# Una marca de agua inteligente aplicada al dinero electrónico

Patricia Jaimes, Gabriel Hermosillo, Gomez Roberto

▶ **To cite this version:**

Patricia Jaimes, Gabriel Hermosillo, Gomez Roberto. Una marca de agua inteligente aplicada al dinero electrónico. 5th Ibero-American Congress on Information Security, CIBSI 09, Universidad de la República, Nov 2009, Montevideo, Uruguay. pp.225-239. inria-00436678

**HAL Id: inria-00436678**
**https://inria.hal.science/inria-00436678**

Submitted on 27 Nov 2009

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Una marca de agua inteligente aplicada al dinero electrónico

Patricia Jaimes[1*], Gabriel Hermosillo[2], Roberto Gómez[1]

[1]ITESM-CEM/Dpto. Ciencias Computacionales, Edo. de Mexico, Mexico.
[2]University of Lille- LIFL- INRIA Project ADAM, Villeneuve d'Ascq, France

[1]{jpatricia,rogomez}@itesm.mx, [2]{gabriel.hermosillo}@inria.fr

**Resumen** El uso de las marcas de agua se ha incrementado, principalmente por la necesidad de proteger los derechos de autor, detener copias ilegales o medir la integridad de los datos de ciertos archivos. Es bien sabido que se puede insertar código ejecutable en imágenes, pero hasta ahora solamente se ha estudiado como una amenaza de seguridad para el usuario. Nosotros proponemos utilizar esta característica de manera segura para expandir las aplicaciones actuales de las marcas de agua, dándoles flexibilidad a través del código ejecutable. Presentamos el modelo de marca de agua inteligente para resolver problemas de incompatibilidad de funciones y demostramos cómo se puede aplicar este modelo a un escenario de dinero electrónico. En dicho escenario el beneficiario puede manejar diferentes implementaciones de dinero electrónico mediante una aplicación estándar. Como parte de este escenario, también proponemos una máquina expendedora de dinero electrónico para ofrecer una opción de pago a los usuarios que no tienen cuenta bancaria.

Palabras clave: marca de agua en imágenes, dinero electrónico, billete electrónico, código ejecutable oculto, esteganografía.

## 1. Introducción

La venta y el intercambio de archivos multimedia (software, imágenes, vídeo, audio) se ha incrementado en los últimos años. Una de las técnicas para brindar seguridad a dichos archivos son las marcas de agua. Estas se han utilizado principalmente para proteger los derechos de autor y detectar copias ilegales, lo cual implica el uso de marcas de agua de tipo robusto. El objetivo de esta categoría es no perder la información oculta ante ataques como distorsiones geométricas, cambios de compresión, resolución, entre algunos otros.

Lo opuesto a las marcas de agua robustas son las de tipo frágil, su objetivo es que la información oculta se destruya ante cualquier ataque, también se usan

[*] El trabajo de este autor fue realizado en el Institut National de Recherche en Informatique et en Automatique (INRIA), Lille, Nord Europe, Francia.

para medir la integridad de los datos, detectar cambios y autenticar imágenes [1]. Una de sus principales aplicaciones es en la medicina, por ejemplo se puede ocultar la información del paciente evitando distorsionar la región de interés de la imagen (electrocardiograma, rayos-x, etc). Nuestra propuesta esta basada en este tipo de marca de agua, de esta forma el dinero electrónico se vuelve inválido ante cualquier modificación.

Las marcas de agua son una rama de la esteganografía, su diferencia reside en el uso que se les asigne. La esteganografía esconde información independientemente del objeto que se utilice para esconder los datos y debe ser estadísticamente indetectable, en cambio en las marcas de agua, es de importancia el objeto en donde se insertarán los datos a ocultar y normalmente es de conocimiento público que el objeto contiene una marca. En nuestro caso al aplicar una marca de agua al dinero electrónico es de importancia la imagen que se usa para esconder los datos, ya que para el usuario final es la representación de dinero físico y al mismo tiempo facilita la portabilidad de los datos. Lo anterior puede ser de gran utilidad para incentivar la adopción del dinero electrónico en los usuarios finales.

Existen diversas clasificaciones de marcas de agua, dependiendo del tipo de archivos multimedia (vídeo, audio, imágenes), la técnica de inserción (sustitución, adición, transformación del dominio) [2], la percepción y pérdida de información (calidad perceptual, métricas basadas en píxeles) [3], la necesidad de la cubierta original para extraer el contenido (ciego, semi ciego, no ciego) [1], el objeto portador (tarjeta inteligente, software), el nivel de anonimato en caso de usar una huella digital (simétrica, pública, anónima) [4], la cantidad de información insertada [3], la integridad contra ataques maliciosos (frágil, semi frágil, robusta) [5], y el propósito de la aplicación (control de copias, propiedad intelectual, monitoreo de transmisiones, imágenes médicas) [2].

Nuestra propuesta es una marca de agua inteligente, la cual contiene una imagen con código ejecutable y mecanismos de seguridad que mantienen la integridad y autenticidad de la imagen. Esta marca puede ser utilizada en problemas de incompatibilidad de funciones, como es el caso de los modelos de dinero electrónico o las diferentes reglas de negocio en los boletos de avión. Ejemplificamos el uso de nuestro modelo aplicándolo a un escenario de dinero electrónico, dentro del cual proponemos una nueva opción de pago que no necesita de cuenta bancaria por parte del usuario.

