UNIVERSIDAD DON BOSCO.

VICERRECTORÍA DE ESTUDIOS DE POSTGRADO.

TRABAJO DE GRADUACIÓN

MODELO DE FIRMA DIGITAL BAJO CERTIFICADO DIGITAL PARA PUNTOS DE VENTA. (POS)

PARA OPTAR AL GRADO DE

MAESTRO EN SEGURIDAD Y GESTIÓN DE RIESGOS INFORMÁTICOS.

ASESOR:

MG. ERMIDES URRUTIA LÓPEZ

PRESENTADO POR:

MARIO EDGARDO PLANAS ORELLANA.

Antiguo Cuscatlán, La Libertad, El Salvador, Centroamérica.
Agosto de 2016.

Índice.

Agradecimiento.

Dicha investigación no hubiera sido posible sin la ayuda de Dios y mi Madre María Auxiliadora y Don Bosco, por haberme dado una prueba de vida, que cada día es una lucha sin fin y que cada éxito se disfruta al final de la jornada, con los agradables momentos que podemos vivir.

Agradezco a mis padres (Mario Planas y Reyna de Planas) no solo por el hecho de darme la vida, sino por el hecho de haber sido su hijo. Por cada día que luchan y que comparten para mí y mis hermanos el pan, para que seamos mejores profesionales, pero más que todo, mejores seres humanos.

También agradecer a mis hermanos (Andrea, Rodrigo y mi cuñado Milomir) por ayudarme y compartir conmigo este triunfo y los que se vengan.

A mis demás familiares (tíos, tías y primos) agradecerles su apoyo y sus palabras de aliento a no darme por vencido, si tener miedo ante la adversidad y los obstáculos que se puedan presentar.

A mis amigos que siempre han estado a mi lado, en las buenas y en las malas, también se los agradezco de todo corazón y pido al señor por cada uno, para que pueda alcanzar sus metas de vida sin olvidar las obras que podamos hacer a los demás.

Y finalmente mis profesores de las distintas materias, y asesor de tesis, por haberme dado las herramientas para convertirme en un nuevo instrumento de la filosofía salesiana:

"DAME ALMAS, LLEVATE LO DEMAS"- Don Bosco.

MODELO DE FIRMA DIGITAL BAJO CERTIFICADO DIGITAL PARA PUNTOS DE VENTA. (POS)

Mario Edgardo Planas Orellana,

Universidad Don Bosco, La Libertad, El Salvador.

marioedgardo10@hotmail.com

***Resumen*—El uso de tarjetas de crédito y débito se ha convertido en el método de pago más requerido a nivel comercial. Pero también es uno de los medios de fraude económico que afecta en millones de dólares a los clientes y las empresas. Es por eso que se invierten en el desarrollo de su seguridad, aunque no garantiza un nivel de autenticidad con respecto a la compra, ni el no repudio ante una adquisición de bien o servicio. Por ello, en éste articulo se propone un mecanismo de autenticación que garantice la fiabilidad del pago por éste medio, tanto para el cliente que genera la adquisición, como para la empresa que trabaja en el manejo bancario del plástico y la marca.**

***Índice de Términos*— Autenticación, Certificado, Clave Privada, Clave Pública, Criptografía, Huella, Integridad, POS.**

I. INTRODUCCIÓN.

El mundo del comercio se ha desarrollado de cierta manera un gran auge en el uso de métodos de pago modernos, los cuales ya no son de carácter desconocido para todos. Y en ese nacimiento de nuevos tipos de remuneración, el uso de las tarjetas de crédito y débito toma la vanguardia, ya que forma parte de nuestra vida diaria. Este método ha tomado una enorme fuerza de la cual formamos parte todos, desde el cliente que hace uso de estos, pasando por el comerciante, las empresas emisoras de tarjetas y bancos que utilizan esta infraestructura.

Pero también, dicho proceso forma parte de una puerta más a la generación de fraudes financieros que pueden sufrir los participantes, el cual se estiman en millones de dólares a nivel mundial. Esto en base a diversos métodos, la mayoría de carácter informático que trae como consecuencia el robo de fondos, de tarjetas y de identidad de la persona.

En este contexto, las medidas de seguridad deberán tener un impacto en el manejo de la información. Esto se soluciona mucho con normativas que obligan a las instituciones a poder resguardar la confidencialidad, mantener la integridad y exigir la disponibilidad de este servicio. Pero en este punto es donde además se debe establecer que se pueda garantizar que la información de la transacción sea realizada por el emisor y pueda ser comprobada por el sistema autorizador. Esto se refleja, justamente, en los pilares de la autenticación y el no repudio de origen, los cuales son parte de la seguridad de la información.

Existen métodos donde se pueden realizar este tipo de validaciones, como es el caso de la firma autógrafa, la cual se estampa dentro de un documento generado en el punto de venta del comercio donde se hace la transacción, con lo que representa que el cliente ha realizado la compra. Pero no brinda una mayor eficiencia en la autenticación y el no repudio del proceso comercial.

Es por eso que se presentara una propuesta de firma electrónica basado en el uso de certificado digital, la cual se buscara demostrar que su aplicación pueda ser tomada en diferentes sistemas con la característica de Punto de Venta (POS), de forma física o virtual, en donde se realiza dicho proceso. Para esto se dará a conocer en esta investigación primeramente el funcionamiento de este sistema autorizador de transacciones (Capitulo 2), tomando en cuenta la forma como se transporta la información durante el proceso (Capitulo 3), seguidamente se explicaran los elementos criptográficos del cual se harán uso (Capitulo 4), y se determinara el método de autenticación propuesto (Capitulo 5) para ser implementado bajo una demo de desarrollo (Capitulo 6) y finalmente se demostrará su aplicabilidad en casos de uso (Capitulo 7) con resultados (Capitulo 8) así como su viabilidad bajo normativas y regulaciones relacionadas (Capitulo 9) para demostrar su cumplimiento en relación a los pilares necesarios en la seguridad de la información.

II. FUNCIONAMIENTO DEL SISTEMA POS. ATAQUES Y MITIGANTES.

A. Sistema POS.

Si se quiere conocer todo el proceso nos enfatizamos primeramente en definir el punto de venta (POS), que según [1], como al punto físico donde se ejecuta la transacción económica, es decir, la caja o líneas de cajas. Es en este punto el cual sirve de un intermediario de pago entre el cliente y el comercio, el cual da como ventaja el uso de diferentes sistemas de pago de acuerdo a la necesidad o requerimiento del establecimiento que brinda el bien o servicio. Pero para que estos puedan ejecutarse, se necesita hacer uso de diversos elementos de carácter informáticos especializados mediante una interfaz accesible para los vendedores, donde permiten la creación e impresión del ticket de venta mediante las referencias de productos, realizan diversas operaciones durante todo el proceso de venta, así como cambios en el inventario. También generan diversos reportes que ayudan en la gestión del negocio, conocido como **Terminal de Punto de Venta**, de acuerdo a [2]. Los cuales pueden ser de carácter hardware (dispositivos físicos) y software (sistema operativo y programa de gestión). Y en algunos casos, como ya las empresas consagradas, permite la autorización para el pago con tarjetas de crédito y débito que posteriormente es transferida a las entidades bancarias. Es en este método de pago donde se realizara el estudio.

Se puede identificar en este proceso 5 participantes, como lo explica [3], que tienen diversas funciones a fin de desarrollar exitosamente la transacción:

-Cardholder: Se conoce como la persona dueña de la tarjeta, ya sea de débito o de crédito.

-Mercante: Es considerado así a la tienda departamental la cual posee el POS a la par de la caja registradora donde realizara todas las transacciones originadas por la tarjeta.
Cabe recalcar que en este punto es donde el cliente puede tener un dispositivo físico (PIN PAD), o un software donde se simula las mismas características (POS virtual).

-Adquirente: También conocido como un banco del negocio afiliado. Un adquirente es una institución financiera autorizada que ayuda a un negocio afiliado a cumplir con su obligación de procesamiento para aceptar tarjetas. El adquirente conecta y procesa las transacciones en la red de autorizaciones que le corresponda. El contrato con el negocio afiliado también debe tener el nombre del banco claramente identificado.

-Red de Tarjetas Bancarias: La organización que ofrece esquema de pago con tarjeta y permite a los emisores para emitir tarjetas a los titulares de cuentas. VISA y Mastercard son dos redes de tarjetas bancarias, más famosos. También están Discover, American Express y Star entre otras.

-Emisor: Emisor es la organización bancaria o financiera que mantiene la cuenta del titular de la tarjeta y emite una tarjeta al dueño de la cuenta a nombre de la red de tarjetas bancarias. El proceso del emisor es de una organización financiera que recibe los datos de los titulares de tarjetas y procesa la transacción en nombre del emisor. Un procesador puede procesar para muchos emisores y un emisor puede obtener los servicios de muchos procesadores. La figura 1 muestra la interacción de estos cinco miembros en el proceso de pago, mostrando que entre ellos existe un proceso de petición entre los miembros del sistema y también una respuesta, ya sea de aceptación o de rechazo.

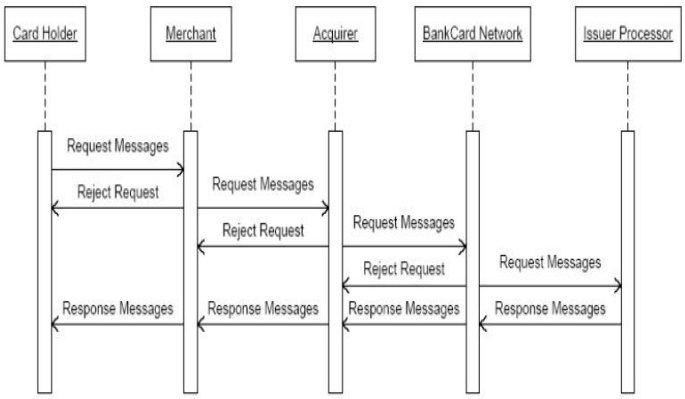


Fig. 1. Diagrama de Secuencia.

Para poder desarrollar un proceso de pago con tarjetas, nuestro punto de venta puede tener integrado un dispositivo **PIN PAD**, el cual es un dispositivo físico modular integrado compuesto, como mínimo, por pantalla, teclado, lector de banda magnética, lector de chips para las tarjetas y conexiones entre cable, inalámbrica o de línea telefónica. Todo esto según los requerimientos establecidos por las certificaciones del Security Standards Council (PCI) y del estándar de interoperabilidad de tarjetas IC o tarjetas con microprocesador (EMV). La figura 2 muestra este dispositivo.

Fig. 2. Dispositivo modular PIN PAD.

Otro método de ingreso al pago con tarjetas es con el uso de software típicamente web o una aplicación Windows que pueden ser utilizados por los comerciantes de tarjetas para autorizar manualmente las transacciones con tarjetas y que están conectadas a los sistemas bancarios, de la misma forma que el PIN PAD, conocido como **POS virtual**, como lo dicta [4]. Esto se puede desarrollar en un PC que tenga los mismos elementos que un sistema modular, con la diferencia que la estructura física será un poco mayor. Pero la ventaja en el uso de esta clase de elemento radica en que se pueden desarrollar sistemas multitasking con el ingreso de las ventas y pueden diseñarse de acuerdo al contexto del negocio.

De estas dos formas se realiza el ingreso de la información de las tarjetas de crédito o débito, los cuales están diseñados bajo componentes físicos y lógicos, como lo dicta [5].
Entre los componentes físicos que conlleva una tarjeta podemos mencionar:

-Logo de la Empresa Emisora.
-Colores o imagen de fondo.
-Número Primario de cuenta embozada. (PAN). Este número es el que lo relaciona con la cuenta de la persona.
-Fecha de vencimiento.
-Nombre del Propietario.
-Valor de verificación de la tarjeta. (CVV2 para VISA, CVC2 para Master Card, CID o 4DBC para AMEX).

La parte lógica de la tarjeta consiste en una banda magnética al reverso, la cual dicha banda se compone de 3 tracks que poseen información sensible no encriptado de la tarjeta. Esto es uno de los puntos débiles a la hora de duplicar dicha información para la ejecución de clonación de tarjetas. La figura 3 muestra la distribución de los tracks en la banda magnética de la tarjeta.

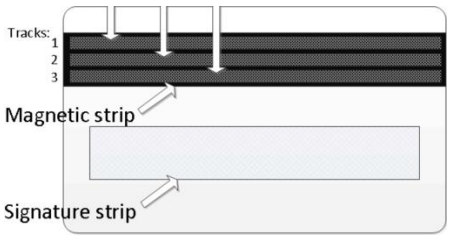


Fig. 3. Lado reverso de Tarjeta de Crédito o Débito con los tracks de banda magnética.

Según la normativa ISO/IEC 7813, el track 1 se compone del PAN, nombre del propietario, fecha de vencimiento en formato mes/año (MM/YY) y datos de discreción.
En el track 2 se diferencia en que no se incluye el nombre del propietario. Fuera de eso, se tiene la misma información. La figura 4 y 5 muestra esa distribución de la información.

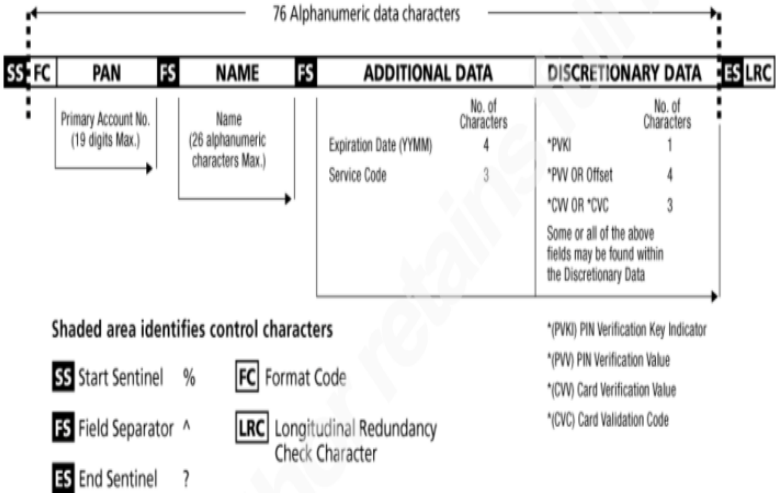


Fig. 4. Track 1 de banda magnética del plástico.

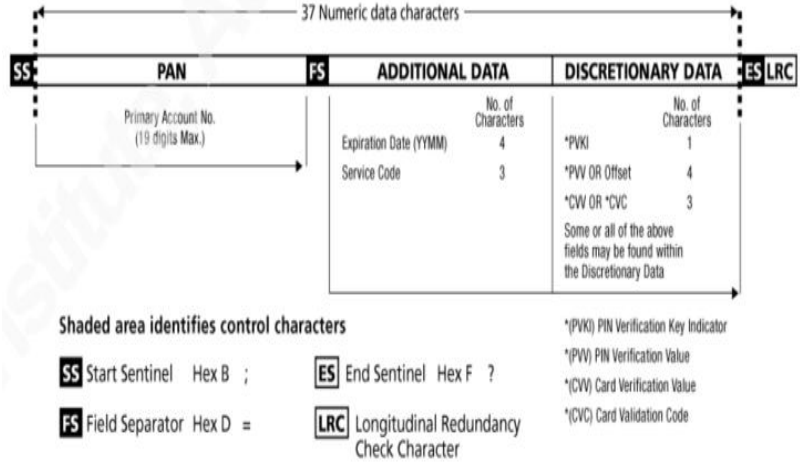


Fig. 5. Track 2 de banda magnética del plástico.

