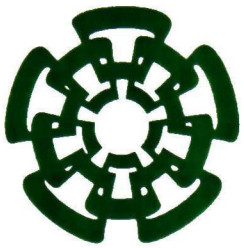
XX(11 6562.1)



CINVESTAV

Centro de Investigación y de Estudios Avanzados del I.P.N.
Unidad Guadalajara

2PSW: Algoritmo de Escritura Segura para Tarjetas Inteligentes Sin Contacto

Tesis que presenta:
Carlos Alberto Franco Reboreda

para obtener el grado de:
Maestro en Ciencias

en la especialidad de:
Ingeniería Eléctrica

Directores de Tesis
Dr. Félix Francisco Ramos Corchado
Dr. Ricardo Raúl Jacinto Montes

Guadalajara, Jal., febrero de 2004.

2PSW: Algoritmo de Escritura Segura para Tarjetas Inteligentes Sin Contacto

Tesis de Maestría en Ciencias
Ingeniería Eléctrica

Por:
Carlos Alberto Franco Reboreda
Ingeniero en Computación
Universidad de Guadalajara 1994-1998

Becario del CONACyT, expediente no. **129206**

Directores de Tesis
Dr. Félix Francisco Ramos Corchado
Dr. Ricardo Raúl Jacinto Montes

CINVESTAV del IPN Unidad Guadalajara, febrero de 2004.

Agradecimientos

A Dios, por darme la gracia de la vida, por los dones que me fueron concedidos y por haberme bendecido con salud todos estos años.

A mi madre, por su ejemplo de vida y apoyo incondicional.

A Luis Alejando, Rosa María, Luz Alicia, Mónica, Carmen y Alejandro, por su sincera y desinteresada amistad, que está a prueba de todo y es totalmente correspondida.

A Luis Alberto, que si bien no es mi hermano de sangre, lo es por elección propia.

A Rosana, el amor verdadero en mi vida, por compartir y hacer conmigo en estos años, los que hasta hoy son los días más felices de mi existencia.

A mis directores de Tesis, que sin su apoyo y gestión no hubiera sido posible concluir este proyecto.

Al CINVESTAV, por la intensa labor de promoción y superación de la enseñanza en diversas áreas del conocimiento científico que hacen posible elevar el nivel de vida de los mexicanos e impulsar el desarrollo de este país.

Al CONACYT, por su aporte para incrementar la capacidad científica y tecnológica de México y el apoyo económico que por su conducto he recibido para cursar mis estudios de maestría.

2PSW: Algoritmo de Escritura Segura para Tarjetas Inteligentes Sin Contacto

Resumen

Las tarjetas inteligentes son ya parte del modo de vida y la cultura de muchos de los países del mundo. Su aplicación en todas las áreas en donde se haga necesaria la identificación automática las hace de una versatilidad poco igualable por otros dispositivos actuales.

Dentro de las aplicaciones más comunes se encuentran los sistemas de identificación, sistemas de control de acceso a edificios, transacciones en tarjetas de crédito, sistemas de monedero electrónico, sistemas de recompensas a clientes frecuentes, prepago de servicios, almacenamiento de expedientes médicos de pacientes en forma segura, mejoramiento de la seguridad en la telefonía inalámbrica o en la prevención de accesos no autorizados a señales de cable o satélite.

En la arquitectura electrónica de una tarjeta inteligente, de acuerdo al estándar MIFARE, existe una zona de almacenamiento de datos con confirmación en la actualización conocida como monedero electrónico. En esta zona se almacenan hasta 4 bytes de manera confiable, basándose en un mecanismo de confirmación por hardware. Esta zona está orientada al manejo de puntos, por lo que puede no resultar adecuada para otras aplicaciones.

Un problema recurrente en la tecnología de tarjetas sin contacto, sin importar el fabricante de las mismas, es la poca confiabilidad que se tiene cuando se requiere la escritura de datos complejos en zonas de memoria diferentes a la zona del monedero electrónico, lo cual limita de manera importante las aplicaciones en las que son utilizadas, impactando de manera directa su utilización masiva.

La propuesta de este trabajo de tesis pretende contribuir al aumento en la confiabilidad de las transacciones de escritura realizadas en zonas de memoria no seguras de las tarjetas inteligentes. Esto mediante la propuesta y prueba formal de un mecanismo de software basado en un protocolo de commit a dos fases que funciona de manera similar al mecanismo de transacciones seguras implementadas en hardware por el sistema de monedero electrónico.

Este trabajo realizado se denomina 2PSW, que es un algoritmo de escritura segura a dos fases para tarjetas inteligentes sin contacto.

Índice General

Resumen *1*

Índice de Figuras *7*

Capítulo 1. Objetivo y Definición de la Tesis *9*

1.1 Objetivo **9**
1.1.1 Alcance 9
1.1.2 Descripción 9

1.2 Motivación **10**

1.3 Estructura de la Tesis **11**

1.4 Conclusiones del capítulo **11**

Capítulo 2. Estado del Arte *13*

2.1 Introducción **13**

2.2 Historia de las Smartcards **16**

2.3 Estándares **17**
2.3.1 ISO 18
2.3.1.1 Estandarización Internacional de las Smartcards 20
2.3.2 MIFARE 20
2.3.2.1 Estructura de Memoria de la Tarjeta Mifare 21
2.3.2.2 Condiciones de Acceso 22
2.3.2.3 Bloques de Valores y Bloques de Datos 22
2.3.2.4 Claves MIFARE 22

2.4 Tipos de Tarjetas **23**
2.4.1 De Relieve 23
2.4.2 Banda Magnética 23
2.4.3 Smartcards 23
2.4.3.1 Tarjetas de Memoria 24
2.4.3.2 Tarjetas de Microprocesador 24
2.4.3.3 Tarjetas de Coprocesador Criptográfico 24
2.4.3.4 Tarjetas de Proximidad 24
2.4.3.5 Tarjetas de Memoria Óptica 25

2.5 Propiedades Funcionales de las Smartcards **25**
2.5.1 Propiedades Físicas y Eléctricas 25
2.5.2 Sistema Operativo 26
2.5.2.1 MULTOS 26
2.5.3 Tecnología de Java Cards 27
2.5.4 Capacidades Criptográficas 28
2.5.5 Transmisión de Datos 28
2.5.6 Transmisión Inductiva 31
2.5.7 Transferencia de Energía 31
2.5.8 Transferencia de Datos 32
2.5.9 Características de Seguridad 32
2.5.10 Conjunto de Instrucciones 33
2.6 Conclusiones del Capítulo **34**

Capítulo 3. Descripción del problema y Solución Desarrollada *35*
3.1 Contexto General del Trabajo de Tesis **35**
3.2 Transacciones **35**
3.3 Protocolo de Commit a 2 fases **36**
3.4 Transacciones en Tarjetas Inteligentes **38**
3.4.1 Consistencia 38
3.4.2 Aislamiento 39
3.4.3 Durabilidad 39
3.4.4 Atomicidad 39
3.5 Soporte electrónico de las Tarjetas Inteligentes **40**
3.5.1 Estructura de memoria de la Tarjeta 40
3.5.2 Memoria Flash 42
3.5.3 Zona de Monedero Electrónico 43
3.5.4 Procesador 43
3.6 Descripción del Problema **44**
3.7 Solución propuesta **45**
3.7.1 Algoritmo 2PSW 45
3.8 Prueba del Algoritmo **47**
3.8.1 Limitaciones Teóricas 48
3.9 Conclusiones del capítulo **48**

Capítulo 4. Pruebas y Resultados *51*

4.1 Formalización del protocolo **51**

4.1.1 Estelle 51

4.1.1.1 Modelado en ESTELLE 52

4.1.1.2 Introducción al lenguaje ESTELLE 52

4.1.1.3 Análisis en Estelle 54

4.1.2 Máquina de estados 54

4.1.3 Arquitectura 58

4.2 Resultados **59**

4.2.1 Pruebas experimentales 59

4.2.2 Ambiente de experimentación 60

4.2.3 Evaluación inicial 61

4.2.4 Evaluación del 2PSW 62

4.2.5 Comparación de resultados 62

4.3 Conclusiones del Capítulo **64**

Capítulo 5. Conclusiones y Propuestas de Trabajo Posterior *67*

Referencias Bibliográficas *71*

Anexos *75*

Anexo 1. Implementación del Protocolo 2PSW en Código Estelle Extendido (Ver. 4.16) **75**

Índice de Figuras

Figura 1. Arquitectura típica de una tarjeta de contacto con lógica de seguridad incluida 14

Figura 2. Arquitectura típica de una tarjeta de microprocesador con coprocesador e interfase sin contacto 15

Figura 3. Estructura general y organización de los grupos de trabajo ISO/IEC involucrados en el desarrollo de estándares internacionales para smartcards 19

Figura 4. Estructura general y jerarquías de los diversos grupos de trabajo europeos para la estandarización de las smartcards 19

Figura 5. Dimensiones físicas de una smartcard 26

Figura 6. Transferencia necesaria de energía y datos entre la terminal y la tarjeta sin contacto 30

Figura 7. Transferencia inductiva para energizar una smarcard sin contacto 31

Figura 8. Diagrama del protocolo de Commit a dos fases 37

Figura 9. Mapa de memoria típico de una smartcard 42

Figura 10. Pseudocódigo del Algoritmo 2PSW 46

Figura 11. Máquina de estados del protocolo 2PSW 55

Figura 12. Análisis de las primitivas de comunicación del protocolo 2PSW 56

Figura 13. Arquitectura del protocolo 2PSW..... 58

Figura 14. Diagrama del laboratorio de pruebas utilizado..... 59

Figura 15. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 1A..... 62

Figura 16. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 1B..... 63

Figura 17. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 2A..... 63

Figura 18. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 2B..... 64

Figura 19. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 2C..... 64

Capítulo 1. Objetivo y Definición de la Tesis

1.1 Objetivo

El objetivo de este trabajo de tesis es aumentar la confiabilidad de las transacciones de escritura realizadas en zonas de memoria no seguras de las tarjetas inteligentes.

1.1.1 Alcance

El alcance de esta tesis cubre el desarrollo de un algoritmo que permita el intercambio confiable de información. La información que se busca proteger es información estructurada, que no puede ser almacenada en el área de monedero electrónico (e-purse) que se encuentra normalmente como parte de la arquitectura de una tarjeta inteligente.

El algoritmo se presenta con una prueba formal, además se desarrolló un script para realizar diversas pruebas del funcionamiento del algoritmo propuesto, obteniendo con ello muestras de su desempeño en escenarios reales.

1.1.2 Descripción

Las tarjetas inteligentes sin contacto son un elemento con cada vez mayor presencia en la vida cotidiana. Su aplicación en todas las áreas en donde se haga necesaria la identificación automática las hace de una versatilidad poco igualable por otros dispositivos actuales. Dentro de las aplicaciones más comunes se encuentra la de monedero electrónico, la cual es posible por el grado tan alto de seguridad que las tarjetas inteligentes ofrecen.

En la arquitectura electrónica de una tarjeta inteligente, de acuerdo al estándar MIFARE, existe una zona de almacenamiento de datos con confirmación en la actualización conocida como e-purse. En esta zona se almacenan hasta 4 bytes de manera confiable, basándose en un mecanismo de confirmación por hardware. Esta zona está orientada al manejo de puntos, por lo que puede no resultar adecuada para otras aplicaciones.

Las aplicaciones que requieran más de los 4 bytes de información y con otro tipo de representación que los puntos deben usar las zonas de memoria generales, las cuales no cuentan con un mecanismo de confirmación de la actualización.

Esta tesis presenta la propuesta y prueba semiformal de un mecanismo de software basado en un protocolo commit a dos fases que funciona de manera similar al mecanismo de transacciones seguras implementadas en hardware por el sistema de monedero electrónico.

1.2 Motivación

Las tarjetas inteligentes son ya parte del modo de vida y la cultura muchos de los países del mundo. Son utilizadas para una gran variedad de aplicaciones: sistemas de identificación, sistemas de control de acceso a edificios, transacciones en tarjetas de crédito, sistemas de monedero electrónico, sistemas de recompensas a clientes frecuentes, prepago de servicios, almacenamiento de expedientes médicos de pacientes en forma segura, mejoramiento de la seguridad en la telefonía inalámbrica o en la prevención de accesos no autorizados a señales de cable o satélite.

La explosión de Internet, la comunicación digital inalámbrica así como el rápido surgimiento del mercado de negocios electrónicos no sólo abren nuevas avenidas para el comercio, sino que también generan vastas oportunidades en la industria para contactar a sus clientes e introducir nuevos valores agregados a los productos o servicios. Lo anterior, ha incrementado la necesidad de implementar medidas de seguridad en las aplicaciones que realizan transacciones electrónicas con tarjetas inteligentes.

Recientemente, con los avances en la tecnología de los chips y la criptografía moderna, las tarjetas inteligentes se han vuelto más poderosas. Obtener información de una tarjeta inteligente requiere la posesión física de la tarjeta, además de conocimiento a detalle de software y hardware de la misma, así como equipo adicional.

Las características de seguridad de las tarjetas inteligentes son fortalecidas por funciones criptográficas. Los datos almacenados en la tarjeta pueden ser encriptados para salvaguardar su privacidad en la memoria física, y el intercambio de datos entre la tarjeta y el mundo exterior puede ser firmado y encriptado.

Adicionalmente, acceder a una tarjeta inteligente usualmente requiere que el poseedor de la tarjeta ingrese un número de identificación personal, lo cual previene que la tarjeta sea utilizada por una persona no autorizada. En general es mucho más difícil irrumpir en una tarjeta inteligente que en una computadora tradicional de escritorio [Carlson 99].

Sin embargo, a pesar de los avances que se tienen en cuestiones de seguridad, las tarjetas inteligentes han presentado ciertos problemas de confiabilidad cuando se emplean en ambientes hostiles, como ambientes industriales, con interferencia o ruidosos, donde la confiabilidad en las operaciones de lectura o escritura realizadas con ellas se reduce notablemente [Weikmann 98].

La motivación de este trabajo de tesis es contribuir con soluciones formales que incrementen la confiabilidad de las operaciones sobre tarjetas inteligentes que son realizadas en cualquier ambiente, incluyendo los ambientes hostiles y ruidosos.

1.3 Estructura de la Tesis

Esta tesis está conformada por cinco capítulos, anexos, y referencias bibliográficas, estructuradas de la siguiente forma:

El primer capítulo presenta una breve introducción a las tarjetas inteligentes, situándonos en el contexto particular a tratar en este trabajo de tesis, definiendo el objetivo y la importancia del problema y la solución propuesta.

El segundo capítulo presenta el estado del arte de las tarjetas inteligentes, explica los conceptos básicos necesarios para desarrollar el trabajo de investigación.

En el tercer capítulo se presenta la descripción del problema y la solución desarrollada en este trabajo de tesis.

En el capítulo cuarto se presentan las pruebas y resultados de la implementación de la solución diseñada.

En el quinto capítulo se presentan las conclusiones del trabajo de tesis y las propuestas de trabajo posterior en esta investigación.

1.4 Conclusiones del capítulo

En este primer capítulo se describió el objetivo de la tesis, definiendo su alcance y dándole al trabajo propuesto un contexto en el que puede ser evaluado.

La importancia de este tema es evidente, la proliferación en el uso de estas tarjetas para aplicaciones como el control de acceso, control de asistencia, monedero electrónico, transmisión segura de datos sin utilizar una red, sistemas de inventarios, pasaportes, licencias de manejo, etc., son fiel parámetro de medición de la importancia de las smartcards.

De igual forma, se hizo una breve descripción del funcionamiento general de la zona de monedero electrónico de las smartcards, denotando la existencia del protocolo base del trabajo propuesto.

El contenido de este capítulo permite tener un panorama general del problema a tratar, presenta la estructura del trabajo de tesis, esboza el acercamiento a la solución planteada y a los resultados obtenidos.

Capítulo 2. Estado del Arte

2.1 Introducción

Del mismo tamaño que una tarjeta de crédito, una tarjeta inteligente es una pequeña computadora portátil, resistente a la intemperie pero sin interfaz hombre – máquina, que almacena y procesa información entre los circuitos electrónicos situados en el cuerpo de la tarjeta y una terminal. El resultado de la interacción entre la terminal y la tarjeta inteligente provee la funcionalidad de la aplicación en cuestión.

El interés en las tarjetas inteligentes es resultado de los beneficios que provee: poder computacional integrado, seguridad, portabilidad y facilidad de uso. Sin embargo, la razón real del uso de un chip en una tarjeta plástica es la seguridad: las tarjetas inteligentes y los circuitos integrados utilizados en ellas tienen diferentes características que les permiten no sólo almacenar datos en forma segura, sino también asegurar los datos almacenados en otros sistemas de cómputo [Bellare 95a].

El universo de las aplicaciones de las tarjetas inteligentes es bastante amplio, pueden ser utilizadas como mecanismos de identificación automática, como monederos electrónicos, como mecanismos para intercambio de información, en aplicaciones bancarias, como control de acceso, como control de asistencia. Estas aplicaciones tienen como característica principal su necesidad por la seguridad, la privacidad y la confianza en la información guardada en la tarjeta.

Un esquema de seguridad debe considerar diferentes niveles, un nivel de autentificación que asegura la membresía del elemento al sistema y un elemento de encripción que garantiza la privacidad y la confiabilidad de la información. Este punto implica asegurar la no intrusión y la integridad de la información mientras esta se encuentra en tránsito.

El hecho de que una smartcard sea en realidad un circuito electrónico compuesto por microprocesadores y memoria, hacen posible el que un procesador se dedique exclusivamente a efectuar algoritmos de autentificación y encripción que aseguran la privacidad y la seguridad en el acceso a la tarjeta. El empleo de tecnologías EEPROM para las memorias permiten que las tarjetas puedan almacenar perennemente una cierta cantidad de información (512 bits, 1kbit,...), de manera confiable ya que pueden soportar más de 30,000 ciclos de escrituras lo que representa una vida útil de por lo menos 10 años para la mayoría de las aplicaciones [Dhem 96] [Carlson 99].

Además la seguridad en la tarjeta no radica en el algoritmo utilizado sino en las llaves que se utilizan para autentificación y encripción de la información. Si las llaves son suficientemente grandes la potencia de cálculo necesaria para romper la seguridad es tanta que resulta incosteable el realizar el esfuerzo para quebrantar la seguridad. Sin embargo los esfuerzos para incrementar la seguridad no paran y cada día existen nuevas técnicas que aumentan el nivel de seguridad en la información intercambiada.

Considerando solamente el método de comunicación, existen dos tipos de tarjetas inteligentes:

1. Tarjetas de contacto.

Las smartcards de contacto requieren una conexión física con un circuito eléctrico denominado terminal, a través del cual reciben la alimentación y las señales de control.

En la siguiente figura se muestra la arquitectura típica de una tarjeta de memoria de contacto con lógica de seguridad. En la imagen se muestran únicamente los flujos de energía y datos y no es un diagrama esquemático detallado.

