los estándares de comunicación ISO/IEC 7816 [10], PC/SC [11], y partiendo de la problemática de las limitantes de los recursos de

hardware de las tarjetas inteligentes con bajas prestaciones, se desarrolló una plataforma que permite la implementación y ejecución de operaciones para la extensión de las capacidades de procesamiento y memoria de aplicaciones *Java Card* en tarjetas inteligentes. Los principios que se definieron en el diseño de la plataforma son los siguientes:

1) La comunicación entre los componentes debe basarse en un modelo de comunicación síncrono (comando - respuesta) como se define en el estándar ISO/IEC 7816.

2) Para generar menor impacto en el desarrollo de nuevas aplicaciones, el terminal y la tarjeta inteligente deben abstraerse de la ejecución de operaciones de extensión fuera de la tarjeta inteligente.

3) Se deben implementar estándares de seguridad, que garanticen un ambiente de ejecución seguro fuera de la tarjeta inteligente.

4) Se debe contar con un ambiente conectado, que posea un buen enlace de red, de manera que la comunicación con el servidor web no constituya un costo adicional significativo.

III. Arquitectura de la plataforma de extensión

La arquitectura de la plataforma de extensión en su vista más abstracta es un modelo Cliente- Servidor. Este modelo está basado en el principio fundamental de que un cliente realiza peticiones a otro programa, el servidor, que le da respuesta.

Para la solución propuesta el componente *Java Card Applet Extension Proxy* que se encuentra en el terminal se comporta como cliente, manejando todas las peticiones que le llegan tanto de la tarjeta inteligente (con el componente *Java Card Applet Extension Client*) como del servidor web (con el *Extension Service*). Una representación de la arquitectura se puede ver en la Fig. # 1.

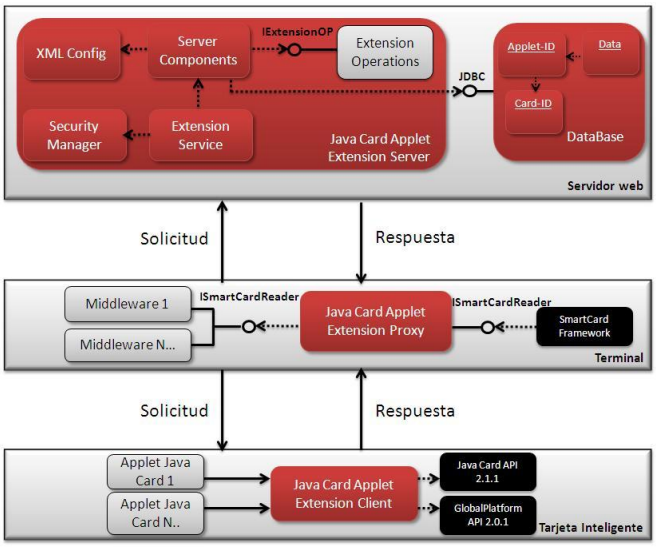

Fig. 1 Arquitectura de la Plataforma de desarrollo para la extensión de las capacidades de procesamiento y memoria de aplicaciones *Java Card* en tarjetas inteligentes.

En la plataforma también puede verse el uso de una arquitectura basada en componentes, donde se definen componentes funcionales o lógicos que exponen interfaces de comunicación bien definidas. Esto provee un nivel de abstracción mayor y se desacoplan las responsabilidades de todos los componentes, lográndose simplicidad en la ejecución de las operaciones de extensión para *applets Java Card*, que por sus características requieran mayor capacidad de memoria o procesamiento de la que la tarjeta es capaz de proveer.

Se destacan de color rojo los componentes que provee la plataforma para el desarrollo y ejecución de aplicaciones que requieran extender las capacidades de procesamiento y memoria de una tarjeta inteligente; y en gris aquellos componentes que deben ser creados por el desarrollador, y se integran a la plataforma en una solución para brindar un servicio final a un usuario. Los componentes que deben ser creados de manera independiente son el *middleware* y su *applet*, de igual manera que en el desarrollo tradicional, y la operación de extensión (*Extension Operation*) con la lógica que se quiera ejecutar fuera de la tarjeta inteligente. Una breve descripción de los componentes que provee la plataforma se brinda a continuación.

