(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 548 838**

(51) Int. CI.:

H04L 9/32 (2006.01)
H04L 9/30 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **09.05.2012 E 12722319 (6)**

(97) Fecha y número de publicación de la concesión europea: **01.07.2015 EP 2707990**

(54) Título: **Procedimiento para una firma digital múltiple**

(30) Prioridad:

13.05.2011 ES 201130777

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
21.10.2015

(73) Titular/es:

TELEFÓNICA S.A. (100.0%)
C/ Gran Vía 28
28013 Madrid, ES

(72) Inventor/es:

HERNÁNDEZ ENCINAS, LUIS;
MUÑOZ MASQUÉ, JAIME;
DURÁN DÍAZ, JOSÉ RAÚL;
HERNÁNDEZ ÁLVAREZ, FERNANDO y
GAYOSO MARTÍNEZ, VÍCTOR

(74) Agente/Representante:

ARIZTI ACHA, Monica

Procedimiento para una firma digital múltiple

DESCRIPCIÓN

Campo de la técnica

La presente invención se refiere, en general, a un procedimiento para una firma digital múltiple, basándose en la generación de una firma múltiple a partir de dichas firmas parciales y la verificación de las mismas, y más particularmente a un método que comprende usar una clave pública común para realizar dicha verificación.

Estado de la técnica anterior

En la actualidad hay diferentes métodos y algoritmos para realizar, de manera segura, firmas electrónicas o digitales por medio de redes informáticas. La mayor parte de estos protocolos se basan en criptografía de clave pública (PKC), (véase [MOV97]). La característica principal de este tipo de criptografía es que cada individuo tiene dos claves, una clave pública, denominada *e*, y una clave privada, denominada *d*. La clave pública permite a cualquier usuario cifrar los mensajes dirigidos al propietario de la clave, usando un procedimiento de cifrado, *E*. Por tanto, esta clave se conoce públicamente. Por otro lado, la clave privada solo la conoce su propietario y es la que permite descifrar los mensajes cifrados recibidos, a través de un procedimiento de descifrado, *D*.

En general, el único requisito para un criptosistema de clave pública es que el cifrado de un mensaje M, con la clave pública, *Ee*, seguido de su descifrado con la clave privada, *Dd*, debe dar como resultado el mensaje original, es decir,

$$Dd\,(Ee\,(M)) = Dd\,(c) = M.$$

Considerando el caso de las firmas digitales, los procedimientos de cifrado y descifrado, *E* y *D*, en los que se basan, deben verificar condiciones adicionales. Una de ellas es que el procedimiento de cifrado realizado con la clave privada, seguido del procedimiento de descifrado realizado con la respectiva clave pública deben tener como resultado el mensaje original, es decir, los procedimientos *E* y *D* deben verificar que:

$$De\,(Ed\,(M)) = De\,(c') = M.$$

Además, para que los procedimientos de firmas digitales y su transmisión electrónica sean más eficaces, se usan funciones *hash* (véase [FGHMM04], [MOV97], [NIST02]). Estas funciones se conocen públicamente y permiten firmar un resumen del documento original en lugar de todo el documento. En este documento estas funciones se indicarán por $H(\cdot)$.

El procedimiento general para realizar la firma digital de un documento, M, sigue las siguientes etapas (véase la figura 1):

1. Seleccionar el criptosistema de clave pública, (*E, D*), y la función *hash*, *H*, que van a usarse en el procedimiento.
2. Generar las claves pública y privada, *e*, *d*, del usuario que va a firmar el documento.
3. Calcular el resumen del mensaje que va a firmarse:

$$H\,(M) = m.$$

4. Firmar digitalmente el resumen del mensaje usando la clave privada del firmante:

$$Ed\,(m) = f.$$

5. Publicar el documento original y su correspondiente firma digital: (*M, f*).
6. Verificar y autenticar la firma del documento. Esta verificación se lleva a cabo usando el mensaje original, *M*, y la clave pública del emisor, *e*:

$$De\,(f) = De\,(Ed\,(m)) = m,$$

$$H(M) = m'$$

$$m' \,¿=?\, m.$$

El desarrollo y la simplicidad con respecto al uso de ordenadores y el acceso a Internet para los ciudadanos han conducido a la aparición de nuevas necesidades y requisitos que la tecnología debe satisfacer, de modo que ahora

debe ser posible llevar a cabo protocolos de firma digital que no se consideraban antes. Éste es el caso de las firmas múltiples.
Una firma múltiple es un protocolo de firma digital en el que un grupo de *t* firmantes,

$$G = \{F_1, F_2, \ldots, F_t\},$$

firma el mismo documento con la idea de que la firma digital del documento solo será válida si todos ellos participan en el protocolo (véase, por ejemplo, [Abo07], [AA07], [BN06], [Boy88], [HK89], [IN83], [KH90], [OO91], [Oka88], [PPKW97], [PLL85]).

La manera más sencilla de llevar a cabo una firma múltiple para un mensaje es considerar como tal firma la lista formada por todas las firmas parciales de cada uno de los firmantes. Sin embargo, esta firma no es práctica ya que su longitud es proporcional al número de firmantes.

En general, la mayoría de protocolos de firma múltiple basados en criptosistemas de clave pública se realizan de la siguiente manera:

1. El firmante F_1 firma un resumen del mensaje original, calculado a partir de una función *hash* que se conoce públicamente. Esta firma se realiza usando la clave privada del firmante y siguiendo el protocolo establecido por el criptosistema de clave pública que esté usándose.
2. A continuación, cada uno de los firmantes siguientes, de manera ordenada, firma el documento, ya firmado por el que le antecede en el grupo, siguiendo el mismo protocolo de firma ya establecido en la primera etapa.
3. Finalmente, el último miembro del grupo de firmantes, F_t, firma el correspondiente documento firmado que le ha enviado el firmante previo. Esta firma se determina usando su clave privada y, si es necesario, con la clave pública del verificador. Posteriormente, F_t envía al verificador no solo el mensaje sino también la firma múltiple calculada por el grupo de firmantes.

El procedimiento de verificación se realiza como sigue:

1. El verificador recibe el mensaje y la firma múltiple calculada por el grupo de firmantes.
2. El verificador realiza la verificación de la firma múltiple comprobando cada una de las firmas parciales del grupo de firmantes, siguiendo el protocolo y manteniendo el orden en el que se firmaron.

Hay diversas invenciones relativas a métodos de firma digital. Ninguna de ellas tiene una relación directa con el esquema de firma múltiple propuesto en esta invención.

Por ejemplo, varias patentes proponen métodos para elaborar una firma digital usando criptografía de clave pública, tal como RSA ([DHM05], [KOKS09]), curvas elípticas sobre campos finitos ([Shi01], [TK02]), y por medio de correlaciones bilineales ([Gen10]). Además, se presenta un método de firma de grupo basado en firmas anulares en [MFM04] y se presenta un esquema de firma digital usando firmas agregadas de identidad en [GR10].

La mayoría de las invenciones anteriores proponen métodos de firma digital individual y se basan en herramientas matemáticas distintas, pero todas ellas son diferentes de las herramientas usadas en esta invención.

Hay otros esquemas de firma propuestos para un grupo de firmantes. Por ejemplo, se presenta un sistema de firmado de múltiples etapas en [SFH01]. Este método usa múltiples dispositivos de firmado para asociar una firma individual que puede verificarse usando una única clave de verificación pública. En este caso, cada dispositivo de firmado tiene una cuota de la clave de firma y asocia una firma parcial en respuesta a la autorización de una pluralidad de agentes de autorización. Además, esta patente no da a conocer el uso de una clave privada para cada uno de los firmantes, es decir, no es exactamente un esquema de firma múltiple.

La invención presentada en [OO01] es un método que permite a un verificador efectuar una verificación en bloque de firmas individuales, múltiples o superpuestas asociadas electrónicamente por una pluralidad de firmantes a uno o más documentos.

Estos métodos, propuestos todos ellos para varios firmantes, difícilmente pueden considerarse esquemas de firma múltiple.