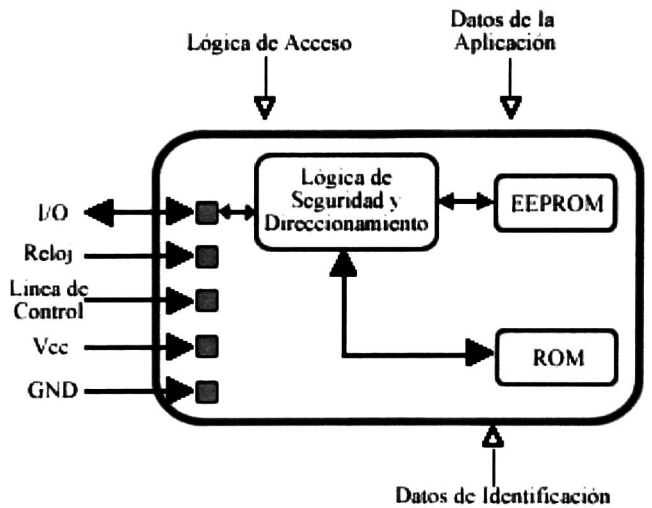


Figura 1. Arquitectura típica de una tarjeta de contacto con lógica de seguridad incluida.

Como se puede inferir por lo mostrado en la figura 1, las smartcards de contacto se comunican con el mundo exterior utilizando la interfase de comunicación serial, a través de sus regularmente ocho puntos de contacto.

Por su propia naturaleza, las tarjetas de contacto deben ser introducidas a un dispositivo lector en la forma correcta y en la orientación adecuada, por lo que resultan incómodas y poco prácticas para ciertas aplicaciones donde se requieran transacciones rápidas.

2. Tarjetas de proximidad o sin contacto.

Las smartcards de proximidad reciben la energía y establecen la comunicación cuando ingresan a un campo de radio frecuencia (RF) emitido por una terminal especial. En la figura 2 se muestra la arquitectura típica de una tarjeta de microprocesador con un coprocesador y una interfase sin contacto. La figura 2 sólo muestra los flujos básicos de energía y de datos y no es un diagrama

esquemático detallado. Esta tesis analiza principalmente el caso de las tarjetas de proximidad o sin contacto.

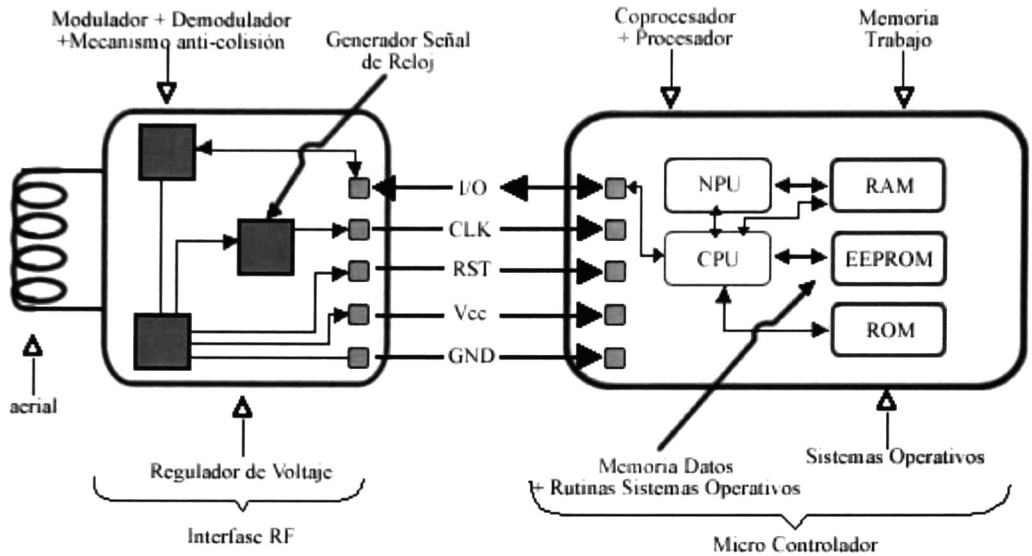


Figura 2. Arquitectura típica de una tarjeta de microprocesador con coprocesador e interfase sin contacto.

La comunicación con la tarjeta siempre es iniciada por la terminal. La tarjeta siempre responde a instrucciones de la terminal, lo que significa que la tarjeta nunca envía información sin una solicitud externa. Esto resulta en una relación maestro-esclavo, con la terminal como maestro y la tarjeta como esclavo.

El intercambio de datos con las tarjetas inteligentes toma lugar digitalmente, lo que significa que solo emplea los valores lógicos 0 y 1. La transferencia de datos entre la tarjeta y la terminal se desarrolla asincrónicamente, lo que significa que cada byte enviado debe ser provisto con bits complementarios de sincronización.

Cuando la tarjeta entra en el campo RF de la terminal, el inicio de la transmisión de datos entre la tarjeta y la terminal debe realizarse dentro de un tiempo límite. Si la terminal no recibe el mensaje de inicio por parte de la tarjeta dentro de este intervalo, repite la secuencia de activación varias veces (usualmente hasta 3 veces) para tratar de detectar un mensaje de activación de la tarjeta [LIM 99]. Si todos estos intentos fallan, la terminal asume que la tarjeta presenta falla y responde de acuerdo a ella.

La amplia gama de aplicaciones en las que son utilizadas las tarjetas inteligentes hace que en ocasiones sea necesario almacenar información que excede la capacidad de la zona de escritura confiable de la tarjeta. Estas aplicaciones cuando se presentan en ambientes con interferencia electromagnética o en ambientes ruidosos, propician que en algunos casos se incremente la cantidad de intentos de escritura requeridos para lograr escrituras exitosas, decrementando la confiabilidad de este tipo de transacciones en las tarjetas en estos ambientes [Vedder 97].

Se dice que las smartcards algún día serán tan importantes como lo son las computadoras hoy [Janson 95]. Se podría pensar que esta afirmación contiene un pequeño margen de error porque las smartcards no son computadoras, pero en realidad si lo son.

En este capítulo se describirá la historia de las smartcards, algunos tipos diferentes, sus propiedades de bajo nivel, los estándares y en general el estado del arte y los conceptos básicos necesarios para entender el trabajo de tesis que se presenta.

Dado que las smartcards son pequeñas computadoras es difícil predecir la variedad de aplicaciones que podrán albergar en un futuro. Posiblemente siga en la tendencia del rápido incremento del poder de procesamiento que tienen las computadoras actuales y quizás dupliquen su desempeño y decrementen por mitad su costo cada 18 meses.

Las Smart Cards han probado ser bastante útiles en los países europeos como un medio de transacción-autorización-identificación. Mientras crecen sus capacidades pueden convertirse en clientes delgados, eventualmente reemplazar todas las cosas que cargamos en nuestros bolsillos incluyendo tarjetas de crédito, licencias, efectivo e incluso fotografías familiares. Por la capacidad de contener certificados de identificación las smartcards pueden ser utilizadas para identificar atributos de nosotros mismos sin importar donde estemos o en cuál computadora en la red estamos trabajando.

No es la intención predecir el futuro de las aplicaciones de las smartcards ni el impacto en la sociedad, en su lugar nos concentraremos en el estado del arte y su utilización en los sistemas de cómputo y esquemas de seguridad.

2.2 Historia de las Smartcards

Las raíces de las actuales smartcards pueden remontarse a inicios de los años 50's en los Estados Unidos cuando se produjeron las primeras tarjetas de plástico para realizar algún tipo de pago [Janson 95]. Se fabricaban con material sintético PVC, que permitían que las tarjetas tuvieran mayor duración que las tarjetas de papel utilizadas hasta entonces.

En este nuevo sistema se permitía a los usuarios pagar utilizando sólo su nombre en lugar de efectivo, de tal forma que las tarjetas además de servir como un mecanismo de identificación permitían al usuario formar parte de un selecto grupo de compradores que era aceptado por ciertos restaurantes u hoteles.

Visa y Master Card entraron entonces al mercado, pero eventualmente los costos originados por fraudes e inconsistencias del sistema hicieron necesaria una máquina lectora de tarjetas. Se introdujo la banda magnética y esto permitió digitalizar datos que eran almacenados en las tarjetas en un formato que podía ser leído e interpretado por las máquinas lectoras. Hoy este tipo de tarjetas con banda magnética es el método de pago mas comúnmente utilizado.

La tecnología de banda magnética sufre una debilidad crítica, en la cual cualquiera con acceso al dispositivo adecuado puede leer, re-escribir o borrar la información. Esto convierte a las tarjetas de banda magnética en una mala opción para cierto tipo de aplicaciones que requieren un mayor nivel de seguridad en cuanto a la verificación y procesamiento de la información [Waidner 95] [Burmester 92].

En 1968 en Alemania se tienen los primeros antecedentes de las tarjetas con circuito integrado. Aplicaciones similares se dieron en Japón en 1970 y en Francia en 1974. En 1984 los servicios postales y de telecomunicaciones en Francia incursionaron con éxito en el campo de las tarjetas telefónicas. Para 1986 había en circulación varios millones de tarjetas telefónicas cuyo número alcanzó los sesenta millones en 1990, y ciento cincuenta millones para 1996.

Mientras la criptografía tuvo grandes progresos desde 1960 y los mecanismos de seguridad pudieron ser probados matemáticamente, las smartcards demostraron ser un medio ideal para almacenar de manera segura claves y algoritmos criptográficos. Los bancos franceses fueron los primeros en utilizar en una tarjeta bancaria un chip en 1984. Los bancos alemanes comenzaron a introducirlas alrededor de 1997. En esa misma época en Alemania se utilizaron alrededor de setenta millones de smartcards que contenían la información de seguridad social y de salud de la mayor parte de la población [BIS 96].

2.3 Estándares

Como en la mayoría de las tecnologías de reciente introducción, existe una tendencia promovida por un cierto número de cuerpos de estandarización internacional quienes se han preocupado de desarrollar un conjunto de estándares que gobiernen los atributos físicos y lógicos de las smartcards.

Podemos entender que un estándar es un documento producido por consenso y adoptado por una institución reconocida, el cual, para aplicaciones generales y recurrentes define reglas, guías o características para actividades o resultados, donde el objetivo es lograr el máximo grado de regulación en un contexto dado. En virtud de lo anterior, está claro que la preparación de un estándar usualmente toma varios años.

La mayoría de estos estándares, sin embargo, se han quedado muy atrás de las realidades del progreso técnico, y no se han dirigido a resolver los problemas de aplicación y de interoperabilidad de manera suficiente para permitir que el desarrollo de software proceda con confianza.

Esta situación ha permitido que los actores internacionales desarrollen productos de acuerdo a especificaciones que ellos desean sean reconocidas como estándares de facto. Algunos sectores del mercado que han actuado en esa dirección son GSM (Global System for Mobile communications) y las principales instituciones de crédito en el consorcio EMV (Europay, MasterCard, Visa).

Muchos países con gobiernos centrales fuertes están también imponiendo estándares para esquemas nacionales. Mientras que existen países que no han reaccionado siquiera a la necesidad

cada vez más imperiosa de estandarizar estas tecnologías, lo que ha permitido el desarrollo de sistemas propietarios sin control.

El efecto de esta fragmentación será el de imponer más y mayores barreras de interoperabilidad entre tarjetas a través de las fronteras nacionales y de los diferentes esquemas.

En el largo plazo, los actores con mayor peso – como Microsoft- tienen mayores probabilidades de desplegar implementaciones de tarjetas y software en un esfuerzo de saturar el mercado internacional con una topografía o una arquitectura particular.

2.3.1 ISO

La organización internacional para la estandarización es una asociación mundial compuesta por alrededor de 100 agencias nacionales de estándares, con una por país. ISO fue fundado en 1948 y es una organización no nacional. Su tarea es promover el desarrollo de estándares en todo el mundo, con el objetivo de simplificar el intercambio internacional de bienes y servicios asi como desarrollar la cooperación en los campos de la ciencia, tecnología y economía. Los resultados del trabajo de ISO son acuerdos que se publican como estándares ISO [Balic 98].

Las organizaciones miembros cumplen cuatro tareas básicas:

> Informar a las partes potencialmente interesadas en sus propios países sobre actividades relevantes y posibilidades de estandarización internacional.
> Formar opiniones nacionales en una base democrática y representar estas opiniones en negociaciones internacionales.
> Conformar secretarias para comités ISO en las cuales el país tenga intereses particulares.
> Pagar la contribución financiera del país para mantener la organización central ISO.

ISO no es la única Organización Internacional de estándares. Para prevenir la duplicación de esfuerzos, ISO coopera de una forma muy cercana con la IEC (Internacional Electrotechnical Commission). Las áreas de responsabilidad se reparten de tal forma que la IEC cubre los campos de Tecnología eléctrica y electrónica, mientras que ISO es responsable de los demás campos.

También se forman grupos de trabajo combinados para abarcar temas de interés común, y estos grupos producen estándares combinados ISO-IEC. La mayor Parte de estándares para las Smart Cards pertenecen a esta categoría. ISO y el Comité de Estandarización Europeo CEN (Comité Européen de Normalisation) también se ponen de acuerdo en reglas para el desarrollo de estándares que son reconocidos como estándares, tanto internacionales como Europeos.

En la figura 3 se muestra de manera general la estructura y organización de los grupos de trabajo involucrados en el desarrollo de los estándares internacionales de las smartcards.

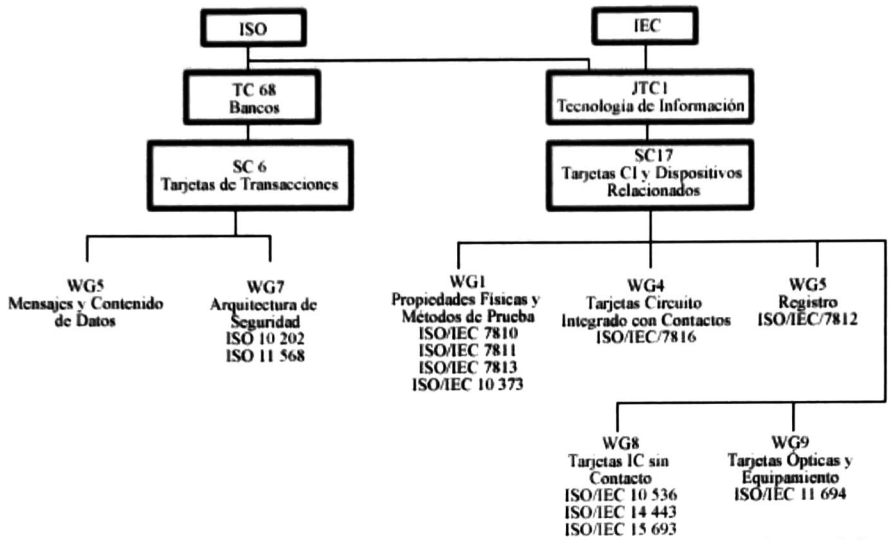


Figura 3. Estructura general y organización de los grupos de trabajo ISO/IEC involucrados en el desarrollo de estándares internacionales para smartcards.

En figura 4 se puede apreciar la jerarquía de los grupos de trabajo del CEN. El trabajo conjunto entre ISO y el CEN se traduce en ahorro de tiempo y costos.

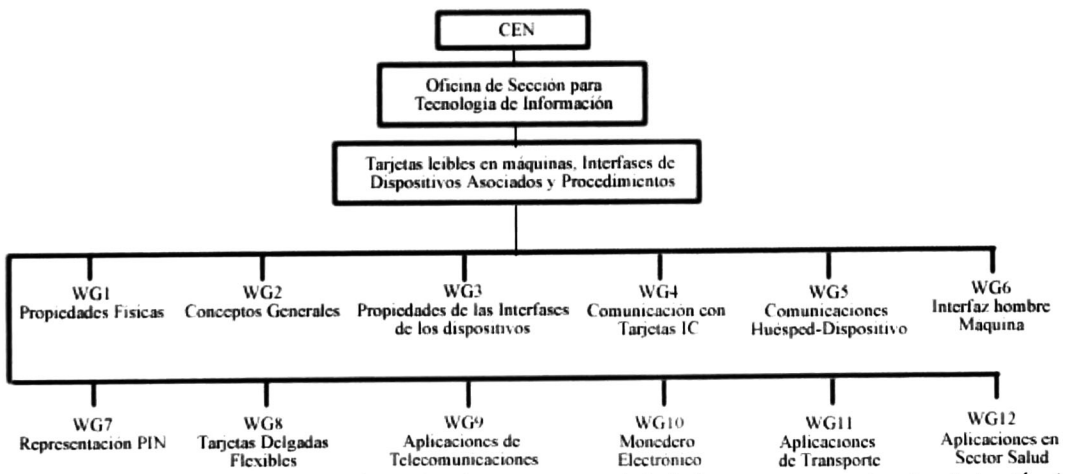


Figura 4. Estructura general y jerarquías de los diversos grupos de trabajo europeos para la estandarización de las smartcards.

2.3.1.1 Estandarización Internacional de las Smartcards

El desarrollo de estándares Internacionales para las smartcards se logra mediante el trabajo de los grupos ISO-IEC y/o el CEN. El la figura 3 se puede apreciar de manera general la estructura de los grupos de trabajo de ISO, IEC, CEN, así como los estándares asociados a cada uno de ellos.

Debido a su importancia y la variedad de aplicaciones, en materia de smartcards existen actualmente diferentes estándares de derecho y de facto, entre los más relevantes para este trabajo de tesis están los siguientes:

ISO 7810-7813 Cubre la mayoría de las tarjetas actuales de banda magnética utilizadas en aplicaciones bancarias y financieras.

ISO 7816 Es una especificación de bajo nivel que define una tarjeta de contacto que contiene un microprocesador en cuanto a sus características físicas, dimensiones y ubicación de contactos, señales electrónicas, protocolos y comandos.

ISO 10536 Define las características físicas, dimensiones, ubicación de las áreas de acoplamiento, señales electrónicas, procedimientos de reinicio y protocolos de las tarjetas de microprocesador sin contacto de acoplamiento cercano.

ISO 14443 Define las características físicas de las tarjetas de microprocesador de proximidad.

ISO15693 Cubre las tarjetas de microprocesador sin contacto de proximidad en cuanto a la interfase aérea e inicialización.

ETSI Estándares para el uso de smartcards en sistemas de telefonía pública y celular.

PC/SC and OCF Especificaciones desarrolladas para ofrecer a los programadores acceso a los datos de las smartcards y manipular sus funciones mediante manejadores de dispositivos y sistemas operativos de PC.

La mayoría de estos estándares enfrentan el dilema de no sobre-especificar para permitir el avance de la tecnología, pero resultan sin ninguna duda una base sólida de desarrollo.

2.3.2 MIFARE

La tecnología Mifare [MIFARE] pertenece a Philips Electronics. Este corporativo no hace tarjetas o lectores, sino que fabrican y venden los chips de las tarjetas y lectores en el mercado abierto. En un principio estos lectores escritores de tarjetas sin contacto fueron desarrollados para manejar transacciones de pago del sistema de transporte público. Evolucionaron debido a que los lectores sin contacto eran mas rápidos y fáciles de utilizar y los lectores virtualmente no necesitaban mantenimiento, igualmente con las tarjetas ya que no existía desgaste en los conectores.