El dinero electrónico, también conocido como *e-cash* o *e-money*, es solamente uno de los muchos métodos de pago en el mercado y su principal ventaja es el nivel de anonimato del usuario. Uno de los problemas con los modelos de dinero electrónico es la incompatibilidad entre ellos, ya que su técnica para generar y cobrar el dinero es diferente. Podemos encontrar un ejemplo de esto en Japón, donde existen modelos incompatibles de dinero electrónico [6,7,8,9], por lo tanto los comerciantes necesitan tener diferentes sistemas para poder recibir los pagos de cada modelo y los clientes deben poner atención en comprar solamente en tiendas donde su modelo de dinero electrónico es aceptado [10]. Con la marca de agua inteligente, los comerciantes necesitarán tener una única aplicación están-

dar que manejará cualquier tipo de dinero electrónico y los emisores de dinero podrán implementar cualquier modelo.

Otro uso de la marca de agua inteligente puede ser en los boletos de avión, en donde se puede tener un mostrador compartido entre todas las aerolíneas en el cual se reciban los boletos electrónicos, los cuales contienen la marca de agua inteligente con la información relacionada al vuelo. La aplicación estándar se encargará de validar, extraer y ejecutar el código que contiene la lógica para interactuar con el usuario y realizar la comunicación necesaria con la línea aérea. El modelo de marca de agua inteligente simplifica la interacción entre diferentes entidades y expande las aplicaciones actuales de las marcas de agua, dándoles flexibilidad a través del comportamiento configurable en el código ejecutable.

El resto de este artículo está organizado de la siguiente manera: La sección 2 presenta la motivación y algunos trabajos relacionados. La sección 3 muestra nuestro modelo de marca de agua inteligente. La sección 4 muestra la aplicación del modelo en un escenario de dinero electrónico y nuestra propuesta de una máquina expendedora de dinero electrónico. Finalmente, la sección 5 concluye y discute el trabajo futuro.

## 2. Motivación y trabajos relacionados

El sistema de dinero electrónico ideal, descrito por Okamoto [11], debe ser anónimo, independiente de cualquier condición física, capaz de prevenir copias no autorizadas, divisible, transferible a otros usuarios y no debe requerir en todo momento de una conexión. El dinero electrónico intenta ser el equivalente al dinero físico. En los últimos años se han propuesto diversos modelos de dinero electrónico, sin embargo solamente algunos se han comercializado, esto se debe a que el usuario final no adopta este tipo de pago o en ocasiones no satisface las leyes del país. En algunos otros casos, el modelo suele ser incompatible con las aplicaciones de pago existentes o simplemente no ha sido implementado. Algunos ejemplos de dinero electrónico que se utiliza actualmente son: Chipknip (Holanda) [12], Proton (Bélgica) [13], Geldkarte (Alemania) [14], Moneo (Francia) [15], Felica (Japón) [6] y Suica (Japón) [7].

Un proyecto de dinero electrónico que utiliza código ejecutable y un certificado es X-Cash [16]. Tanto el código como el certificado son enviados a través de la red, con el fin de encontrar en algún lugar la mejor opción para el objeto y el precio que el usuario busca. Sus autores lo llegan a comparar con un tipo de virus amigable, el cual está basado en un paradigma de agente móvil. El cliente paga al vendedor con un X-Cash firmado que contiene los elementos para comprar y la cantidad de dinero a ser pagado. El vendedor puede entonces intercambiar el X-Cash por dinero físico en el banco del cliente. El banco ejecuta el programa de X-Cash, utiliza los elementos como entradas y determina la cantidad de dinero que se le debe al vendedor. En este modelo todos los elementos se tratan de manera separada y no están insertados en ningún archivo.

Una de las características de nuestra propuesta es reunir toda la información en un solo archivo, lo cual puede otorgar al dinero electrónico mayor portabilidad.

Lo anterior ataca el problema de adopción por parte del usuario, ya que ayudaría a crear una fácil abstracción del dinero físico.

Algunos trabajos de investigación han encontrado una amenaza de seguridad cuando se esconde código ejecutable dentro de un archivo [17,18,19]. Esta vulnerabilidad puede ser explotada si previamente se engaña al usuario para que instale un software que se encargue de extraer el código ejecutable en la computadora del usuario. En algunos casos, el código ejecutable puede insertarse a sí mismo en nuevos archivos sin que el usuario se de cuenta. Algunos ejemplos de este tipo de virus son Perrun y Trojan Frog [20,21], los cuales fueron creados para demostrar que un antivirus no puede detectar el código ejecutable insertado en las imágenes. Otra manera de ejecutar código insertado en una imagen es explotando una antigua vulnerabilidad de desbordamiento de búfer en el procesamiento de imágenes JPEG dentro de la interfaz de dispositivo gráfico (GDI por sus siglas en inglés) de Microsoft[22]. En este caso no se necesita instalar previamente un software que extraiga el código, ya que un atacante puede utilizar esta vulnerabilidad si guarda correctamente código ejecutable y algunas direcciones de memoria en una imagen JPEG, por lo tanto cuando un usuario abra la imagen, el GDI ayudará a ejecutar el código insertado.

Nuestra propuesta es una marca de agua inteligente que contiene código ejecutable insertado en una imagen, además de una firma digital e información sobre el objeto (en el contexto de dinero electrónico la información son los datos, como nombre del banco, número de identificación, etc). Este tipo de marca de agua se puede aplicar sobre diferentes contextos, por ejemplo, billetes de lotería, billetes de avión, o sobre distintos modelos de dinero electrónico. En este último caso el emisor de dinero puede implementar el código insertado que contendrá la lógica del modelo. El usuario final únicamente tendrá una imagen, la cual puede ser fácilmente relacionada con el dinero físico y nunca tendría que manejar los datos en crudo del dinero electrónico, ni las aplicaciones para interpretarlos. Además, las entidades emisoras de dinero electrónico podrían implementar comportamientos especiales en algunos billetes, como premios, encuestas, estadísticas, etc.