Es decir, que al deslizar la tarjeta por banda magnética tomara esa información junto a la data que se ingrese de forma manual como el monto de la compra y los dígitos del número de embozo con el fin de colocar en el paquete que se enviará a la siguiente etapa.

El adquirente tomara la información brindada de los dispositivos de los que estén conectados, ya sea de forma modular o del aplicativo del cual se encuentre conectado y realizará su función. Esta consiste en determinar primeramente si la institución financiera es la misma donde se encuentra el sistema emisor. De ser así, le enviará la solicitud a dicho sistema para que haga el movimiento de la cuenta. Caso contrario, se procede a enviar a la red de tarjetas bancarias a fin de que ellos determinen el banco emisor del plástico.

En la red de tarjetas bancarias, o mejor conocida como la **"marca"**, se determinará la procedencia de dicha tarjeta. Muchas veces lo que se hace cuando se llegan esas peticiones es que se guían por el número embozado de la tarjeta (véase figura 6), en la parte delantera de la misma.

Fig. 6. Número embozado e impreso en tarjeta para identificarlo.

Dicha identificación, establecida bajo la normativa ISO/IEC 7812, está compuesta por 16 dígitos y dividida en 4 grupos de 4 números cada uno. Pero de dicha composición, se toman los primeros 6 dígitos a fin de identificar la marca dueña de la tarjeta. Dicho conjunto se le denomina como el Número de Identificación del Emisor (INN). Del INN, se necesitan los primeros 3 dígitos a fin de identificar la marca de procedencia de la tarjeta. Por ejemplo, la marca VISA en todas sus tarjetas se identificará con el primer dígito (número 4), como se menciona en [6]. La tabla 1 muestra las marcas más reconocidas en el uso de tarjetas y sus respectivos rangos de identificación de tarjetas en el INN.

TABLA I
RANGO DE INN PARA MARCAS DE TARJETAS.

Red de la Marca.	Rango INN.
American Express	34, 37
Diners Club International	36, 38, 39, 300-305, 309
Master Card	51, 55
VISA	4
China UnionPay	62

Con esta validación determinan si la transacción proviene de otro emisor conectado a su red o de otra marca de tarjetas. Se envía entonces al destino correspondiente. Cuando finalmente se llega al sistema emisor, este primeramente tomará los 6 últimos dígitos del INN, y junto a la fecha de vencimiento y el pin ingresado, realizará un algoritmo matemático con el fin de determinar que la transacción provee de la tarjeta correcta. Esto lo verifica contra el valor que este ya ingresado en las bases del emisor, pero con el inconveniente que no determina si la persona dueña de este desarrolló la compra.

Luego revisa la cuenta del cliente para verificar su conformidad y responde aprobando o negando la operación. Esta respuesta la recibe a través de su banco Adquirente en el punto de venta, donde el cliente deberá firmar el comprobante de pago en forma de estar de acuerdo con la compra. La aprobación implica que el banco emisor acuerda reembolsar el monto de la compra al banco adquirente, quien a su vez lo depositará en la cuenta del comerciante. A todo este sistema se le denomina como Autorizaciones. La figura 7 muestra de mejor manera este procesó.

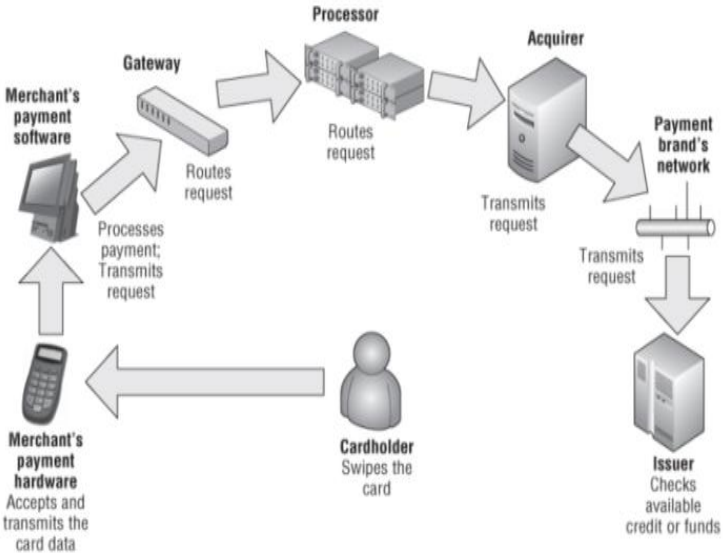


Fig. 7. Sistema Autorizador.

El depósito en la cuenta es realizado al final del día, el punto de venta envía al banco Adquirente el resumen de las ventas efectuadas a través de un proceso batch. También es posible hacerlo a través de resúmenes de venta que el comerciante llena manualmente y que deposita en las ventanillas del banco, a este resumen se anexa los comprobantes firmados por cada uno de sus clientes portadores de la tarjeta habiente. Ya con estos resúmenes se envían a los emisores de tarjetas a fin de determinar los movimientos que se harán con los clientes dueños de las tarjetas. Este proceso se denomina como Asentamiento.

Para el estudio se enfocará en el proceso autorizador, ya que es donde se realiza la mayoría de ataques y robo de información, específicamente en la generación de la firma a fin de que compruebe la identidad del dueño del plástico.

B. Ataques Informáticos y Mitigantes.

En las industrias del sistema de pagos el mayor riesgo que se tiene es sobre la información que se maneja en la transacción, donde la captura de estar información es la fuente que utilizan los delincuentes con el objetivo de generar fraudes. Y dicha captura de información se puede desarrollar, según [7], en varios puntos de la cadena.

-Hackeo en los dispositivos POS: Este tipo de ataque se puede desarrollar mucho antes de la implementación, por ejemplo en la planta del proveedor del producto. Los dispositivos son resguardados por empleados, lo cual dificulta para un atacante instalar un malware, skimmers o un hardware que pueda instalarse. Para esos casos se necesitaría la ayuda de un empleado descontento que tenga acceso a esos puntos o un atacante bien disimulado, con el fin de instalar de manera manual estos elementos que puedan robar la información. Los atacantes pueden hacer uso de terminales de "auto servicio" y de ubicaciones que no estén monitorizados como otras estaciones. Años atrás existían tipos de dispositivos falsos que generaban recibos, por defecto, que informaban a la víctima que un error ha impedido la operación en proceso, cuando en realidad ese mismo sistema ya había sacado toda la información de la tarjeta.

-Hackeo a nivel de Red: La piratería informática a nivel de red puede ser posible a través de diferentes métodos. Uno puede ser a través de conexiones compartidas entre los sistemas en un establecimiento, tales como los POS que comparten la misma conexión con el punto de acceso WI-FI disponible para los clientes. Estos sistemas pueden estar usando WI-FI como por ejemplo en una zona de "back-office" para comunicarse con los servidores. Los puntos de venta podrían, también, utilizar una red WI-FI cerrada, pero los atacantes todavía pueden ser capaces de romper su frase de contraseña. Los atacantes también pueden encontrar un puerto abierto en un interruptor y agregar su propio punto de acceso WI-FI.

Tales agujeros de seguridad son producto del incumplimiento. El estándar de seguridad para el procesamiento de tarjetas de pago (como es el caso de PCI) requiere una conexión segura para dichos puntos, el cifrado de datos de tarjeta, la autenticación para el acceso remoto desde y hacia las maquinas de punto de venta, y muchos otros métodos que garanticen que las operaciones se mantengan a salvo de accesos no autorizados.

Se cabe de recalcar que será en esta área donde nuestro modelo será realizado, como una ayuda a ir mejorando la autenticación del proceso y el no repudio de la transacción.

-Hackeo en Servidores Específicos: A diferencia de una violación en dispositivos o en el nivel de red, una infiltración exitosa en el servidor dará acceso a los atacantes no solo a un punto de venta especifico o una red de sistemas de terminales en un solo lugar, dependiendo de la arquitectura, sino a todos los sistemas de terminales controlados por el minorista en varias ubicaciones. Esto no está exento de dificultades adicionales, sin embargo, necesitan tener acceso a la red antes que puedan llegar a los servidores, además de poder traspasar los firewalls. También puede llevar algo de trabajo para que los piratas informáticos conozcan los softwares disponibles en los servidores y los medios para explotarla.

Entre las recomendaciones de protección en los dispositivos POS se debe:

-Implementar hardware basado en encriptación punto a punto.

-Limitar acceso de internet.

-Deshabilitar acceso remoto.

-Eliminar sistemáticamente datos de titulares de tarjetas.

-Implementar la última versión del sistema operativo con los parches actualizados.

-Emplear listas blancas con el fin de bloquear los sistemas de punto de venta, solamente para sus usos previstos.

-Limitar el acceso interno al dispositivo físico.

-Hacer cumplir las políticas con respecto a la reparación física o actualización del POS.

-Implementar software de seguridad y mantenerlo actualizado con las últimas firmas.

Y para las redes de comunicación con las que se interconectan, se debe:

-Restringir la comunicación dentro y fuera de su entorno para que lo que se requiere.

-Asegúrese de que está constantemente protegido contra vulnerabilidades en los sistemas y aplicaciones, incluso en el medio de ciclos de conexión.

-Identificar cuando un componente del sistema ha cambiado.

-Proteger contra el malware y URL's maliciosas.

-Comunicación encriptado entre aplicaciones y data.

-Continuamente escaneo de aplicaciones WEB para las vulnerabilidades potenciales.

III. ISO 8583.

La ISO 8583 es un estándar que explica el formato y diseño del mensaje de las transacciones financieras originados por las tarjetas. Este formato se debe utilizar, sin importar los diferentes sistemas en que participe. La mayoría de transacciones en ATM's y POS lo utilizan, por lo cual es la base con la cual se formara el protocolo propuesto.

La estructura del mensaje, básicamente, se compone de 3 partes, las cuales explicaremos a continuación:

-Identificador del tipo de Mensaje (MTI): Conformado por 4 dígitos numéricos, los cuales determinan el tipo de mensaje con el cual se identifica el paquete y la función que se realizara con este. Cada dígito tiene función:

*Posición 1 de MTI: Indica la versión del estándar ISO 8583.

*Posición 2 de MTI: Indica el tipo de transacción. Por ejemplo: Financiero, Reversión, Mensaje de Gestión de Red.

*Posición 3 de MTI: Específica la función del mensaje, donde se define la forma que se trabajara dentro del sistema.

*Posición 4 de MTI: Define la localización del origen del mensaje dentro de la cadena de pago.

Con esto se puede definir, en la tabla 2, los tipos de mensaje que se manejan en la ISO 8583 y su significado [8].

TABLA II
MTI's estándar y su significado.

MTI.	Significado.	Uso.
0100	Requerimiento de autorización.	Requerimiento para autorizar compra de POS.
0120	Aviso de Autorización.	POS está roto y Ud. debe firmar un voucher.
0200	Requerimiento Financiero del Comprador.	Requerimiento de fondos, usualmente de un ATM.
0210	Respuesta al Requerimiento Financiero del Comprador.	Respuesta del mensaje de requerimiento de fondos (aprobada o denegada).
0220	Aviso Financiero del Comprador.	e.g. Checkout de un hotel.
0230	Respuesta al Aviso Financiero del Comprador.	Respuesta al mensaje 0220.
0400	Requerimiento de Reverso del Comprador.	Reversa una transacción.
0420	Aviso de Reverso del Comprador.	Aviso de que se realizó un reverso.
0430	Respuesta del Aviso de Reverso del Comprador.	Respuesta al aviso 0420 del Aviso de Reverso.

-Bitmap: Es un mapa compuesto por 64 bits, de forma primaria, hasta un máximo de 192 bits que representa que campos están presentes en el paquete. Este Bitmap se representa en formato hexadecimal, por lo cual tendría una composición de 48 valores en total.

Tomando como ejemplo el Bitmap "4210001102C04804", se puede determinar que los campos que contendrán elementos en el paquete serán el 2, 7, 12, 28, 32, 39, 41, 42, 50, 53, 62.

-Campos de Datos: Estos se componen de 192 campos, los cuales poseen la información requerida de la transacción. Por el estándar, ya se tiene definido cada campo, la información que tendrá y el tipo de dato que se puede aceptar. Los tipos de datos que se utilizan en cada uno pueden ser numéricos, alfa numéricos, cadenas, caracteres especiales y hasta binario. La tabla 3 muestra los campos más comunes que se utilizan en los paquetes, sin importar el MTI asignado, y por lo cual, se utilizarán a fin de demostrar el protocolo que se propondrá [9].

TABLA III
CAMPOS COMUNES EN ISO 8583

Nº Campo.	Nombre.	Descripción.
2	Número Primario de la Cuenta. (PAN)	Dígito que identifica la tarjeta miembro y su cuenta relacionada.
3	Código de Procesamiento.	Describe el efecto de la transacción en la cuenta y que se realizara con esta.
4	Monto de la Transacción.	El valor de la transacción con el cual se pagara.
7	Fecha y Hora de Transmisión.	Especifica el tiempo en que fue ingresada la transacción en el POS.
11	STAN.	Número asignado por el POS de la transacción con el cual se puede auditar.
42	Código de Identificación del Vendedor.	Con dicho código se puede determinar al dueño del POS.
123	Código de datos de POS.	Explica el contexto de la transacción y del punto de venta.

La figura 8 muestra la composición del paquete en total, cabe recalcar que este lleva una cabecera para identificarlo, la cual no se toma en cuenta para este protocolo.

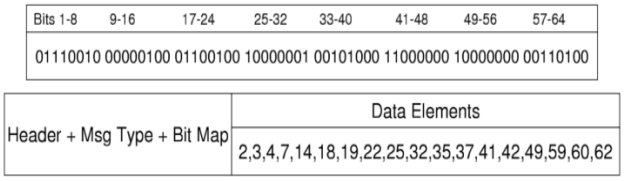


Fig. 8. Composición de paquete ISO 8583.

IV. FUNDAMENTOS CRIPTOGRÁFICOS.

A. Generalidades.

Según [10] la derivación de criptografía se da de la palabra griega "**Krypto**", la cual contiene el significado de "oculto", y la palabra griega "**Graphein**", que significa "escritura". Y se define esta como el conjunto de métodos de protección de la información utilizando técnicas matemáticas que hagan posible el intercambio de mensajes, de manera que solo puedan ser leídos por las personas a quienes van dirigidos.

La criptografía se clasifica históricamente en dos etapas: clásica y moderna.

El periodo clásico se desarrolló desde antes de la época actual hasta la mitad del siglo XX. En dicho periodo no estaba implementado la digitalización lo cual hacia que los métodos, algunos simples y otros complicados, fueran muy difíciles de dar un criptoanálisis para esa época.

En la segunda etapa la criptografía hace ingreso a la era informática, esto en base de varios autores. El primero, desarrollado por Claude Shannon, denominado "A MathematicaTheory of Communication" (1948) y "CommunicationTheory of SecrecySystems" (1949), sienta las bases de la teoría de la información y de la criptografía moderna. Los autores WhitfieldDiffie y Martin Hellman, en su publicación "New Directions in Cryptography" (1976), introducen el concepto de criptografía de llave pública, abriendo un camino a una nueva era en seguridad.

Según [11], los sistemas de cifrado o criptosistemas se definen como una quíntupla (M, C, K, E, D) donde:

-M representa el conjunto de los mensajes sin cifrar (lo que se denomina texto claro, o plaintext) que pueden ser enviados.

-C representa el conjunto de todos los posibles mensajes cifrados o criptogramas.

-K representa el conjunto de claves que se pueden emplear en el criptosistema.