Mifare se ha convertido en un estándar en la industria debido a que el 80% de las aplicaciones de las tarjetas sin contacto utilizan como base esta tecnología [MIFARE]. Posicionándonos en un escenario de aplicación real, por ejemplo en un sistema de transporte público, el estándar Mifere funcionaría de la siguiente forma:

Una tarjeta se entrega al pasajero que se dirige a una terminal automatizada y utiliza una tarjeta de crédito o efectivo para incorporar un valor en la tarjeta. El valor se almacena en un "monedero electrónico" en la tarjeta, desde el cual es sustraída la tarifa adecuada cada vez que el pasajero hace uso de una autobús o un tren. Cuando el valor almacenado se agota el pasajero se dirige a la terminal automatizada y recarga el valor del "monedero electrónico".

Con base en algunas publicaciones de resultados derivados de proyectos con implementaciones reales de este estándar [MIFARE], se recomienda el uso de las tarjetas Mifare y su tecnología asociada para aplicaciones de colección automática de tarifas, esquemas de lealtad de clientes, estacionamientos, como identificación, tarjetas universitarias, prepago de servicios como el telefónico o el bancario, pago de casetas de peaje o aplicaciones en líneas aéreas.

A pesar que las tarjetas Mifare cuentan con características de seguridad como de transmisión de radiofrecuencia encriptada, autenticación mutua, y llaves de seguridad, la mayoría de los bancos no considera que las tarjetas Mifare contenga la capacidad suficiente para procesar el tipo de encripción requerida por las transacciones bancarias y de tarjetas de crédito [Anderson 94].

La tarjeta Mifare tiene hasta 16 sectores separados, los cuales pueden ser configurados como monederos o bien para almacenamiento general de datos. El primer sector típicamente se utiliza como un directorio para el resto de la tarjeta, dejando 15 segmentos libres para datos o monederos.

En una tarjeta Mifare pueden ser almacenadas hasta 15 aplicaciones diferentes las cuales están separadas y seguras una de la otra mediante la utilización de llaves únicas o contraseñas para cada sector. Cada sector tiene dos llaves denominadas A y B, permitiendo diferentes privilegios de acceso en ese sector. Este par de llaves pueden ser designadas como de lectura y lectura/escritura o bien como decremento e incremento/decremento.

La tarjeta Mifare cuenta además con un número aleatorio único de 32 bits el cual es permanentemente codificado en cada chip por parte del fabricante. Este número se le conoce regularmente como número de seria de la tarjeta (CSN) o identificador universal (UID) y puede ser leído por cualquier lector Mifare sin la necesidad de conocer cualquiera de las llaves utilizadas para proteger el resto de la información de la tarjeta.

2.3.2.1 Estructura de Memoria de la Tarjeta Mifare

Cada uno de los 16 sectores de la tarjeta Mifare consta de 4 bloques de 16 bytes que son numerados del 0 al 3.

El sector 0 bloque 0 contiene el código del fabricante de la tarjeta y un identificador de 32 bits. No puede contener ningún dato de usuario y no puede ser modificado. Este dato puede ser leído sin

las claves de seguridad Mifare. En todos los demás sectores el bloque cero puede ser programado con datos de usuario.

Los bloques 0-2 de cualquier sector contienen datos de usuario. Dependiendo de cómo sea el formato de los datos, un bloque puede ser un dato o un valor almacenado.

El bloque 3 contiene claves y condiciones de acceso para todos los cuatro bloques incluyéndose él mismo. Solo existe un par de claves por sector pero pueden existir condiciones de acceso únicas en cada bloque.

Contar con dos claves por sector le permite al administrador del sistema estructurar la codificación de las tarjetas de tal forma que diferentes usuarios tengan diferentes privilegios respecto a los datos.

2.3.2.2 Condiciones de Acceso

Las condiciones de acceso para un segmento dato pueden ser únicas para cada bloque. Estas condiciones de acceso son expresadas como un número binario de 3 bits, lo cuál permite ocho diferentes formas de configurar el acceso de cada par de claves en cada bloque.

Las condiciones de acceso para los bloques de datos pueden permitir o prevenir que los datos puedan ser leídos, escritos, incrementados o decrementados utilizando una o ambas claves.

2.3.2.3 Bloques de Valores y Bloques de Datos

Un bloque de datos puede ser o bien un bloque de lectura/escritura conteniendo 16 bytes de datos generales o un bloque de valor conteniendo 4 bytes de un valor dado. Sólo los bloques de valor pueden sen incrementados, decrementados, transferidos o restaurados.

Los bloques de valor consisten en 4 bytes de información de dirección, 4 bytes del valor almacenado, 4 bytes del valor almacenado y 4 bytes del valor repetido.

El valor es almacenado 3 veces un bloque de valor para permitir la detección de errores y capacidades de corrección. Un sector puede contener cualquier combinación de bloques de valor o de datos en los bloques 0 al 2.

2.3.2.4 Claves MIFARE

Básicamente las claves Mifare son contraseñas numéricas utilizadas para controlar el acceso a la información contenida en las tarjetas. Una clave es un campo de datos de 48 bits, típicamente expresados como 12 caracteres hexadecimales.

Las claves Mifare son asociadas en pares. Nos referiremos a una como clave A y a la otra como clave B. Cada sector en la tarjeta Mifare tienen un par de claves lo que significa que existen 16 pares de claves en una tarjeta, donde cada par de claves controla el acceso a los datos del sector donde se encuentra.

2.4 Tipos de Tarjetas

La organización internacional para la estandarización (ISO) en el estándar 7810 define las características físicas y propiedades de las tarjetas de identificación como son la flexibilidad, resistencia a temperatura y dimensiones de tres diferentes formatos de tarjetas (ID-1, ID-2, ID-3). El estándar de las Smart Cards ISO-7816 se basa en el formato ID-1.

A continuación se incluye una breve descripción de diferentes tipos de tarjetas en este formato (ID-1) [Balic 98] [Frank 96]:

2.4.1 De Relieve

Las tarjetas de relieve permiten que información textual o un diseño específico en la tarjeta pueda ser llevado a papel utilizando un dispositivo simple y barato. El estándar ISO-7811 especifica las marcas de relieve cubriendo su forma, tamaño, altura de relieve y posición en la tarjeta. La transferencia de información a través de la presión de un papel en el relieve puede parecer primitiva, pero la simplicidad del sistema hizo posible su proliferación.

2.4.2 Banda Magnética

La ventaja principal que la tecnología de banda magnética ofrece sobre la de relieve es la reducción en la cantidad de papeles y documentos. Las partes 2, 4 y 5 del estándar ISO-7811, especifican las propiedades de la banda magnética, las técnicas de codificación y posicionamiento relativo. La capacidad de almacenamiento de la banda es aproximadamente 1000 bits y cualquiera con el dispositivo de lectura-escritura adecuado puede ver o alterar los datos.

2.4.3 Smartcards

Las tarjetas de circuito integrado se denominan convencionalmente smartcards. Estas son la más reciente e inteligente adición a la familia de tarjetas de formato ID-1 y sus detalles técnicos se especifican en el estándar ISO-7816. Este tipo de tarjetas puede almacenar una gran cantidad de información. Actualmente hay disponibles tarjetas de 20 o hasta 32 Kbytes.

Adicionalmente y tal vez lo más importante los datos almacenados pueden ser protegidos contra accesos no autorizados. Las funciones de memoria como lectura, escritura y borrado pueden ser

ligadas a condiciones específicas, controladas tanto por hardware como por software. Otra ventaja de las smartcards sobre las tarjetas de banda magnética es que son más confiables y su expectativa del tiempo de vida es mucho mayor.

2.4.3.1 Tarjetas de Memoria

Las tarjetas de memoria son típicamente mucho más económicas, pero mucho menos funcionales que las tarjetas de microprocesador. Contienen memoria EEPROM y ROM, así como algunas direcciones y lógica de seguridad. En los diseños más sencillos la lógica existe para prevenir escritura y borrado de los datos. En diseños más complejos permiten tener accesos restringidos a memoria. Aplicaciones típicas de las tarjetas de memoria son las tarjetas de prepago telefónico y las tarjetas de seguridad social.

2.4.3.2 Tarjetas de Microprocesador

Los componentes de este tipo de arquitectura incluyen un CPU, RAM, ROM y EEPROM. El sistema operativo usualmente es almacenado en la ROM, el CPU utiliza la RAM como su memoria de trabajo y la mayoría de los datos se almacenan en la EEPROM. Una regla que casi por lo general se cumple en cuanto al diseño es que la RAM requiere casi cuatro veces el espacio que requiere la EEPROM y a su vez, esta requiere cuatro veces mas espacio que la ROM.

2.4.3.3 Tarjetas de Coprocesador Criptográfico

A pesar de que técnicamente estas tarjetas deberían clasificarse con las tarjetas de microprocesador se hace una diferencia por las diferencias en costo y funcionalidad. Dado que los algoritmos criptográficos asimétricos comunes utilizados hoy en día requieren cálculos matemáticos de enteros muy largos [Bellare 95b], un microprocesador de 8 bits con muy poca RAM puede requerir de varios minutos para realizar una operación de llave privada de 1024 [Anderson 95].

Sin embargo, si se agrega un coprocesador criptográfico a la arquitectura el tiempo requerido para realizar la misma operación se reduce a pocos cientos de microsegundos. Los coprocesadores incluyen unidades aritméticas adicionales desarrolladas específicamente para cálculos con enteros largos y exponenciales. Existe una desventaja, el costo. La adición de un coprocesador criptográfico puede aumentar el costo de la Smart Card entre un 50 y un 100 %, pero el costo se reducirá conforme se diversifique el uso de los coprocesadores.

2.4.3.4 Tarjetas de Proximidad

A pesar de que la confiabilidad de las smartcards de contacto se ha incrementado considerablemente hasta alcanzar niveles bastante aceptables al paso de los años, los contactos son uno de los puntos de falla más frecuentes de cualquier sistema electromecánico, debido a la suciedad, el uso, etc. Dado que los contactos que están ubicados en la superficie de la tarjeta se conectan directamente a las entradas del circuito integrado empotrado en la tarjeta, hay un riesgo

de daño o destrucción del circuito integrado por cargas electrostáticas y no son raras cargas de varios miles de volts.

Los problemas de confiabilidad y otros problemas técnicos de las tarjetas de contacto mencionados con anterioridad son resueltos de una manera eficiente por las tarjetas sin contacto. Además de sus ventajas técnicas, la tecnología de las tarjetas sin contacto ofrece además, tanto al emisor de la tarjeta como al portador un rango muy interesante de aplicaciones potenciales nuevas. Las tarjetas sin contacto no necesariamente se insertan en algún lector dado que hay sistemas que trabajan a distancias de hasta un metro [Thomasson 96]. Esta es una gran ventaja en sistemas de control de acceso, transporte público local, etc.

La tecnología sin contacto también ofrece ventajas sobre sistemas que requieren la inserción deliberada de la tarjeta en un lector, dado que no importa como se inserte ésta en lector, a diferencia de las tarjetas de banda magnética o tarjetas de contacto. La libertad en las restricciones de orientación simplifica la operación e incrementa la aceptación del cliente a esta tecnología.

Desde el punto de vista mercadológico las tarjetas sin contacto ofrecen el beneficio de que no tienen visibles en la superficie de la tarjeta elementos técnicos de tal forma que su diseño visual no está limitado por bandas magnéticas o superficies de contacto.

La tecnología de manufactura para la producción en masa de las tarjetas sin contacto ha madurado al punto que productos de calidad están disponibles a precios que no difieren significativamente el precio de aquellos productos de contacto.

2.4.3.5 Tarjetas de Memoria Óptica

El estándar ISO-IEC11693 y 11694 definen los estándares para tarjetas de memoria óptica. Estas tarjetas pueden llevar muchos Megabytes de datos pero solo pueden ser escritas una vez y nunca borradas con la tecnología actual. Los dispositivos de lectura y escritura para las tarjetas óptica aún son muy caros y estas pueden utilizarse en aplicaciones médicas donde debe almacenarse gran cantidad de información.

2.5 Propiedades Funcionales de las Smartcards

2.5.1 Propiedades Físicas y Eléctricas

El tamaño físico de una Smart Card se describe con detalle en el estándar ISO-7810. Las dimensiones son 85.6 mm por 54 mm, con una esquina redondeada de 3.18 mm y una anchura de .76mm. Cuando el estándar ISO-7810 se creó en 1985, no mencionaba una ubicación para el chip. La ubicación del chip se define en la parte 2 del estándar ISO-7816, creado en 1988. Estas características se describen en figura 5.

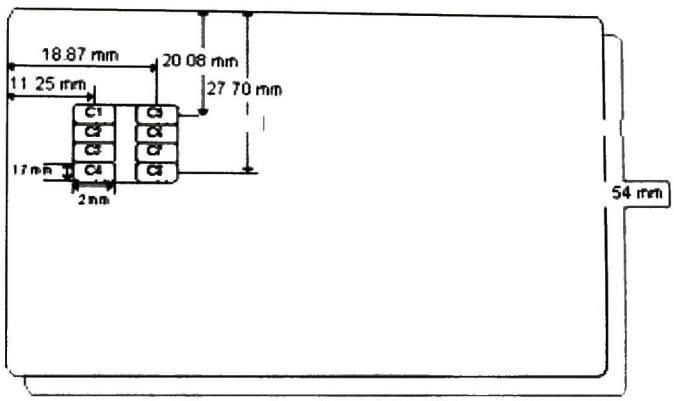


Figura 5. Dimensiones físicas de una smartcard

2.5.2 Sistema Operativo

A pesar de contar con sólo unos cuantos miles de bytes de código de programa, el sistema operativo del microprocesador de la smartcard debe manejar tareas como:

Transmisión de datos por la interfase serial bidireccional
Cargar, operar y administrar aplicaciones
Control de ejecución y procesamiento de instrucciones
Acceso protegido a datos
Administración de memoria
Administración de archivos
Administración y ejecución de algoritmos criptográficos

A diferencia de los sistemas operativos de las computadoras personales, los sistemas operativos de las Smart Cards no cuentan con interfaces de usuario o la habilidad de acceder a dispositivos periféricos externos o medios de almacenamiento. El tamaño es típicamente entre los 3 y los 24 Kbytes. El límite inferior es utilizado por aplicaciones especializadas y el límite superior por sistemas operativos multi-aplicación.

Dado que el espacio de memoria de las smartcards está severamente limitado, no todas las instrucciones estandarizadas y estructuras de archivos pueden ser implementadas en todos los sistemas operativos de smartcards. Por esta razón en los estándares ISO-7816 parte 4 y EN 726-3 se ha introducido el concepto de "perfil". Ahí se definen los requerimientos mínimos para estructuras de datos y comandos.

2.5.2.1 MULTOS

MULTOS (que significa sistema operativo múltiple) es un sistema operativo que permite que múltiples programas de aplicación sean instalados y residan separadamente y de manera segura en

una smartcard. Cada programa es aislado por el sistema operativo de tal forma que ninguna aplicación pueda interferir con alguna otra.

Mientras que los sistemas anteriores de smartcards no permitían que se instalaran nuevas aplicaciones, o en su caso borraban las aplicaciones existentes, MULTOS hace esto posible. Las actualizaciones o parches también pueden ser instaladas como sea requerido.

Cada aplicación es independiente de la plataforma debido a la implementación de una máquina virtual. Los desarrolladores escriben aplicaciones para las smartcards MULTOS utilizando el lenguaje ejecutable MULTOS (MEL).

Antes de MULTOS, los desarrolladores de aplicaciones tenían que escribir versiones separadas de una misma aplicación para cada tipo de smartcard y el consumidor requería una smartcard diferente para cada aplicación. Con MULTOS, muchas aplicaciones pueden residir en una smartcard sin tomar en cuenta el microchip utilizado [multos].

Las seguridad para el sistema operativo MULTOS se logra por la certificación de autoridad MULTOS (CA), la cuál utiliza llaves criptográficas para cada smartcard MULTOS y todas las aplicaciones MULTOS. Estas llaves previenen que aplicaciones no autorizadas sean cargadas en una tarjeta o bien que sean borradas sin la autorización del usuario.

El consorcio MAOSCO, un grupo de organizaciones líderes internacionales, licencia abiertamente las especificaciones MULTOS. Los sistemas más importantes de transacciones con tarjetas de crédito como MasterCard, Mondex, Discover y Europay utilizan MULTOS.

2.5.3 Tecnología de Java Cards

La tecnología Java Card ofrece una forma de derribar los obstáculos que impiden la completa aceptación de las smartcards. Permite a las smartcards y otros dispositivos de memoria integrada ejecutar aplicaciones (llamadas applets) escritas en el lenguaje de programación Java [Stocker 98]. Esencialmente la tecnología de Java Cards define una plataforma para smartcards segura, portable y multi-aplicación que incorpora las principales ventajas del lenguaje Java [Gosling 99].

Los beneficios de la tecnología Java Cards pueden enumerarse [Yellin 96] como sigue:

Facilidad de desarrollo de aplicaciones
Seguridad
Independencia de hardware
Capacidad para almacenar y administrar múltiples aplicaciones
Compatibilidad con los estándares existentes de las smartcards

Las smartcards representan una de las plataformas de cómputo más pequeñas que están en uso. La configuración de memoria de una smartcard está en el orden de 1K de RAM, 16K de EEPROM, 24K de ROM. El mayor reto de la tecnología Java Card es diseñar software de sistema Java que quepa en una smartcard y conservar suficiente espacio para aplicaciones.

La solución es soportar solo un subconjunto de características del lenguaje Java y aplicar un modelo para implementar la máquina virtual de Java. La máquina virtual Java Card se divide en dos partes: una se ejecuta fuera de la tarjeta y la otra en la tarjeta. Muchas tareas de procesamiento que no se restringen a ejecutarse en una rutina como la carga de clases, verificación de bytecode, resolución y enlace y optimización son dedicadas para la máquina virtual que se ejecuta fuera de la tarjeta donde los recursos usualmente no son una preocupación [Hendry 97].

Las smartcards difieren de las computadoras de escritorio en muchas formas. Además de proveer soporte al lenguaje Java. La tecnología de Java Cards define un entorno de ejecución que soporta la operación del modelo de aplicación de las smartcards, de acuerdo con el estándar internacional ISO-7816.

2.5.4 Capacidades Criptográficas

Actualmente las Smart Cards contienen suficientes capacidades criptográficas para soportar las aplicaciones y protocolos de seguridad más populares. Las firmas y verificaciones soportadas fluctúan entre los 512, 768 o 1024 bits de longitud.

Los algoritmos regularmente utilizan el teorema CRT para acelerar el procesamiento. Aún en las llaves de 1024 bits de longitud el tiempo requerido para llevar a cabo una firma es típicamente menos de un segundo [Biham 91]. Con frecuencia están presentes funcionalidades de monedero electrónico aunque estas regularmente se basan en tecnologías de llave simétrica como DES y triple DES [Wiener 93].

Los protocolos de comunicación en las Smart Cards a nivel de comandos muchas veces tienen un protocolo de seguridad inter-construido, que regularmente se basa en tecnología de llaves simétricas y permite a la propia tarjeta autentificar a la terminal de lectura/escritura o viceversa, sin embargo los criptogramas y algoritmos para estos protocolos usualmente son específicos para determinada aplicación y conjunto de terminales [Schneier 96].