El componente *Java Card Applet Extension Client* (JCAEC) provee un conjunto de métodos a utilizar por el desarrollador *Java Card* como conformar una petición de extensión con los datos necesarios para realizarla fuera de la tarjeta, o procesar el resultado final de una operación de extensión.

El componente *Java Card Applet Extension Proxy* (JCAEP) es el encargado de comunicarse en tiempo de ejecución con el *applet* mediante el lector de tarjetas y con el servicio publicado en el servidor web mediante un mensaje SOAP. Su principal responsabilidad es servir de intermediario para dirigir las transmisiones de comandos y respuestas APDU[1] entre los componentes de la plataforma.

El componente *Java Card Applet Extension Server* (JCAES) integra varios componentes cuya lógica se describe a continuación.

Extension Service: Es un servicio web SOAP genérico, que se utiliza para la ejecución de una operación de extensión publicada en el servidor.

Server Components: Provee un conjunto de clases para funcionalidades auxiliares que utiliza el *Extension Service* durante la ejecución de una operación de extensión. Entre estas funcionalidades se encuentra el acceso a la base de datos para las operaciones de almacenamiento o consulta de datos, así como la implementación de la carga dinámica de las operaciones de extensión previamente implementadas y publicadas en el servidor.

[1] Unidad de datos del protocolo de aplicación (traducido de las siglas en inglés APDU: *Application protocol data unit*).

Extension Operation: Operaciones que implementan una lógica determinada y deben ser implementadas y publicadas en el servidor a priori, para que sea posible su invocación. Pueden ser de almacenamiento y/o consulta de información, o de ejecución de operaciones de alto costo computacional que sean más factible de realizar fuera de la tarjeta inteligente.

Security Manager: Se encarga de la implementación del canal seguro de comunicación entre la tarjeta y el servidor.

DataBase: Estructura de datos para el almacenamiento de información asociada a una aplicación de una tarjeta dada. Por lo básico de los tipos de datos que se manejan y su conversión en arreglos de *bytes* para su salida o entrada de esta, no es necesaria la definición de tipos de datos complejos, sino una estructura simple compuesta por propiedades (clave – valor) que permita almacenar variables y sus valores. Esta estructura se encuentra agrupada por tarjeta y aplicaciones.

IV. Despliegue de la plataforma

Para el despliegue de la plataforma se utilizaron tres nodos de ejecución en ambientes diferentes: la tarjeta inteligente, un ordenador que funciona como terminal de servicio y un servidor *Java Apache TomCat* con *Axis 2*, que expone un servicio web SOAP.

En la tarjeta se encuentran instalados el JCAEC de conjunto con los *applets* que requieren su uso. En el terminal se encuentran el JCAEP y los *middlewares* para el acceso a los *applets* que se encuentran en la tarjeta inteligente. Por último, en el servidor web van publicados los componentes del JCAES que se exponen como un servicio web. Por la simplicidad de los datos que se manejan en la base de datos esta se encuentra alojada en el mismo servidor web.

La comunicación entre el lector de tarjetas y el *applet* en la tarjeta inteligente se realiza mediante el envío de comandos APDU. Los lectores se conectan a cada una de las terminales de servicio mediante el puerto USB y utilizan el protocolo PC/SC para la comunicación con estos ordenadores. Para la comunicación con el servicio web desplegado en el servidor TomCat se utiliza el protocolo SOAP.

Las tarjetas inteligentes deben poseer un mínimo de 5 kB de memoria libre en EEPROM[2] (para la instalación del *Java Card Applet Extension Client* y el *applet*). Además se requiere que cuenten con el sistema operativo *Java Card* 2.1.1 o una versión superior, así como *GlobalPlatform* 2.0.1 o superior. El terminal de servicio debe contar con un lector de tarjetas USB con los drivers PC/SC instalados.