-E es el conjunto de transformaciones de cifrado o familia de funciones que se aplica a cada elemento de M para obtener un elemento C. Existe una transformación diferente Ek para cada valor de la clave k.

-D es el conjunto de transformaciones de descifrado, análogo a E.

De esta quíntupla deberá cumplir las condiciones siguientes para considerarse un criptosistema:

1. La parte del cifrado de la información representado por:

$$Ek(m) = c \quad (1)$$

Donde m sea parte de M. Esto quiere decir que al convertir un mensaje m, por medio de un sistema de cifrado e, perteneciente a E, por con una llave k, generara un texto cifrado c, perteneciente a C.

2. La parte del descifrado de la información representado por:

$$Dk\,(c) = m \quad (2)$$

Esto quiere decir que al convertir un cifrado c, por medio de un sistema de cifrado d, perteneciente a D, por con una llave k, generara un texto plano m, perteneciente a M.

Los criptosistemas en la actualidad se dividen en dos grandes ramas, los sistemas simétricos y los criptosistemas asimétricos, el protocolo a proponer en esta investigación utiliza un sistema asimétrico, por lo cual se profundizará un poco sobre el tema.

B. Sistema Asimétrico y RSA.

Según [12] El proceso de cifrado asimétrico se caracteriza principalmente en que cada usuario ha de poseer un par de claves:

-Clave Pública: se caracteriza que esta clave será conocida por todos los usuarios.

-Clave Privada: será custodiada por su dueño y no se deberá revelar, ni compartir a ninguna otra persona.

Está pareja de llaves se caracteriza en ser complementarias, es decir, que lo que una llave llega a cifrar, la otra llave lo podrá descifrar y viceversa.

Estas claves se obtienen mediante métodos matemáticos complicados de forma que por razones de tiempo de cómputo, es imposible conocer una clave a partir de la otra.

Para explicar mejor este método de cifrado tomamos el siguiente ejemplo:

1-Ana y Bernardo (en la figura 9) tienen sus pares de claves respectivas: una clave privada que solo ha de conocer el propietario de la misma y una clave pública que está disponible para todos los usuarios del sistema.

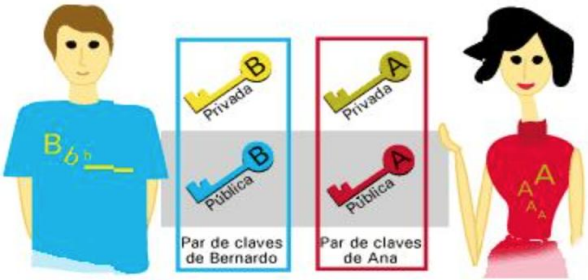


Fig. 9. Par de llaves Pública y Privada.

2-Ana escribe un mensaje a Bernardo y quiere que solo él pueda leerlo. Por esta razón lo cifra con la clave pública de Bernardo, accesible a todos los usuarios.

3-Se produce el envío del mensaje cifrado no siendo necesario el envío de la clave.

4-Solo Bernardo puede descifrar el mensaje enviado por Ana ya que solo él conoce la clave privada correspondiente. La figura 10 muestra el proceso en sí.

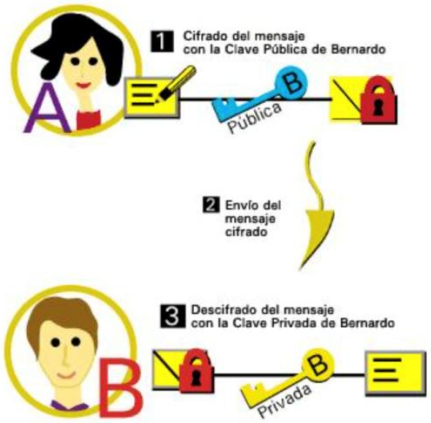


Fig. 10. Cifrado Asimétrico.

Según [13], Los algoritmos asimétricos emplean generalmente llaves de mayor longitud que los sistemas simétricos, recomendablemente de mayor o igual a 1024 bits. Estos algoritmos son relativamente recientes, comenzando desde el año 1975. El más representativo de estos es el RSA.

Este algoritmo es empleado en si tanto para el cifrado como para la autenticación, y su base se desarrolla en el problema de factorización de números grandes, del cual se desarrollan sus 2 llaves. De acuerdo a [14] el proceso de generación de llaves se hace de esta forma:

1. Se necesita primeramente dos números primos grandes *p* y *q*, de igual longitud de bits. Con estos se podrá generar un valor *n*, tal que:

$$n = p.q \tag{3}$$

Donde n se usara como módulo de ambas claves. Cabe mencionar que conocer esos dos números es imposible, lo que dificulta el criptoanálisis de este sistema.

2. Se calcula la función de Euler de φ(n), tal que:

$$\varphi(n) = (p - 1)\,(q - 1) \tag{4}$$

3. Se escoge un entero positivo *e* menor que $\varphi(n)$, que sea coprimo o primo relativo de $\varphi(n)$. Con estos números (e, n) se hará referencia a la clave pública del sistema de cifrado.

4. Se determina un *d* (mediante aritmética modular) que satisfaga la congruencia:

$$e \cdot d = 1 \ (mod \varphi(n)) \quad (5)$$

En otras palabras, que d sea el multiplicador modular inverso de *e modφ(n).* Con estos números *(d, n)* se hará referencia a la clave privada del sistema de cifrado.

Con las llaves obtenidas, para obtener el mensaje de cifrado *C* a partir de un mensaje en claro *M* se realiza la siguiente operación:

$$C = M^e (mod\ n) \quad (6)$$

En éste caso se cifra con la clave pública.

$$M = C^d (mod\ n) \quad (7)$$

En este caso se descifra con la clave privada.
Dependiendo de la función criptográfica a utilizar, el manejo de RSA puede cambiar, pero se identifica en 3 procesos de importancia:
-Cifrado y Descifrado: Tal como se vio en el ejemplo de Ana y Bernardo.
-Firma Digital: En esta función el cifrado se hará con la llave privada del firmante y el receptor podrá verificar con la llave privada.
-Intercambio de Llaves: dos partes cooperan en el intercambio de una llave de sesión.

C. Función HAS o Picadillo.

Una función hash H, es una función computable que convierte un mensaje M en un resumen h, tal que:

$$h = H(M) \quad (8)$$

Tal y como se indica en [15], la función H deberá asegurar las siguientes características:

1. **Unidireccionalidad**. Conociendo un resumen *h*, debe ser computacionalmente imposible encontrar *M* a partir de dicho resumen
2. **Compresión**. A partir de un mensaje *M* de cualquier longitud, el resumen *h* debe tener una longitud fija. Lo normal es que la longitud del digesto *h* sea menor.
3. Facilidad de cálculo. Debe ser fácil calcular *h* a partir de un mensaje *M*.
4. **Difusión**. El resumen *h* debe ser una función compleja de todos los bits del mensaje *M*. Si se modifica un bit del mensaje *M*, el hash *h* debería cambiar aproximadamente la mitad de sus bits.
5. **Colusión Simple**. Conocido *M*, será computacionalmente imposible encontrar otro *M'* tal que *H(M) = H(M').* Se conoce como resistencia débil a las colisiones.
6. **Colusión fuerte**. Será computacionalmente difícil encontrar un par *(M, M')* de forma que *H(M) = H(M')*. Se conoce como resistencia fuerte a las colisiones.

Cabe de mencionar en [16], que estos resúmenes tiene diferentes aplicaciones entre las cuales se pueden conjuntar en:

-Herramientas básicas para la construcción de utilidades más complejas, como estructuras de datos, algoritmos de cifrado o descifrado, acumuladores criptográficos, filtros de Bloom, sellos de tiempo confiables, etc.
-Herramientas de protección de la integridad, como la firma digital y suma de verificación (checksum).
-Herramientas vinculadas a la autenticación y control de acceso, como los códigos de autenticación de mensajes (hash + llave secreta), protección y derivación de claves.
-Herramientas para la identificación y rápida comparación de los datos, como las Huellas Digitales.

Entre las más reconocidas funciones podemos mencionar:
-MD5. Este algoritmo fue desarrollado en 1991. Su codificación de 128 bits es representada típicamente como un número de 32 hexadecimales.
-SHA. Este algoritmo es una familia de funciones, publicadas por el Instituto Nacional de Estándares y Tecnología (NIST). SHA posee 4 versiones, de 0 a 3. Las primeras versiones manejaban salidas de 160 bits hasta la de 512. Para nuestro estudio se hará uso de SHA de 256 bits.

D. Firma Electrónica.

La función de la firma electrónica es una de las aplicaciones resultantes de la combinación de los sistemas mencionados anteriormente. Para el desarrollo de este se deberá tener por parte del emisor y receptor sus respectivas llaves y haber compartido la llave pública a su contraparte.

Se explicara a continuación este proceso.

1. El emisor al poseer un mensaje M, deberá desarrollar con una función H un digesto h, tal que:

$$h = H(M) \quad (9)$$

2. Con la obtención de h se pasa al proceso de cifrado, pero en este caso la llave a utilizar será la clave privada (d, n) del emisor haciendo quedar la función de cifrado así:

$$S(h) = h^d \ (mod\ n) \quad (10)$$

Cabe de recalcar que ahora el texto cifrado, se denominara como firma digital S, y este representa como todo el conjunto de firmas que se pueden generar, usualmente con una longitud fija.

3. Se envía tanto el mensaje M, como la firma S(h), al receptor. Con esto se recalca que la firma digital no garantiza la confidencialidad de la información, a menos que la persona desea implementar el cifrado asimétrico al documento.

4. El receptor recibe la información y lo primero a realizar será generar un nuevo digesto del mensaje M, tal que:

$$h' = H(M) \quad (11)$$

5. Luego para poder obtener la huella digital dentro de la firma, el receptor empleara la llave pública (e, n) del emisor, quedando la ecuación:

$$h = (S(h))^e \ (mod\ n) \quad (12)$$

6. Al tener el picadillo h de la firma digital, se deberá comparar con el digesto h' generado por el receptor al recibir el mensaje M, de esta manera:

$$h' = h \quad (13)$$

7. Si los valores obtenidos de h y h' concuerdan se concluye que el documento se considerara de carácter valido. En caso contrario, el documento será tomado como no valido.

Con este método se aseguran 3 pilares de suma importancia para la Seguridad de la Información:

-Autenticidad: donde se demuestra que el emisor ha generado este documento, ya que este utilizó su llave privada, la cual solo el propietario de este lo debe conocer.

-Integridad: donde al poder comprobar la validez del documento, se demuestra que este no fue modificado en otra parte de la comunicación, que no haya sido el emisor.

-No repudio del Origen: donde el emisor no puede negar haber enviado este mensaje.

La figura 11 nos muestra el esquema de firma digital con el que aseguramos estos pilares. Pero aun así se da el problema que no podemos comprobar que la llave que nos comparten está relacionada al emisor que lo ha generado, o no.

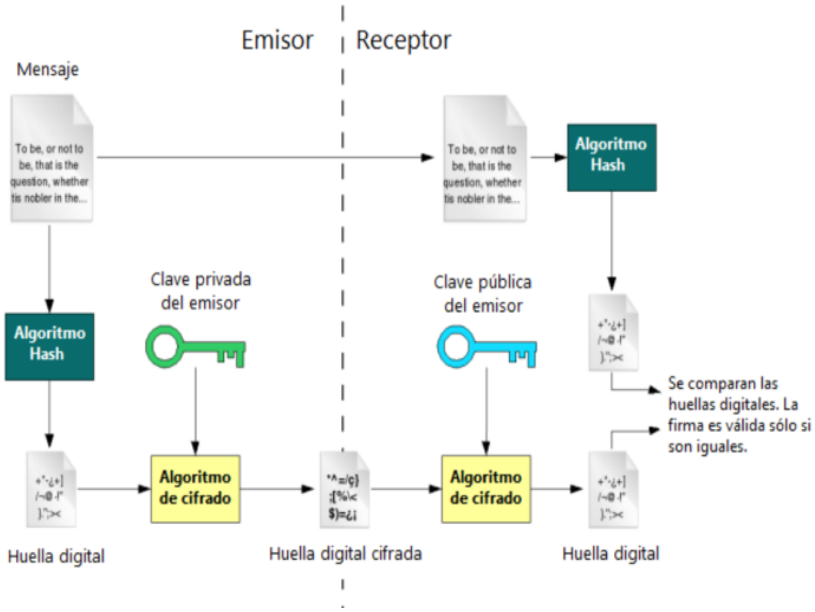


Fig 11. Proceso de Firma Digital.

E. Infraestructura de PKI.

El modelo de protocolo propuesto se deberá basar en este sistema, debido a que se busca que la firma sea autentica y no repudiable tanto en la generación, como en la verificación. Pero es en este segundo donde se tiene el dilema que si la llave pública pertenece al emisor del mensaje o no. Esto también porque en el viaje de la información, un tercero podría tomar estos datos, modificarlos, firmarlos y hacer creer al receptor que la llave que se comparte es la genuina del emisor.

Para lograr eso se necesitara un nuevo elemento digital, el cual es la base de esta estructura. Según [17], **los certificados digitales** son documentos digitales que dan fe de la vinculación entre una clave pública y un individuo o entidad. De este modo, permiten verificar que una clave pública específica pertenece, efectivamente, a un individuo determinado. Los certificados ayudan a prevenir que

alguien utilice una clave para hacerse pasar por otra persona.

En su forma más simple, el certificado contiene una clave pública y un nombre. Habitualmente, también contiene una fecha de expiración, el nombre de la Autoridad Certificante que la emitío, un número de serie y alguna otra información. Pero lo más importante es que el certificado propiamente dicho está firmado digitalmente por el emisor del mismo. Su formato está definido por el estándar internacional ITU-T X.509 (ver figura 12). De esta forma, puede ser leído o escrito por cualquier aplicación que cumpla con el mencionado estándar.

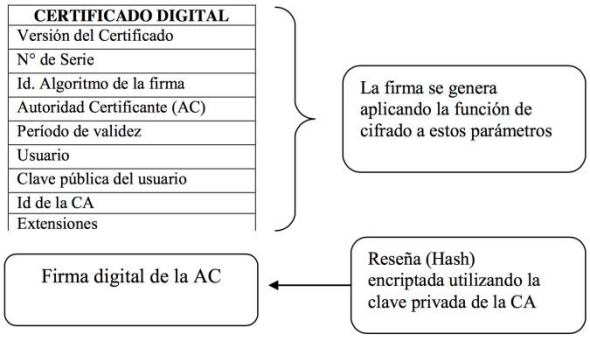


Fig. 12. Formato X.509 para Certificado Digital.

La labor de la Autoridad Certificadora, de acuerdo a [18], por sí misma o a través de la intervención de un Prestador de Servicios de Certificación, verifica la identidad del solicitante de un certificado antes de su expedición. Además, tiene la facultad de autorizar, almacenar, revocar, suspender o extinguir los certificados de llave pública.

La operación de una AC se sustenta en una infraestructura tecnológica que le permite la emisión, administración y registro de Certificados Electrónicos, así como de la disposición de herramientas que permitan la consulta de la validez de los mismos en cualquier momento por parte de los servicios que hagan uso de la Firma Electrónica.

Dado lo anterior, y bajo el principio que la confianza que se tenga en la AC es vital para la emisión de documentos y operación de servicios con firma electrónica, es necesario el establecimiento de procedimientos, políticas y lineamientos que estén apegados a estándares reconocidos en términos de seguridad, encriptación, confidencialidad, continuidad, entre otros.