2.5.5 Transmisión de Datos

A pesar de que este trabajo se enfoca en la descripción del funcionamiento de un algoritmo de escritura segura para tarjetas sin contacto en esta sección se presenta la descripción de los protocolos más comúnmente utilizados para la transmisión de datos en las tarjetas inteligentes de contacto pero que servirán como base para comprender el trabajo propuesto en esta tesis.

En las comunicaciones de las tarjetas de contacto, prácticamente todas las comunicaciones funcionan bajo una relación del tipo cliente servidor entre la tarjeta y la terminal. Una vez que la tarjeta se inserta en la terminal, ésta es energizada.

La tarjeta ejecuta una operación denominada "power-on-reset" y envia un mensaje ATR "answer-to-reset" a la terminal. El mensaje ATR es recibido, varios parámetros son extraídos y entonces la

terminal envía la instrucción inicial a la tarjeta. La tarjeta genera una respuesta y la envía de regreso a la terminal. La relación cliente servidor continúa de esta manera hasta que el procesamiento se completa y la tarjeta es removida de la terminal [Anderson 96b].

Lógicamente hay diferentes protocolos para el intercambio de información en una relación cliente servidor. Estos protocolos se designan como "T=" mas un número:

T=0 Es un intercambio de información asíncrono, donde solo una parte puede comunicarse a la vez, es orientado a bytes y se define en el estándar ISO/IEC 7816-3.

T=1 Es un intercambio de información asíncrono, donde solo una parte puede comunicarse a la vez, es orientado a bloques y se define en el estándar ISO/IEC 7816-3, Adm1.

T=2 Es un intercambio de información asíncrono, donde ambas partes pueden comunicarse a la vez, es orientado a bloques y se define en el estándar ISO/IEC 10536-4.

T=3 Es un intercambio de información, donde ambas partes pueden comunicarse a la vez, y aún no está cubierto por ningún estándar.

T=4 Es un intercambio de información asíncrono, donde solo una parte puede comunicarse a la vez, es orientado a bytes (expansión de T=0).

T=5 a T=13 están reservados para uso futuro.

T=14 Reservado, no es estándar ISO.

T=15 Reservado para uso futuro.

Los dos protocolos más comúnmente vistos son T=0 y T=1, siendo T=0 el más popular.

En el protocolo T=0 la terminal inicia la comunicación enviando un encabezado de instrucción de cinco bytes que incluye un byte de clase (CLA), un byte de instrucción (INS), y tres byes de parámetro (P1, P2, P3). Esto es seguido de manera opcional por una sección de datos.

La mayoría de los comandos pueden ser de entrada o salida desde la perspectiva de la tarjeta y el byte P3 especifica la longitud del dato de entrada o salida. La verificación de errores es manejada exclusivamente por un bit de paridad agregado a cada byte transmitido.

Si la tarjeta recibe correctamente los cinco bytes regresará un mensaje de un byte equivalente al byte INS recibido. Si la terminal está enviando más datos (comando de entrada) enviará el número de bytes especificados en P3. En este punto la tarjeta ha recibido una instrucción completa y puede ya procesarla y generar una respuesta.

Todos los comandos tienen dos bytes de código de respuesta denominados SW1 y SW2 los cuales reportan condiciones de éxito o de error. Si un comando exitoso debe regresar bytes adicionales, el número de bytes se especifica en el byte SW2. En este caso se utiliza el comando "GET RESPONSE", el cuál es por si mismo una instrucción de cinco bytes conforme al protocolo.

En la instrucción "GET RESPONSE" P3 será igual al número de bytes especificados en el byte previo SW2. "GET RESPONSE" es un comando de salida desde el punto de vista de la tarjeta. La terminal y la tarjeta se comunican de esta forma utilizando comandos de entrada o salida hasta completar el procesamiento

A diferencia de las tarjetas de contacto las tarjetas sin contacto no requieren ninguna conexión eléctrica entre la tarjeta y la terminal para transferir datos en una distancia corta. Las técnicas que se utilizan en las tarjetas sin contacto no son nuevas, se han conocido por muchos años en sistemas RFID (Sistemas de identificación de radiofrecuencia), los cuales han sido utilizados en una amplia variedad de aplicaciones, como implantes a animales, sistemas de seguridad para vehículos, etc.

Existe una gran variedad de métodos para identificar personas u objetos en cortas o largas distancias, basados en técnicas de radio y en particular en técnicas de radar. Entre la gran variedad de posibilidades técnicas, solo algunas pueden utilizarse en las tarjetas inteligentes del formato ID-1, dado que los elementos funcionales deben ser albergados en una tarjeta flexible de solo 0.76mm de grueso. Por ejemplo no hay baterías disponibles con ese ancho que puedan ser utilizadas para proveer energía a la circuitería electrónica de la tarjeta.

Al igual que las tarjetas de contacto un sistema que utiliza tarjetas sin contacto consiste en al menos dos componentes, denominados la tarjeta y la terminal de verificación, que puede funcionar como un lector o un lector escritor según la tecnología utilizada. Como una regla la terminal incluye una interfaz adicional a través de la cual se puede comunicar con otros sistemas.

Deben cubrirse cuatro necesidades para que una tarjeta sin contacto pueda comunicarse con la terminal:

1. Transferencia de energía que energice el circuito integrado
2. Transmisión de la señal de reloj
3. Transferencia de datos a la tarjeta
4. Transferencia de datos de la tarjeta

Se han desarrollado muchos métodos diferentes para resolver los problemas que se han comentado, basados en la experiencia o en los sistemas RFID. En la figura 6 se muestra la transferencia necesaria de energía y datos que debe existir de forma mínima entre una terminal y una smartcard sin contacto.

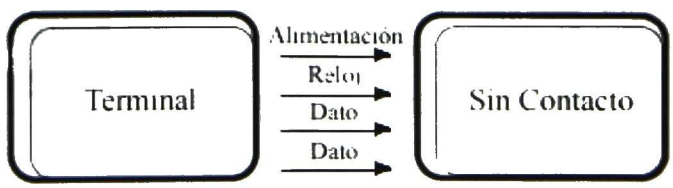


Figura 6: Transferencia necesaria de energía y datos entre la terminal y la tarjeta sin contacto

2.5.6 Transmisión Inductiva

Actualmente es la técnica utilizada más ampliamente. Puede ser utilizada para transferir energía y datos. Permite soluciones técnicamente simples debido al bajo consumo de energía (unas cuantas decenas de micro watts), el rango utilizable de estas tarjetas se limita a un metro y la capacidad de memoria es de solo algunos cientos de bits.

Si los datos deben ser escritos además el consumo de energía se eleva a más de 100 micro watts. Como consecuencia el rango se limita alrededor de 10cm. en el modo de escritura.

Las tarjetas de microprocesador sin contacto tienen mayor consumo de energía de alrededor de 100 mili watts. La distancia de la terminal se limita entonces a algunos milímetros independientemente del rango de alcance y el consumo de energía, todas las tarjetas deben emplear transmisión inductiva, basados en el mismo principio [Anderson 96b].

2.5.7 Transferencia de Energía

Hasta hoy no hay muchas baterías disponibles que sean lo suficientemente delgadas y flexibles para incorporarse en una tarjeta inteligente. Además las consideraciones ambientales desalientan el uso de baterías en virtud de que contienen sustancias venenosas.

Por estas razones toda la energía requerida para la operación del chip en la tarjeta inteligente debe ser transferida del lector a la tarjeta. En la figura 7 se muestra cómo se realiza una transferencia inductiva para energizar una tarjeta inteligente sin contacto.

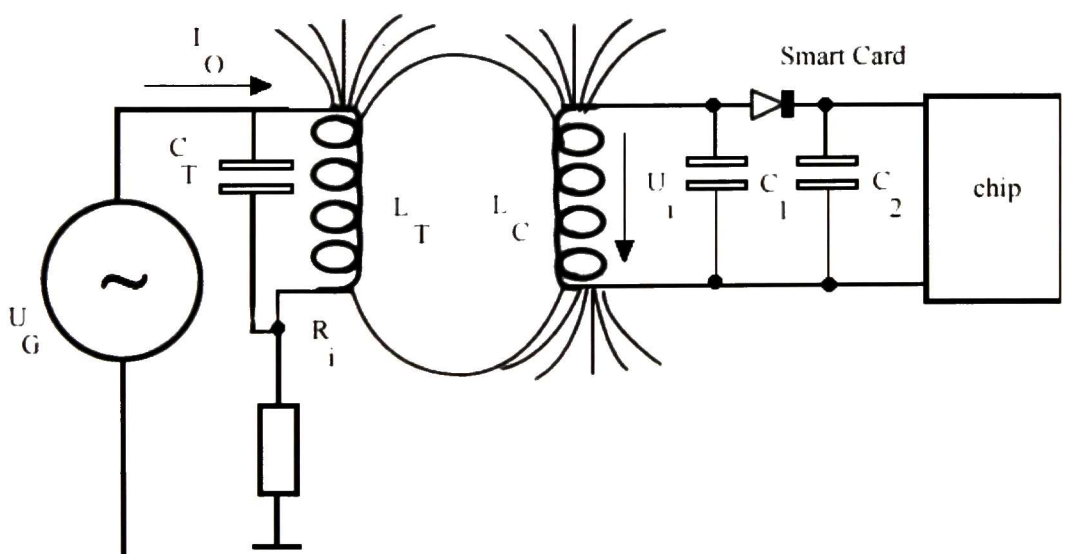


Figura 7. Transferencia inductiva para energizar una smarcard sin contacto.

La transferencia de energía se basa en el principio de un "LOOSELY COUPLED TRANSFORMER". Un fuerte campo magnético de alta frecuencia se genera por una bobina en la terminal con el fin de transferir energía. Las frecuencias utilizadas con mayor frecuencia son las de 125Khz y la de 13.56Mhz.

Si una tarjeta sin contacto se acerca a los linderos de la terminal, una porción del campo magnético de la terminal pasa a través de la bobina en la tarjeta, lo cuál induce un voltaje en esta bobina. Este voltaje es rectificado para servir como la fuente de energía para el chip.

Dado que la transferencia entre las bobinas en la terminal y las tarjetas es muy débil, la eficiencia de este arreglo es muy baja, por lo cual, es necesario tener niveles muy altos en la bobina de la terminal para lograr un nivel suficientemente fuerte en el campo.

2.5.8 Transferencia de Datos

Para transferir datos de la terminal de la tarjeta pueden utilizarse todas las técnicas de modulación conocidas. Las técnicas más comúnmente utilizada son ASK (Amplitud de Shift Keying), FSK (Frecuency Shift Keying) y PSK (Phase Shift Keying). ASK y PSK se utilizan frecuentemente dado que son especialmente sencillas de remodular.

En la otra dirección, de la tarjeta inteligente a la terminal se utiliza un tipo de modulación de amplitud. Este es generado utilizando la señal del dato para modificar digitalmente una carga en la tarjeta (modulación de carga). Si una tarjeta inteligente que es sintonizada a la frecuencia de resonancia de la terminal se acerca al campo magnético de la terminal, cede energía de su campo como se describió anteriormente.

2.5.9 Características de Seguridad

El componente clave de una tarjeta inteligente es el circuito integrado incorporado en ella [Schaumüller-Buchl 91]. Este chip puede ser una memoria, una memoria protegida, un microprocesador o un FPGA (Field Programmable Gate Array) que es un dispositivo semiconductor que genera sus salidas directamente de los estados de entrada de acuerdo a un programa definido por el usuario.

Varios productores de semiconductores elaboran circuitos integrados específicamente diseñados para tarjetas inteligentes, mismo que generalmente utilizan:

Tecnología CMOS de bajo consumo, operando a voltajes entre 5 y 1.8V, y actualmente se encuentran en desarrollo chips que operan con 0.8V y voltajes inferiores.

Características de tamaño muy pequeño, el ancho de la línea más pequeña o separación en el chip de 0.5 micras es común ahora, y las líneas de producción de circuitos integrados más avanzadas trabajan a 0.15 micras.

Los microprocesadores más comunes en las tarjetas inteligentes aún son de 8 bits, usualmente basados en el diseño del motorota 8005 o del intel 8051, y con velocidades de reloj de 3.5 o 5 Mhz. Los productores están cambiando a tamaños de palabra de 32 bits, pero sus chips más poderosos no es común verlos en smartcards, particularmente para utilizarse en sistemas operativos multiaplicación. Los procesadores de 16 bits son menos comunes para estos efectos.

Con el fin de incrementar la velocidad del procesamiento, se requieren velocidades de reloj más grandes. Estudios demostraron que si la velocidad del reloj externo era incrementada significativamente se experimentaban efectos de interferencia en radio frecuencia.

Además, los cambios realizados al estándar ISO que trataban sobre las velocidades de reloj originaron grandes dificultades, así que los productores decidieron generar su propio reloj en la tarjeta, sincronizado de vez en cuando con el reloj externo. Esto les permite operar a velocidades mucho mayores (20Mhz para los chips más rápidos).

La última etapa de desarrollo es la introducción de microcontroladores con un conjunto reducido de instrucciones (RISC), los cuales ofrecen alto desempeño y bajo consumo de energía. Algunos productores están intentando con procesamiento "Pipeline" y conjuntos de instrucciones optimizados para la ejecución directa de código Java.

El efecto en la velocidad de las transacciones, en particular donde la autentificación es una función dominante es muy significativo: los casos típicos muestran una reducción de 3 segundos a 0.5 segundos. El factor limitante ahora es la velocidad de comunicación entre la tarjeta y la terminal, que comúnmente es de 9600 bps.

2.5.10 Conjunto de Instrucciones

Existen cuatro estándares internacionales que definen los conjuntos de instrucciones típicos de una smartcard [Lim 99]. En estos estándares se definen más de 50 instrucciones y sus correspondientes parámetros de ejecución.

A pesar de ser encontrados en cuatro estándares diferentes las instrucciones son altamente compatibles. Las especificaciones son GSM 11.11 (prETS 300608), EN 726-3, ISO/IEC 7816-4 y el estándar CEN preeliminar prEN 1546. Las instrucciones pueden ser clasificadas por función como sigue:

Selección de archivos
Lectura y escritura de archivos
Búsqueda de archivos
Operaciones de archivos
Identificación
Autentificación
Funciones criptográficas
Administración de archivos

Instrucciones para monedero electrónico o tarjetas de crédito
Sistema operativo
Prueba de hardware
Instrucciones especiales para funciones específicas
Soporte para protocolo de transmisión

Típicamente una smartcard solo implementará un subconjunto de las instrucciones posibles, específico a su aplicación. Esto se debe a las limitaciones de la memoria o al costo.

2.6 Conclusiones del Capítulo

En este capítulo se han descrito de una manera simple los conceptos relacionados con las smartcards, que son necesarios para poder comprender con mayor precisión el trabajo de tesis que se presenta.

Se presentó una introducción general al tema de las tarjetas inteligentes, particulamente a las smartcards sin contacto o contactless. De igual forma se destacó el aporte de las organizaciones como ISO, IEC y CEN, así como de la tecnología MIFARE como referentes y promotores de estándares de la industria y de facto, así como la relación de trabajo conjunto y organización jerárquica de dichas organizaciones internacionales para la elaboración de estándares, particulamente reacionados con la tecnología de las smartcards.

Se presentó un breve análisis de las características, capacidades y funcionalidades de las smartcards y se presentó un panorama general del estado del arte en esta tecnología.

Capítulo 3. Descripción del problema y Solución Desarrollada

3.1 Contexto General del Trabajo de Tesis

Las tarjetas inteligentes han encontrado uso en un número importante de aplicaciones, ejemplos de las cuales ya han sido enumeradas a lo largo de este trabajo. Estas aplicaciones podrían ser desarrolladas indistintamente con tarjetas de contacto o con tarjetas de proximidad. La diferencia radica en la flexibilidad de uso y en la velocidad de la transacción.

Por ejemplo, en una aplicación de transporte urbano el uso de una tarjeta de contacto implica un lector con una ranura y un mecanismo de sujeción, el cliente debe insertar la tarjeta en la ranura y esperar a que la transacción se realice, lo que puede tomar aproximadamente unos 30 segundos, causando esperas innecesarias al público que desea acceder al transporte.

Por el contrario, en la misma aplicación el uso de tarjetas sin contacto permite un tiempo de proceso de 150 ms típicamente por transacción lo que permite un procesamiento ágil y eficiente de las transacciones [Vedder 97]. Esto es posible, entre otras cosas porque no se requiere más que una antena, que substituye todo el hardware adicional que se requiere para operar e implementar un sistema basado en tarjetas de contacto.

3.2 Transacciones

Las transacciones, que por cierto tienen su origen en el campo de las leyes contractuales, observan las siguientes propiedades:

Consistencia: las transacciones deben obedecer la ley, la consistencia es el equivalente a la coherencia cuando hablamos de sistemas distribuidos.

Atomicidad: Las transacciones deben ocurrir completamente o entonces no ocurrir. En el campo en las telecomunicaciones la atomicidad es equivalente a un dato digital, discreto y discontinuo con estados bien diferenciados.

Durabilidad: Una vez consolidada una transacción (la consolidación de una transacción significa que esta se a completado de manera exitosa), ya no puede ser cancelada.

En el área de las ciencias computacionales, particularmente en la literatura de las bases de datos, la noción de transacción fue introducida para definir la consistencia, entre múltiples usuarios, de una base de datos común. Una transacción puede ser definida como una secuencia de comandos de lectura y escritura enviados por un cliente a un sistema de archivos.

En general el término transacción describe una secuencia de operaciones en uno o más objetos de la base de datos que trasforman un estado actual consistente en el sistema a un nuevo estado consistente. No todos los estados en el sistema son consistentes y por tanto algunos cambios no están permitidos para ocurrir.

Una transacción es una colección arbitraria de conexiones enmarcadas por dos indicadores - inicio de transacción (Begin Transaction) y fin de transacción (End Transaction).

Una generalización del concepto de transacción puede ser entendida como una acción atómica. Una acción es atómica si el proceso que la lleva a cabo no esta consciente de la existencia de algún otro proceso activo y ningún otro proceso esta consciente de la actividad del proceso inicial durante el tiempo en que dicho proceso llevan a la acción.

Una acción es atómica si se cumple lo siguiente:

El proceso que la lleva a cabo no se comunica con otros procesos mientras la acción es desarrollada.

El proceso que la lleva a cabo no puede detectar cambios de estado a excepción de aquellos desarrollados por el mismo y si no revela estos cambios de estado hasta que la acción es completada.

Si puede ser considerada indivisible e instantánea, de tal forma que los efectos en el sistema son como si estas se desarrollaran de una forma secuencial.

3.3 Protocolo de Commit a 2 fases.

La discusión de la comunicación en los sistemas operativos distribuidos basados en transacciones y acciones atómicas se concluye con la presentación del algoritmo general de consolidación a dos fases (protocolo de Commit a 2 fases).

El protocolo Commit a 2 fases es una técnica básica para implementar acciones atómicas y es aplicable para casi cualquier operación multiprocesos [Bobak 95].

En la figura 8 se muestra el diagrama del protocolo Commit a dos fases. En este diagrama se puede apreciar el intercambio de mensajes entre el proceso maestro y los procesos esclavos, en las dos fases claramente diferenciadas.