V. Resultados y discusión

Se realizó la implementación de operaciones de extensión sobre la plataforma desarrollada, que permitieron valorar la extensión de las capacidades de procesamiento y memoria de tarjetas inteligentes *Java Card* en diferentes escenarios. Con este propósito se utilizaron tarjetas inteligentes del modelo Gemplus GemXpresso Pro R3.2 E32 PK [12].

Estas tarjetas poseen la máquina virtual y el API *Java Card* en su versión 2.1.1. Además tienen soporte para los algoritmos criptográficos DES, 3DES, SHA, MD5, RSA 1024, RSA 2048, DAP. No soportan el tipo de dato *integer* de 32 *bits* y los recursos de *hardware* que poseen son los siguientes.

- Tamaño del *buffer*: 256 *bytes* (+ 5 *bytes* de encabezado del comando)
- Tamaño de pila persistente (ROM): 29,6 kB.
- Tamaño de pila transitoria (RAM) disponible para *applets*: 1,2 kB.
- EEPROM disponible: 29,7 kB

Para el servidor web se utilizó un ordenador Dell con un procesador Intel Core 2 Quad CPU Q9400 y 2GB de memoria RAM. Todos los experimentos se realizaron en un ambiente controlado con un buen enlace de red, para que no fuese un costo adicional elevado la comunicación con el servidor web.

Como primer escenario de prueba se seleccionó una operación aritmética simple como lo es el factorial de un número entero positivo. A pesar de que es un algoritmo sencillo de implementar, las limitaciones de esta operación para su implementación en *Java Card* usando las tarjetas GemXpresso Pro R3.2 E32 PK radican, en primer lugar, en que no es posible calcular un factorial que sea expresable en un número mayor a un entero de 16-*bits* (*short*), debido a que este el tipo de dato mayor que soporta la tarjeta. Además de la limitación de los tipos de datos y la memoria que estos ocupan, también se encuentra la limitación asociada al costo computacional e incluso de memoria volátil (RAM) dentro de la tarjeta, debido a que el factorial de un número crece exponencialmente muy rápido.

Con la ejecución de la operación factorial en el servidor, es posible obtener el factorial de números expresables en 32 *bits* y 64 *bits*. En la Tabla # 1 se muestran los resultados obtenidos en pruebas donde se hizo el cálculo del factorial tanto en la tarjeta inteligente como en el servidor. Como es posible observar, en el caso de las operaciones que se pueden ejecutar tanto en la tarjeta como en el servidor, el tiempo de respuesta de las operaciones en la tarjeta es significativamente menor. Este resultado se debe fundamentalmente al tiempo asociado al establecimiento de una conexión con el servidor web y a que el flujo de comunicación se complejiza, teniendo que regresar este resultado a la tarjeta, para finalmente ser devuelto al *middleware* en el terminal.

Tabla 1. Resultados en el cálculo del factorial

Operación	Tiempo en el servidor	Tiempo en la tarjeta	Instrucciones en servidor	Instrucciones en la tarjeta
Factorial(5)	949 ms	21 ms	9 instrucciones	14 instrucciones
Factorial(10)	969 ms	-		
Factorial(20)	1 s 4 ms	-		

Respecto al número de instrucciones que se requieren para esta ejecución fuera de la tarjeta usando la plataforma, en la

[2] Memoria de solo lectura programable y borrable eléctricamente (traducido de las siglas en inglés EEPROM: *Electrically Erasable Programmable Read-Only Memory*)

Tabla # 1 se observa que no varía mucho con respecto a las que requiere la tarjeta de manera tradicional. Este resultado se debe a que los componentes implementados como parte de la plataforma fueron diseñados teniendo en cuenta que se minimizase el esfuerzo que implicaba el procesamiento fuera de la tarjeta inteligente. Esta es una fortaleza de la plataforma, debido a que es importante para los desarrolladores que sea mínimo el impacto que genera el desarrollo de nuevas aplicaciones que requieran operaciones de extensión.