Por otro lado, la invención dada en [KOKS09] se basa en el criptosistema de clave pública RSA y es un verdadero método de firma múltiple mediante el cual una pluralidad de firmantes realizan de manera sucesiva un proceso de generación de firma de un documento dado para generar de ese modo una firma. En este caso, la firma se calcula usando el sistema RSA. Finalmente, en [FSNT09], la invención proporciona un sistema de verificación de firma múltiple sumando nuevos datos adicionales a los datos originales con una firma asociada a los mismos y verificando la validez de los datos originales y los datos adicionales.

El artículo [WCW96] propone un esquema de firma múltiple basado en la identificación de los firmantes. En este caso, cada firmante tiene su propio par de claves pública/privada, de las que la clave pública se obtiene a partir de información pública del firmante, relacionada en general con su identidad, tal como su dirección de correo electrónico, por ejemplo.

Otra propuesta para esquemas de firma múltiple se ha propuesto en [QX10]. En esta propuesta, la herramienta matemática depende de un mapa bilineal. La propuesta es bastante teórica ya que no se proponen implementaciones eficaces y prácticas de mapas bilineales.

La patente [ZWW10] da a conocer un método para proteger la seguridad de documentos de firma digital de múltiples verificadores designados firmemente por múltiples firmantes, donde cada firmante tiene su propia clave pública.

La patente [MS10] da a conocer un método para firmar digitalmente datos que incluye recoger un grupo de firmantes, teniendo cada uno una clave pública y una clave secreta correspondientes, un subgrupo de firmantes cada uno produciendo una firma digital parcial de los datos y obtener una firma combinada de los datos combinando las firmas parciales de los datos.

La patente [MS11] del mismo autor de [MS10] da a conocer un método particular para certificar datos que incluye tener un subgrupo de autoridades, contribuyendo cada uno una firma digital parcial de los datos para posibilitar el cálculo de una firma combinada, donde el subgrupo incluye algunos, pero no todos, del número total de autoridades que pueden aplicar una firma parcial a los datos, emitir un certificado para los datos y almacenar información para mantener al subgrupo de autoridades responsables de los datos que el subgrupo de autoridades contribuye a certificar.

Finalmente, el documento [CJ97] presenta un esquema de firma en grupo donde un gestor de grupo calcula dos pares de claves públicas-privadas, y publica las dos claves públicas como las claves públicas para el grupo.

Problemas con las soluciones existentes

Los protocolos genéricos descritos anteriormente presentan habitualmente algunas deficiencias, tales como:

1. Ya se ha demostrado que muchos de los esquemas propuestos no son seguros.
2. Implican un gran esfuerzo computacional.
3. En la mayoría de los esquemas de firma múltiple, el tamaño de la firma de un mensaje aumenta en la misma medida que el grupo de firmantes, lo que no es en absoluto deseable.
4. La necesidad de que todos los firmantes deban estar presentes simultáneamente para llevar a cabo la firma puede provocar un retraso en la obtención de la firma múltiple.
5. El hecho de que la firma múltiple tenga que llevarse a cabo en un orden determinado del grupo de usuarios obliga a verificar la firma de cada firmante en el orden inverso.
6. Dado un grupo de firmantes y una firma múltiple para un mensaje dado, el protocolo de firma múltiple tiene que realizarse una vez más por todos los miembros del grupo cada vez que un nuevo firmante se une al grupo.
7. En varios esquemas de firma múltiple, son necesarias algunas condiciones adicionales, más restrictivas sobre las claves del firmante. Éste es el caso para esquemas basados en RSA ([DHM05], [RSA78]). En estos esquemas, el módulo RSA de cada firmante debe aumentar a medida que aumenta el orden de cada firmante. De lo contrario, o bien la firma no puede realizarse o bien debe dividirse en bloques y entonces debe firmarse cada uno de estos bloques. El hecho de firmar varios bloques implica un mayor tamaño de la firma y más esfuerzo computacional.

Como ya se ha mencionado, la mayoría de los esquemas de firma múltiple se basan en criptosistemas de clave pública. Los dos criptosistemas más generalizados en la actualidad son aquellos cuya seguridad se basa o bien en factorización numérica (RSA [RSA78]), o bien en el problema de logaritmo discreto (ElGamal [ElG85], Elliptic curves [HMV04], [Men93]), y por tanto las características generales de los esquemas de firma múltiple se basan en estos sistemas. En esta invención, se usará un sistema ligeramente diferente, que representa una novedad y proporcionará algunas ventajas en comparación con los sistemas mencionados anteriormente.