Dentro del escenario de aplicación de nuestro modelo, proponemos un nuevo enfoque para el dinero electrónico basado en las necesidades de los países en desarrollo, donde en los últimos años los pagos a través de Internet sin tener una cuenta bancaria han creado un reto. En un estudio reciente, las Naciones Unidas reportaron que la mayoría de las transferencias electrónicas de dinero en los países en desarrollo se realizan a través de giro postal o telefonía móvil, dado que mucha gente no tiene una cuenta bancaria [23]. Esto significa que los pagos son prácticamente anónimos, pues en los países en desarrollo los teléfonos móviles no están ligados a una persona y los giros postales pueden tener cualquier dirección. Un estudio mexicano realizado en el 2008 muestra que el comercio electrónico ha crecido 78 % en el último año y podría ser mejorado si la gente tuviera más opciones de pago [24]. El mismo estudio mostró que el 50 % de la gente que tiene una tarjeta de crédito nunca ha comprado en Internet, debido a que se sienten inseguros al dar su información bancaria. Al utilizar nuestro modelo de dinero electrónico, los usuarios serán capaces de comprar en Internet sin tener

que exponer su información personal y financiera e incluso sin necesidad de tener una cuenta bancaria.

## 3. El modelo de marca de agua inteligente

Nuestra propuesta de una marca de agua inteligente puede ser utilizada para resolver problemas de incompatibilidad de funciones. Como se muestra en la Fig. 1, en este tipo de problemas se tienen varios distribuidores y cada uno puede crear distintos tipos de objetos. Estos objetos son conjuntos de datos que toman valor y sentido cuando son procesados por el software correcto, dichos objetos son enviados al usuario, quien necesita una aplicación diferente para utilizar cada objeto.

El distribuidor necesita proveer al usuario de la aplicación correspondiente o renovar la actual para cada versión y tipo de objeto. Lo anterior representa un problema para el usuario, pues necesita manejar un número creciente de aplicaciones para poder aceptar los diferentes objetos.

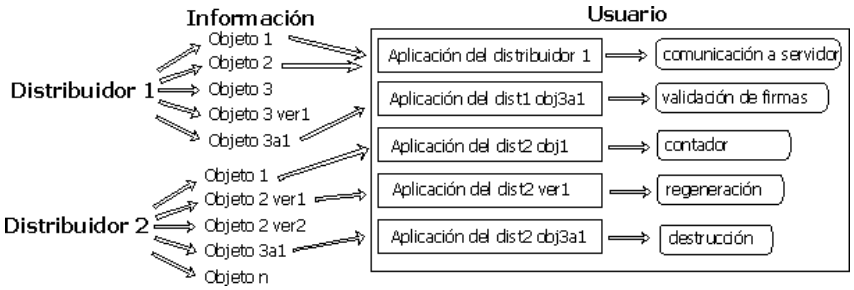


**Figura 1.** Problemas de incompatibilidad de funciones

Es cierto que si todos los datos que actualmente se intercambian se enviaran junto con la aplicación que los procesa se crearían archivos de gran cantidad de bytes, inadministrables para el usuario y sin sentido ya que se podrían reutilizar. Sin embargo existen situaciones en donde las aplicaciones son pequeñas y los diferentes objetos se pueden agrupar bajo un mismo tema, los cuales tienen una gran variedad de versiones, en donde se añaden y disminuyen constantemente los objetos. También puede ser que los usuarios cambien con frecuencia y que muchos solo utilicen el software sola una vez. Lo anterior además de consumir tiempo puede ser inseguro, ya que los usuarios no verifican que el software que instalan sea integro y que provenga de una entidad válida. Por lo tanto en este tipo de casos conviene que los datos se encuentren acompañados de su aplicación.

Un ejemplo se puede observar en el dinero electrónico ya que su objetivo final es que sea transferible entre usuarios finales tal como lo es el dinero físico. Debido a que existen muchos modelos de dinero electrónico es probable que un usuario casual no vuelva a cobrar un billete del mismo modelo y versión dentro

de mucho tiempo. Para el caso de comercios pueden tener previamente instalado el software necesario para manejar cada modelo de billete electrónico y de esta forma evitar extraer el código ejecutable de cada billete, además con la ventaja de poder recibir nuevos modelos o cambios de versiones hasta que se realice la instalación del nuevo software fijo.

Nuestra propuesta de marca de agua inteligente puede ayudar a integrar en un solo archivo un conjunto de datos con su aplicación de manera portable y segura para el usuario. Primero explicaremos la creación de la marca de agua inteligente y luego detallaremos la aplicación estándar encargada de manejar cualquier tipo de objeto basado en dicha marca.

La meta común de cada distribuidor $D$ es crear objetos $O$, los cuales contienen distinta información $info$ (i.e. un número único, el nombre del distribuidor, etc.) y una o varias firmas digitales. En nuestro modelo, en lugar de obtener $O$, obtendremos una marca de agua inteligente $I_{marcaInte}$, como se puede observar en la Fig. 2. Cada $D$ desarrolla su propio código ejecutable $C$, el cual contiene la lógica de la aplicación para manejar los datos de $info$ y puede realizar diversas tareas, como comunicarse con $D$, generar un nuevo objeto, destruir un objeto, interactuar con un usuario $U$, manejar una marca de agua o una huella digital adicional, etc.

$D$ utiliza una función de inserción $E()$ la cual introduce $C$ dentro de la imagen original $I_{orig}$, y como resultado obtenemos una imagen que contiene el código ejecutable a la que le llamaremos $I_{marca}$. Lo anterior se expresa como $I_{marca} = E(I_{orig}, C)$. La función $E()$ dependerá del tamaño del código, del objetivo de la aplicación y del desempeño deseado. $D$ no necesita crear una $I_{marca}$ por cada objeto, porque puede reutilizar una existente, como se explicará más adelante. Cada $D$ tiene su propio certificado $K$, creado por una Autoridad Certificadora (AC) utilizando sus llaves asimétricas $(sk_{dist}, pk_{dist})$.