Como otra de las implementaciones realizadas se encuentra la generación de llaves usando algoritmos como ECDH y RSA. ECDH está basado en la utilización de Curvas Elípticas [13] de conjunto con las operaciones de *Diffie-Hellman*, posibilitando generar claves más cortas dado el aumento significativo de la complejidad de las operaciones matemáticas que se realizan. Este algoritmo de cifrado es ampliamente usado en aplicaciones relacionadas con los documentos electrónicos de viaje, para el control de acceso extendido responsable de la protección de los datos biométricos de estos documentos.

Debido a que las tarjetas GemXpresso Pro R3.2 E32 PK no soportan este algoritmo de cifrado, no es posible generar las llaves de este tipo dentro de la tarjeta como se puede requerir. A través de la implementación de una operación de extensión sobre la plataforma, se logró la generación de llaves de sesión para la tarjeta, usando el algoritmo ECDH de 128, 192 y 256 *bits*. De esta manera se demuestra que es posible realizar un Control de Acceso Extendido versión 2 europea, usando tarjetas con bajas prestaciones que no soportan Curvas Elípticas.

Aunque las tarjetas de prueba soportan el algoritmo RSA, es solamente para llaves con tamaños hasta 2048 *bits*, no pudiéndose generar llaves de mayor tamaño dentro de la tarjeta inteligente, las cuales son significativamente más seguras. Se implementó una operación de extensión que permitió generar un par de llaves RSA de 3072 y 4096 *bits* en el servidor web, usando la plataforma de extensión. De esta manera se solventa la limitación de la generación de llaves de mayor tamaño en este modelo de tarjetas. En la Tabla # 2 se resumen las operaciones implementadas y los resultados en cuanto a los tiempos de respuesta de la ejecución de estas operaciones en el servidor.

Tabla 2. Resultados en la generación de llaves ECDH y RSA

Operación	Tamaño	Tiempo en el servidor
ECDH Session Keys	128 bits	987 ms
ECDH Session Keys	192 bits	994 ms
ECDH Session Keys	256 bits	1 s
RSA keys	3072 bits	8 s 299 ms
RSA keys	4092 bits	12 s 850 ms

En cuanto a la extensión de la capacidad de almacenamiento, la memoria EEPROM disponible en la tarjetas GemXpresso Pro R3.2 E32 PK es de tan solo 29.7 kB, luego de ser personalizada por el emisor. Este tamaño limita el número de aplicaciones que pueden ser instaladas en la tarjeta, así como las variables e información que estas pueden manejar. La plataforma de extensión por su parte, provee un mecanismo para el almacenamiento en una base de datos relacional, donde el tamaño está en el orden de los *terabytes* y está asociado a los recursos del servidor. Mediante el almacenamiento y la recuperación de datos, se evidencia que también es posible extender la capacidad de memoria de las tarjetas inteligentes *Java Card*. Esta posibilidad ofrece ventajas en diversos escenarios, como por ejemplo donde se necesite que los datos de las aplicaciones estén disponibles para realizar un backup en caso de pérdida de la tarjeta inteligente.

En la Tabla # 3 se resumen las características seleccionadas en los escenarios descritos con anterioridad, que evidencian la extensión de las capacidades de procesamiento y memoria de las tarjetas inteligentes Gemplus GemXpresso Pro R3.2 E32 PK.

Tabla 3. Extensión de las capacidades de procesamiento y memoria de las tarjetas Gemplus GemXpresso Pro R3.2 E32 PK

Características	Modelo tradicional de desarrollo	Implementación de operaciones de extensión
Tipos de datos soportados	boolean, byte, short	boolean, byte, short, **char, double, float, int, long, BigNumbers**
Algoritmos criptográficos soportados	DES, 3DES, SHA, MD5, RSA 1024, RSA 2048, DAP	DES, 3DES, SHA, MD5, RSA 1024, RSA 2048, **RSA 4096**, DAP, **ECDH (posibilidad de otras implementaciones)**
Capacidad de almacenamiento	Hasta 29.7 kB disponible en memoria EEPROM	**En el orden de los terabytes** (según los recursos del servidor)