El primer esquema de firma múltiple fue diseñado por Itakura y Nakamura ([IN83]). Desde entonces no se ha propuesto ninguna solución al problema de firmar un documento conjuntamente de manera eficaz. En esa propuesta, se realizó una modificación de RSA de tal manera que el módulo considerado era el producto de tres primos en lugar de solo dos. Además, cada uno de los firmantes, en orden, firma la firma del firmante que le precede en el grupo, de modo que el último miembro del grupo es quien realmente calcula la firma de todo el grupo ya que firma el resultado de la firma de todos los firmantes anteriores que él. Para verificar la firma múltiple, el verificador actúa verificando la firma de cada firmante del grupo en el orden correcto.

Okamoto ([Oka88]) propuso otro esquema de firma múltiple, basado también en RSA. En este esquema, la longitud de la firma es similar a la longitud obtenida a partir de un esquema de firma simple, y más corta que la firma obtenida a partir del esquema propuesto por Itakura y Nakamura. Además, esta propuesta puede usarse solo si el criptosistema es biyectivo; por ejemplo, en el caso de RSA, pero este hecho no se verifica en criptosistemas basados en el problema de logaritmo discreto.

El esquema de Harn y Kiesler ([HK89]) propone una modificación de RSA que permite a un grupo de usuarios firmar un documento y enviarlo a un receptor o verificador, que debe conocerse de antemano. En este caso, la longitud del texto cifrado es fija y no depende del número de firmantes del grupo. Sin embargo, los firmantes deben firmar el documento de manera consecutiva; por tanto es necesario aplicar varias transformaciones a las firmas parciales.

Más tarde, Kiesler y Harn ([KH90]) propusieron otras opciones para solucionar las limitaciones que los autores mencionan en la utilización del criptosistema RSA como un esquema de firma digital, es decir: bloquear la expansión de bits del mensaje, problema del tamaño de módulo para firmas múltiples, y el problema de módulo relacionado con la firma digital, y confidencialidad.

En [PPKW97] se proponen dos esquemas que mejoran los presentados en [Oka88] y [KH90]. El primero implica un aumento en el número de bits en la firma múltiple aunque este aumento no excede el número de firmantes. El segundo esquema no implica ningún aumento en el número de bits, pero se requiere que cada firmante tenga módulos RSA con el mismo número de bits y con el mismo patrón en los bits más significativos, lo que induce un defecto importante en su seguridad.

Para intentar resolver algunas de las desventajas anteriores, otra propuesta es el uso de rebloqueo ([PLL85]), pero esta opción todavía presenta el problema de mantener el orden de los firmantes.

En [AA07] se propone otro esquema más en el que cada firmante puede usar un módulo RSA de diferente tamaño. Esta opción produce un aumento en el número de bits, pero esta vez, el aumento se refiere al número de firmantes y no al tamaño del módulo usado.

Por lo que respecta a los esquemas de firma múltiple basados en el problema de logaritmo discreto, es importante mencionar el esquema propuesto por Laih y Yen ([LY96]). En este esquema, el grupo de firmantes debe cooperar para firmar el mensaje y enviarlo a un grupo dado de verificadores. Entonces, solo la unión de todos los verificadores puede validar la firma múltiple. Además, los firmantes no solo deben usar sus propias claves privadas, sino también la clave pública de todos los verificadores. En cualquier caso, el uso de este esquema no es recomendable ya que se han detectado ciertas debilidades ([He02], [Yen96]).

Hwang, Chen y Chang ([HCC98]) diseñaron otro esquema de firma múltiple para un grupo dado de verificadores, que proporciona autenticidad y confidencialidad; sin embargo, en este esquema, el mensaje solo puede recuperarse si todos los verificadores junto con su correspondiente firma múltiple se unen entre sí.