$D$ generará una firma digital $F_{dw}$ utilizando su llave privada $sk_{dist}$, los datos $info$, la $I_{marca}$ y la función $Sig$ en la cual se implementan los algoritmos. Dicha firma digital se obtiene a partir de $F_{dw} = Sig_{sk_{dist}}(info||I_{marca})$. Esto le confirmará a todo usuario $U$ la integridad de $C$ y de $info$, mientras que también le asegurará que la $I_{marca}$ fue creada por un $D$ válido, utilizando el $K$ de $D$ para comprobar la firma. Finalmente, la marca de agua inteligente $I_{marcaInte}$ se obtiene utilizando una función de unión $J()$, la cual se expresa $I_{marcaInte} = J(F_{dw}||info||I_{marca})$.

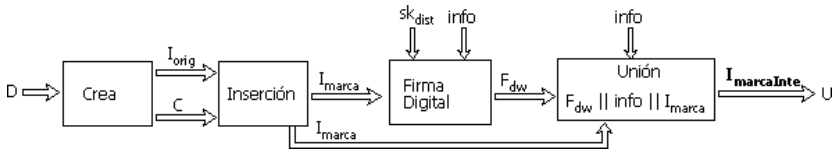


**Figura 2.** Generación de la marca de agua inteligente

Los distribuidores necesitan definir las implementaciones de las funciones $E()$ y $J()$, así como el tamaño y el orden de la información común que se va a utilizar en el campo $info$, en el cual también se puede agregar información específica de cada distribuidor que no requiere ser acordada por el conjunto de distribuidores. Los datos $info$ pueden ser diferentes para cada $I_{marcaInte}$, dependiendo de la aplicación, pero la $I_{marca}$ puede ser la misma para todos, ya que lógica del código ejecutable insertado no cambia a menos de que se trate de un nuevo tipo o versión de $I_{marcaInte}$ . Esto quiere decir que $D$ solamente necesitará crear una $I_{marca}$ para crear cualquier número de $I_{marcaInte}$. La función $J()$ puede ser utilizada tanto por $D$ como por $U$, pero lo mejor es que sea utilizada por $U$, ya que esto permite distribuir el poder de cómputo necesario para la función, en lugar de centralizar todo en $D$.

Una vez creada, la $I_{marcaInte}$ puede ser utilizada por $U$ para interactuar con cualquier otro usuario $U$, a quien en este caso llamamos $U_2$. Para lograr esta interacción, $U_2$ necesita tener una única aplicación estándar $A$ que sea compatible con los estándares definidos por el grupo de distribuidores $D$. Para utilizar la información contenida en la $I_{marcaInte}$, la aplicación $A$ emplea la función de separación $S()$, expresada como $S(I_{marcaInte}) = info, F_{dw}, I_{marca}$, de la cual se obtienen los elementos contenidos en la imagen. Enseguida $A$ verifica la firma digital $F_{dw}$ aplicando la llave pública del distribuidor $pk_{dist}$ en la función $Verifica_{pk_{dist}}(F_{dw}, info||I_{marca})$, y si la verificación es exitosa entonces $U_2$ puede confiar en la integridad de la información y puede estar seguro de que el objeto fue creado por un $D$ autorizado. La aplicación estándar $A$ puede contener diversas implementaciones de $F_{dw}$, dependiendo en los estándares definidos por el grupo de $D$. Para extraer el código $C$ incrustado en $I_{marca}$, $U_2$ deberá utilizar la función de extracción $X(I_{marca}) = C$, para luego ejecutar $C$ utilizando los datos de $info$ como entrada. La Fig. 3 muestra cómo funciona la aplicación estándar.

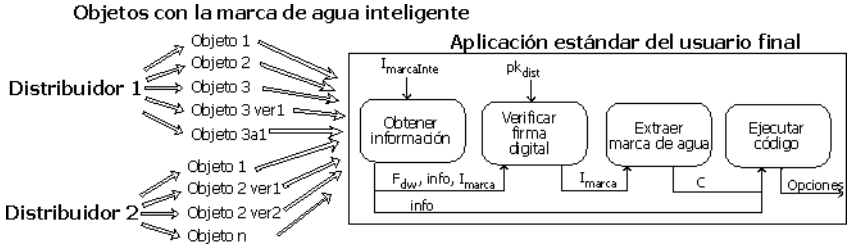


**Figura 3.** Una aplicación estándar para la marca de agua inteligente

## 4. Aplicando el modelo de marca de agua inteligente al dinero electrónico

Para probar nuestro modelo de marca de agua inteligente, utilizaremos un escenario de dinero electrónico en el que participan tres diferentes bancos, llamados $B_1$, $B_2$ y $B_3$, quienes tiene cada uno su propia implementación de dinero electrónico. Para utilizar la marca de agua inteligente, los bancos han elegido el formato de imagen TIFF (Tagged Image File Format), para crear los archivos de dinero electrónico [25]. Para la implementación elegimos este tipo de archivo sin embargo la marca de agua inteligente también se podría implementar en archivos de tipo PNG, JFIF o cualquier archivo que pueda contener una sección de datos extra a la imagen.

En las etiquetas del archivo TIFF almacenamos la firma digital y los datos, que son: la denominación, el nombre del banco, el tamaño del código ejecutable y la información de la imagen. Cada banco puede agregar la información necesaria en la sección de etiquetas según el modelo de dinero electrónico que elija. En el siguiente segmento, de acuerdo con la especificación de los archivos TIFF, deberá estar la imagen que en este caso es de tipo JPEG, ya que permite modificar su calidad y tamaño de acuerdo a la necesidad de cada banco, esta imagen es la que contiene el código ejecutable.