Al devolver el certificado al usuario este podrá ser uso de dicha herramienta en el proceso de firma digital (ver figura 13) y así considerar la firma como certificada.

Para el caso del protocolo que se propone se hará uso de esta infraestructura pero con alguna modificación de acuerdo al contexto del sistema de pago por tarjetas.

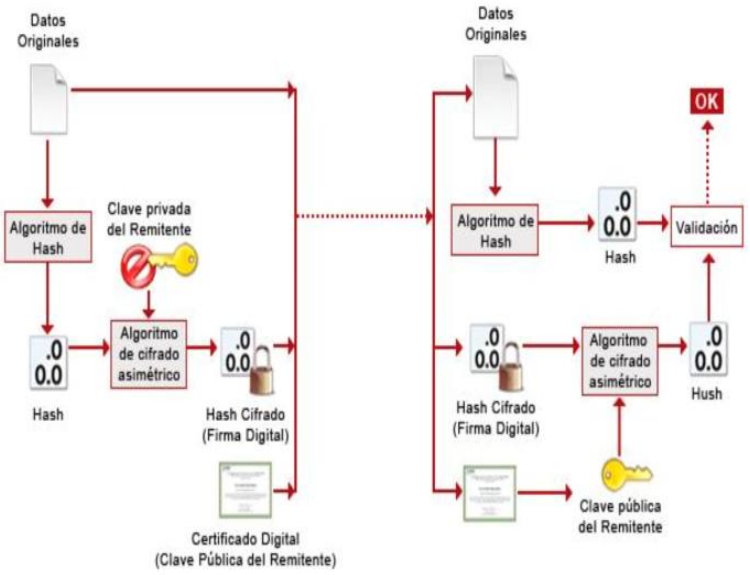


Fig 13. Proceso de Firma Digital con Certificado.

V. PROTOCOLO DE PAGO PROPUESTO.

El protocolo que se propone es especialmente para garantizar la autenticidad, el no repudio de origen y la integridad. Ya que el caso de la confidencialidad es una materia desarrollada por las marcas de tarjetas, bancos adquirentes y emisores, bajo los protocolos y normativas, como el caso de PCI, que se lo exigen, aunque se dará confidencialidad a la firma.

El protocolo a proponer se basa en el uso de la firma digital con la colaboración de certificados, el cual en este caso se llevara a un nivel de paquete, en este caso el uso de la normativa ISO 8583. Habrá algunas variaciones con respecto a la infraestructura tradicional con la que se maneja. Pero se mantendrá la base del modelo, esto con el fin de buscar que sea más eficiente el desarrollo de este en el pago de tarjetas.

En esta propuesta se hará uso de tres participantes que intervendrán en el proceso, tanto en la generación de elementos, como en la ejecución del proceso, los cuales son:

-El Cliente: que deberá realizar la compra y que tendrá a disposición el elemento que lo autenticara como el desarrollador de la transacción financiera.

-Banco: el cual realizara la verificación de la transacción generada por el cliente y así aprobar o no la autenticidad de este, cabe de mencionar que en este punto el sistema

emisor deberá realizar dicha función por ser el dueño generador del plástico.

-Autoridad Certificadora: Este tercero brindara la herramienta que ayudara a dar la validez de la compra desarrollada.

A. Generación de Llaves (Kpub, Kpri).

Tradicionalmente el desarrollo de las llaves, en este caso la privada (d, n) y pública (e, n) lo realiza cada miembro que participa en el proceso de cifrado/descifrado o firma/verificación.

Pero observando el contexto de desarrollo de nuestro protocolo proponemos que la generación de las claves deberá ser realizado por la entidad emisora de la tarjeta de crédito o débito que maneje la cuenta asignada a dicho plástico.

Esto se realizara de esta manera, ya que el cliente no tiene la capacidad tecnológica y conocimiento del desarrollo de estos elementos criptográficos.

Obviamente la entidad bancaria dentro de su infraestructura no debería conocer el contenido de la llave privada del cliente, es por eso que este deberá determinar procedimientos internos a fin de no revelar esta dentro de la empresa, esto primeramente utilizando elementos de protección de acceso a los servidores en donde se resguarde, y seguidamente con la estricta imposición de políticas empresariales y normativas que restrinjan al robo, modificación y eliminación de los mismos. Una de las formas para lograr esa protección será colocando una clave al llavero que solo el cliente deberá conocer.

Además, el cliente podrá exigir dentro del contrato de negociación que su clave privada no sea usada para otro fin que no sea la generación del certificado digital CER, el cual será entregado al cliente. Esta primera llave participara en el proceso de firma digital.

La clave pública del cliente, que estará en el certificado CER, deberá ser resguardada dentro de la infraestructura interna del banco, para su posterior uso en la verificación de la concordancia de la huella digital *h*, pero también respetando su acceso por algún tercero.

La parte privada del banco tendrá que resguardarse dentro de los servidores del sistema que controla las transacciones que manejan el paquete del estándar ISO 8583, para realizar las operaciones de descifrado, siempre de manera protegida y fuera del alcance del personal interno que trabaje dentro de esa área de servicio.

Finalmente, la llave pública debería ser incluida dentro de los servidores de manejo de los POS, tanto para los de característica modular, como los PIN PAD, como para los que se manejan a nivel web, que en este caso son los POS virtuales. Su función específica será de cifrar los paquetes.

Para poder mantener la seguridad de estas llaves se deberá realizar cambios, de acuerdo a un periodo establecido por la entidad bancaria, ya sea como mínimo cada 2 semanas para las llaves que manejen el banco. Y en caso de los clientes, cada 2 meses.

Otra de las propuestas a dar a conocer es que se haga uso del sistema de cifrado de RSA, con longitud de 2048 bits a fin de proteger las llaves y los cifrados correspondientes.

B. Generación de Certificado Digital(CER).

Como se menciona en la teoría, el certificado digital es un elemento de autenticidad en donde un tercero valida la llave que se brinda como la llave la cual tendrá como dueño la persona que posea ese certificado.

En nuestra propuesta, el banco deberá tener una relación con un tercero que pueda autenticar la información que este le brinde. En este caso una clave publica del cliente, que junto a la información que se pueda proporcionar en la entrevista, se pueda elaborar un elemento de petición CSR, que será enviado a la autoridad que lo certifique, por un medio seguro ya sea SSL o un cifrado de llave simétrica entre las dos entidades. Esto haciendo tal que:

$$CSR\ cliente = (e, n)cliente + Info. \quad (14)$$

La Autoridad Certificadora, la cual deberá ser validada en nuestro contexto por una entidad gubernamental, deberá tomar esta petición y autenticarlo generando una huella digital del mismo y firmando con su propia clave privada (d, n), lo cual será anexado al documento que pasara a ser un certificado CER y devuelto por un el mismo medio que recibió esa petición. Esto garantizando la protección del mismo. Tal que:

$$\text{CER } cliente = H(CSR)^d \ (mod\ n) + CSR \quad (15)$$

Finalmente, el banco al recibir este, deberá hacer entrega al cliente, junto a su llave privada, el cual podría portarlo por un medio electrónico, ya sea USB o un dispositivo móvil. Para combinar esos elementos el archivo que se recomienda a usar es el formato PKCS12. Pero considerando que ese medio se ocupe solo para el proceso de pago y no se utilice para otra función y maquina que no sea un PIN PAD o un POS virtual. Ya que este medio al contener un virus, podrá convertirse en un punto de entrada al robo de información o robo de certificado que afecte al cliente, del cual el banco se desligara ante dicho acto.

C. Proceso de Firma del Pago.

Para explicar el proceso de firma, mencionamos a continuación algunos de puntos a tomar en cuenta:

-El cliente posee su propio certificado electrónico CER con su llave respectiva y dentro del periodo de validación del mismo.
-La clave pública (e, n) del banco está colocada en el lugar respectivo de acuerdo al proceso de generación de llaves.
-El dispositivo POS (sea modular o virtual) tiene las características, físicas y lógicas, para la aceptación de un dispositivo de almacenamiento.

Ya realizando el cumplimiento de estos puntos, el proceso de firma se hará de la siguiente manera:

1-Cliente al realizar pago, entregara tarjeta, de débito o crédito, y dispositivo de almacenamiento a mercante, el cual deslizara en sistema POS.
2- PIN PAD o POS virtual desarrollara un paquete ISO 8583, realizando petición para desarrollo de transacción.
3- Sistema POS realizara mensaje M de los campos expuestos en tabla 3, tal que:

$$M = MTI + Camp2 + Camp3 + Camp4 + Camp7 + Camp\ 11 + Camp\ 42 + Camp\ 123 \quad (16)$$

Cabe de recalcar que la regla del mensaje puede modificarse con respecto a los campos que más interesen a las marcas internacionales.
4-El mensaje obtenido será colocado en una función picadillo H en a fin de generar un digesto h (como se explica en la formula (9)).

5-Habiendo obtenido h se procede a firmar, pero tomado en este punto el hecho que se ocupara el archivo PKCS12, donde está la llave privada (d, n) que se tendrá acceso con la clave que haya colocado el cliente, de acuerdo a:

$$S(h) = PKCS12(h) \quad (17)$$

6-Para brindar la seguridad a la firma obtenida el sistema POS realizara un cifrado del mismo a fin de que este no sea observado por un tercero, pero para eso ocupara una llave secreta ks, haciendo que:

$$Cf = AES\ (ks(\ S\ (\ h\ )\ )) \quad (18)$$

Y luego:

$$Cll = (ks)^e\ (mod\ n) \quad (19)$$

Donde (e, n) será la clave pública del banco, del archivo CER, obtenida en el dispositivo POS.
7- Finalmente, los cifrados serán colocados en campos dentro de la ISO 8583, que puede ser determinado por entidad que controla la marca del plástico y enviado a través de la red pasando por los sistemas adquirentes, y la red de tarjetas bancarias.

D. Proceso de Verificación del Pago.

Esta etapa también deberá tener premisas que deberán de cumplirse, las cuales mencionamos:
-La llave pública (e, n) del cliente está depositada dentro de la entidad bancaria de forma resguardada.
-La llave privada (d, n) del banco está depositada dentro del sistema autorizador, fuera del alcance del personal interno y externo, de la institución.
-La verificación se desarrollara en el sistema emisor de la tarjeta, debido a que este hace la transacción bancaria, por ende no se deberá realizar ninguna operación de validación en los demás participantes del proceso de pago de tarjetas.
Teniendo en cuenta estos puntos, la verificación se desarrollara de la siguiente forma:

1-El paquete de petición llega al sistema emisor, en donde al obtener la información se generara un digesto h' con los campos de la ISO 8583, fueron usados a fin de generar la firma digital, quedando:

$$h' = H(MTI + Camp2 + Camp3 + Camp4 + Camp7 + Camp\ 11 + Camp\ 42 + Camp\ 123) \quad (20)$$

2-De los cifrados obtenidos, Cf y Cll, se aplicara la llave privada (d, n) del banco, localizada dentro del sistema emisor para sacar la llave secreta, el cual al aplicar el algoritmo:

$$ks = Cll^{d} \ (mod\ n) \quad (21)$$

Y luego:

$$S(h) = AES\ (ks(\ S\ (\ h\ )\ )) \quad (22)$$

Se obtendrá la firma electrónica S(h), el cual pasara al proceso de verificación en sí.

3-Por medio de un método de identificación de la tarjeta (determinado por el banco), se obtendrá la llave pública (e, n) del cliente y se aplicara el algoritmo inverso de la firma:

$$h = (S(h))^{e} \ (mod\ n) \quad (23)$$

4-Finalmente, se hará la comparación, igual como se aplica en la fórmula 13, con lo cual podremos determinar si procedo o no la transacción por validación en firma digital.

Ahora para poder determinar la funcionalidad de este protocolo se deberá dar a entender por medio de un desarrollo y de escenarios o casos de uso del funcionamiento del mismo.

VI. DEMO DEL PROTOCOLO PROPUESTO.

Para la elaboración de la demo que presentamos como prueba de la investigación, se ha desarrollado bajo el ambiente de 2 aplicativos.

-OpenSSL: Utilizado específicamente para el desarrollo de llaves, CSR's, CER's y paquetes PKCS12 (.p12).

-Java: Lenguaje de programación que nos ayudara a la simulación del proceso que se genera dentro de un POS, tanto en la firma como la verificación del paquete.

-Paquete JPOS: Este sistema de archivos, desarrollados para lenguajes abiertos, son utilizados a fin realizar sistemas de puntos de venta. De este hemos adquirido la sección de generación del paquete bajo norma ISO8583.

A. Generación de Autoridad Certificadora, llaves, CER y archivos .p12.

La generación de la autoridad certificadora se determina dentro del anexo 1, donde como primer punto fue la elaboración de sus llaves y la petición csr, la cual es firmada.

En el punto 2, se configura la fuente, conocida como openssl.cnf, para colocar en la sección "CA_default":

-El directorio de origen ($dir).

-La ubicación del certificado (certificate).

-La ubicación de la llave privada (privatekey).

Ya con esto en los puntos 3 y 4, se podrán generar los llaveros que serán protegidos por una contraseña que proponga tanto el banco y el cliente respectivamente. Además de las peticiones CSR a ser enviadas al tercero que lo validara.

Con el punto 5 y 6 se hace la generación de los archivos CER, ya firmados por la autoridad certificadora para ser regresados al banco emisor

Y finalmente en el punto 7 y 8, se generan los archivos PKCS12 para resguardar los certificados digitales y la llave privada los cuales serán entregados a los dueños en un dispositivo que ellos puedan tener.

En el desarrollo de esta prueba se colocan 3 carpetas representando los elementos que obtendrá cada uno de los participantes, los cuales son:

-"Cliente-Demo": donde se tendrá la llave privada de este, con su certificado (CLIpaquete.p12).

-"POS-Demo": donde estará depositado el certificado con la llave publica del banco (BACcer.pem).

-"Banco-Demo": donde estará depositado el certificado con la llave publica del cliente (CLIcer.pem) la llave privada de este, con su certificado (BACpaquete.p12).

B. Generación de Firma Digital en POS.

En el anexo 2, podemos ver en el primer punto el demo para generar la firma, su cifrado y su participación dentro del paquete bajo normativa ISO 8583 que será enviado en la red. La secuencia lógica de funcionamiento se determina por estos puntos:

1-Se hace uso del botón "Info. Tarjeta", donde se obtendrá el número PAN, el cual simula el paso de tarjeta por el POS.

2-Los siguientes 8 campos, representan a la información que se menciona en la tabla 3, los cuales a excepción del monto y el PAN, serán escritos por el POS.

3-El segundo botón ("Cadena") genera la cadena de texto a ser firmada. El sistema POS lo hará automáticamente.

4-Antes de usar en botón "Generar Paquete", el cliente deberá ingresar su clave para obtener acceso al llavero a así a su clave privada.

5-Cuando es presionado el botón en el punto 4, se toma la clave ingresada y el sistema adquiere la llave privada del

cliente (en este caso CLIpaquete.p12), que está insertado físicamente por un dispositivo y genera la firma digital de la cadena que se obtiene del punto 3.
6-El sistema, automáticamente, genera una clave secreta y cifra la firma (utilizando cifrado AES). Se muestra el cifrado obtenido.
7-La llave publica del banco (BACcer.pem), existente en el POS, se utilizara a fin de cifrar con RSA la llave secreta.
8-Se ingresa la firma cifrada y la llave secreta cifrada en los campos 125 y 126, del paquete ISO 8583, respectivamente. Y finalmente, se envía el paquete que en este caso se representa por el archivo ISO8583.txt.