Supongamos que existen un proceso maestro y N procesos esclavos. El algoritmo muestra dos fases. En la primera fase el proceso maestro envía peticiones a los N procesos esclavos requiriendo la ejecución de algunas operaciones.

Cada esclavo verifica si puede llevar a cabo sus peticiones. Si puede almacena la petición y el estado inicial del objeto relevante, asegura este objeto (de tal forma que ninguna petición de otros procesos maestros pueden interferir) y envía un mensaje confirmando la capacidad de efectuar el trabajo solicitado. De otra forma envía un mensaje rehusándose a ejecutar el trabajo solicitado.

La segunda fase comienza cuando todas las respuestas de los esclavos han llegado. En este punto el maestro verifica si todos los esclavos pueden desarrollar las tareas requeridas. Si pueden el maestro les informa que las lleven a cabo. En otro caso, esto es, cuando uno o más esclavos rechazan efectuar una operación solicitada, el maestro aborta la acción. Todos los esclavos se ven forzados a desbloquear su objeto y restaurarlo en su estado inicial. De esta forma no se efectúa trabajo alguno y ningún objeto es cambiado.

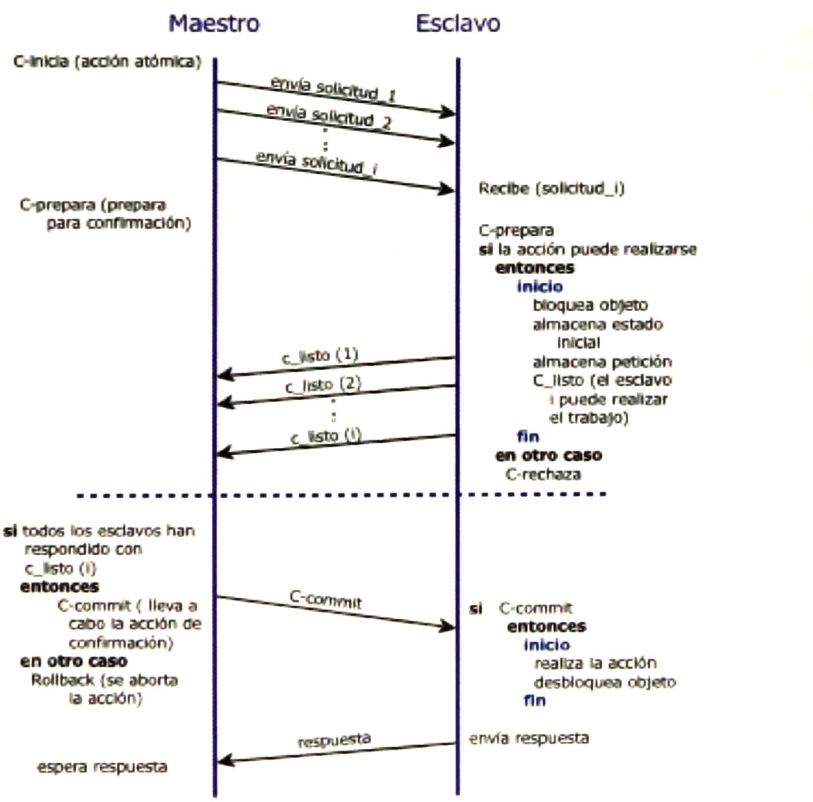


Figura 8. Diagrama del protocolo de Commit a dos fases.

Existe solamente un problema de confiabilidad, cuando un esclavo tiene problemas después de recibir una petición y antes de llevar a cabo su trabajo. En este caso un esclavo puede recuperarse utilizando su estado inicial y petición.

3.4 Transacciones en Tarjetas Inteligentes

Una transacción en una tarjeta electrónica debe reflejar las mismas propiedades y conservar las mismas características que una transacción en un ambiente distribuido [Bellare 95a], esto es, debe presentar la característica ACID de cualquier transacción: atomicidad, consistencia, aislamiento (isolation) y durabilidad.

En esta discusión, se asume que del lado de la aplicación (el lector de la tarjeta que permite registrar los datos en el sistema global) el manejo de la información se realiza de manera transaccional.

Este trabajo sólo trata con el aspecto de manejo de transacciones entre el lector y la tarjeta inteligente. Más aún exclusivamente con el aspecto de mantener la información coherente del lado de la tarjeta.

Para un análisis lo más sistemático posible, se analiza cada una de las propiedades ACID de la transacción, cuando esta se realiza en una tarjeta electrónica. El orden de las propiedades se alteró para dejar la atomicidad al final, ya que es sobre esta propiedad que depende el buen funcionamiento de la transacción.

3.4.1 Consistencia.

La consistencia se define como la habilidad de un sistema de pasar de un estado consistente de una base de datos, a otro estado consistente. La coherencia de los datos es dependiente de la aplicación que los interpreta, pero bajo un conjunto de funciones de transformación predecibles los datos también deben ser predecibles.

Bajo esta óptica, la coherencia de los datos en la tarjeta es una función de la escritura correcta o incorrecta de los mismos y no de un proceso propio de la tarjeta.

Se basa en el hecho de que el procesamiento se realiza en el lado del lector, es decir en el lado de la aplicación central y del hecho de que los resultados se transmiten a la tarjeta en este sentido. En cuanto a los resultados de la aplicación, si los datos se escriben correctamente en la tarjeta son coherentes.

El problema pues, es el de garantizar la estructura correcta de los datos en la memoria de la tarjeta electrónica, lo cual solo es posible si la tarjeta permanece dentro del campo magnético del lector de tarjetas. El algoritmo propuesto asegura que un error en la escritura de los datos no destruye la

coherencia de los mismos si la memoria flash se escribe en bloques y la zona de monedero electrónico es confiable.

3.4.2 Aislamiento

El aislamiento de una transacción implica que si existe un conjunto de transacciones ejecutándose de manera concurrente, el efecto de su ejecución es equivalente a si se hubieran ejecutado de manera serial.

Esta propiedad se mantiene de manera natural porque solo una transacción se realiza a la vez entre la aplicación central y una tarjeta inteligente. Entonces no depende de nada más y por lo mismo esta propiedad se da por garantizada. Si las transacciones son acciones atómicas entonces se observa un efecto de serialización en el orden de ejecución de las transacciones.

3.4.3 Durabilidad

La durabilidad implica que los datos una vez almacenados permanecen sin cambios. Esta propiedad implica un mecanismo de almacenamiento persistente, el cual del lado de la tarjeta inteligente se garantiza por la memoria Flash.

Esta propiedad es intrínseca a la tarjeta y depende de la confiabilidad de la tecnología de la memoria de la tarjeta eléctrica. Esto es, la confiabilidad de los materiales de las cuales esta constituida, así como de sus propiedades físicas y eléctricas, las cuales se asumen como de un alto grado de confiabilidad. Es común encontrar tarjetas con memorias que almacenan eficientemente información durante un lapso promedio de diez años y pueden llegar a tener hasta cien mil ciclos de lectura/escritura.

3.4.4 Atomicidad

La Atomicidad es la base de las otras propiedades. Debido a la posibilidad siempre latente de fallas en el canal de comunicación, a errores aleatorios inducidos por el movimiento de las tarjetas al ser leídas por un lector que requiere que la tarjeta este dentro de su campo, es complicado garantizar esta propiedad.

En las tarjetas inteligentes del tipo MIFARE se cuenta con una zona llamada de Monedero Electrónico, la cual cuenta con mecanismos electrónicos de respuesta rápida, casi instantánea de actualización de la información. Esta zona es muy confiable, aún cuando no puede decirse que sea 100% libre de posibles errores en su manejo.

Sin embargo, esta zona está orientada al uso de puntos, si la aplicación requiere del manejo de información más estructurada tiene que usar algunos de los bloques de memoria que existen en la arquitectura de una Tarjeta Inteligente. Estos bloques no presentan el nivel tan alto de confiabilidad de la zona de Monedero Electrónico y son más susceptibles a errores.

Debido a que en las condiciones normales de trabajo no puede garantizarse la atomicidad de la escritura de los datos por los errores eventuales debidos a problemas con la comunicación, se requiere de uno o varios mecanismos que aseguren ya sea que la transacción se realiza de manera completa o que entonces mejor no se efectúe en lo absoluto.

La identificación de estos problemas y la contribución para su solución ha sido la motivación más importante para el trabajo de esta tesis.

3.5 Soporte electrónico de las Tarjetas Inteligentes

3.5.1 Estructura de memoria de la Tarjeta

En las smartcards, la memoria es el equivalente al disco duro de una PC. La mayoría de las smartcards utilizan memoria EEPROM ya que les brinda una gran flexibilidad de operación.

La memoria es la parte más variable en el diseño de una tarjeta inteligente [Dhem 96]. Es dividida en áreas de acuerdo con el tipo de semiconductor de la memoria:

La memoria de solo lectura ROM se utiliza para almacenar el programa fijo de la tarjeta, también se le conoce como su máscara. En ocasiones se piensa que la máscara actúa como el sistema operativo de la tarjeta, pero casi siempre es solo un programa.

Parte de la memoria ROM conocida como ROM de usuario puede estar disponible para programas de aplicación que se ejecutan en la tarjeta. Esta es eficiente en cuanto a requerimientos de espacio y energía.

La memoria programable de solo lectura PROM es utilizada para cargar números seriales en la tarjeta u otros valores fijos. Esta parte de memoria usualmente es muy pequeña (hasta 32 bytes) y requiere poco espacio y energía.

La memoria flash algunas veces se utiliza para almacenar programas adicionales ejecutados en la tarjeta, así como bloques de almacenamiento de datos que serán leídos y extraídos todos al mismo tiempo.

Esto es debido a que la memoria flash puede escribirse utilizando la interfaz normal de comunicaciones de la tarjeta, pero puede ser borrada sólo como un bloque simple. La memoria flash es más eficiente en espacio y energía que la EEPROM, pero menos que la ROM. Actualmente no es muy utilizada en las tarjetas inteligentes, sin embargo su uso es cada vez más común.

La memoria EEPROM, es utilizada más comúnmente para almacenamiento de datos. Es el equivalente al disco duro de una PC en una tarjeta inteligente, puede ser leída y escrita en cualquier momento, pero es muy costosa en espacio y en energía. Sin embargo, es lo que en mayor medida determina la flexibilidad de funciones de la tarjeta inteligente, incluso hoy las tarjetas son provistas con hasta 64Kbytes de EEPROM.

Para muchos chips de tarjetas inteligentes, la EEPROM ocupa una gran porción del área total del chip. Existen 2 parámetros importantes para la vida y confiabilidad de un chip de una tarjeta: el número de ciclos de escritura que acepta de manera confiable la EEPROM y el periodo de retención de datos (el periodo durante el cual la memoria retiene datos de manera confiable.

Aunque las cuotas puestas para estos parámetros son un poco elevadas (típicamente 10,000 ciclos y 10 años respectivamente, ambos se determinan estadísticamente y en una gran población de tarjetas algunas comenzarán a mostrar poca confiabilidad antes de los límites establecidos.

Memoria de Acceso Aleatorio (RAM), es utilizada para almacenamiento temporal. Los datos en la RAM se pierden cuando la tarjeta es removida. A diferencia de las computadoras donde normalmente se tiene acceso a decenas de Megabytes de memoria, las tarjetas inteligentes tienen tamaños de memoria de 128 a 1024 bytes.

RAM Ferro-eléctrica (FRAM o FERAM), comienza a utilizarse en tarjetas inteligentes. Este tipo de memoria consiste en RAM con una capa adicional, que tiene el efecto de hacerla no volátil. Por lo tanto puede ser utilizada en lugar de la EEPROM, sobre la cual tiene dos grandes ventajas:

1) El tiempo de escritura es el mismo que el de lectura: nanosegundos comparados con milisegundos de la EEPROM.

2) La energía requerida para la escritura es mucho menor que para la EEPROM, esto es particularmente importante para las tarjetas sin contacto porque reduce el requerimiento total de energía del chip.

La FRAM tiene mucha mayor densidad de empaquetamiento que la EEPROM y puede ser memoria volátil o no volátil. Esto le permite al fabricante diseñar la memoria completa como un solo bloque designando áreas como ROM y RAM de manera personalizada.

La FRAM por muchos años fue limitada por el estricto control del poseedor de la patente; ahora ha sido desarrollada por otros productores y se utiliza en chips de tarjetas inteligentes estando disponible desde 2001. Los tamaños de memoria de 256Kbytes y superiores se prevén con el uso de la FRAM.

Las diferentes aplicaciones requieren variar las proporciones de estos diferentes tipos de memoria, por consecuencia, las tarjetas se dividen en aplicaciones dependiendo de estas cantidades, así como también por la energía y funciones incluidas en la máscara.

Por ejemplo, una tarjeta con mucha memoria flash puede ser utilizada para almacenar un historial de operaciones, almacenar expedientes de salud o telemetría. Aplicaciones tales como las bancarias requieren una gran cantidad de ROM por la complejidad de los programas, mientras que la nueva generación de módulos de seguridad para teléfonos móviles requiere de gran cantidad de áreas de EEPROM, para almacenar perfiles del suscriptor así como contraseñas, agendas de teléfono e historial de llamadas.

Existen dos parámetros importantes que determinan la vida y confiabilidad de los chips de las smartcards: el número de ciclos de escritura que la memoria puede aceptar y el periodo de tiempo durante el cual la memoria puede retener los datos almacenados. En este caso, la memoria puede tener aproximadamente 100,000 ciclos confiables de escritura y tiene un periodo de retención de datos promedio de 10 años.

3.5.2 Memoria Flash

La capacidad de memoria usual de una tarjeta sin contacto es de 2048 bits, dispuestos en 32 bloques de 64 bits cada uno, por lo que cada bloque de memoria está constituido por 8 bytes (0 – 7).

De los 32 bloques disponibles (0 – 31), en el bloque 0 se almacena de fábrica un número de serie del chip; los siguientes 5 bloques (1 5) se reservan para definir la cantidad de aplicaciones que tendrá la tarjeta, se especifican los límites de la zonas de memoria para cada aplicación, se definen las claves con las que serán accedidas cada una de las zonas de memoria de las aplicaciones: llave de débito y llave de crédito, se incluyen datos del fabricante así como otros parámetros de la tarjeta y configuraciones adicionales del chip.

Es importante señalar que en este caso, el bloque 2 de la memoria es la zona de escritura segura de la tarjeta. Los 26 bloques restantes se utilizan como zonas de crédito o débito, según la aplicación lo requiera.

	Byte number within a block							
Block	0	1	2	3	4	5	6	7
0	Serial Number (64 bits)							
1	Applications Limit	Application 16-bit OTP Area		Block Write Lock	Tuning Cap	1Fh	E.A.S	Fuses
2	Secure Stored Value Area							
3	Debit Key							
4	Credit Key							
5	Application Issuer Area							
6	Block Write Lockable Application Area							
7								
8								
9								
10								
11								
12								
13	Application Area							
14								
-								
31								

Figura 9. Mapa de memoria típico de una smartcard.

La figura 9 muestra el mapa típico de memoria de una smartcard. En este caso coincide con el mapa de memoria de la tarjeta utilizada para la prueba del algoritmo propuesto en este trabajo.

3.5.3 Zona de Monedero Electrónico

Dadas las características de la memoria, el bloque 2 es la zona donde se almacena el valor actual del contador del monedero electrónico. Con un tamaño de 64 bits, dicho bloque suele ser suficiente para aplicaciones sencillas, y su escritura resulta en general confiable, al menos para la mayoría de los casos.

Sin embargo, en ciertas ocasiones es necesario realizar la escritura de datos complejos fuera de esta zona de memoria, lo que trae como consecuencia escrituras no seguras que pueden poner en peligro la confiabilidad del sistema.

La zona del monedero electrónico está formada por dos áreas de 32 bits cada una. Esta zona puede ser utilizada con un valor de hasta 65534, recargable 65535 veces.

La zona de monedero electrónico es una zona segura y resistente a la ausencia de energía debido a su principio de trabajo de dos etapas.

Podemos asumir que si se cumple que se ha llevado a cabo una autentificación exitosa y los comandos UPPDATE y READ se ejecutan y son terminados por el certificado criptográfico correcto, entonces es imposible recargar accidentalmente, de manera deliberada, o bien, perder unidades del contenido de la zona del monedero electrónico.

3.5.4 Procesador

El procesador es el subensamblaje más importante de todo microcontrolador. Ejecuta las instrucciones de máquina en el orden definido por el programa y realiza accesos a memoria. El termino CPU (Unidad Central de Procesamiento) es utilizado comúnmente como sinónimo de procesador.

Los procesadores utilizados en las smartcards no son desarrollos especiales sino módulos probados que han sido utilizados en otras áreas por mucho tiempo. No es usual desarrollar nuevos procesadores para áreas de aplicación específica, en virtud de que esto generalmente es muy costoso. Adicionalmente podría resultar en un desarrollo de un procesador completamente desconocido para el cual no existirían quizás librerías de funciones disponibles para los creadores de sistemas operativos.

Además los procesadores de las smartcards deben ser extremadamente confiables. Por esto es mejor confiar en procesadores más antiguos que han sido probados en la práctica, que experimentar con los desarrollos más recientes de los fabricantes de semiconductores.

Los procesadores tecnológicamente más avanzados no son utilizados en las smartcards por las razones que acaban de enunciarse. Una cantidad total de 200 mil transistores en un chip se considera muy alta en el contexto de las smartcards.

Dado que el tamaño de memoria dimensionable en una smartcard queda entre 5kb y un máximo de 30, el uso de un bus de memoria de 8 bits no implica restricciones significativas. Los procesadores se basan en una arquitectura CISC (Computadoras con un conjuto de Instruccione Complejo).

Esta arquitectura requiere varios ciclos de reloj por instrucción máquina y usualmente tiene conjuntos de instrucciones muy extensos. El rango de direcciones de procesadores de 8 bits comúnmente es de 16bits con lo cual pueden ser direccionados un máximo de 65536 bytes. Los conjuntos de instrucciones del procesador utilizados se basan en la arquitectura del Motorola 6805 o del Intel 8051, sin embargo pueden ser agregadas instrucciones suplementarias por parte del fabricante de semiconductores.

Existe un procesador de smartcards que es una excepción a las arquitecturas recién mencionadas. Se trata del chip Hitachi H8 el cual emplea un procesador de 16bits cuya arquitectura y conjunto de instrucciones es similar a una máquina RISC (Computadora con un conjunto reducido de instrucciones).

Los procesadores actuales comienzan a tener arquitecturas de 32 bits especialmente utilizando código de programa interpretado por software como las implementaciones actuales con tecnología Java.

3.6 Descripción del Problema

El problema radica en el manejo de datos complejos que exceden las capacidades de almacenamiento de la zona del monedero electrónico y que requieren de una gran confianza en su escritura.

El manejo de los datos en la zona de monedero electrónico se soluciona por hardware lo que reduce las probabilidades de error en la escritura. Sin embargo reducir no quiere decir eliminar y aun cuando para la mayoría de situaciones prácticas el problema no existe, siempre está presente la posibilidad del error.