En nuestro escenario, los bancos acordaron en usar la técnica F5 para insertar y extraer el código en la imagen [26], ya que dicha técnica opera sobre los coeficientes de la transformada discreta del coseno (DCT por sus siglas en inglés), no tiene pérdidas en los datos ocultos y distribuye la información a lo largo de toda la imagen disminuyendo las distorsiones. Este escenario es solo un ejemplo de aplicación y podrían utilizar JSteg o cualquier otra técnica esteganográfica [27].

La anterior técnica la estamos utilizando como una marca de agua frágil, ya que se desea que ante cualquier modificación de la imagen (rotación, translación, recorte, etc.) el dinero electrónico sea inválido. Estamos utilizando el modelo de marca de agua inteligente únicamente con una función de inserción, pero los bancos pueden estipular más funciones siempre y cuando se entregue también la función de extracción en la aplicación estándar del usuario.

Para obtener la firma digital de cada billete, los bancos insertan solamente una vez el código ejecutable en la imagen JPEG y luego pueden reutilizarla al concatenarla con los datos únicos de cada billete y a partir de esto calcular su firma. El banco siempre envía la firma digital y los datos al usuario, en algunos casos también envía la imagen que contiene el código ejecutable dependiendo si el usuario no cuenta con una copia de la imagen. La aplicación estándar del usuario une toda la información y crea el archivo TIFF, en el cual se insertan los datos del billete, la firma digital y la imagen con el código. La Fig. 4 muestra la información contenida dentro de un archivo TIFF.

En la Fig.5 se muestra la interacción general de las entidades involucradas aplicando la marca de agua inteligente en la mayoría de modelos de dinero electrónico. El dinero electrónico creado por $B_1$ utiliza como modelo de dinero electrónico una firma parcialmente ciega, basada en el problema del logaritmo dis-

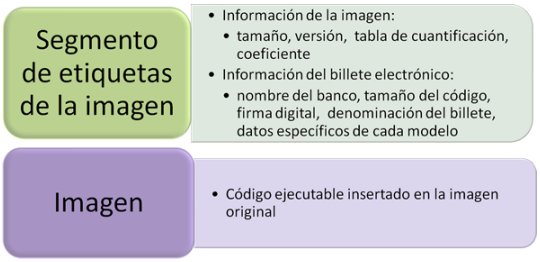


**Figura 4.** Campos del dinero electrónico con la marca de agua inteligente

creto [28]. En este modelo el usuario se registra la primera vez ante una tercera entidad de confianza, la cual crea un certificado digital con su llave pública. El usuario al momento de retirar un billete electrónico de su banco, envía una parte de los datos del billete cegados mediante una firma especial y otra en claro. Al aplicar la marca de agua inteligente el banco tiene lista la imagen con el código oculto mediante F5 y los datos. El banco firma ciegamente los datos de acuerdo a este modelo y envía la firma digital general de la marca de agua inteligente, la cual como acordaron los bancos, se calcula con SHA y RSA. El usuario recibe la información y la une en el archivo TIFF por medio de la misma aplicación con la que realizó la petición de billetes. Finalmente este archivo es el billete electrónico.

Posteriormente el usuario compra un bien al beneficiario $P$ y le paga con el billete, además entrega también su certificado. $P$ tiene instalada la aplicación estándar la cual separa la información del archivo TIFF y obtiene los datos de las etiquetas y la imagen JPEG. Luego la aplicación localiza el certificado digital que contiene la llave pública del banco y valida la firma digital. Si la firma es válida extrae el código de la imagen mediante el algoritmo de extracción F5.

El código implementa el modelo de cobro elegido por el banco $B_1$, el cual valida el certificado del usuario mediante la llave pública del banco y genera un reto para el usuario a partir del certificado. Si el usuario contesta correctamente su pago es aceptado porque significa que realmente es el dueño del billete. Finalmente, el código ejecutable le pregunta a $P$ si desea depositar el dinero en su cuenta o si desea hacerlo más tarde. Esto es posible ya que este es un modelo sin conexión, si el usuario llegara a gastar otra vez este billete, se termina su anonimidad y podría ser acusado de robo. En el momento en que $P$ decide realizar el depósito, la aplicación envía el dinero electrónico al banco de $P$, donde es validado nuevamente y depositado.

Como segunda opción en este escenario proponemos el uso del billete electrónico condicional [29], el cual sería creado por $B_2$. Para ejemplificarlo más fácilmente podemos tener un comercio de apuestas el cual realiza una petición de billetes al banco. El banco genera el billete bajo la marca de agua inteligente por lo tanto ya tiene imágenes con código oculto mediante el algoritmo F5 y crea los datos necesarios, que ya hemos mencionado (tamaño, versión, nombre