Otro nuevo esquema (véase [ZX04]) permite realizar una firma múltiple si los verificadores de la firma pertenecen a un grupo previamente especificado. No obstante, para este esquema se han encontrado también ciertas debilidades ([LWK05], [YY05]).

En [MS10] cada firmante en el grupo tiene una clave pública y una clave secreta correspondiente y existe una clave pública combinada (CPK = (PK1, ..., PKn)), como resultado, si un nuevo participante se une al grupo obligará un cambio en CPK.

En todos los artículos y patentes anteriores, cada firmante tiene un par de claves pública/privada, lo que no sucede en esta propuesta. Como se mencionó anteriormente, este esquema de firma múltiple tiene la propiedad y la ventaja de que cada firmante tiene su propia clave privada, pero todos ellos comparten la misma clave pública. Este hecho simplifica y evita en gran medida los problemas mencionados anteriormente en relación con el esfuerzo computacional para el cálculo, ancho de banda y, por tanto, la eficacia global del protocolo propuesto.

Finalmente, un artículo de gran interés es el publicado por Bellare y Neven ([BN06]), porque los autores presentan un esquema de firma múltiple general para un modelo genérico de clave pública. Desde su publicación, esta propuesta se ha considerado una referencia esencial para los esquemas de firma múltiple. No obstante, en este modelo se han considerado obligatorios algunos requisitos, tales como los siguientes:

- Cada uno de los firmantes debe tener una clave pública certificada, con su correspondiente clave privada, que debe generarla el propio firmante.
- Los firmantes deben interaccionar en un número dado de ciclos. En cada ciclo, cada firmante recibe un mensaje, realiza varios cálculos y envía otro mensaje al siguiente firmante.
- Debe ser computacionalmente inviable falsificar una firma múltiple si existe un firmante honesto.

Hasta la fecha no se ha propuesto ningún esquema ni patente eficaz, que pueda solucionar todos los problemas mencionados anteriormente, es decir, no se ha diseñado ningún esquema con los siguientes requisitos:

1. Seguridad.
2. Eficacia.
3. Independencia del tamaño de la firma múltiple con respecto al número de firmantes.
4. Posibilidad de firma fuera de línea, es decir, no todos los firmantes deben estar en línea simultáneamente.
5. Firma del mensaje por todos los firmantes en cualquier orden.
6. Simplicidad en el procedimiento para añadir nuevos firmantes.
7. Simplicidad en el procedimiento verificación de la firma múltiple (que no sea necesario verificar la firma parcial de cada miembro del grupo de firmantes).

Descripción de la invención

Es necesario ofrecer una alternativa al estado de la técnica que supere los problemas mencionados anteriormente que sufren las soluciones existentes.

Con este fin, la presente invención proporciona un procedimiento para una firma digital múltiple, que comprende:

i) generar, por una tercera parte de confianza, un conjunto de parámetros, su propia clave privada y una clave privada para cada firmante o miembro de un grupo de firmantes;
ii) generar, por cada uno de dichos firmantes, una firma parcial en un resumen de un mensaje, o documento, usando sus claves privadas;
iii) generar una firma múltiple a partir de dichas firmas parciales; y
iv) verificar, por un verificador, dicha firma múltiple.

A diferencia de las propuestas conocidas, el procedimiento de la invención comprende:

- determinar, por dicha tercera parte de confianza, una clave pública única y común para todos de dichos firmantes en el grupo de firmantes, calculando dos números enteros, en $\mathbf{Z}_n$,

$$P = \alpha^{a_0} \cdot \beta^{b_0} \pmod{n},$$
$$Q = \alpha^{c_0} \cdot \beta^{d_0} \pmod{n};$$

y

- determinar, por dicha tercera parte de confianza, claves privadas individuales de los firmantes del grupo de firmantes, asociadas a dicha determinada clave pública única y común, calculando (a_i, b_i, c_i, d_i), para $i = 1, ..., t$, en el que:

  (a_0, b_0, c_0, d_0) son cuatro números enteros aleatorios que pertenecen a $\mathbf{Z}_r$ que definen la clave privada de la tercera parte de confianza (T);
  (b_i, d_i), para $i = 1, ..., t$, son t pares de números enteros aleatorios en $\mathbf{Z}_r$, y (a_i, c_i), para $i = 1, ..., t$, son t pares de números enteros en $\mathbf{Z}_r$ que verifican las siguientes condiciones:

$$a_i = (h - s \cdot b_i) \pmod{r},$$

$$c_i = (k - s \cdot d_i) \pmod{r};$$

  y h y k son dos números enteros secretos, en $\mathbf{Z}_r$, definidos por

$$h = (a_0 + s \cdot b_0) \pmod{r},$$

$$k = (c_0 + s \cdot d_0) \pmod{r},$$

y

- generar, por dicha tercera parte de confianza (T), un conjunto de parámetros (n, r, α, β, p, q, s) de modo que publica n, r, α y β y mantiene p, q y s secretas, donde

$$n = p \cdot q,$$

$p = u_1 \cdot r \cdot p_1 + 1$ y $q = u_2 \cdot r \cdot q_1 + 1$ son dos números primos grandes,

u_1 y u_2 son dos números enteros pares, cuyo máximo común divisor (mcd) verifica

$$\text{mcd}(u_1, u_2) = 2,$$

p_1, q_1, r, son números primos,
α es un elemento reversible en el grupo de los enteros módulo n, $\boldsymbol{Z}_n$, con orden multiplicativo r, que verifica la condición

$$\text{mcd}(\alpha, (p - 1)(q - 1)) = 1;$$

$\beta = \alpha^s \pmod{n}$ y
s es un número secreto aleatorio en el subgrupo generado por α.

Por tanto, este procedimiento permite generar las claves de una tercera parte de confianza, que generará la clave pública común y las claves privadas de un determinado grupo de usuarios. Estos usuarios firmarán conjuntamente un documento de tal manera que las firmas de todos los miembros del grupo constituirán la firma múltiple de tal grupo para el documento dado.

Una vez realizada la firma múltiple, cualquiera que conozca la clave pública común del grupo de usuarios podrá verificar la autenticación de la firma múltiple o bien declararla inválida.

La presente invención se basa en un nuevo esquema de firma múltiple, que no se ha propuesto ni publicado antes, que cumple todos los requisitos mencionados anteriormente. Por tanto, garantiza que todos los problemas descritos en el apartado anterior se solucionan de manera precisa.

Cabe destacar que este nuevo esquema no coincide con el modelo propuesto en [BN06] ya que el procedimiento se lleva a cabo en un único ciclo en el que participan todos los firmantes. Además, no todos los firmantes tienen que tener su propio par de claves certificadas (pública y privada). De hecho, en el protocolo propuesto en esta invención todos los firmantes comparten la misma clave pública, pero cada uno tiene una clave privada diferente, que se mantiene en secreto y que solo conoce él mismo y el centro generador de claves o tercera parte de confianza.

En particular, se proponen dos protocolos diferentes para obtener una firma múltiple para un documento. El primero, que requiere una tercera parte de confianza no solo para generar las claves sino también para realizar la firma múltiple, es más eficaz que el segundo, que no necesita una tercera parte de confianza para realizar la firma.

Como se mencionó en la sección de estado de la técnica anterior, el esquema general de los dispositivos que llevan a cabo el procedimiento de firma múltiple considera las siguientes etapas (véase la figura 2):

1. Generar las claves por una tercera parte de confianza.
2. Verificar las claves del firmante.
3. Generar la firma múltiple por medio de la firma parcial de cada miembro del grupo de firmantes.
4. Verificar la firma múltiple.

En esta invención, se presenta un procedimiento para llevar a cabo los procesos de generación de claves, generación de la firma múltiple y verificación de la firma múltiple.

Esta invención no solo garantiza la generación de una firma digital múltiple de un documento por un grupo de firmantes, sino que también mejora sustancialmente otros protocolos publicados anteriormente de muchas formas diferentes. Por ejemplo, es más fácil de usar, es más eficaz computacionalmente y permite ahorrar tiempo y ancho de banda.

Otras realizaciones del procedimiento de la invención, para algunas de las cuales se requiere la colaboración de la tercera parte de confianza mientras que para otras no, se describen según las reivindicaciones 2 a 9 adjuntas, y en una sección posterior relativa a la descripción detallada de varias realizaciones.