VI. CONCLUSIONES

La plataforma de extensión desarrollada permite la implementación y ejecución de operaciones para el almacenamiento de datos o la ejecución de algoritmos costosos fuera de la tarjeta inteligente, para lo cual fueron concebidos componentes con responsabilidades bien definidas y con el objetivo de minimizar el desarrollo que implica la ejecución de nuevas operaciones de extensión fuera de la tarjeta inteligente. El componente de la tarjeta (JCAEC) permite abstraer al desarrollador de la codificación de las operaciones necesarias para la solicitud de una operación de extensión y su posterior procesamiento en la tarjeta. El componente en el terminal (JCAEP) sirve de intermediario de la comunicación entre los componentes de la plataforma, abstrayendo al *middleware* en

el terminal de la ejecución de alguna operación fuera de la tarjeta. Por su parte, el *Extension Service* es un servicio web genérico que permite la ejecución de las operaciones de extensión previamente implementadas y publicadas.

Las tecnologías y herramientas seleccionadas para el desarrollo de la plataforma brindan la posibilidad de ser utilizada en los sistemas operativos Linux y Windows. La independencia del servidor web facilita una posible migración.

La primera versión de la plataforma fue sometida a pruebas de software que permitieron la corrección de algunas fallas y demostrar su capacidad de procesar los comandos recibidos, así como ejecutar operaciones de extensión para los comandos que así lo requirieran de forma satisfactoria. La implementación de operaciones de extensión sobre la plataforma permitió evidenciar la extensión de las capacidades de procesamiento y memoria de las tarjetas inteligentes Java Card Gemplus GemXpresso Pro R3.2 E32 PK en diferentes escenarios, sirviendo de guía además para el desarrollo de otras aplicaciones de este tipo con el uso de la plataforma.

REFERENCIAS

[1] A. C. Legón, S. M. Ramírez, J. Landrian, D. Almeida, A. Surós, and F. A. Carratala, "Sistema de gestión de servicios para documentos de identificación nacional electrónicos," in *XI Seminario Iberoamericano de Seguridad en las Tecnologías de la Información. XV Convención Informática*, 2013.

[2] K. E. Mayes and K. Markantonakis, *Smart cards, Tokens, Security and Applications*. London, UK: Springer, 2008.

[3] C. Bobineau, L. Bouganim, P. Pucheral, and P. Valduriez, "PicoDBMS : Scaling down Database Techniques for the Smartcard," *The VLDB Journal*, vol. 10, no. 2–3, pp. 120–132, 2001.

[4] C. H. Cap, N. Maibaum, and L. Heyden, "Extending the data storage capabilities of a Java-based smartcard," in *Computers and Communications, 2001. Proceedings. Sixth IEEE Symposium on*, 2001, pp. 680–685.

[5] W. Rankl, W. Effing, and K. Cox, *Smart card handbook*, Fourth Edi. John Wiley & Sons, 2010.

[6] Smart Card Alliance, "Smart Card Technology in US Healthcare: Frequently Asked Questions," no. September. 2012.

[7] S. M. Ramírez, A. C. Legón, and Y. Valle, "Modelo para la extensión de las capacidades de procesamiento y memoria de tarjetas inteligentes Java Card," *Revista Cubana de Ciencias Informáticas*, vol. 8, no. 1, pp. 112–126, 2014.

[8] M. Blaze, "High-Bandwidth Encryption with Low-Bandwidth," in *Proceedings of the Fast Software Encryption Workshop*, 1996, pp. 33–40.

[9] M. Blaze, J. Feigenbaum, and M. Naor, "A formal treatment of remotely keyed encryption," in *Proceedings of the 10th annual ACM-SIAM symposium on Discrete algorithms*, 1999, pp. 868–869.

[10] ISO/IEC, "ISO/IEC 7816. Identification cards — Integrated circuit cards-Part 4: Organization, security and commands for interchange." 2005.

[11] PC/SC Workgroup, "PC/SC Specification Version 2.01.11 Release." (En linea), 2013.

[12] Gemalto, "GemXpresso Pro R3.x. Reference Manual," France, 2004.

[13] Federal Office for Information Security, "Elliptic Curve Cryptography versión 2.0," Bonn, Germany, 2012.