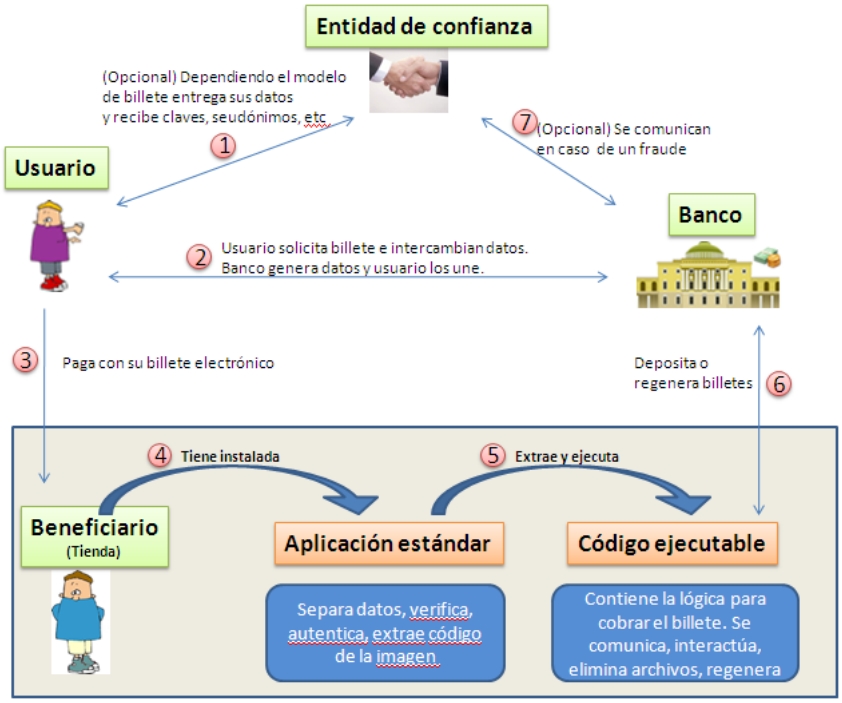


**Figura 5.** Aplicación de la marca de agua inteligente de manera general en los modelos de dinero electrónico

del banco, tamaño del código, denominación, etc). El banco firma usando SHA y RSA y envía lo anterior al comercio de apuestas.

La aplicación que realizó la petición del billete del comercio de apuestas une los datos en el archivo TIFF además de aumentar cierta información propia y un secreto según lo indicado en este modelo. Como ejemplo el comercio de apuestas tiene identificado cierto tipo de secretos para el equipo de fútbol 1 y otros para el equipo de fútbol 2. El comercio entrega a los clientes los billetes según su equipo favorito. Los clientes deben tener instalada la aplicación estándar para cobrar el dinero. Si los clientes intentan cobrar el billete sin el secreto correcto, solo podrán comprobar que el billete es auténtico mediante la la firma digital, sin embargo cuando se ejecute el código oculto no podrán cobrar el billete porque no tienen el secreto requerido. Cuando se conoce el resultado del encuentro de fútbol el comercio publica el secreto de los billetes del equipo ganador. Por lo tanto ahora los clientes al procesar el billete con la aplicación estándar pueden introducir el secreto cuando se extrae y ejecuta el código oculto. Al coincidir el secreto, el código realiza una conexión segura con el banco y pide la cuenta del usuario para depositar el billete.

Como podemos ver hasta ahora el modelo de marca de agua inteligente se puede incorporar a los anteriores modelos de dinero electrónico. Como tercera opción en este escenario, nosotros proponemos una máquina expendedora de dinero electrónico, la cual también está basada en el modelo de marca de agua inteligente. Esta solución puede ayudar a incrementar el comercio electrónico en países en desarrollo, donde hay mucha gente que quiere realizar compras en Internet pero que no cuenta con una cuenta bancaria, lo cual es un requisito para la mayoría de las formas de pago en línea. Esta tercera opción se implementó de forma práctica en el lenguaje JAVA.

En este caso, el usuario introduce el dinero físico en la máquina donde $B_3$ distribuye el dinero, la cual solicita un nuevo billete electrónico a un servidor utilizando una conexión segura. En nuestra implementación simulamos la máquina expendedora con un software, en el cual el usuario selecciona las opciones de dinero físico. El servidor que representa al banco, tiene implementados diferentes servicios web, una base de datos para guardar la información de cada billete y previamente oculta el código ejecutable en ciertas imágenes. Dicho código se encuentra compactado en un archivo JAR, el cual contiene la lógica de cobro del billete y ofrece distintas opciones al usuario, está imagen se encuentra guardada en la máquina expendedora y se puede actualizar según lo indique el banco.

El servidor al recibir una petición de nuevo billete, le responde a la máquina expendedora con los datos del billete (el nombre del banco, el tamaño del código ejecutable, la denominación, la versión de la implementación, un número único de identificación) y la firma digital de dichos datos. La firma se implemento utilizando SHA y RSA. La máquina expendedora como ya tiene previamente guardadas las imágenes de las diferentes denominaciones con el código ejecutable insertado, solo realiza la copia de una imagen y la agrega al archivo TIFF junto con los datos y firma del billete que recibe. Finalmente el usuario guarda el dinero electrónico en algún dispositivo de almacenamiento, como una memoria USB o en su teléfono celular. En la Fig. 6 se muestra la máquina expendedora en el escenario de dinero electrónico.

Cuando el usuario entrega este tipo de dinero electrónico a $P$, este último también puede utilizar la máquina expendedora para cambiar dinero electrónico por dinero físico. En este caso, la máquina expendedora contactará al servidor para verificar el dinero electrónico y registrarlo como cobrado.

Otra opción que $P$ tiene para cobrar el dinero, es utilizar la aplicación estándar, la cual en nuestra implementación lee y muestra al usaurio la información de las etiquetas del archivo TIFF que se encuentran identificadas por números y se obtiene la imagen JPEG, para finalmente verificar la firma digital. La aplicación estándar tiene una sección para dar de alta o baja certificados digitales los cuales hayan sido creados por una autoridad bancaria y a partir de estos certificados se obtiene la llave pública necesaria para verificar la firma.

Si la firma es válida y el usuario decide cobrar el billete, se extrae el código ejecutable de la imagen JPEG por medio de la técnica F5 y se ejecuta el código. En este punto la lógica de cobro que da sentido a los datos del billete es la que se