En el segundo punto del anexo 2 se podrá apreciar la codificación realizada en el lenguaje de programación JAVA, para que finalmente en el punto 3 del mismo se visualice una imagen del paquete ISO 8583 que viajara en la red, representado por un archivo de texto.
En el anexo 4, punto 1, presentamos el flujo del proceso para un mayor entendimiento del mismo.

C. Verificación de Firma Digital de POS.

En el anexo 3, podemos ver en el primer punto el demo para verificar la firma, su cifrado y su participación dentro del paquete bajo normativa ISO 8583 que es recibido desde la red. La secuencia lógica de funcionamiento se determina por estos puntos:
1-Se hace uso del botón "Desempaquetar", donde se obtendrá el paquete que viene de la red, que en este caso es representado por el archivo ISO8583.txt.
2-Al recibirse, el sistema hace el desglosamiento de la información del texto y lo coloca en su casilla correspondiente, en este se incluye la firma digital cifrada que se coloca en el campo "Cifrado".
3-Antes de usar en botón "Descifrar", el banco deberá ingresar su clave para obtener acceso al llavero a así a su clave privada.
4-Cuando es presionado el botón en el punto 3, se toma la clave ingresada y el sistema adquiere la llave privada del banco (en este caso BACpaquete.p12), que deberá estar resguardado en la infraestructura interna.
5-Con la llave privada obtenida en el punto anterior, el sistema descifrara, utilizando RSA, a fin de obtener la clave secreta.
6-Ya con la clave secreta, se revelara la firma digital por medio del descifrador AES. Dicha firma se muestra en el formulario.
7-Utilizando el botón "Verificar Firma" se vuelve a generar la cadena con los valores que se obtengan del paquete ISO8583 y se procede a la función de verificación.
8-Ya en esa función se toma el certificado del cliente (CLIcer.pem), con el cual obtenemos la llave pública y obtenemos la huella digital de la firma.
9-La cadena obtenida se le aplica la función de picadillo a fin de obtener un digesto, que será comparado con la huella obtenida en el punto 8.
10-Se envía una conclusión que se muestra en el formulario, que para la imagen del punto 1 del anexo 3 revela que la firma es válida, con lo cual podemos concluir que la petición es legítima.
Caso contrario, como en el punto 2 del mismo anexo que presenta la frase "Error en Firma Digital", donde se concluye que hubo un ataque que pudo modificar la información del paquete en su viaje a la red, con lo cual no procede la solicitud de transacción.

En el tercer punto del anexo 3 se podrá apreciar la codificación realizada en el lenguaje de programación JAVA. Y finalmente en el anexo 4, punto 2, presentamos el flujo del proceso para un mayor entendimiento del mismo.

VII. CASOS DE USO DEL PROTOCOLO.

Para la determinación de la funcionalidad del protocolo propuesto, se hace presentación de casos donde se aplicara el protocolo y el resultado que este nos da en los métodos de pagos con tarjeta de crédito y débito.

Estos casos se definen como posibles amenazas que se producen a fin de obtener un beneficio de carácter económico. Destacamos que para le ejecución de estos ocupamos la siguiente nomenclatura:

Kpriv = Llave Privada.
Kpub = Llave Pública.
Ks = Llave Secreta.
h = Digesto o picadillo.
h' = Digesto o picadillo en sistema emisor.
S = Firma digital.
POS = Punto de venta. (Modular o Software).
AC = Autoridad Emisora de Certificado.
CERT = Certificado Digital. En formato PCKS12 (P12).

A. Firma Digital Comprobada.

1. Cliente paga con tarjeta y brinda documento y USB con su CERT.
2. USB se coloca en POS y solicita clave de llavero para adquirir Kpriv de cliente.
3. POS toma campos y genera h.
4. h se firma por Kpriv de cliente generando S.
5. POS usa Ks y cifra S.
6. POS usa su Kpub y cifra Ks.
7. S y Ks, cifrados, se anexa a paquete ISO8583.
8. Emisor recibe paquete.
9. Emisor genera h', de campos de paquete ISO 8583.
10. Emisor descifra Ks con Kpriv de Banco.
11. Emisor descifra S con Ks.
12. Emisor revela h con Kpub de cliente.
13. Se compara h con h'
14. Digestos iguales: proceder con otras validaciones.
15. Notificar al cliente de aceptación de la transacción.
16. Elaboración de txt, en USB, especificando comprobante de pago.

B. Firma Digital: Cliente Falso.

1. Cliente ilegitimó paga con tarjeta y brinda documento y USB con su CERT.
2. USB se coloca en POS y solicita clave de llavero para adquirir Kpriv de cliente.
3. Cliente ilegitimó no conoce clave de acceso a llavero.
4. No procede la transacción.

C. Firma Digital: Modificación de Huella Digital.

1. Cliente paga con tarjeta y brinda documento y USB con su CERT.
2. USB se coloca en POS y solicita clave de llavero para adquirir Kpriv de cliente.
3. POS toma campos y genera h.
4. h se firma por Kpriv de cliente generando S.
5. POS usa su Ks y cifra S.
6. POS usa su Kpub y cifra Ks.
7. S y Ks, cifrados, se anexa a paquete ISO8583.
8. Atacante adquiere paquete e intenta modificar firma electrónica.
9. Atacante no accede a información por no tener Kpriv de banco emisor.
10. No procede el ataque.

D. Firma Digital: Información Modificada.

1. Cliente paga con tarjeta y brinda documento y USB con su CERT.
2. USB se coloca en POS y solicita clave de llavero para adquirir Kpriv de cliente.
3. POS toma campos y genera h.
4. h se firma por Kpriv de cliente generando S.
5. POS usa su Ks y cifra S.
6. POS usa su Kpub y cifra Ks.
7. S y Ks, cifrados, se anexa a paquete ISO8583.
8. Atacante adquiere paquete y modifica información sensible que genere un beneficio.
9. Emisor recibe paquete.
10. Emisor genera h', de campos de paquete ISO 8583.
11. Emisor descifra Ks con Kpriv de Banco.
12. Emisor descifra S con Ks.
13. Emisor revela h con Kpub de cliente.
14. Se compara h con h'
15. Digestos diferentes: no proceder con otras validaciones.
16. Notificar al cliente de rechazo de la transacción.

E. Firma Digital: Negación de Transacción.

1. Cliente paga con tarjeta y brinda documento y USB con su CERT.
2. Se realiza el proceso de firma digital comprobada.
3. Cliente niega proceso de transacción en POS.
4. Banco autentica la identidad del cliente.
5. Banco solicita a AC el CERT del cliente.
6. AC realiza envío del CERT del cliente vigente.
7. Cliente envía CERT a banco.
8. Banco revela, con Kpub de AC, el h de CERT de cliente.
9. Banco revela, con Kpub de AC, el h' de CERT de AC.
10. Se compara h con h'
11. Digestos iguales: procede la transacción, ya que el cliente tiene el CERT legítimo. Y como dicho elemento no es transferible, este realizó la transacción.

F. Firma Digital: Phising.

1. Cliente paga con tarjeta y brinda documento y USB con su CERT.
2. Cliente/Mercante verifica que POS virtual se desarrolle bajo protocolo HTTPS.
3. Cliente/Mercante verifica que POS virtual posea CERT de autenticación en la web.

4. Sitio no posee CERT de autenticidad.
5. No procede a hacer la compra/venta.

VIII. ANÁLISIS DE SEGURIDAD Y VIABILIDAD DEL PROTOCOLO.

A. Pilares de Seguridad de la Información.

El protocolo desarrollado y presentado participa en la definición de seguridad en el proceso de autorizaciones, no en un 100%, pero si en algunos pilares no muy comunes.

a. Confidencialidad.

En esta primera característica, nos muestra que los sistemas deben ser capaces de no revelar información sensible a personas que no tengan las credenciales necesarias para conocer los datos.

Nuestra propuesta solo se enmarca en la protección de la firma en si por medio del cifrado de forma simétrica y asimétrica de la clave secreta entre el POS y el banco emisor, esto a fin de evitar una modificación que altere la firma desarrollada.

También se hace énfasis en el cuidado que podemos tener con las llaves privadas, tanto el banco como el cliente mismo.

Obviamente el cliente deberá tener el suficiente cuidado de no revelar sus accesos, ni dispositivos de almacenamiento (USB o tarjetas).

La protección de la información, depende en gran medida de la infraestructura que puedan desarrollar los bancos, junto a reglamentos, políticas y normativas que se implementen en los sistemas.

b. Disponibilidad

En este segundo pilar se deberá tener la viabilidad, departe del sistema, de mantener la continuidad del funcionamiento, sin importar el contexto en el que se desarrolle.

Nuestra propuesta no abarca este pilar, ya que este depende en su totalidad de la capacidad de la infraestructura tecnológica y de la respuesta que esta pueda desarrollar ante un incidente o evento que interrumpa el proceso.

c. Integridad.

En este tercer pilar, nos muestra que los sistemas deben ser capaces de no permitir la modificación sin autorización de información sensible de personas, a menos que tengan las credenciales necesarias.

Esta característica participa mucho el protocolo presentado, desde el hecho que se hace bajo el sistema de firma digital. Esta técnica nos ayuda a demostrar la no modificación de la información a través de la red.

La integridad es tomada muy en cuenta desde la generación de llaves, la cual será la base de uso en un certificado digital generado por un tercero con el cual validara al propietario. Para que en este documento se compruebe que la información no se ha modificado, se genera su huella digital.

Con este elemento se puede ser capaz de comprobar que el documento no ha sido alterado en su recorrido al banco, y con su entrega al cliente. Esto generando nuevamente el digesto del documento y comparándolo con la huella que la Autoridad Certificadora haya generado.

También está característica se verá reflejada en el picadillo que se envíe del sistema POS al sistema que emite/autoriza las transacciones de tarjetas. Ya que al revelar la igualdad del hash, contra el que se genere departe de la información de la tarjeta que podrá determinar que la data no ha sufrido ninguna modificación en su trayecto por la red pública.

d. Autenticación.

Esta característica permite identificar al generador de la información como el origen legal del mismo. En este contexto consistiría en verificar su identidad.

En la búsqueda de determinar la identidad del individuo se busca cumplir con un sistema de multi factor que asegure a este como dueño legítimo y generador de la transacción. Estos elementos se ven en tres pilares:

-"Lo que tengo": donde se agrupan elementos físicos que este en posesión del usuario y que ayuden a identificarlo. Como por ejemplo un dispositivo tockenizador, una USB con llave de cifrado, entre otros.

-"Lo que sé": donde los elementos son más de carácter cognitivo de conocimiento, tales como un usuario, una contraseña, una clave, etc.

-"Lo que soy": este tercer elemento es más desarrollado en el uso de la biométrica. Donde interactúan elementos provenientes del cuerpo humano que ayudan a identificar a la persona.

De estos pilares, los procesos de seguridad recomiendan el uso de dos factores de manera combinada, a fin de asegurar la identidad de la persona.
Para nuestra propuesta hacemos uso de los elementos físicos y de conocimiento, ya que la biometría es un elemento no muy desarrollado en el contexto de nuestra sociedad.
La autenticación se inicia en la generación de llaves de cifrado y descifrado para el cliente y el banco, donde cada uno tendrá un par de llaves, de las cuales solo será intercambiable la llave que puede ser de carácter público. Es de ahí, que se pueda usar cualquiera de los sistemas de criptografía, que en este caso se utilizara la firma digital.

Este pilar se notara en diversos aspectos del proceso:
-Autenticación de Certificado Digital: donde al ser firmado el certificado recibido por la Autoridad Certificadora y verificado con la llave pública del emisor comprobamos que dicho tercero es legítimo.
-Autenticación de acceso a Certificado Digital del Cliente: donde el cliente al realizar entrega de su dispositivo que posee su certificado electrónico, el cual representa el factor físico, y la contraseña de acceso para obtención de la llave privada, el cual nos representa el factor cognitivo, nos muestra la veracidad de su identidad por dos de tres factores.
-Autenticación en la Firma Digital: el cual se comprueba al ser cifrado la huella digital de los campos del paquete, con la llave propietaria del cliente y verificado en el sistema autorizador, con la clave pública, nos demuestra que el cliente es quien ha realizado el pago, ya que es el único con el acceso a generar la transacción, con esas llaves.

e. No Repudio del Origen.
Esta característica nos debe demostrar la capacidad de un sistema de responder ante una negación del cliente en el desarrollo de cualquier transacción, en este caso del pago con tarjetas de crédito o débito.

Esto se refleja con el Certificado Digital, ya que el comprador que lo posea es el auténtico generador del proceso de pago. Y si es el mismo certificado vigente que posee la Autoridad Certificadora, más el conocimiento al acceso de la clave que determinara la firma de la huella digital, son elementos suficientes para que el cliente no niegue el desarrollo de la adquisición de un bien o servicio.

B. Viabilidad del Protocolo.

La viabilidad del protocolo propuesto abarca varios aspectos que se toman en cuenta y por los cuales muestra que dicho sistema puede ser implementado.

a. Viabilidad Tecnológica.
La implementación de este protocolo depende de gran medida del tipo de infraestructura tecnológica que pueda tener la empresa. En este caso la entidad bancaria que maneje el proceso de pago de tarjetas.

Cabe recalcar que la mayoría de bancos posee esa infraestructura de manera exigida, por el proceso que desarrollan y por las normativas internacionales que lo demandan a fin de poder trabajar con ese tipo de cartera bancaria. Dentro de estas estructuras hay elementos con los que cuentan, lo cual los considera aptos para utilizar la propuesta presentada:

-Software de Desarrollo: los cuales poseen diferentes funciones que manejan la información de manera rápida y eficiente.
Estos programas, específicamente deberán manejar una serie de contenedores de funciones que ayuden a hacer trabajar un aplicativo. Como en el caso del manejo de puntos de ventas, se toma como ejemplo la paquetería JPOS.
JPOS es una librería desarrollada especialmente para los sistemas operativos de código abierto como Java, Android, entre otros, con el fin de desarrollar frameworks en la generación de paquetes con el formato de intercambio de mensajes por tarjetas de pago. Dentro de sus funciones posee la de desarrollar los paquetes con formato ISO8583 que serán utilizados con el fin de transportar la información desde el punto de venta al sistema emisor.
-Sistemas de Desarrollo y Administración de Llaves: en donde se tiene la capacidad de generar claves que serán utilizadas en las diferentes funciones de cifrado que se mencionan en el proyecto. Además del desarrollo de peticiones de certificación para comprobar la autenticidad de sus dueños.
El protocolo Secure Socket Layer (SSL) facilita la autenticación y la privacidad de la información que viaja por internet, esto por medio del funcionamiento que se hace en el uso de sistemas como OpenSSL, el cual consiste en un robusto paquete de herramientas de administración y bibliotecas relacionadas con la criptografía. En esta propuesta, este protocolo debe ser considerado entre banco

emisor y la Autoridad Certificadora, esto con el fin de obtener una congruencia en la petición y respuesta de esto.
SSL nos permite además poder desarrollar diferentes formatos donde guardar las llaves, siendo el formato PKCS12, el más indicado para brindar seguridad a la llave privada que se le entregue al cliente y su certificado en formato X.509.

-Servidores de Base de Datos y Aplicativos: en donde se encuentran resguardados las rutinas de los sistemas y las llaves con las que se trabajara.

Los bancos tienen la capacidad de desarrollos sobre estos, los cuales se conectan a los POS distribuidos en su red bancaria.