El problema se presenta cuando la tarjeta no permanece el tiempo suficiente en el campo de acción del lector, típicamente de 150 milisegundos, en este tiempo debe efectuarse la autentificación de la tarjeta y del sistema, el intercambio de información de la tarjeta al lector, el proceso de la misma y el envío de la información de regreso a la tarjeta. Todo esto utilizando el protocolo descrito en los estándares.

En general esto implica el uso de procesadores dedicados para el manejo de la comunicación con la tarjeta y lo mismo para el proceso de la información, se asume que los procesadores tienen suficiente potencia de cómputo, donde suficiente depende de la aplicación y no acepta una definición única.

Si bien la zona de monedero electrónico funciona satisfactoriamente para aplicaciones basadas en puntos, cuando la aplicación requiere el manejo de datos complejos como puede ser registro de la hora, manejo de diferentes servicios, se requiere el almacenamiento en las zonas de usuario definidas.

En el monedero electrónico existen dos copias del contador, una copia activa y otra de respaldo, el mecanismo de hardware asegura (en lo posible) que las dos copias se modifican o que ninguna se modifica. La solución se orienta a implementar un mecanismo equivalente por medio de un protocolo que garantice este comportamiento.

3.7 Solución propuesta

La solución propuesta se inspira en el protocolo de commit a 2 fases, el cual ha sido ampliamente usado en aplicaciones que requieren garantizar un nivel muy alto de confiabilidad.

Igualmente aprovecha el mecanismo de control del monedero electrónico. Esta solución eventualmente debe ser implementada por medio de hardware para incrementar su fiabilidad.

Retomando lo expuesto anteriormente en este mismo capítulo, el protocolo de Commit a dos fases consiste en una primera ronda de intercambio de mensajes que aseguran la preparación de los datos para ser consolidados. La segunda fase es la fase de consolidación (Commit) en esta fase se envía la orden de consolidar. Los detalles del protocolo se presentan a continuación.

3.7.1 Algoritmo 2PSW

El protocolo propuesto al cual llamaremos 2 Phases Secure Writing o 2PSW considera que:

a) El mecanismo del monedero electrónico es atómico y que solo permite decrementar o recargar completamente el valor del contador de puntos.

b) Existen dos zonas de escritura dentro de la estructura de la tarjeta, en la práctica se requieren al menos dos bloques de memoria.

En la figura 10 se muestra el pseudocódigo del algoritmo 2PSW.

Bajo estas consideraciones el protocolo funciona de la siguiente manera:

Sean bloque 0 y bloque 1 los bloques necesarios para este protocolo. Sea S1 el contador de puntos del monedero electrónico.

El contador S1 solo puede decrementarse, y si es par señala como activo al bloque 0; si es impar el bloque activo es el bloque 1.

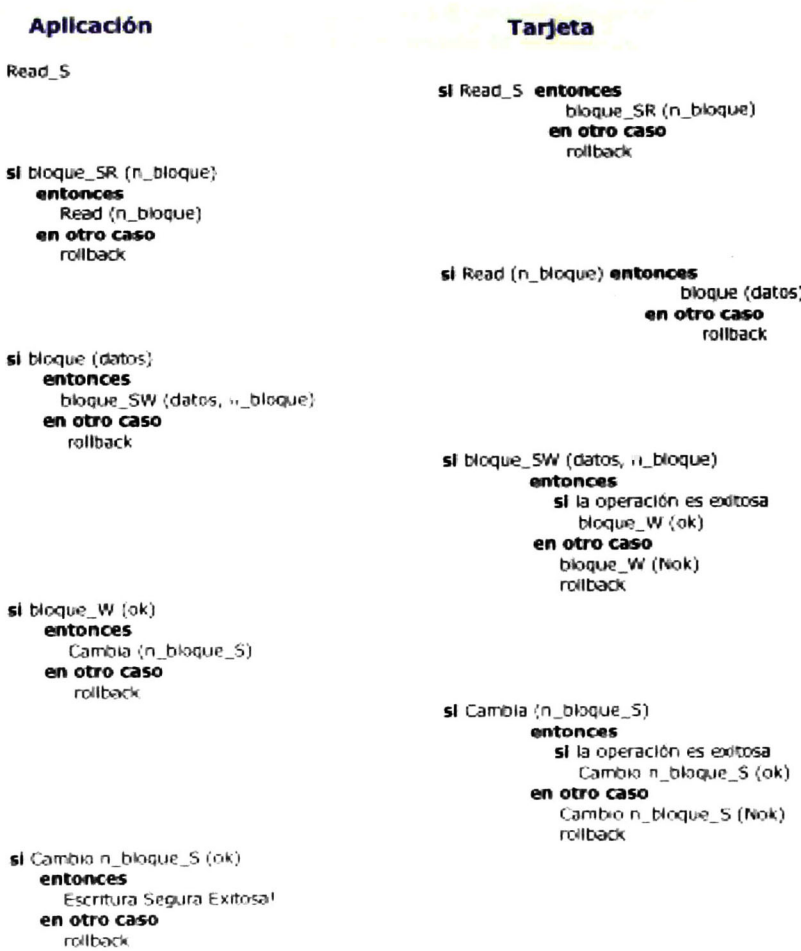


Figura 10. Pseudocódigo del algoritmo 2PSW.

Existe la fase de tratamiento, en esta fase se lleva a cabo la autentificación del sistema, si es exitosa se leen los datos sujetos de tratamiento y se lee el contador del monedero electrónico.

Suponga que S1 es par, en este caso se escribe en el bloque 0, si la operación es exitosa se altera S1 el cual apunta ahora como bloque activo al bloque 1.

Asuma que no es posible escribir en el bloque 0, el contador sigue apuntando al bloque 0 y no hay pérdida de información.

Suponga que exista una falla y no es posible escribir en el bloque 1, en este caso la información del bloque 0 sigue vigente. Si la escritura es exitosa entonces se altera el contenido de S1 y el nuevo bloque activo es el bloque 0.

La solución se basa en la hipótesis de que la escritura de S1 es atómica, y que igualmente no existe la posibilidad de alterar parcialmente el contenido de ninguno de los bloques de memoria. Esto es posible de garantizarse por el uso de memoria flash dentro de la tarjeta.

3.8 Prueba del Algoritmo

Este problema se puede asimilar al problema del consenso en sistemas distribuidos cuando el canal de comunicación puede fallar o el tiempo de comunicación es impredecible.

Bajo estas hipótesis, el consenso es imposible y en consecuencia no existe ninguna solución a este problema. Por lo tanto, no existe tal cosa como una prueba formal de este protocolo. Sin embargo a pesar de esta limitación es posible establecer bajo la siguiente hipótesis H1, una prueba informal de la correctud del protocolo.

Hipótesis H1: Una tarjeta T que inicia una transacción se mantiene en el campo o regresa a él sin que exista una transacción intermedia.

Sea una transacción iniciada por la presencia de una tarjeta T, por simplicidad considere que el bloque activo es el uno, i.e. el contador S1 es par, la aplicación solicita la lectura del bloque 1, lo procesa y solicita la escritura, la tarjeta lo escribe en el bloque 2, si no hay error modifica S1 el cual ahora indica que el bloque activo es el bloque 2. No hubo errores por lo que la información es correcta.

Ahora suponga que un error existe en el momento de escribir en el bloque 2, la transacción se anula, los datos activos siguen siendo los del bloque 1, los cuales no han sido alterados i.e. son validos. La aplicación puede de manera segura anular la transacción.

Suponga que se escribe en el bloque 2 y al actualizar el contador S1 la operación falla, el contador S1 sigue apuntando al bloque 1, los datos son coherentes y la aplicación es informada del error y puede anular la transacción.

Esta discusión puede extenderse al caso del bloque 2, sin perdida de generalidad.

En el evento de que la tarjeta no se encuentre en el campo en algún momento puede ocurrir que el contador S1 sea actualizado pero la aplicación reciba un mensaje de no tarjeta en el campo, la probabilidad de este evento es tan remota que prácticamente no se pudo detectar en los experimentos realizados. Sin embargo, esta posibilidad impide ofrecer una prueba absoluta de este algoritmo, la cual ría en contradicción con los resultados obtenidos y reportados para sistemas asíncronos.

3.8.1 Limitaciones Teóricas

La solución propuesta sin embargo es una solución práctica a este problema. No lo elimina porque es imposible garantizar que la tarjeta siempre y en todas condiciones estará dentro del campo de la tarjeta, más aún es imposible asegurar un límite en el tiempo de procesamiento, por ejemplo en el tiempo máximo de comunicación, lo que nos lleva a tener un sistema asíncrono.

Más aún, ni siquiera el mecanismo del monedero electrónico puede garantizar un 100% de confiabilidad en la escritura del contador de puntos, debido obviamente a este problema. Claro está, que en la mayoría de las situaciones prácticas este problema no se presenta y el uso del monedero electrónico es bastante confiable.

3.9 Conclusiones del capítulo

El uso de tarjetas inteligentes sin contacto se basa en asegurar que las transacciones se han realizado completamente, lo cual no es tarea fácil por las condiciones de uso de las mismas. Este capítulo presenta un algoritmo que incrementa la fiabilidad en las transacciones cuando estas involucran datos complejos. El algoritmo asume un medio poco fiable y propone un mecanismo para asegurar que la transacción mantuvo la característica de la Atomicidad.

En particular este trabajo se dedica a proponer un mecanismo que permita establecer una transacción confiable para tarjetas sin contacto. El problema nace por la comunicación inalámbrica entre el lector y la tarjeta, este tipo de mecanismo asume que la tarjeta permanecerá en el campo del lector el tiempo suficiente para que la transacción pueda efectuarse.

Desafortunadamente esta condición no puede ser siempre satisfecha, sobre todo en condiciones de paso de usuarios como en sistemas de control de acceso o sistemas de transporte. Más aún no sólo existe este problema sino que en ambientes ruidosos la comunicación puede verse afectada. Debido a este problema, es que se propone el algoritmo descrito en este texto.

El protocolo 2PSW presentado en este trabajo se inspiró en el muy conocido protocolo de COMMIT a dos fases (2PC), el cual se utiliza primordialmente en sistemas distribuidos.

Como en el 2PC, el 2PSW se compone de dos fases, una de preparación para la escritura que en ambos casos consiste en tener los datos en una zona de donde puede pasarse a su consolidación, y de una etapa de consolidación o commit, en la cual se asegura que la versión escrita es la final y es correcta.

A diferencia del 2PC la confirmación en el 2PSW se basa en el hecho de que es posible cambiar la versión que se guarda en la zona de Monedero Electrónico. Confiando ciegamente en este mecanismo, lo que se traduce en una posible debilidad en su implementación.

La diferencia básica entre ambos protocolos radica que el 2PC en el evento de una falla se bloquea asumiendo que eventualmente los sitios regresarán y las operaciones podrán ser deshechas o consolidadas, asegurando la coherencia y la consistencia de los datos.

Esta situación no puede garantizarse en las tarjetas inteligentes, una vez que la tarjeta salió del campo de lectura no es posible garantizar su regreso y por lo mismo no es posible almacenar la información para eventualmente restituirla a un estado coherente con el de la aplicación.

Debido a esta característica el protocolo que se presenta aquí requiere de al menos un mecanismo confiable que garantice que cualquier error en la escritura de la información se detecta. Este mecanismo afortunadamente existe y es el que se encuentra implementado en la zona de Monedero Electrónico.

Capítulo 4. Pruebas y Resultados

4.1 Formalización del protocolo

Este capítulo está orientado a presentar los resultados de la formalización del algoritmo propuesto en este trabajo. La formalización es un paso necesario para asegurar la corrección del trabajo, dentro de los límites teóricos de su aplicabilidad.

Como ya se estableció en el capítulo 3 no existe un algoritmo que garantice el consenso en un ambiente distribuido en el que el medio puede fallar, o que presente un comportamiento asíncrono. Este es el caso del trabajo de esta tesis.

Aún cuando pudiera pensarse en utilizar un modelo de asincronía virtual [Anderson 94] este no es el caso en el ambiente analizado, ya que aún con el bajísimo tiempo de procesamiento de una transacción en la práctica se detectaron problemas debido principalmente a esta asincronía.

Por esta razón no es posible presentar una prueba de la corrección del algoritmo, entendiendo la corrección como el comportamiento transaccional en todas las circunstancias. Debido a esta imposibilidad, el trabajo de esta sección se orienta a formalizar el trabajo de la tesis, utilizando para ello técnicas de descripción formal y una evaluación práctica de la eficiencia del algoritmo en condiciones reales.

Para este fin se realiza una presentación del lenguaje de descripción formal. Con Estelle se realizó una especificación formal, la cual se validó lo más extensamente posible corroborando las conclusiones teóricas previas. Para complementar este estudio se realizó una serie de experimentos que permiten evaluar el comportamiento del algoritmo en condiciones prácticas.

4.1.1 Estelle

ESTELLE, Extended State Transition Langage o lenguaje de transición de estados extendidos es un lenguaje de descripción formal basado en el lenguaje PASCAL y las máquinas de estados extendidos, en el que la extensión implica la adopción de mecanismos para expresar la comunicación entre máquinas [Buttzac 94].

ESTELLE ha sido objeto de estandarización por parte de la ISO y es ampliamente utilizado para la definición de estándares de protocolos de comunicación.

Un sistema distribuido se compone de niveles de descripción anidados, cada nivel pudiendo contener varios módulos comunicantes a la ayuda de canales. El nivel de descripción más elevado representa

al sistema bajo la forma de un módulo único conocido como la especificación. El cual demanda, en general, la creación de uno o varios niveles de refinamiento suplementarios (módulos hijos)

En consecuencia un sistema distribuido se representa:

Por un lado en términos de arquitectura, es decir por medio de un conjunto de módulos que intercambian informaciones a través de canales de comunicación vía puertos de comunicación llamados puntos de interacción.

Por otro lado en términos de componentes, quienes describen la evolución del sistema (las partes declaración, inicialización y transición).

Uno de los puntos fuertes de la técnica de descripción formal ESTELLE es su potencia de expresión del paralelismo y de la comunicación, así como la estructuración dinámica de un sistema en módulos, los módulos en submódulos y así consecutivamente. Esta característica hace de esta técnica una herramienta adecuada para la modelación de mecanismos que puedan ser puestos en práctica en aplicaciones reales.

4.1.1.1 Modelado en ESTELLE

En esta sección se proporciona un panorama general del lenguaje ESTELLE, no es un tutorial y no pretende cubrir todos los aspectos del lenguaje. Mayores detalles pueden encontrarse en [Buttzac 94].

4.1.1.2 Introducción al lenguaje ESTELLE

Un sistema distribuido se especifica en ESTELLE por una arquitectura jerarquizada de componentes secuenciales no deterministas (llamadas instancias de módulo), quiénes hacen intercambios de mensajes a través de puntos de interacción asociados a cada módulo.

Los objetos principales de la arquitectura son:

a) Los canales.

Los canales definen al conjunto de interacciones que pueden sen intercambiadas entre módulos. Toda definición de canal comprende dos grupos de interacción, un grupo por cada módulo asociado al canal.

b) Los puntos de interacción.

Los intercambios de mensajes entre módulos se efectúan a través de los puntos de interacción, una fila FIFO no limitada se asocia para el almacenamiento de los mensajes recibidos.
La definición de un punto de interacción debe considerar al menos la referencia del indicador de canal y un identificador del rol que define cuales son las interacciones que pueden emitirse.

c) Los módulos.

Un módulo se especifica en Estelle por:

a. La definición del encabezado del módulo.

La vista externa de un módulo está definida por la declaración del encabezado que contiene al menos la palabra clave MODULE seguida de su identificador. El encabezado del modulo puede contener también el nombre la clase (SYSTEMPROCESS, SYSTEMACTIVITY, PROCESS, ACTIVITY) así que las declaraciones de los puntos de interacción y las variables exportadas (variables internas visibles desde el exterior). En una especificación ESTELLE el atributo SYSTEM para un módulo implique el módulo es el padre de la especificación.

b. La definición de un cuerpo de módulo.

Al menos una definición de cuerpo de un módulo está declarado para cada definición de encabezado de módulo. Una definición de cuerpo de módulo está compuesta de tres elementos:

i. **La parte declaración.**
Esta parte contiene de un lado, las declaraciones Pascal y de otro lado las declaraciones que son propias a Estelle para tener en cuenta los sistemas distribuidos (canales, módulos, variables de módulo, puntos de interacción). Una declaración de cuerpo puede contener declaraciones de otros módulos, la aplicación reiterada de este principio induce una estructuración jerárquica de la definición de módulos.

ii. **La parte inicialización.**
La parte inicialización especifica los valores iniciales de ciertas variables del módulo con las cuales cada nueva instancia de ese módulo se genera.

iii. **La parte transición.**
La parte transición describe en detalle la máquina a estado extendidos que representa el comportamiento interno de los módulos. Una transición Estelle puede ser considerada como siendo compuesta de tres partes:

1. La parte precondición.
Que permite definir las condiciones de tiro de la transición considerada. La cláusula FROM indica el estado mayor de salida. La cláusula WHEN permite describir las condiciones de tiro de una transición a la llegada de un mensaje. La cláusula PROVIDED permite definir una expresión booleana como condición de tiro de la transición.

2. La parte postcondición.
Que permite definir el estado mayor siguiendo la ejecución de la transición (cláusula TO).

3. La parte acción.
Caracterizada por un bloque BEGIN END en la cual es posible incluir instrucciones PASCAL y ESTELLE. La palabra clave OUTPUT traduce la emisión del mensaje a partir del punto de interacción. La ejecución de esta parte es atómica. Una especificación Estelle es dinámica en el sentido que, a la vez y de manera independiente, la arquitectura de las instancias del módulo y la configuración de las ligas entre los puntos de interacción de estas instancias pueden ser modificadas en el curso de la ejecución.

4.1.1.3 Análisis en Estelle

La simulación se utiliza como complemento a la verificación. La simulación permite seguir uno o varios caminos del grafo de accesibilidad, permitiendo analizar los objetos de la especificación.

Evidentemente, entre más fuerte sea la etapa de simulación mas eficaz es la validación de la especificación. Sin embargo, en la realidad el alto grado de no determinismo no permite la validación exhaustiva por simulación.

En resumen, el análisis de la especificación de este trabajo se basó en el análisis del grafo de accesibilidad y en la simulación de la especificación.

4.1.2 Máquina de estados

Para analizar el grafo de accesibilidad del protocolo propuesto se creó una máquina de estados, donde se puede apreciar con claridad cada transición o cambio de estado del protocolo, así como la comunicación que se establece entre la aplicación y la tarjeta a través del medio.

El intercambio de información entre la aplicación y la tarjeta se realiza mediante primitivas de comunicación que pueden tener o no parámetros. En la máquina de estado propuesta se definen operadores que le dan sentido al flujo de los mensajes, que pueden ser de entrada o de salida y mediante otro operador se consigue que cada mensaje conserve una estructura única que facilita la comprensión del algoritmo.

La estructura básica de los mensajes tiene la siguiente estructura:

Condición / Acción

Una condición puede ser:
Recepción de mensaje.
Condición Booleana

Una acción puede ser:

- Emisión de un mensaje
- Ejecución de un procedimiento

Los mensajes pueden emitirse del lado de la aplicación al lado de la tarjeta o viceversa, al igual que ocurre con la emisión de los mensajes, que puede hacerse del lado de la tarjeta al lado de la aplicación o viceversa.