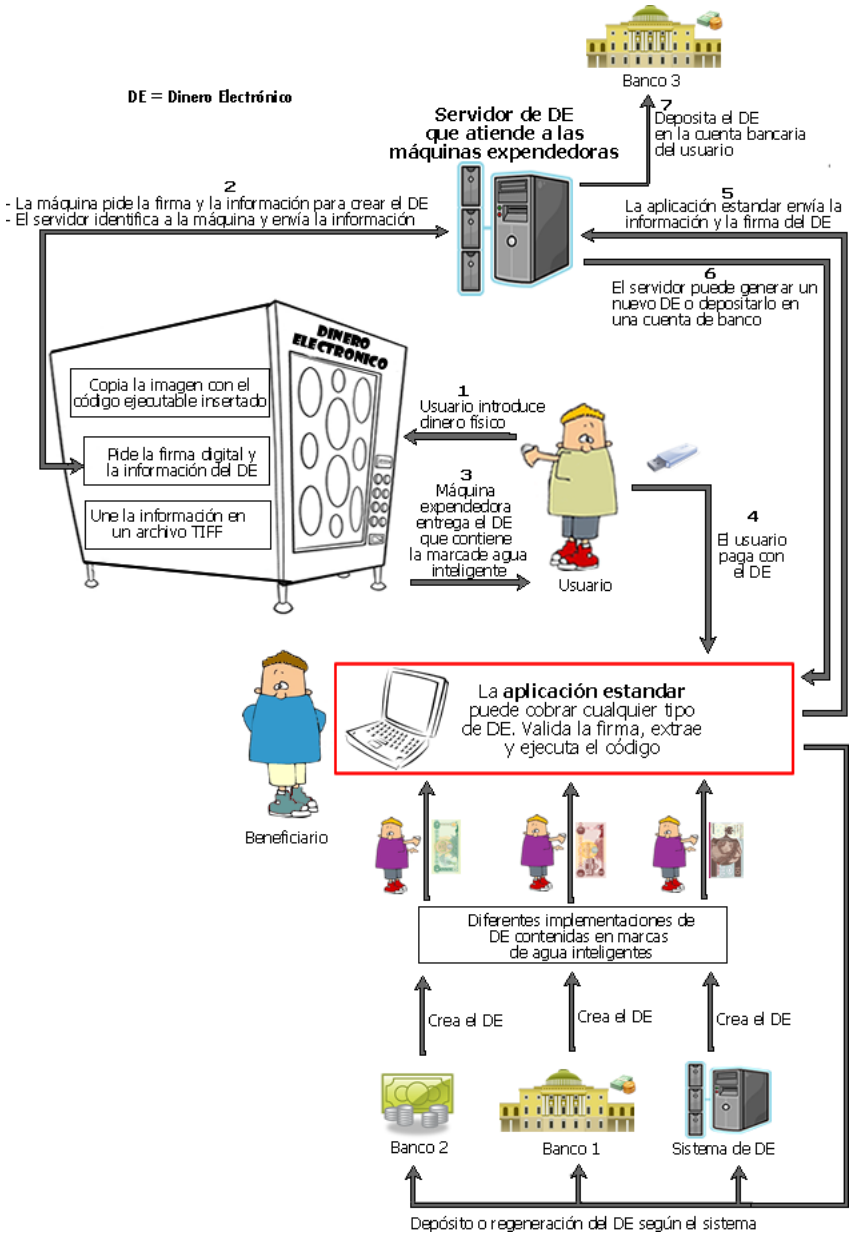


**Figura 6.** Máquina expendedora de dinero electrónico

encuentra en el código ejecutable, el cual ofrece al usuario dos opciones: generar un nuevo billete o depositarlo en una cuenta bancaria.

En ambos casos el código se conecta al servicio web ofrecido por el banco y envía los datos y la firma digital. El servidor verifica la autenticidad de los datos y revisa que el identificador único del billete no haya sido cobrado previamente. En caso de la opción de generar un nuevo billete, el servidor regresa un nuevo segmento de datos y una firma digital, los cuales los recibe el código en ejecución y se utilizan para generar un nuevo archivo TIFF de la misma forma que lo hizo la máquina expendedora. Dicho billete volverá a tener la función de dinero electrónico y puede usarse para pagar a otro usuario. En caso de elegir depositarlo en una cuenta bancaria, la implementación del código solicita a $P$ los datos de la cuenta y envía la petición al banco correspondiente.

Se realizaron pruebas instalando la máquina expendedora y la aplicación estándar en diferentes máquinas, se regeneraron entre ellas varios billetes de diferentes denominaciones y se depositaron a diferentes cuentas los billetes. Se realizaron modificaciones a los billetes, las cuales fueron detectadas por la aplicación estándar al avisar que la firma digital no era válida. También se hicieron pruebas con diferentes códigos ejecutables ocultos los cuales algunos ofrecían premios a los usuarios. El código ejecutable en el archivo JAR midió aproximadamente 23Kb dependiendo de las opciones que se ofrecieron y las imagenes usadas fueron de 474 x 308 píxeles.

## 5. Conclusiones

Hemos presentado un nuevo modelo llamado marca de agua inteligente, el cual simplifica la interacción entre distintas entidades y expande las aplicaciones actuales de las marcas de agua, dándoles flexibilidad a través de la implementación del código ejecutable. La marca de agua inteligente es segura para los usuarios, y el código es únicamente ejecutado cuando la aplicación estándar instalada en la computadora del usuario valida que el archivo es íntegro y que fue creado por un distribuidor válido. La función de inserción es realizada solamente una vez por el distribuidor para crear tantos objetos como desee y la función de unión puede ser realizada por el usuario, distribuyendo así el poder de cómputo necesario.

Aplicamos este concepto para resolver incompatibilidades entre diferentes implementaciones de dinero electrónico, de esta forma los beneficiarios pueden aceptar, de manera transparente, cualquier implementación creada bajo el mismo modelo de marca de agua inteligente. Otra des sus ventajas es agrupar toda la información de los modelos de dinero electrónico en un solo archivo brindando portabilidad y una fácil adopción de la tecnología al usuario.

Los diferentes distribuidores de dinero electrónico deben acordar previamente el tipo de archivo y datos comunes que utilizarán. Sólo necesita ser instalada una única aplicación estándar en la máquina del beneficiario, para poder manejar todos los archivos de dinero electrónico que los usuarios entreguen. Para mostrar el uso del modelo de marca de agua inteligente, lo aplicamos en un escenario que

contiene tres implementaciones distintas de dinero electrónico. La última de estas es nuestra propuesta de una máquina expendedora de dinero electrónico, la cual es una opción para evitar el uso de una cuenta bancaria y así ayudar a cubrir las necesidades que tienen los países en desarrollo en el comercio electrónico. La marca de agua inteligente puede simplificar la necesidad de tener diferentes aplicaciones, es segura para los usuarios y cada distribuidor puede implementar libremente sus propias reglas y lógica en el código ejecutable de la imagen.