Breve descripción de los dibujos

Las anteriores y otras ventajas y características se entenderán más completamente a partir de la siguiente descripción detallada de realizaciones, con referencia a los dibujos adjuntos, que deben considerarse de manera ilustrativa y no limitativa, en los que:

La figura 1 muestra un esquema general de un procedimiento de firma digital representativo del protocolo de un procedimiento de firma digital convencional;
La figura 2 es un diagrama de flujo que ilustra un procedimiento convencional usado para realizar una firma múltiple;

La figura 3 ilustra, por medio de un diagrama de flujo, una realización del procedimiento de la invención que usa una tercera parte de confianza para realizar la firma múltiple;
La figura 4 es un diagrama de flujo que ilustra una realización del procedimiento de la invención para el que no requiere la colaboración de una tercera parte de confianza para realizar la firma múltiple; y
La figura 5 muestra una arquitectura que implementa el procedimiento de la invención para una realización.

Descripción detallada de varias realizaciones

Siendo $\{F_1, F_2, \ldots, F_N\}$ un grupo de N usuarios, una firma digital múltiple es un procedimiento mediante el cual cada subgrupo de t miembros del mismo,

$$G = \{F_1, F_2, \ldots, F_t\},$$

puede realizar, en cualquier momento, a través de un protocolo especificado, una firma común de un documento o de un mensaje previamente fijado. Además, una firma de este tipo puede verificarse y validarse por cualquier otro usuario. Para llevar a cabo la verificación es necesario conocer el número de usuarios, t, el documento original o un resumen de tal mensaje generado mediante una función *hash*, m, la firma múltiple, y la(s) clave(s) pública(s) usada(s) en el protocolo.

Tal como se mencionó anteriormente, estos protocolos de firma digital necesitan un centro de claves o una tercera parte de confianza, T, que genere tanto sus propias claves como las claves de cada uno de los firmantes.

En esta propuesta, se describirán dos protocolos para realizar la firma. Los dos protocolos proporcionan la misma firma múltiple cada vez que se usan las mismas entradas del protocolo. Por tanto, ambos son esencialmente el mismo, aunque presentan algunas diferencias con respecto a las acciones que llevan a cabo tanto la tercera parte de confianza como los usuarios que participan en el protocolo.

Por otro lado, la tercera parte de confianza, T, puede formar parte del grupo de firmantes o no, lo que no afecta ni al procedimiento de firma ni a la propia firma múltiple.

Las fases en las que se divide este protocolo son las siguientes (véanse las figuras 3-4):

1. Generación de claves.
2. Verificación de claves.
3. Generación de la firma digital múltiple.

    a) Con la colaboración de T.
    b) Sin la colaboración de T.

4. Verificación de la firma digital múltiple.

A continuación se describe más detalladamente cada una de las fases:

1. Generación de claves

En primer lugar, deben generarse la clave de T y después las claves de los firmantes. Las etapas para generar la clave de T son las siguientes:

1. T elige dos números primos grandes p y q, que deben cumplir las siguientes condiciones:

$$p = u_1 \cdot r \cdot p_1 + 1,$$

$$q = u_2 \cdot r \cdot q_1 + 1,$$

donde r, p_1, q_1 son números primos y u_1, u_2 número enteros pares, cuyo máximo común divisor (mcd) es

$$\text{mcd}(u_1, u_2) = 2,$$

es decir, $u_1 = 2 \cdot v_1$ y $u_2 = 2 \cdot v_2$.

2. T calcula

$$n = p \cdot q,$$

$$\varphi(n) = (p-1)(q-1) = u_1 \cdot u_2 \cdot r^2 \cdot p_1 \cdot q_1,$$

$$\lambda(n) = \text{mcm}(p-1, q-1) = 2\, v_1 \cdot v_2 \cdot r \cdot p_1 \cdot q_1,$$

donde mcm representa el mínimo común múltiplo, φ(*n*) es la función de Euler y λ(*n*) es la función de Carmichael. Para garantizar la seguridad del protocolo, el tamaño de *r*, es decir, su longitud de bits, debe ser suficientemente grande para hacer computacionalmente irresoluble el problema de logaritmo discreto en subgrupo (SDLP), con orden r de los enteros módulo *n*, Z^*_n.