Los operadores utilizados son los siguientes:

?	Implica la recepción de un mensaje
!	Implica la emisión de un mensaje

En la figura 11 se muestra la máquina de estados del protocolo propuesto, y posteriormente se presenta su análisis correspondiente.

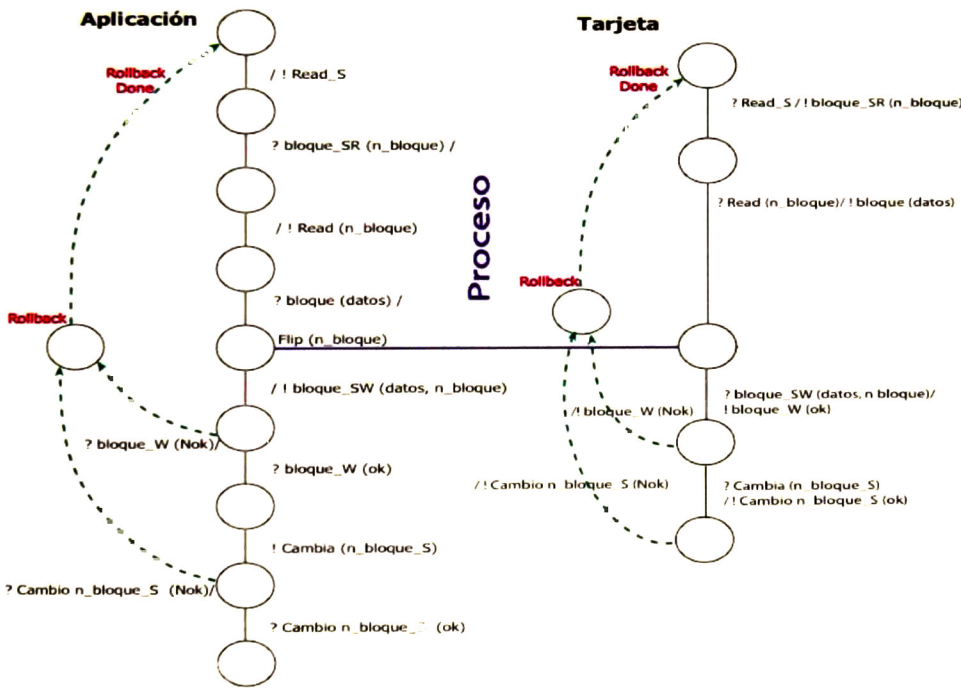


Figura 11. Máquina de estados del protocolo 2PSW.

Para poder hacer un análisis más profundo y exacto de la máquina de estados presentada anteriormente, es necesario describir las primitivas utilizadas, la constitución de las mismas y definir los parámetros que utilizan. En la figura 12 se presenta un análisis de las primitivas utilizadas en la máquina de estados y posteriormente se explica su funcionamiento.

PRIMITIVA	DESCRIPCIÓN	PARÁMETROS
Read_S	Procedimiento que indica la ejecución de una acción de lectura segura.	Ninguno
bloque_SR	Indica que se va a realizar una acción de lectura en el bloque determinado por el parámetro n bloque. Esta lectura se hará en conjunto con una posterior operación de escritura segura.	n_bloque
Read	Lee el bloque determinado por el parámetro n bloque.	n_bloque
bloque	Accede a la información que se le ha indicado. Dicha información se asocia con el parámetro datos.	datos
Flip	Indica el cambio al bloque alterno de la tarjeta.	n_bloque
bloque_SW	Indica la realización de una acción de escritura segura de la información representada por el parámetro datos, en la localidad designada por el parámetro n bloque.	datos, n_bloque
bloque_W	Indica si se ha realizado con éxito una operación de escritura. Cuenta con un parámetro tipo booleano.	Tipo Booleano: ok, Nok
Cambia	Cambia de posición el indicador de zona activa de memoria. Esta primitiva es la que consolida la seguridad en la operación de escritura del protocolo.	n_bloque_S
Cambio n_bloque_S	Indica si la primitiva Cambia se ha realizado con éxito. Cuenta con un parámetro tipo Booleano.	Tipo Booleano: ok, Nok

Figura 12. Análisis de las primitivas de comunicación del protocolo 2PSW.

Una vez presentada la máquina de estados y la descripción de las primitivas, queda claro el funcionamiento del protocolo.

La comunicación siempre es iniciada del lado de la aplicación, donde se hace una solicitud de lectura segura a la tarjeta, para su posterior escritura segura en la zona de memoria correspondiente.

La tarjeta recibe la petición de lectura segura y responde solicitando a la aplicación le informe acerca del bloque al cual se debe acceder para tal efecto, la aplicación indica el bloque seleccionado y la tarjeta le envía el dato correspondiente.

En este punto todavía no se ha comenzado a utilizar el protocolo de escritura segura propuesto, y el desarrollo anterior no es relevante para este trabajo de tesis, por lo que se puede considerar como un proceso previo.

Suponiendo el caso de que ocurra algún error en cualquier fase de operación de los hasta ahora descritos, tenemos que aún no se ha solicitado la operación de escritura y obviamente no se ha requerido el protocolo de escritura segura propuesto, por lo tanto, un análisis no sería requerido en esta etapa.

Sin embargo, a partir de la ejecución del proceso Flip, que cambia al bloque alterno de la zona de memoria, comienza el protocolo de escritura segura a dos fases. La primera fase comienza con la emisión de un mensaje de escritura segura por parte de la aplicación a la tarjeta, indicando el dato a escribir y el bloque correspondiente.

En caso de que del lado de la tarjeta se haya logrado escribir correctamente el dato, se envía un mensaje de escritura exitosa, en cualquier otro caso se genera un error y se reinicia el proceso mediante un procedimiento denomidado "rollback". Si la aplicación no recibe el mensaje de escritura exitosa por parte de la tarjeta, se reinicia mediante el proceso de rollback, abortando la escritura y generando el mensaje al usuario correspondiente.

Si la aplicación recibe el mensaje de confirmación de escritura exitosa por parte de la tarjeta, entonces le solicita que realice un cambio de bloque de memoria, dando inicio a la segunda fase del protocolo.

La tarjeta recibe este mensaje y en caso de que la realización del cambio de bloque haya sido exitosa envía otro mensaje de confirmación. Si en cualquier punto ocurre un error o no se recibe el mensaje de éxito, se aborta la escritura.

Si la aplicación recibe un mensaje de éxito de parte de la tarjeta, informando que se ha realizado el cambio de bloque, entonces se considera consolidada la escritura segura. El protocolo funciona, porque garantiza que únicamente cuando se realiza una escritura exitosa, entonces se cambiará el bloque activo de la memoria, en otro caso, el bloque nunca cambiará y el dato existente en el bloque (que corre el riesgo de perderse si no existe la verificación), queda intacto en caso de algún error.

4.1.3 Arquitectura

La arquitectura del protocolo de escritura segura se puede apreciar en la figura 13, donde se distinguen claramente tres zonas: la tarjeta, la aplicación y el medio. Todas ellas dentro del ámbito del protocolo.

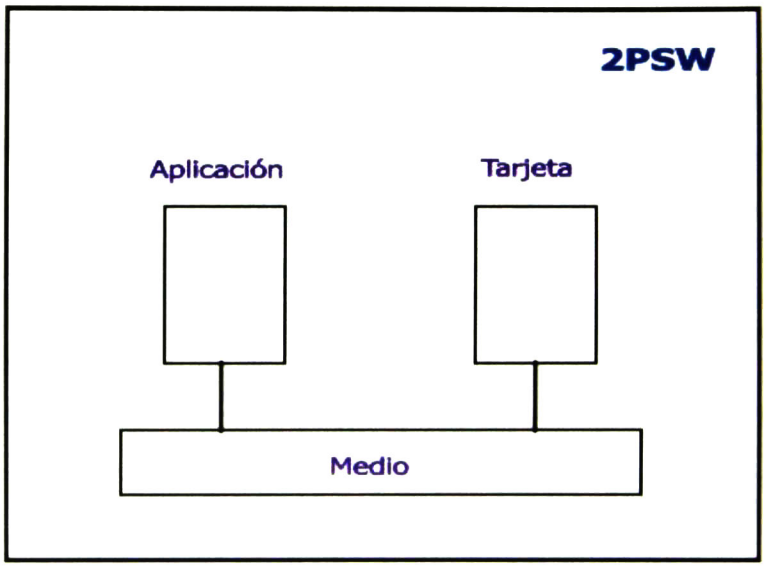


Figura 13. Arquitectura del protocolo 2PSW

La aplicación y la tarjeta se comunican por medio de las primitivas, intercambiando los mensajes necesarios para la consolidación de una escritura segura en dos fases, interactuando como se describe en la máquina de estados.

El medio representa todas las condiciones ajenas al protocolo, a la aplicación y a la tarjeta. Representa todas las condiciones de operación y actores del algoritmo, como pueden ser un ambiente ruidoso o el mismo usuario que de alguna forma incide de manera directa en la operación del mismo en un escenario real, porque como ya se demostró en el capítulo anterior, hay variables que no se pueden controlar por el protocolo y sí por el usuario, como el asegurar que la tarjeta siempre permanezca en el campo del lector durante el tiempo que dure la operación.

4.2 Resultados

4.2.1 Pruebas experimentales

Las pruebas realizadas se llevaron a cabo mediante la implementación de un pequeño laboratorio de pruebas, donde se configuró un mecanismo giratorio conformado por un motor con brazos extensores, un lector de tarjetas y varias tarjetas sin contacto, mismo que se puede apreciar en la figura 14.

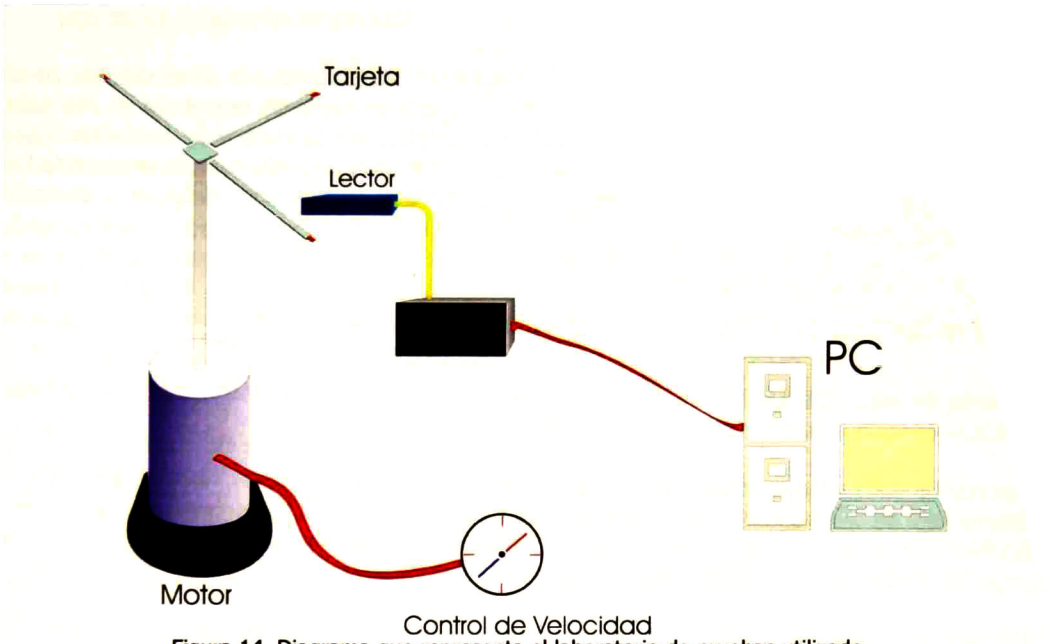


Figura 14. Diagrama que representa el laboratorio de pruebas utilizado.

Se desarrolló un script donde se programaban transacciones que se traducían en operaciones de escritura en la zona de memoria de las tarjetas. Estas transacciones consistían en incrementar o decrementar un valor dado, que estaba almacenado en la zona de memoria de las tarjetas sin contacto.

Las pruebas eran muy sencillas, ya que consistían en colocar una tarjeta sin contacto en el brazo extensor del motor. Dentro del rango de pruebas se consideró en un primer caso, mantener la tarjeta siempre presente en el campo del lector. Esto con el objeto de que el único factora considerar como variable no controlable fuera el tiempo de proceso.

En un segundo caso, el brazo extensor se hacía girar a diversas velocidades entrando y saliendo del campo del lector.

Es muy importante señalar que el objetivo principal de las pruebas experimentales realizadas era encontrar los errores críticos en las transacciones, esto es, contabilizar las transacciones erróneas que sucedían. De ninguna manera se pretendía contabilizar los mensajes de error generados por no haber podido efectuar una transacción por alguna razón justificable o entendible, como el que la tarjeta no esté presente en el campo del lector.

Lo que sí se pretendía detectar era la cantidad de veces que fallaban los algoritmos empleados en las pruebas. Sabemos que por el tipo de aplicaciones en las que se emplean las tarjetas sin contacto, donde cada vez más se requiere escribir de manera segura datos en las zonas no protegidas de la memoria, es crítico el que existan inconsistencias en los datos y mecanismos de escritura de los mismos.

Si consideramos por ejemplo, un sistema de monedero electrónico, la idea de las pruebas experimentales realizadas es detectar cuántas veces el algoritmo evaluado regresaba un mensaje de tarjeta no presente o escritura no exitosa cuando en realidad la escritura sí se había realizado, o bien detectar si el algoritmo regresaba un mensaje de escritura exitosa cuando en realidad ésta nunca había ocurrido. Este tipo de inconsistencias son las que se buscaron detectar y contabilizar con las pruebas realizadas.

4.2.2 Ambiente de experimentación

El ambiente de experimentación fue un ambiente controlado, ya que se manipularon y controlaron las variables en todo momento. Las transacciones fueron atómicas.

Es importante señalar que cada transacción implicaba la autentificación, lectura de los datos, procesamiento de los mismos, así como su correspondiente escritura dentro de la tarjeta. En condiciones normales, esto es, en condiciones normales de operación, con la tecnología empleada, el tiempo típico que toma el realizar una transacción es de 150 ms.

En ambientes hostiles, como un ambiente industrial, un ambiente ruidoso o donde existan condiciones atmosféricas, de temperatura o presión que sean adversas, no se realizaron pruebas significativas, puesto que no se consideró relevante para el propósito de este trabajo. Además, no se contaba con los medios y mecanismos para efectuar las pruebas en dichos ambientes de experimentación.

Se realizaron pruebas en dos escenarios, con algunas variantes en cada uno de ellos. Los escenarios eran los siguientes:

Escenario 1
En este escenario la tarjeta sin contacto siempre permaneció dentro del campo del lector, esto quiere decir que no se presentaba el problema de falla en el canal de comunicación, al menos porque la tarjeta no estuviese presente por tiempo suficiente en el campo del lector.

En este escenario general, se hicieron pruebas utilizando un algoritmo de escritura convencional de datos en las tarjetas. Asimismo, se hicieron pruebas utilizando el algoritmo de escritura segura presentado en este trabajo de tesis.

Escenario 2
En el segundo escenario la tarjeta sin contacto entraba y salía del campo del lector. En este escenario se agregó la complejidad de que la tarjeta no siempre estaba en el campo del lector el tiempo suficiente para terminar la transacción.

Se realizaron pruebas para probar el comportamiento y eficiencia del algoritmo de escritura segura presentado en esta tesis, al igual que el mismo algoritmo de escritura convencional utilizado en el escenario anterior. En este escenario se manejaron algunas variantes en cuanto al tiempo y se registraron los resultados del desempeño de los algoritmos en cada uno de los escenarios empleados.

Para la realización de las pruebas, el script empleado básicamente consistía en una aplicación sencilla que se ejecutaba del lado del lector y que se encargaba de realizar diversas acciones en la tarjeta, como cargar un valor determinado en la tarjeta, y luego irlo decrementando o incrementando en cada transacción realizada.

El script contabilizaba la cantidad total de intentos de transacciones, la cantidad de transacciones que el algoritmo consideraba como exitosas, los mensajes de error generados por intentos no exitosos y se tenía como referencia el dato que se encontraba en la tarjeta. Todo esto se contrastaba y se obtenían los resultados.

4.2.3 Evaluación inicial

La evaluación inicial se realizó empleando el algoritmo convencional de escritura, bajo las condiciones descritas en los escenarios 1 y 2, definidos anteriormente. Para el registro de los resultados se utilizó el script antes mencionado, desarrollado expresamente para estos fines.

En el escenario 1, como la tarjeta estaba siempre presente en el campo del lector, la prueba consistía únicamente en realizar una transacción cada cierto tiempo, situación que estaba contemplada en la aplicación de prueba del algoritmo.

La aplicación cargaba en la tarjeta un valor igual a la cantidad de transacciones que estaban programadas por realizar, y cada vez que el algoritmo efectuaba una transacción exitosa, se decrementaba ese valor, o en su defecto se generaba un mensaje de error. Se registraron los resultados de cada una de las transacciones.

Caso similar fue lo que se realizó en el escenario 2, donde la prueba consistía en que la tarjeta sin contacto se colocaba en el brazo extensor del motor, que giraba a distintas velocidades, entrando y saliendo del campo del lector. El script cargaba un valor inicial en la tarjeta y cada transacción incrementaba el valor cargado.

De manera adicional, el script de aplicación generaba un mensaje de error para cada transacción que no fuera completada de manera exitosa, sin importar las causas y registraba la cantidad de transacciones exitosas que el algoritmo generaba.

4.2.4 Evaluación del 2PSW

La evaluación del algoritmo de escritura segura descrito en este trabajo de tesis fue sometido a las mismas pruebas que el algoritmo de escritura convencional utilizado como contraparte. Se realizaron algunas modificaciones al script para que pudiera realizar las transacciones conforme lo establece el protocolo 2PSW.

Al igual que con la prueba del algoritmo anterior, el script registraba las transacciones que el algoritmo consideraba exitosas, los mensajes de error o de transacción no realizada (sin importar la causa), las transacciones que se tenían registradas en la tarjeta y la cantidad total de transacciones que efectuaban.

4.2.5 Comparación de resultados

Los resultados obtenidos por el script VB de prueba, permitieron comparar el desempeño del algoritmo de escritura convencional y el algoritmo 2PSW en los dos escenarios propuestos, con dos variantes en el primer escenario (denominadas 1A y 1B)y tres variantes en el segundo escenario (denominadas 2a, 2b y 2C).

Los resultados obtenidos del primer escenario donde se consideraron las variantes 1A y 1B, se muestran en las figuras 15 y 16. En este primer escenario, como se ha dicho, la tarjeta no sale del campo del lector y se consideró en la primera variante un muestreo de 30 transacciones por minuto, y para la segunda variante, una frecuencia de muestreo de 60 transacciones por minuto.

Estos resultados se muestran a continuación:

Escenario 1A. Tarjeta siempre presente en el campo del lector. **30** Transacciones por minuto.			
Algoritmos Evaluados	Parámetros Evaluados		
	Cantidad de errores por cada 5,000 Transacciones	Cantidad de errores por cada 10,000 Transacciones	Cantidad de errores por cada 20,000 Transacciones
Algoritmo Convencional	0	1	2
2PSW	0	0	0

Figura 15. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 1A.