Además, la entidad bancaria deberá implementar medidas de seguridad con respecto a los elementos de criptografía (Llave Privada del Banco, Llave Publica del Cliente y Certificado Digital del Cliente), a fin de no revelar esa información ninguna tercero.

-Comunicación de la Información: El manejo de la información a través de la red, debe ser de efectiva importancia de los sistemas bancarios y con un alto sentido de la seguridad.

IPsec es un claro ejemplo de esto, al ser un protocolo de tunnel que proporciona un cifrado y autenticación de los paquetes IP. Pero además de eso establece una comunicación en los nodos, con el fin de evitar la presencia de un tercero que pueda interceptar la información.

Para el caso de los certificados digitales entre el banco y el tercero que certificara, se puede brindar una mayor seguridad de estos con el sistema de cifrado simétrico, por medio del cifrado DES, implementado entre las base de datos en donde se guardan a fin de cuidar la información enviada en la red a través de esos dos nodos.

b. Viabilidad Legal y Normativa.

-Legalidad.

La base legal del desarrollo del proyecto se fundamenta en la Ley de Firma Digital de la República de El Salvador dónde se define que la misma, según Art. 23 y 24 de [19], tendrá los siguientes efectos:

a) Vincula un mensaje de datos con su titular, de manera exclusiva;

b) Permite la verificación inequívoca de la autoría e identidad del signatario; y,

c) Asegura que los datos de la firma estén bajo control exclusivo del signatario.

Con esto se demuestra que la Firma Electrónica Certificada tendrá igual validez y los mismos efectos jurídicos y probatorios que una firma manuscrita en relación con los datos consignados en un documento o mensaje de datos electrónicos en que fuere empleada.

En el desarrollo del proceso, la Autoridad de Certificación para el desarrollo de sus funciones deberá, según Art. 43 de mencionada ley, poseer la capacidad tecnológica, financiera, humana y de seguridad ya sea para la prestación de sus servicios, así como de los certificados electrónicos que proporcione, revoque, suspenda o cancele y las restricciones o limitaciones apliquen a estos. Punto que varias empresas se les exige, si aplican a convertirse en entidades de validación

En el caso del contenido del certificado, de acuerdo al Art. 58 de [19], la estructura del mismo deberá poseer:

a) Identificación del titular del certificado electrónico.

b) Identificación del proveedor de servicios de certificación.

c) Fecha de la acreditación y caducidad asignada al proveedor de servicios de certificación por la Unidad de Firma Electrónica;

d) Fecha de emisión y expiración del certificado;

e) Número de serie o de identificación del certificado;

f) La firma electrónica certificada del prestador de servicios de certificación que emitió el certificado;

g) Datos de verificación de la firma.

h) Cualquier información relativa a las limitaciones de uso, vigencia y responsabilidad.

i) Indicación de la ruta de certificación; y,

j) Acreditación de la persona natural que recibe certificado digital.

Esta estructura se apega en si al formato de certificado X.509, con lo cual se apega dicho requerimiento.

Específicamente a nivel de banco, en los Art. 62 y 63 especifican sus derechos como dueños de los certificados que serán brindados los clientes, los cuales deberán hacer valer sobre la autoridad certificadora. Además de poseer obligaciones, en donde más impera:

a) Proporcionar la información veraz a la AC, de los dueños.
b) Custodiar adecuadamente los mecanismos de seguridad del funcionamiento del sistema de certificación que le proporcione el prestador y

actualizar sus datos en la medida que éstos vayan cambiando.

c) Solicitar oportunamente la suspensión o revocación del certificado, ante cualquier circunstancia que pueda haber comprometido la privacidad de los datos de creación de firma electrónica certificada.

Los cuales por sus respectivas Políticas de Seguridad de la Información lo pueden implementar.

-Normativa.

Las normas internacionales juegan un papel de control de los procesos que se manejan a nivel bancario. Más si es un proceso de carácter delicado como el uso de tarjetas de crédito y débito.

Esto ha llevado a marcas, como VISA y MASTERCARD, a exigir el desarrollo de estos reglamentos, como condicionantes para la autorización de emisión de los plásticos en las entidades bancarias, tal como el caso de la normativa **PCI DSS.**

Según [20], el Estándar de Seguridad de Datos para la Industria de Tarjeta de Pago (PaymentCardIndustry Data Security Standard) o PCI DSS fue desarrollado por un comité conformado por las compañías de tarjetas (débito y crédito) más importantes, comité denominado PCI SSC (PaymentCardIndustry Security Standards Council) como una guía que ayude a las organizaciones que procesan, almacenan y/o transmiten datos de tarjetahabientes (o titulares de tarjeta), a asegurar dichos datos, con el fin de evitar los fraudes que involucran tarjetas de pago débito y crédito.

En el requerimiento 4 de dicha normativa se exige una encriptación durante la transmisión en la red pública, dificultando el acceso a la información que se maneje, ya que la mala protección de estos medios puede representar una vulnerabilidad que pueda ser aprovechada por un atacante. Y como parte de este requerimiento se exige el uso de llaves y certificados para la protección de la información, tal como se usa en el proyecto presentado. Por lo cual es aplicable desde el punto de vista normativo.

IX. RECOMENDACIONES.

Para el desarrollo de dicho protocolo, donde se ha desarrollado una demo de prueba de investigación, existen algunos puntos a mencionar que servirán de gran ayuda al mismo:

a) La elaboración del mensaje a firmar, donde se exige la selección de los campos a conformarlo y el orden de diseño deberá ser de forma consensuada entre las marcas, a mantener un estándar que pueda ser utilizado si el cliente llega a remunerar un bien o servicio en un punto de venta que no posea un sistema de pago de tarjeta apegado a la marca del plástico que este posea, ni del banco emisor.

b) La entrega de certificado digital puede ser desarrollada en base a dispositivo de almacenamiento seguro, que sea validado por el banco emisor de la tarjeta y con función específica para esas transacciones y no para otras actividades que no correspondan a eso.

c) Desarrollar el formato de entrega de paquete, sobre el estándar ISO 8583, a fin de mantener un medio de transporte que sea congruente en los diferentes sistemas adquirentes y de emisión.

d) Diseñar una infraestructura de seguridad de los elementos de cifrado, la cual, sea restringida para un personal no autorizado, tanto de manera lógica como físicamente.

e) Elaborar bajo el formato X.509, el desarrollo de los certificados digitales y que puedan ser transportado en un repositorio junto a la llave de firma del cliente para utilizarse dentro del protocolo. El caso recomendado es el formato PKCS12

f) Poseer un método de intercambio de llaves de carácter público para los dispositivos de POS de la firma a fin de aplicar un cifrado que proteja dicha información. El cual se puede desarrollar entre el sistema adquirente, quien posea las claves y que identifique la emisión del plástico para brindar la llave correspondiente.

g) Utilizar el Protocolo de Transferencia de Hipertexto Seguro (HTTPS), como medio de protección ante Phising o el uso de Certificado Digital para la protección de POS de manera virtual.

h) Determinar el periodo de validación de certificados entre el banco y el AC, así como los métodos de gestión de estos, de acuerdo a lo establecido por la ley.

X. CONCLUSIONES.

Con el desarrollo del sistema criptográfico se ha podido concluir algunas cuestiones sobre el proyecto de investigación:

a) Se ha podido identificar en si el proceso de pago con tarjetas de crédito y débito como un sistema que debe ser resguardado y cuidado. Dicho sistema debe ir

evolucionando de acuerdo al cambio tecnológico con el paso del tiempo, lo cual obliga a implementar nuevas medidas de seguridad, lo que al notar su infraestructura se puede determinar un futuro prometedor.

b) Existen elementos suficientes en la infraestructura tecnológica que se pueden apegar al modelo de firma electrónica y que por ende pueden ser aplicables a nuestro modelo propuesto.

c) Se ha desarrollado un modelo matemático, capaz de ayudar a representar el modelo de cifrado deseado, con funciones que ayudan a resguardar la información en su confidencialidad e integridad, para mostrar la aplicabilidad de este.

d) Existe una forma de desarrollo de una aplicación, bajo herramientas y lenguajes de programación, en donde se pueda implementar dicha propuesta para así validar la investigación realizada.

e) Los casos de estudio desarrollados nos muestran que la aplicación del protocolo en dichos contextos salvaguardan el cuidado de la data en los pilares de la seguridad de la información, y por ende puede ser usado como complemento en los controles que ayudan a mitigar los riesgos que puedan suscitarse.

f) Se posee una viabilidad legal y normativa la cual nos autoriza y exige a poder llevar a la realidad el protocolo propuesto de esta investigación.

BIBLIOGRAFÍA.

a. Libros.

[14] Stallings, W. (2011). (Cryptography and Network Security: Principles and Practice. FifthEdition). 245-297.

b. Escritos Publicados.

[3] M. Khayyam, A.Mian (Standardizing Implementations of ISO-8583,) [En línea]. Available: http://www.cslhr.nu.edu.pk/GCCS/Summer2010/papers/umar.pdf [Último acceso: 15 06 2016]

[5] SANS Institute InfoSec Reading Room (Point of Sale (POS): Systems and Security,) [En línea]. Available: https://www.sans.org/reading-room/whitepapers/bestprac/point-sale-pos-systems-security-35357 [Último acceso: 09 06 2016]

[7] Trend Micro Incorporated (Point-of-Sale System Breaches: Threats to the Retail and Hospitality Industries) [En línea]. Available: http://www.trendmicro.com/cloud-content/us/pdfs/security-intelligence/white-papers/wp-pos-system-breaches.pdf [Último acceso: 14 06 2016]

[9] Electronic Clearing House, Inc. (ECHO ISO 8583, Technical Specification. Version 1.6.5, August 19, 2005) [En línea]. Available: http://www.nationalchecknetwork.net/secure/ECHO-, ISO-8583-Technical-Specification-V1.6.5.pdf [Último acceso: 15 06 2016]

[17] Rocha M., Castello R., Bollo D., (Criptografía y Firma Electrónica/Digital en el Aula) [En línea]. Available: http://www.editorial.unca.edu.ar/Publicacione%20on%20line/CD%20INTERACTIVOS/DUTI/PDF/EJE2/ROCHA%20VARGAS.pdf [Último acceso: 08 07 2016]

c. Reportes Técnicos.

[8] ADMFactory.com (Introduction to ISO 8583 financial transaction message format) [En línea]. Available: http://www.admfactory.com/iso8583-financial-transaction-message-format/ [Último acceso: 21 08 2016]

d. Disertaciones.

[10] UNAM, (Criptografía) [En línea]. Available: http://www.matem.unam.mx/rajsbaum/cursos/web/presentacion_seguridad_1.pdf [Último acceso: 05 07 2016]

e. Legislaturas.

[19] Asamblea Legislativa de El Salvador, (LEY DE FIRMA ELECTRÓNICA) [En línea]. Available: http://www.asamblea.gob.sv/eparlamento/indice-legislativo/buscador-de-documentos-legislativos/ley-de-firma-electronica [Último acceso: 29 07 2016]

f. Normativas.

[20] Payment Card Industry: Data Security Standar, (Requirements and Security Assessment Procedures) version 3.2, April 2016. Norma

g. Recursos en línea.

[1] Whatis (POINT OF SALE TERMINAL,) [En línea]. Available: http://whatis.techtarget.com/definition/point-of-sale-terminal-POS-terminal. [Último acceso: 21 08 2016]

[2] Enciclopedia Culturalia (Punto de Venta – Su Significado, Definición, Concepto e Importancia,) [En línea]. Available: https://edukavital.blogspot.com/2013/10/definicion-de-punto-de-venta.html. [Último acceso: 21 08 2016]

[4] Google (What is a Virtual Terminal?,) [En línea]. Available: http://www.merchantmaverick.com/virtual-terminal/ [Último acceso: 08 06 2016]

[6] Jerry's Journal (BIN de Tarjeta de Crédito,) [En línea]. Available: http://jerryjournal.com/bin-de-tarjeta-de-credito/ [Último acceso: 21 08 2016]

[11] DMA, (Introducción a la Criptografía) [En línea]. Available: http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_modular/criptografia.html [Último acceso: 05 07 2016]

[12] CERT, (Criptografía de Clave Asimétrica) [En línea]. Available: http://www.cert.fnmt.es/curso-de-criptografia/criptografia-de-clave-asimetrica[Último acceso: 07 07 2016]

[13] DMA, (Criptosistemas de clave pública. El cifrado RSA) [En línea]. Available: http://www.dma.fi.upm.es/recursos/aplicaciones/matematica_discreta/web/aritmetica_modular/rsa.html [Último acceso: 07 07 2016]

[15] Google, (Funciones Hash en Criptografía (Tema 6)) [En línea]. Available: http://criptosec.unizar.es/doc/tema_c7_criptosec.pdf [Último acceso: 07 07 2016]

[16] w3ii.com, (Funciones Criptográficas Hash)]En línea]. Available: http://www.w3ii.com/es/cryptography/cryptography_hash_functions.html [Último acceso: 21 08 2016]

[18] Gobierno del Estado de Guerrero, (Firma Electrónica Certificada) [En línea]. Available: http://autoridadcertificadora.guerrero.gob.mx/fec/que-es-la-fec.html[Último acceso: 08 07 2016]

Mario Planas recibe el grado de Ingeniero en Ciencias de la Computación en la Universidad Don Bosco, Soyapango, en San Salvador, El Salvador (2011).
Ha desempeñado como Oficial de Seguridad de la Información, para grupo financiero BAC-Credomatic S. A. de C.V (2015-2016). Además de tener a cargo análisis de riesgo crediticio para Banco Agrícola (2012-2015). Y esto agregar su labor de Analista Programador para Grupo OCTO S.A. de C.V (2011-2012).
Ha sido participe de diversos proyectos como la re certificación de BAC-Credomatic en ISO 9001:2015, implemetación de sistemas de Gestión de Riesgo de SI, y de Inventario de Medicinas para Droguería Santa Lucia y sistemas de validación de contabilidad bajo normas nacionales e internacionales. Ha sido parte de diversos programas de capacitación, con temas variados de los cuales podemos mencionar:
-CCNA's y CCNP.
-Seguridad Informática.
-Finanzas.
-Desarrollo Web.
-Gestión de TI.
-COBIT.

ANEXO 1. Generación de Autoridad Certificadora, Llaves, CSR, CER y archivos P.12.