Como trabajo futuro, proponemos la búsqueda de más aplicaciones de la marca de agua inteligente, para demostrar que nuestro modelo puede ser aplicado en varios escenarios. Se pueden realizar nuevas investigaciones para mejorar las medidas de seguridad que deben ser agregadas a la marca de agua inteligente y así realizar las validaciones de una forma más rápida. Se pueden implementar prácticamente más modelos de dinero electrónico y finalmente, se podrían comparar distintos algoritmos esteganográficos para medir la cantidad de información que logran ocultar, para así obtener el mayor provecho del espacio para el código ejecutable.

## Agradecimientos

Al equipo ADAM en el INRIA Lille Nord Europe (http://adam.lille.inria.fr), por la ayuda y el apoyo recibidos para la realización de este artículo.

## Referencias

1. Petitcolas, F.A.P.: Digital Watermarking. In: Digital Rights Management: Technological, Economic, Legal and Political Aspect. Volume 2770 of LNCS. Springer Berlin - Heidelberg (2003) 81–92
2. Suhail, M., Chun-Shien, L.: Digital Watermarking for Protection of Intellectual Property. In: Multimedia Security: Steganography and Digital Watermarking Techniques for Protection of Intellectual Property. Idea Group Publishing, Covent Garden, London (2004) viii, 255 p.
3. Kutter, M., Petitcolas, F.A.P.: A fair benchmark for image watermarking systems. In Wong, P.W., Delp, E.J., eds.: Electronic Imaging '99, Security and Watermarking of Multimedia Contents. Volume 3657., San Jose, California, U.S.A (1999) 226–239
4. Wang, Y., Lü, S., Liu, Z.: A simple anonymous fingerprinting scheme based on blind signature. In: Information and Communications Security, 5th International Conference, ICICS 2003, Huhehaote, China, October 10-13, 2003. (2003) 260–268
5. Cox, I.: Digital Watermarking: Principles and Practice. Morgan Kaufmann Publishers Inc. (2001)
6. Felica Networks Inc: Mobile felica. http://www.felicanetworks.co.jp
7. JR East: Mobile suica. http://www.jreast.co.jp/e/press/20071201/index.html
8. AEON Group: Waon. http://www.waon.com/
9. Seven and I Holdings Co: Nanaco. http://www.nanaco-net.jp/
10. Callas, J., Desmedt, Y., Nagy, D., Otsuka, A., Quisquater, J.J., Yung, M.: Real electronic cash versus academic electronic cash versus paper cash (panel report). (2008) 307–313

11. Okamoto, T., Ohta, K.: Universal electronic cash. In: CRYPTO '91: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, London, UK, Springer-Verlag (1992) 324–337
12. Netherlands: Chipknip. http://www.chipknip.nl
13. Electronic Payment Certification Institute: Proton. http://www.epci.be
14. EURO Kartensysteme GmbH: Geldkarte. http://www.geldkarte.de
15. BMS Exploitation: Moneo. http://www.moneo.net
16. Jakobsson, M., Juels, A.: X-cash: Executable digital cash. In: FC '98: Proceedings of the Second International Conference on Financial Cryptography, London, UK, Springer-Verlag (1998) 16–27
17. Cochran, J.T.: "Steganographic Computer Warfare". Master's thesis, Air Force University. Wright-Patterson Air Force Base, Ohio, United States. (2000)
18. Gómez, R., Ramírez, G.: Using digital images to spread executable code on internet. In: 6th International Workshop on Innovative Internet Community Systems, I2CS, Neuchâtel, Switzerland (2006)
19. Rogers, M.: Steganographic trojans. In: DEF-CON X, Las Vegas, Nevada (2002)
20. McAfee: Trojan frog on the loose. http://www.avertlabs.com/research/blog/?p=36 (2006)
21. Erdelyi, G.: W32.perrun. http://www.f-secure.com/v-descs/perrun.shtml (2002)
22. Microsoft TechNet: Microsoft security bulletin ms04-028, buffer overrun in jpeg processing (gdi+) could allow code execution. http://www.microsoft.com/technet/security/bulletin/MS04-028.mspx (2004)
23. Tran-Nguyen, A.N.: E-Banking and e-payments: implications for developing and transition economies. In: UNCTAD Information Economy Report 2007-2008. United Nations Publication, New York and Geneva (2007) xxxvi, 348 p.
24. Menendez, P.: Estudio de comercio electronico 2008. http://www.amipci.org.mx/estudios.php (2008)
25. AdobeDevelopers: Tiff revision 6.0. http://partners.adobe.com/public /developer/en/tiff/TIFF6.pdf (1992)
26. Westfeld, A.: F5-a steganographic algorithm: High capacity despite better steganalysis. In: 4th International Workshop on Information Hiding, Springer-Verlag (2001) 289–302
27. Upham, D.: Jpeg-jsteg-v4. http://www.funet.fi/pub/crypt/steganography/jpeg-jsteg-v4.diff.gz (2008)
28. Miyazaki, S., Sakurai, K.: A more efficient untraceable e-cash system with partially blind signatures based on the discrete logarithm problem. In: FC '98: Proceedings of the Second International Conference on Financial Cryptography, London, UK, Springer-Verlag (1998) 296–308
29. Shi, L., Carbunar, B., Sion, R.: Conditional e-cash. In Dietrich, S., Dhamija, R., eds.: Financial Cryptography. Volume 4886 of Lecture Notes in Computer Science., Springer (2007) 15–28