Escenario 1B. Tarjeta siempre presente en el campo del lector. **60** Transacciones por minuto.			
Algoritmos Evaluados	Parámetros Evaluados		
	Cantidad de errores por cada 5,000 Transacciones	Cantidad de errores por cada 10,000 Transacciones	Cantidad de errores por cada 20,000 Transacciones
Algoritmo Convencional	0	2	5
2PSW	0	0	1

Figura 16. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 1B.

Mientras tanto, en el segundo escenario, donde la tarjeta entra y sale repetidamente del campo del lector, se consideraron tres variantes: 2A, 2B y 2C.

En la variante 2A, se consideró una frecuencia de muestreo de 15 transacciones por minuto. Los resultados obtenidos de la comparativa correspondiente se muestran en la figura 17.

En la variante 2B, se consideró una frecuencia de muestreo de 30 transacciones por minuto. Los resultados obtenidos de la comparativa correspondiente se muestran en la figura 18.

En la variante 2C, se consideró una frecuencia de muestreo de 60 transacciones por minuto. Los resultados obtenidos de la comparativa correspondiente se muestran en la figura 19.

Escenario 2A. Tarjeta que entra y sale del campo del lector. **15** Transacciones por minuto.			
Algoritmos Evaluados	Parámetros Evaluados		
	Cantidad de errores por cada 5,000 Transacciones	Cantidad de errores por cada 10,000 Transacciones	Cantidad de errores por cada 20,000 Transacciones
Algoritmo Convencional	1	3	7
2PSW	0	1	2

Figura 17. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 2A.

Escenario 2B. Tarjeta que entra y sale del campo del lector. **30** Transacciones por minuto.			
Algoritmos Evaluados	Parámetros Evaluados		
	Cantidad de errores por cada 5,000 Transacciones	Cantidad de errores por cada 10,000 Transacciones	Cantidad de errores por cada 20,000 Transacciones
Algoritmo Convencional	2	5	13
2PSW	0	1	3

Figura 18. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 2B.

Escenario 2C. Tarjeta que entra y sale del campo del lector. **60** Transacciones por minuto.			
Algoritmos Evaluados	Parámetros Evaluados		
	Cantidad de errores por cada 5,000 Transacciones	Cantidad de errores por cada 10,000 Transacciones	Cantidad de errores por cada 20,000 Transacciones
Algoritmo Convencional	4	11	27
2PSW	1	2	4

Figura 19. Resultado de la comparación del algoritmo de escritura convencional y el algoritmo 2PSW en el escenario 2C.

4.3 Conclusiones del Capítulo

En este capítulo se describe el algoritmo propuesto desde el punto de vista formal, utilizando el lenguaje de descripción formal Estelle para tal fin.

Derivado de la descripción formal del algoritmo, surgió la máquina de estados, el diagrama de primitivas de comunicación utilizadas, con su significado y se presentó la arquitectura del protocolo.

Es necesario señalar que para el caso general, la eficiencia del algoritmo convencional que se utilizó en este capítulo para comparar el desempeño del protocolo 2PSW, es suficientemente buena para la mayoría de los casos. El problema surge cuando es necesario garantizar la escritura en zonas no protegidas de la memoria de la tarjeta.

Como se pudo apreciar en el comparativo de desempeño, la cantidad de errores de consistencia en los datos de la memoria de la tarjeta que presentan los algoritmos de escritura convencionales, derivados principalmente de un problema de configuración en el tiempo de espera por la confirmación de la escritura y los problemas en la comunicación, son comunes a todas las tarjetas sin contacto, independientemente del fabricante de las mismas.

Los resultados mostrados son satisfactorios para el 2PSW, ya que el margen de error es mínimo, por lo que este algoritmo es adecuado en cuanto a la funcionalidad que provee para su implementación con aplicaciones que requieran alta confiabilidad en sus transacciones, particularmente en las operaciones de escritura. Por esta razón, este algoritmo puede ser utilizado de manera confiable para trabajar con aplicaciones que requieran utilizar las zonas de memoria no protegidas de la tarjeta.

Capítulo 5. Conclusiones y Propuestas de Trabajo Posterior

El trabajo de tesis presentado abordó el uso de tarjetas inteligentes sin contacto o de proximidad para la implementación y prueba formal de un algoritmo de escritura segura para esta tecnología.

Una de las desventajas que presenta la tecnología de tarjetas inteligentes sin contacto radica en la confiabilidad que proveen [Anderson 94].

Se ha comprobado que la escritura de datos complejos para cierto tipo de aplicaciones no es confiable en lo general, lo cual reduce de manera considerable la cantidad y tipo de aplicaciones que se pueden implementar con esta tecnología, sobre todo si se considera que la inversión que se debe realizar para participar en proyectos de esta naturaleza es bastante grande.

Pese a esta desventaja, la tecnología de tarjetas sin contacto está siendo muy utilizada, debido al corto tiempo promedio que toma realizar una transacción en una smartcard sin contacto.

Especialmente en los últimos años se han diversificado los escenarios reales de aplicación de esta tecnología. Lo anterior debido a que resultan ideales para cierto tipo de aplicaciones donde se requiere agilidad en el desarrollo de las transacciones y donde obviamente resultaría impráctico el uso de la tecnología de tarjetas de contacto. Recordemos el ejemplo del sistema de transporte público o un sistema de control de acceso a espacios físicos.

La tecnología de tarjetas sin contacto permite que se realicen más transacciones en menor tiempo y su costo, se estima que se reducirá en el corto plazo y será lo suficientemente accesible como para considerar su uso masivo entre la población.

Durante el desarrollo del trabajo de investigación, se puede concluir que el problema encontrado (poca confiabilidad en la escritura) es común y frecuente cuando se trata de tecnología de smartcards sin contacto, sin importar el fabricante o tipo de aplicación de las mismas.

Este trabajo de tesis aborda el problema de la poca confiabilidad en las operaciones de escritura con practicidad y propone el diseño, implementación y prueba de un algoritmo de escritura segura optimizado y adaptado para el manejo de zonas no protegidas de la memoria de las tarjetas.

Este protocolo denominado 2PSW (Two Phases Secure Writing Protocol) resolvió una problemática real que se presenta de manera recurrente en la tecnología de smartcards sin contacto. Esta situación se presenta durante la ejecución de las diferentes aplicaciones que utilizan la escritura de datos en zonas de memoria diferentes a la zona de monedero electrónico.

Se reciben mensajes de error en la escritura cuando ésta en realidad se ha realizado efectivamente, o bien en el caso contrario, cuando se reporta que se ha efectuado una escritura y no es así.

Desafortunadamente, el algoritmo propuesto no garantiza un 100% de confiabilidad, porque como se puede apreciar claramente en el capítulo 4, no puede existir una solución totalmente confiable.

Ni siquiera el mecanismo de confirmación implementado por hardware en el monedero electrónico puede garantizar ese porcentaje de confiabilidad. Esto porque no se pueden controlar particularmente dos variables:

> Que la tarjeta permanezca en todo momento y bajo cualquier condición dentro del campo de la terminal
>
> Que todas las veces la tarjeta permanezca dentro del campo de la terminal durante el tiempo necesario para completar la transacción

Sin embargo, dadas las hipótesis explicadas a detalle en el capítulo 3 y 4, el protocolo 2PSW propuesto es una solución práctica que permite aumentar la confiabilidad.

Mediante la combinación de los mecanismos de escritura confiable de los protocolos analizados (particularmente el protocolo commit a dos fases utilizado en sistemas distribuidos) y el manejo del hardware para el monedero electrónico se obtuvieron resultados interesantes y alentadores, como lograr reducir la tasa de error a 0 en 10,000 transacciones si la tarjeta siempre permanece en el campo de lectura y a 1 en 10,000 transacciones si la tarjeta entra y sale del campo de lectura a razón de 2 transacciones por segundo.

La solución del problema que se presenta con este trabajo es confiable porque se basa en mecanismos probados como lo es el protocolo de Commit a dos fases y en el empleo del hardware especializado para el monedero electrónico y los resultados obtenidos en las pruebas, como se ha visto, reducen de manera importante el problema de garantizar la escritura de datos complejos en tarjetas inteligentes sin contacto, lo cual marca la pauta para poder ampliar el rango de aplicaciones de esta tecnología.

Si consideramos la prueba formal del protocolo, en este trabajo se realizó la especificación del protocolo 2PSW en un lenguaje de descripción formal. En este caso se utilizó un lenguaje denominado Estelle, donde se pudo verificar la correctud del mismo.

Por otro lado, se creó una aplicación que permitió mediante la experimentación descrita en el capítulo 4, corroborar que el algoritmo propuesto sí presenta realmente una mejora respecto a un algoritmo convencional de escritura en zonas de memoria no protegidas en el ámbito de las tarjetas inteligentes sin contacto.

Adicionalmente, si consideramos que en la mayoría de las situaciones prácticas no se presenta el problema de falta de confiabilidad en la escritura y si tomamos en cuenta que en general el uso del monedero electrónico es bastante confiable, podemos concluir que la utilización del algoritmo propuesto es altamente efectivo para prácticamente cualquier situación y ambiente de implementación.

El objetivo de este trabajo de tesis era encontrar un algoritmo genérico, abstracto, libre de implementación, que redujera el porcentaje de fallas y que incrementara la confiabilidad de la escritura en zonas de memoria diferentes a la del monedero electrónico de la tecnología de las tarjetas inteligentes sin contacto.

En general, podemos decir que se cumplió el objetivo del trabajo de tesis, ya que se encontró un algoritmo con las características recién mencionadas que incrementa la confiabilidad de las transacciones de escritura en la tecnología de las smartcards sin contacto.

El trabajo debe continuar para una mejor evaluación de la confiabilidad obtenida, ya que puede no resultar suficiente con los procedimientos experimentales llevados a cabo. Además también resultaría interesante conocer el comportamiento del algoritmo en otros escenarios diferentes a los presentados en este trabajo de tesis.

Además es conveniente continuar el trabajo de investigación en este tema para analizar la conveniencia de trasladar la implementación del algoritmo 2PSW en hardware y abrir así la posibilidad de creación de un nuevo tipo de smartcard con mejores características que las actuales, ya que incorporaría la función de escritura segura en zonas de memoria no protegidas en la tarjeta.

Referencias Bibliográficas

[Anderson 94] Anderson, R.J., "Why Cryptosystems Fail in SC", Communications of the ACM, Vol. 37, No.11, 1994, pp.32-40

[Anderson 95] Anderson, R.J., and S. J. Bezuidenhout, "Cryptographic Credit Control in Pre-Payment Metering Systems" IEEE Symposium on Security and Privacy, 1995, pp. 15-23

[Anderson 96b] Ross J. Anderson, Markus G. Kuhn: Communications of the SC, USENIX Workshop, Noviembre 1996

[Buttzac 94] Buttzac, Valdimir, "Extended State Transition Language", 1994.

[Balic 98] Balic, Mijit, "ISO and rules", The Communication ' s book, 1998.

[Baller 99] Baller, M.J., and Y. Yacobi, "Fully-Fledged Two-Way Public Key Authentication and Key Agreement for Low-Cost Terminals", Electronics Letters)Uk, Vol.29, No.11, pp. 999-1001

[Bellare 95a] Mihir Bellare, Juan Garay,Ralf Hause, Amir Herzberg,Hugo Krawczyk,Michael Steiner, Gene Tsudik,Michael Waidner:
iKP –A Family of Secure Electronic Payment Protocols, SC Workshop, 1995

[Bellare 95b] Mihir Bellare,Philip Rogaway: Optimal Asymmetric Encryption – How to encrypt with RSA, Internet,1996

[Biham 91] Eli Biham, Adi Shamir : Differential Crypto analysis of DES-like Cryptosystems, Journal of Cryptology, Vol. 4, N°.1,1991

[BIS 96] Bank for International Settlements: Security of Electronic Money – Report by the Committee on Payment and Settlement Systems and the Group of Computer Experts of the Central Banks of the Group of Ten Countries. Basel, Agosto 1996

[Bobak 95] Bobak, Angelo R., "Distributed and Multi-Database Systems", AH ed., 1995.

[Burmester 92] Burmester, M., Y. Desmedt, and T. Beth, "Efficient Zero-Knowledge Identification Schemes for Smart Cards", Computer Journal, vol.35, No.1, 1992, pp.21-29

[Carlson 99] Carlson, R. "Utilizing Smart Cards and PKI Technology to Solve E-Business Security Issues" corporate publication, Authentic 8, Doncaster (VIC, Autralia), 1999

[Dhem 96] J.F. Dhem. D. Veithem, J.J. Quisquarter: SCALPS: Smart Card Applied to Limited Payment Systems, UCL Crypto Group Technical Report Series,Université Catholique de Louvain, 1996

[Frank 96] Frank, J.N., "Smart Cards Meet Biometrics", Card Technology, Sep./Oct. 1996, pp. 30-38

[Gosling 99] James Gosling, Henry McGilton:
The Java Card Technology – A White Paper, Sun Microsystems, USA, 1999

[Hendry 97] Hendry M. "Security Is More Than a Card Game", Smart Card´97 Conference Proceedings, QMS, Peterborough, England, 1997

[Holloway 91] Holloway, C., "The IBM Personal Security Card", Smart Card´91 Conference Proceedings, Agestream, Peterborough, England, 1991

[IC Protection 97] Common Criteria for IT Security Evaluation Protection Profile-Smartcard Integrated Circuit Protection Profile, Internet,1997

[Janson 95] Janson, P. And M. Waidner, "Electronic Payment System" SI Informatik/Informatique, Switzerland, Vol.3, 1995, pp.10-15

[Lim 99] Lim, C.H., et al., "Smart Card Reader", IEEE Transactions on Consumer Electronics, Vol.39, No.1, pp.6-12

[MIFARE] http://www.mifare.net

[Mondex 95] Mondex International, "Mondex: Security by Design", corporate publication, London , 1995

[multos] http://www.multos.com/

[Pfaffenberger 97] Bryan Pfaffenberger: Dictionary of Computer Terms, Simon & Schuster/Macmillan, New York,1997

[Rivest 78] Ronald L. Rivest, Adi Shamir, Leonard Adleman: Method for Obtaining Digital Signatures and Public- Key Cryptosystems, Internet, 1976

[Schaumüller-Buchl 91] Schaumüller-Buchl, I., "Card Security: An Overview", Smart Card 2000 Conference Proceedings, North Holland, Amsterdam, 1991, pp.19-27

[Schneier 96] Schneier, B., Applied Cryptography: Protocols, Algorithms and Source Code in C, New Yhork: Wiley, 1996

[Simmons 93] Gustavus J. Simmons: The Subliminal Channels in the U.S Digital Signature Algorithm , Proceedings of Symposium on State and Progress of Research in Cryptography, Rome,1993

[Sociedad 93] Sociedad Interbancaria de Servicios, S.A., "Multibanco Electronic Purse-Description of Scheme", corporate publication, Lisbon, Portugal, 1993,

[Stocker 98] Thomas Stocker: Java for Smart Cards in Tagungsband Smart Cards, Vieweg Verlag, Braunschweig, 1996

[Thomasson 96] Thomasson, J.P., "Advances in Smart Card IC Tecnology" SGS-Thomson technical article TA164, Paris, France, 1996,

[Vedder 97] Klaus Vedder, Franz Weikmann: Smart Cards- Requirements, Properties and Applications, ESAT-COSIC course, Katholische Universität Leuven, 1997

[Waidner 95] Waidner, M., "Secure Billing and Payment over the Internet", Rapperswil Networking Forum, Rapperswi, Switzerland, 1995,

[Weikmann 98] Frankz Weikmann, Klaus Vedder: Smart Cards Requirements, Properties and Applications, in: Tagungsband Smart Cards, Vieweg Verlag, Braunschweig,1998

[Wiener 93] Michael J. Wiener: Efficient DES Key Search,Crypto 93, Santa Barbara, CA,1993

[Yellin 96] Frank Yellin: Low Level Security in Java, Internet, 1996

Anexos

Anexo 1. Implementación del Protocolo 2PSW en Código Estelle Extendido (Versión 4.16)

```
specification t2psw;

default individual queue;
channel U(R1,R2);

by R1: put;
channel S(R1,R2);
  by R1: read_s;
  readn_bloque;
  flipn_bloque;
  bloque_swdatosn_bloque;
  cambian_bloque_s

by R2: bloque_srn_bloque;
bloquedatos;
bloque_wok;
bloque_wnok;
cambion_bloque_sok;
cambion_bloque_snok;

module E systemprocess;
  ip P1: S(R1);
     P2: S(R2);
end;

body E1 for E;
state s0, s1, s2, s3;

initialize to s0 begin output P1.read_s end;
```

```
trans when P2.bloque_srn_bloque
  from s0 to s1
  begin output P1.readn_bloque end;

trans when P2.bloquedatos
  from s1 to s2
  begin output P1.flipn_bloque; output P1.bloque_swdatosn_bloque end;

trans when P2.bloque_wok
  from s2 to s3
  begin output P1.cambian_bloque_s end;

trans when P2.bloque_wnok
  from s2 to s0
  begin output err; initialize end;

trans when P2.cambion_bloque_sok
  from s3 to s0
  begin initialize end;

trans when P2.cambion_bloque_snok
  from s3 to s0
  begin output err;
  initialize end;
end;

module R systemactivity;
  ip P3: S(R2);
     P4: S(R1);
end;

body E2 for R;
state s0, s1, s2, s3;

initialize to s0 begin end;

trans when P4.read_s
  from s0 to s1
  begin output P3.bloque_srn_bloque end;
```

```
trans when P4.readn_bloque
  from s1 to s2
  begin output P3.bloquedatos end;

trans when P4.bloque_swdatosn_bloque;
  from s2 to s3
  begin output P3.bloque_wok end;
  from s2 to s0
  begin output P3.bloque_wnok;
  output err;
  initialize end;

trans when P4.cambian_bloque_s
  from s3 to s0
  begin output P3.cambion_bloque_snok; output err; output  initialize end;
  from s3 to s4
  begin output P3.cambion_bloque_sok; initialize end;
end;

module UNION systemprocess;
  ip JoinMod;
end;

body UNION_BODY for UNION; external;
modvar X: E; Y: R;
initialize
begin
  init X with E1;
  init Y with E2;
  connect X.P1 to Y.P4;
  connect Y.P3 to X.P2;
end;

end.
```

Centro de Investigación y de Estudios Avanzados del IPN
Unidad Guadalajara

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis:

2PSW Algoritmo de Escritura Segura para Tarjetas Inteligentes Sin Contacto

del (la) C.

Carlos Alberto FRANCO REBOREDA

el día 27 de Febrero de 2004.

Dr. Luis Ernesto LÓPEZ MELLADO
Investigador Cinvestav 3A
CINVESTAV GDL
Jalisco

Dr. Félix Francisco RAMOS CORCHADO
Investigador Cinvestav 2B
CINVESTAV GDL
Jalisco

Dr. Antonio RAMIREZ TREVIÑO
Investigador Cinvestav 2A
CINVESTAV GDL
Jalisco

Dr. Ricardo Raúl JACINTO MONTES
Director
Diseño y Desarrollo Tecnológico S.C.
Jalisco