Punto 1-Generación de Autoridad Certificadora.
opensslreq -new -x509 -keyoutAUCllavero.pem -out AUCcsr.pem -set_serial 01 -days 365
openssl x509 -in AUCcsr.pem -outform DER -out AUCcer.der

Punto 2-Configuración de openssl.cnf. Nota: Crear la red de directorios que se soliciten.

```
##########################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

##########################################################################
[ CA_default ]

$dir         = /home/mario/CA5/AC           # Where everything is kept
certs        = $dir/certs                   # Where the issued certs are kept
crl_dir      = $dir/crl           # Where the issued crl are kept
database   = $dir/index.txt        # database index file.
#unique_subject = no                  # Set to 'no' to allow creation of
# severalctificates with same subject.
new_certs_dir        = $dir/newcerts # default place for new certs.
certificate              = $dir/AUCcsr.pem# The CA certificate
serial       = $dir/serial          # The current serial number
crlnumber              = $dir/crlnumber# the current crl number
# must be commented out to leave a V1 C$
crl          = $dir/crl.pem # The current CRL
private_key          = $dir/private/AUCllavero.pem# The private key
RANDFILE         = $dir/private/.rand    # private random number file

x509_extensions = usr_cert# Theextentions to add to the cert
```

Punto 3-Generación Llaves Banco y csr.
opensslreq -new -keyoutBACllavero.pem -out BACcsr.pem

Punto 4-Generación Llaves Cliente y csr.
opensslreq -new -keyoutCLIllavero.pem -out CLIcsr.pem

Punto 5-Generación Certificado por AC para Banco.
openssl ca -in BACcsr.pem -out BACcer.pem

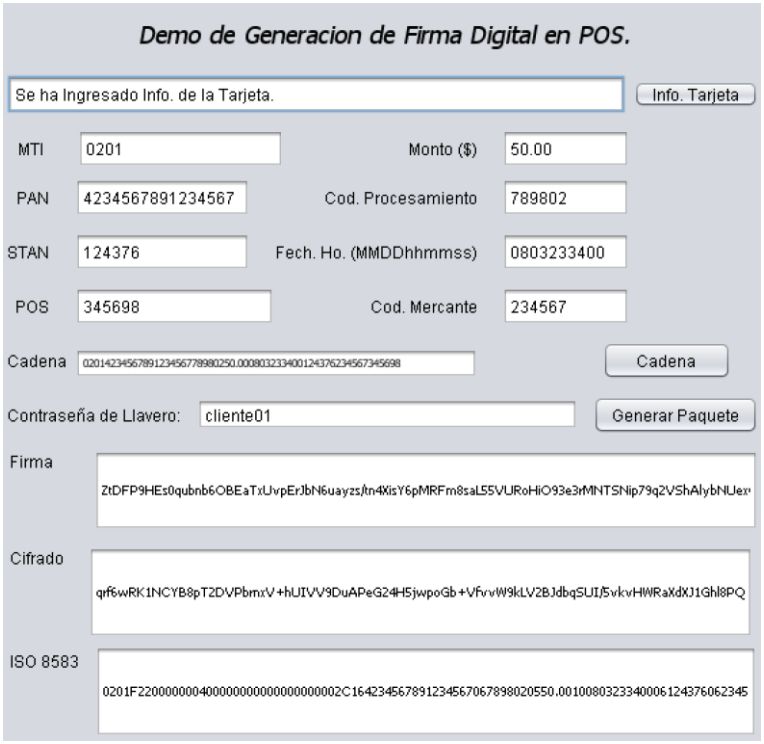
Punto 6-Generación Certificado por AC para Cliente.
openssl ca -in CLIcsr.pem -out CLIcer.pem

Punto 7-Exportación de llavero y certificado a formato p.12 para banco.
openssl pkcs12 -export -in BACcer.pem -inkeyBACllavero.pem -name banco > BACpaquete.p12

Punto 8-Exportación de llavero y certificado a formato p.12 para cliente.
openssl pkcs12 -export -in CLIcer.pem -inkeyCLIllavero.pem -name cliente> CLIpaquete.p12

ANEXO 2. Generación de Firma Digital.

Punto 1-Imagen de Generador de Firma Digital, Cifrado y Paquete ISO 8583.



Punto 2-Codigos en Java para “Demo de Generación de Firma Digital en POS”.

-basic2.xml: Archivo XML que utilizara para delimitar información que se ingresara a paquetería JPOS.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE isopackager SYSTEM "genericpackager.dtd">
<isopackager>
 <isofield
    id="0"
    length="4"
    name="MESSAGE TYPE INDICATOR"
    class="org.jpos.iso.IFA_NUMERIC"/>
 <isofield
    id="1"
    length="128"
    name="BIT MAP"
    class="org.jpos.iso.IFA_BITMAP"/>
 <isofield
    id="2"
    length="16"
    name="PAN"
    class="org.jpos.iso.IFA_LLCHAR"/>
 <isofield
    id="3"
    length="6"
    name="PROCESSING CODE"
    class="org.jpos.iso.IFA_LLCHAR"/>
 <isofield
    id="4"
    length="12"
    name="AMOUNT, TRANSACTION"
    class="org.jpos.iso.IFA_LLCHAR"/>
 <isofield
    id="7"
```

```
    length="10"
    name="TRANSMISSION DATE AND TIME"
    class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
    id="11"
    length="6"
    name="SYSTEM TRACE AUDIT NUMBER"
    class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
    id="42"
    length="10"
    name="ADITIONAL RESPONSE DATA"
    class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
    id="123"
    length="15"
    name="POS DATA CODE"
    class="org.jpos.iso.IFA_LLLCHAR"/>
   <isofield
    id="125"
    length="500"
    name="CIFRADO"
    class="org.jpos.iso.IFA_LLLCHAR"/>
   <isofield
    id="126"
    length="500"
    name="LLAVE"
    class="org.jpos.iso.IFA_LLLCHAR"/>
</isopackager>
```

-Generador.java: Código del formulario con la configuración de los botones.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package iso.pkg8583;

/**
 *
 * @author Mario Planas
 */

import iso.pkg8583.LeerT;
import java.util.StringTokenizer;
import iso.pkg8583.Firma;
import iso.pkg8583.Cifrado;
import iso.pkg8583.ISO8583;
import iso.pkg8583.EnvioPaquete;
public class Generador extends javax.swing.JFrame {
    LeerT InfT=new LeerT();
    //byte[] Firma;
    String CadTarj, CadLlave, MTI, Camp2, Camp3, Camp4, Camp7, Camp11, Camp42, Camp123, Cadena, Resultado, FirmaOb;
    String FirCiOb[];
    ISO8583 Paq=new ISO8583();
    EnvioPaquete x=new EnvioPaquete();
    /**
     * Creates new form Generador
     */
    public Generador() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
```

```
        // TODO add your handling code here:
        //Lectura de la informaciòn de la tarjeta de débito o credito
        CadTarj=InfT.lectura();
        StringTokenizer st = new StringTokenizer(CadTarj, ";");
        Camp2=st.nextToken();
        jTextField1.setText("Se ha Ingresado Info. de la Tarjeta.");
        jTextField7.setText(Camp2);
    }

    private void jButton4ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        //Preparaciòn de la informacion a ser firmada
        MTI=jTextField6.getText();
        Camp4=jTextField2.getText();
        Camp3=jTextField5.getText();
        Camp7=jTextField8.getText();
        Camp11=jTextField9.getText();
        Camp42=jTextField10.getText();
        Camp123=jTextField11.getText();
        Cadena=MTI+Camp2+Camp3+Camp4+Camp7+Camp11+Camp42+Camp123;
        jTextField4.setText(Cadena);

    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        //Ejecucion para enviar a firmar y cifrar informacion
        Firma f=new Firma();
        Cifrado c=new Cifrado();
    try{
        //Firma de la Informaciòn
        FirmaOb=f.PrivateKey(jTextField3.getText(), Cadena);
        jTextField12.setText(FirmaOb);
        //Ejecucion del cifrado de la firma
        FirCiOb=c.Cipher(FirmaOb);
        jTextField14.setText(FirCiOb[0]);
        //Elaboracion de paquete ISO 8583
        Resultado=Paq.GenPaq(MTI, Camp2, Camp3, Camp4, Camp7, Camp11, Camp42, Camp123, FirCiOb[0],FirCiOb[1]);
        jTextField13.setText(Resultado);
        //Envio de Paquete ISO 8583 por txt
        x.envio(Resultado);

        }catch (Exception e){
       System.out.println(e.toString());
    }

    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]) {
        /* Set the Nimbus look and feel */
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Generador().setVisible(true);

            }
        });
    }
// Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton4;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel2;
```

```
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField10;
    private javax.swing.JTextField jTextField11;
    private javax.swing.JTextField jTextField12;
    private javax.swing.JTextField jTextField13;
    private javax.swing.JTextField jTextField14;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    // End of variables declaration
}
```

-leerT.java: Lee la tarjeta de crédito o débito.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
*/
package iso.pkg8583;

/**
 *
 * @author Mario Planas
 */
importjava.io.FileReader;
importjava.io.BufferedReader;

//Clase que representa la lectura de la informaciòn de la tarjeta
public class LeerT {
String texto;



public String lectura(){
try{
FileReader lector= new FileReader ("C:\\Tarjeta.txt");

BufferedReadercontenido = new BufferedReader(lector);

texto=contenido.readLine()+";"/*+contenido.readLine()+";"+contenido.readLine()+";"+
contenido.readLine()+";"+contenido.readLine()+";"+contenido.readLine()+";"+
contenido.readLine()+";"+contenido.readLine()+";"*/;

      }
catch (Exception e){
System.out.println("Error al leer Info. de Tarjeta");
   }
return texto;
}
}
```

-Firma.java: Realiza el proceso de firma con la llave privada del cliente.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package iso.pkg8583;
import java.io.*;
```

```
importjava.security.*;
importjava.security.*;
importjava.security.cert.*;
importjavax.crypto.*;
importjava.security.spec.*;
importjava.nio.charset.StandardCharsets;
import java.util.Base64;
/**
 *
 * @author Mario Planas
 */
publicclass Firma {
   //Clase donde se realizara la firma de la informaciòn.
public String PrivateKey(String clave, String Cadena)
throws Exception {

     Base64.Encoder encoder = Base64.getEncoder();

//Adquisiciòn de la llave privada del Cliente
KeyStoreks=KeyStore.getInstance("PKCS12");
ks.load(new FileInputStream("C:\\CA5\\Cliente-Demo\\CLIpaquete.p12"), clave.toCharArray());
     Key myKey=ks.getKey("cliente", clave.toCharArray());
PrivateKeymyPrivateKey= (PrivateKey)myKey;


     //Ejecucion de la Firma Digital
byte[] bufferPlano=Cadena.getBytes();
     Signature mySign= Signature.getInstance("MD5withRSA");

mySign.initSign(myPrivateKey);
mySign.update(bufferPlano);
byte[] bufferCifrado=mySign.sign();



     String b = encoder.encodeToString(bufferCifrado);

return b;
}
}
```

-Cifrado.java: Realiza el proceso de cifrado de la firma con la llave publica del banco ubicado en el POS, junto a la llave secreta que éste genera.

```
package iso.pkg8583;

import java.io.*;
importjava.security.*;
importjava.security.cert.*;
importjavax.crypto.*;
importjavax.crypto.spec.IvParameterSpec;
importjavax.crypto.spec.SecretKeySpec;
importjava.security.spec.*;
importjava.nio.charset.StandardCharsets;
import java.util.Base64;
/**
 *
 * @author Mario Planas
*/
public class Cifrado {
public String[] Cipher(String Firma) throws Exception {
byte[] FirmaBy;
String[] b=new String[2];
Base64.Decoder decoder2 = Base64.getDecoder();
FirmaBy = decoder2.decode(Firma);

  // Generaciòn de cifrado secreto
String alg="AES";
  String ci="AES/CBC/PKCS5Padding";

  Cipher cifrador1= Cipher.getInstance(ci);
SecretKeySpecLlave = new SecretKeySpec("92AE31A79FEEB2A3".getBytes(),alg);
```

```
IvParameterSpecVar = new IvParameterSpec("0123456789ABCDEF".getBytes());
cifrador1.init(Cipher.ENCRYPT_MODE, Llave, Var);
byte[] FirmaCifrada=cifrador1.doFinal(FirmaBy);


  //Adquisicion de la llave publica del certificado del Banco que se encuentra
//en el POS
FileInputStream is2=new FileInputStream("C:\\CA5\\POS-Demo\\BACcer.pem");
CertificateFactory cf2= CertificateFactory.getInstance("X.509");
X509Certificate certificado2=(X509Certificate)cf2.generateCertificate(is2);

PublicKey myPublickey2 = certificado2.getPublicKey();

    //Cifrado de llave Secreta con la llave publica del banco
Cipher cifrador2 = Cipher.getInstance("RSA");
cifrador2.init(Cipher.ENCRYPT_MODE, myPublickey2);
byte[] LlaveCifrada=cifrador2.doFinal("92AE31A79FEEB2A3".getBytes());

Base64.Encoder encoder2 = Base64.getEncoder();

    //Envio de firma cifrada y llave secreta cifrada.
b[0] = encoder2.encodeToString(FirmaCifrada);
b[1] = encoder2.encodeToString(LlaveCifrada);

return b;
}
}
```

-ISO8583.java: Genera el paquete a ser enviado por la red, ya con la Firma Cifrada y la llave secreta Cifrada.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package iso.pkg8583;

/**
 *
 * @author Mario Planas
 */
importjava.io.IOException;
importorg.jpos.iso.ISOException;
importorg.jpos.iso.ISOMsg;
importorg.jpos.iso.packager.GenericPackager;
publicclass ISO8583 {
 //Clase que maneja la generaciòn del paquete de normativa ISO 8583
public String GenPaq(String MTI, String Camp2, String Camp3, String Camp4, String Camp7, String Camp11, String Camp42, String Camp123, String
FirmaCi, String LlaveCi)
throws Exception {
GenericPackager packager = new GenericPackager("basic2.xml");

ISOMsgisoMsg = new ISOMsg();
isoMsg.setPackager(packager);
isoMsg.setMTI(MTI);
isoMsg.set(2,Camp2);
isoMsg.set(3,Camp3);
isoMsg.set(4,Camp4);
isoMsg.set(7,Camp7);
isoMsg.set(11,Camp11);
isoMsg.set(42,Camp42);
isoMsg.set(123,Camp123);
isoMsg.set(125,FirmaCi);
isoMsg.set(126,LlaveCi);
byte[] data = isoMsg.pack();

return new String(data);
}
}
```

-EnvioPaquete.java: Envía el paquete por la red, el cual será representado en el archivo ISO8583.txt.

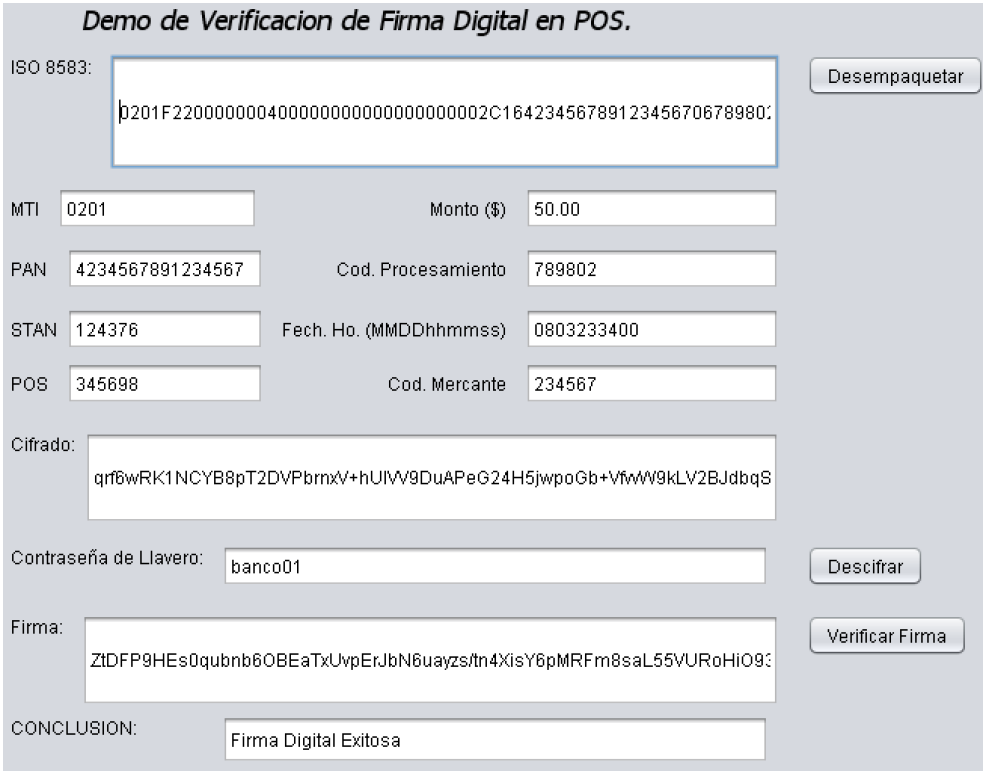
```
/*
```

```
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
*/
package iso.pkg8583;

/**
 *
 * @author Mario Planas
 */
import java.io.*;
public class EnvioPaquete {

public void envio(String mensaje){
//Envio de paquete representado en un txt que funciona en el generador
      //de la informacion del POS
FileWriter fichero=null;
PrintWriter pw=null;
try{
fichero=new FileWriter("C:\\ISO8583.txt");
pw=new PrintWriter(fichero);
pw.println(mensaje);
}catch (Exception e){
e.printStackTrace();
      } finally {
try{
if(null != fichero)
fichero.close();
}catch (Exception e2){
e2.printStackTrace();
      }
      }

   }

}
```

Punto 3-Imagen de paquete ISO8583.txt.

ANEXO 3. Verificación de Firma Digital.

Punto 1-Imagen de Verificador de Firma Digital, Cifrado y Paquete ISO 8583.



Punto 2-Imagen de Verificador de Firma Digital con mensaje de denegación del proceso.

Punto 3-Codigos en Java para “Demo de Verificación de Firma Digital en POS”.

-basic2.xml: Archivo XML que utilizara para delimitar información que se ingresara a paquetería JPOS.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE isopackager SYSTEM "genericpackager.dtd">
<isopackager>
  <isofield
     id="0"
     length="4"
     name="MESSAGE TYPE INDICATOR"
     class="org.jpos.iso.IFA_NUMERIC"/>
  <isofield
     id="1"
     length="128"
     name="BIT MAP"
     class="org.jpos.iso.IFA_BITMAP"/>
  <isofield
     id="2"
     length="16"
     name="PAN"
     class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
     id="3"
     length="6"
     name="PROCESSING CODE"
     class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
     id="4"
     length="12"
     name="AMOUNT, TRANSACTION"
     class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
     id="7"
     length="10"
     name="TRANSMISSION DATE AND TIME"
     class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
     id="11"
     length="6"
     name="SYSTEM TRACE AUDIT NUMBER"
     class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
     id="42"
     length="10"
     name="ADITIONAL RESPONSE DATA"
     class="org.jpos.iso.IFA_LLCHAR"/>
  <isofield
     id="123"
     length="15"
     name="POS DATA CODE"
     class="org.jpos.iso.IFA_LLLCHAR"/>
   <isofield
     id="125"
     length="500"
     name="CIFRADO"
     class="org.jpos.iso.IFA_LLLCHAR"/>
   <isofield
     id="126"
     length="500"
     name="LLAVE"
     class="org.jpos.iso.IFA_LLLCHAR"/>
</isopackager>
```

-Verificador.java: Código del formulario con la configuración de los botones.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package iso.pkg8583;
```

```
/**
 *
 * @author Mario Planas
 */
import java.io.IOException;
import org.jpos.iso.ISOException;
import org.jpos.iso.ISOMsg;
import org.jpos.iso.packager.GenericPackager;
import iso.pkg8583.RecibePaquete;
import iso.pkg8583.Verificacion;
import iso.pkg8583.Descifrado;
import java.nio.charset.StandardCharsets;
import java.util.Base64;

public class Verificador extends javax.swing.JFrame {
String cadena, Resultado, LlaveSec;
byte[] FirmaOb;
RecibePaquete Rec= new RecibePaquete();

    /**
     * Creates new form Verificador
     */
    public Verificador() {
        initComponents();
    }

    /**
     * This method is called from within the constructor to initialize the form.
     * WARNING: Do NOT modify this code. The content of this method is always
     * regenerated by the Form Editor.
     */
    @SuppressWarnings("unchecked")

    private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        try{
        //Recibimiento y desempaquetamiento de txt con ISO 8583
        jTextField1.setText(Rec.lectura());
        cadena=jTextField1.getText();
        GenericPackager packager = new GenericPackager("basic2.xml");
                ISOMsg isoMsg = new ISOMsg();
                                isoMsg.setPackager(packager);
                                isoMsg.unpack(cadena.getBytes());
                jTextField2.setText(isoMsg.getMTI());
                jTextField3.setText(isoMsg.getString(2));
                jTextField6.setText(isoMsg.getString(4));
                jTextField7.setText(isoMsg.getString(3));
                jTextField4.setText(isoMsg.getString(11));
                jTextField8.setText(isoMsg.getString(7));
                jTextField9.setText(isoMsg.getString(42));
                jTextField5.setText(isoMsg.getString(123));
                jTextField10.setText(isoMsg.getString(125));
                LlaveSec=isoMsg.getString(126);
        }
        catch (Exception e)
        {
            System.out.println(e);
        }
    }

    private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        //Clase a Descifrar la Firma
        Descifrado d = new Descifrado();
        //Envio a Descifrar la Firma Recibida
         try{
             jTextField12.setText(d.UNCipher(LlaveSec,jTextField10.getText(),jTextField11.getText()));
        }catch (Exception e){
       System.out.println(e.toString());
    }
    }
```

```
    private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        Base64.Decoder decoder = Base64.getDecoder();
        FirmaOb = decoder.decode(jTextField12.getText());
        //Verificacion de Firma Digital
        Verificacion v = new Verificacion();


        try{ //Envio de Verificacion de Firma Digital
        Resultado=v.PublicKey(FirmaOb, jTextField2.getText()+
                jTextField3.getText()+
                jTextField7.getText()+
                jTextField6.getText()+
                jTextField8.getText()+
                jTextField4.getText()+
                jTextField9.getText()+
                jTextField5.getText());
        jTextField13.setText(Resultado);
        }catch (Exception e){
       System.out.println(e.toString());
    }
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String args[]){
        /* Set the Nimbus look and feel */
        /* Create and display the form */
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Verificador().setVisible(true);
            }
        });
    }

    // Variables declaration - do not modify
    private javax.swing.JButton jButton1;
    private javax.swing.JButton jButton2;
    private javax.swing.JButton jButton3;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel10;
    private javax.swing.JLabel jLabel11;
    private javax.swing.JLabel jLabel12;
    private javax.swing.JLabel jLabel13;
    private javax.swing.JLabel jLabel14;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JLabel jLabel7;
    private javax.swing.JLabel jLabel8;
    private javax.swing.JLabel jLabel9;
    private javax.swing.JTextField jTextField1;
    private javax.swing.JTextField jTextField10;
    private javax.swing.JTextField jTextField11;
    private javax.swing.JTextField jTextField12;
    private javax.swing.JTextField jTextField13;
    private javax.swing.JTextField jTextField2;
    private javax.swing.JTextField jTextField3;
    private javax.swing.JTextField jTextField4;
    private javax.swing.JTextField jTextField5;
    private javax.swing.JTextField jTextField6;
    private javax.swing.JTextField jTextField7;
    private javax.swing.JTextField jTextField8;
    private javax.swing.JTextField jTextField9;
    // End of variables declaration
}
```

-RecibePaquete.java: Código del formulario que lee la información del archivo ISO8583.txt, que será después desglosado.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package iso.pkg8583;

import java.io.BufferedReader;
import java.io.FileReader;

/**
 *
 * @author Mario Planas
 */
public class RecibePaquete {
 String Paquete;

 public String lectura(){
     //Recibe paquete representado en un txt que funciona en el receptor
     //de la informacion del POS
    try{
        FileReader lector= new FileReader ("C:\\ISO8583.txt");

        BufferedReader contenido = new BufferedReader(lector);

        Paquete=contenido.readLine();

        lector.close();

        }
    catch (Exception e){
        System.out.println("Error al leer Paquete ISO");
    }
return Paquete;
}
}
```

-Descifrado.java: Código que descifra primeramente la llave secreta con la llave privada del banco, para luego obtener la firma con la llave secreta.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package iso.pkg8583;
import java.io.*;
import java.security.*;
import java.security.*;
import java.security.cert.*;
import javax.crypto.*;
import java.security.spec.*;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;
import java.nio.charset.StandardCharsets;
import java.util.Base64;
/**
 *
 * @author Mario Planas
 */
public class Descifrado {
    public String UNCipher(String LlaveSecCif, String FirmaCif, String clave) throws Exception {
        Base64.Decoder decoder = Base64.getDecoder();
        byte LlaveSecCifBy[] = decoder.decode(LlaveSecCif);
        byte FirmaCifBy[] = decoder.decode(FirmaCif);

        //Adquisiciòn de la llave privada del Banco
      KeyStore ks=KeyStore.getInstance("PKCS12");
      ks.load(new FileInputStream("C:\\CA5\\Banco-Demo\\BACpaquete.p12"), clave.toCharArray());
      Key myKey=ks.getKey("banco", clave.toCharArray());
      PrivateKey myPrivateKey= (PrivateKey)myKey;
```

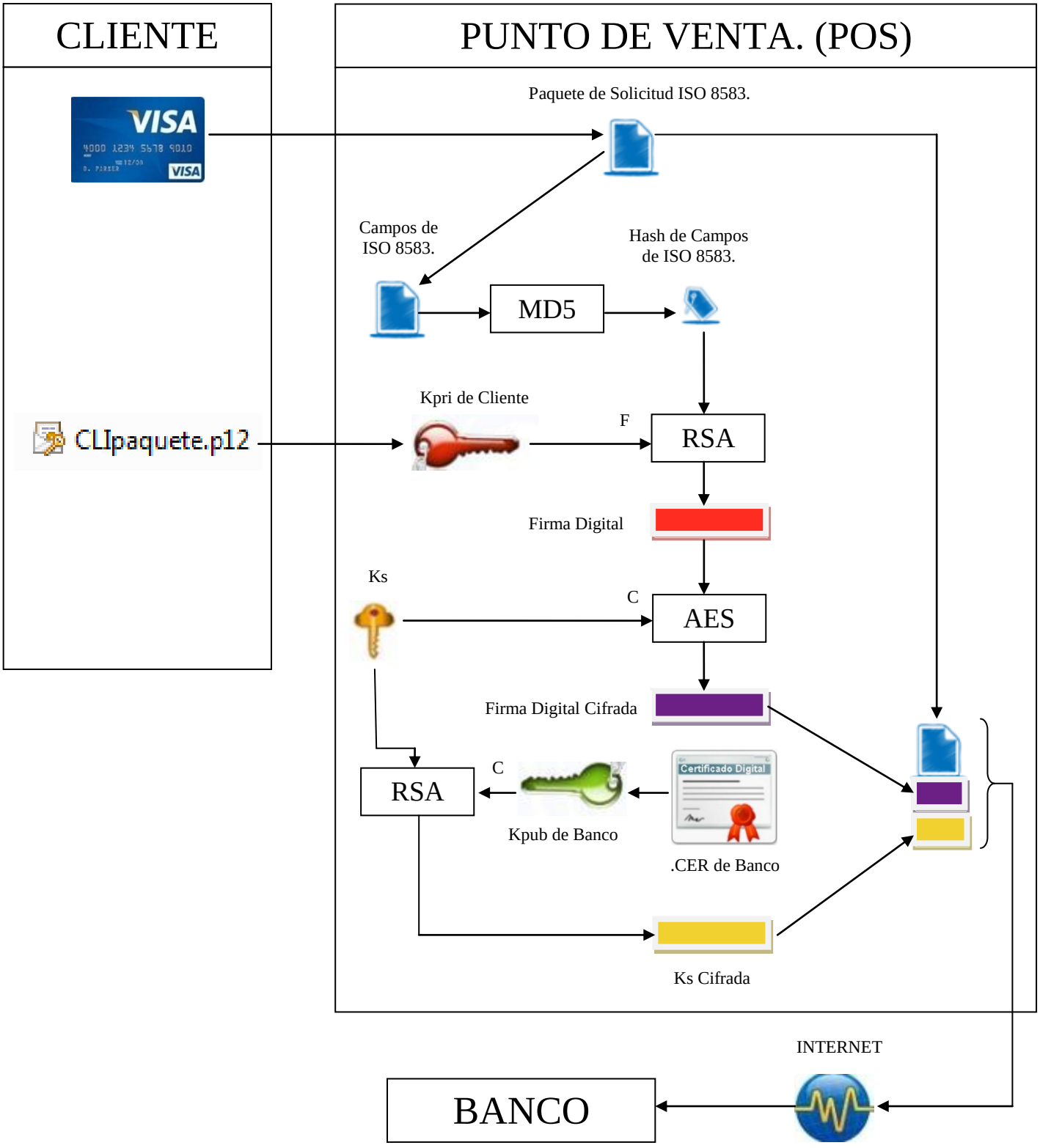
```
      //Descifrado de la llave Secreta con la llave privada del banco
      Cipher cifrador3 = Cipher.getInstance("RSA");
      cifrador3.init(Cipher.DECRYPT_MODE, myPrivateKey);
      byte[] LlaveSec=cifrador3.doFinal(LlaveSecCifBy);

      // Obtencion de la Firma a partir de la llave secreta
        String alg="AES";
        String ci="AES/CBC/PKCS5Padding";

        Cipher cifrador4= Cipher.getInstance(ci);
        SecretKeySpec Llave = new SecretKeySpec(LlaveSec,alg);
        IvParameterSpec Var = new IvParameterSpec("0123456789ABCDEF".getBytes());
        cifrador4.init(Cipher.DECRYPT_MODE, Llave, Var);
        byte[] Firma=cifrador4.doFinal(FirmaCifBy);

        Base64.Encoder encoder3 = Base64.getEncoder();
        return encoder3.encodeToString(Firma);
}
}
```

-Verificacion.java: Código que accede a la llave pública del cliente para luego tomar la huella digital, a fin de comparar el nuevo digesto de la cadena que se le ingresa y validar la veracidad del paquete.

```
/*
 * To change this license header, choose License Headers in Project Properties.
 * To change this template file, choose Tools | Templates
 * and open the template in the editor.
 */
package iso.pkg8583;
import java.io.*;
import java.security.*;
import java.security.*;
import java.security.cert.*;
import javax.crypto.*;
import java.security.spec.*;
/**
 *
 * @author Mario Planas
 */
public class Verificacion {

    //Clase donde se realiza la verificaciòn de la firma digital
    public String PublicKey(byte[] Firma, String Data)
  throws Exception {
    //Adquisicion de la llave publica del certificado del cliente
    FileInputStream is=new FileInputStream("C:\\CA5\\Banco-Demo\\CLIcer.pem");
    CertificateFactory cf= CertificateFactory.getInstance("X.509");
    X509Certificate certificado=(X509Certificate)cf.generateCertificate(is);

    PublicKey myPublickey = certificado.getPublicKey();

    //Obtencion de Huella Digital de la Informacion
    Signature myVerifySign = Signature.getInstance("MD5withRSA");
    myVerifySign.initVerify(myPublickey);
    myVerifySign.update(Data.getBytes());

    //Ejecucion de la verificacion de la firma
    boolean verifySign = myVerifySign.verify(Firma);

if (verifySign == false){
    return "Error en Firma Digital";
}
else
{
    return "Firma Digital Exitosa";
}
}
}
```

ANEXO 4. Flujo del Proceso de Firma y Verificación.

Punto 1-Flujo del proceso de Generación de Firma Digital.
C = Cifrar.
F = Firmar.



Punto 2-Flujo del proceso de Verificación de Firma Digital.
D = Descifrar.
F = Verificar.