3. A continuación, *T* elige un número entero $\alpha \in Z^*_n$ con orden multiplicativo *r*, módulo *n* y que cumpla la condición

$$\text{mcd}(\alpha, \varphi(n)) = \text{mcd}(\alpha, u_1 \cdot u_2 \cdot r^2 \cdot p_1 \cdot q_1) = 1.$$

Siendo S_r el subgrupo de Z^*_n generado por α. La obtención del generador α puede llevarse a cabo de manera eficaz, es decir, en un tiempo polinomial, simplemente siguiendo el lema 3.1 de [Sus09].

En realidad, según este lema, la primera etapa es determinar un elemento $g \in Z^*_n$ cuyo orden es λ(*n*). El procedimiento consiste en elegir un elemento $g \in Z^*_n$ y en verificar que *g* elevado a todos los posibles divisores de λ(*n*), módulo *n*, es diferente de 1 en todos los casos. Este procedimiento es rápido ya que la factorización de λ(*n*) se conoce y solo tiene unos pocos factores primos, de modo que la lista de sus divisores puede calcularse fácilmente. En caso de que el elemento elegido aleatoriamente no satisfaga esta condición, debe elegirse otro y repetirse el procedimiento.
Una vez determinado el elemento *g* con orden λ(*n*), la siguiente etapa es calcular el elemento

$$\alpha = g^{2v1.v2.p1.q1} (\text{mod } n)$$

4. *T* genera un número secreto aleatorio *s* en S_r y calcula

$$\beta = \alpha^s (\text{mod } n).$$

5. Los valores (α, *r*, β, *n*) se hacen públicos, mientras que *T* mantiene los valores (*p, q, s*) en secreto.

Aunque el factor *r* de *p*–1 y *q*–1 se conoce y *n* es el producto de dos primos, *p* y *q*, actualmente no hay ningún algoritmo eficaz que pueda calcular los dos factores de *n*.

Las etapas seguidas por *T* para generar las claves de los firmantes son:

1. *T* determina su clave privada generando aleatoriamente cuatro números enteros

$$a_0, b_0, c_0, d_0 \in Z_r.$$

2. *T* obtiene la clave pública común para todos los firmantes calculando

$$P = \alpha^{a_0} \cdot \beta^{b_0} (\text{mod } n),$$

$$Q = \alpha^{c_0} \cdot \beta^{d_0} (\text{mod } n),$$

3. A partir de las expresiones anteriores, *T* determina

$$P = \alpha^{a_0} \cdot \beta^{b_0} (\text{mod } n) = \alpha^{a_0 + s \cdot b_0} (\text{mod } n),$$

$$Q = \alpha^{c_0} \cdot \beta^{d_0} (\text{mod } n) = \alpha^{c_0 + s \cdot d_0} (\text{mod } n),$$

Por tanto, tanto *P* como *Q* son elementos del subgrupo S_r; es decir, existen números enteros *h*, $k \in Z_r$, que satisfagan

$$h = (a_0 + s \cdot b_0) (\text{mod } r),$$

$$k = (c_0 + s \cdot d_0) (\text{mod } r).$$

4. *T* determina la clave privada para cada firmante $F_i \in G$, con *i* = 1, …, *t*, generando cuatro número enteros para

cada uno de ellos,

$$a_i, b_i, c_i, d_i \in Z_r ,$$

de modo que los firmantes compartirán la misma clave pública (*P, Q*). Por tanto, para los valores de $i = 1, \ldots, t$, deben cumplirse las siguientes dos condiciones Z_r,

$$h = (a_i + s \cdot b_i) \pmod r,$$

$$k = (c_i + s \cdot d_i) \pmod r,$$

o de manera equivalente

$$a_i = (h - s \cdot b_i) \pmod r, \qquad (1)$$

$$c_i = (k - s \cdot d_i) \pmod r. \qquad (2)$$

Por tanto, puesto que *T* conoce los valores *s, h* y *k*, determina *t* claves privadas para los firmantes F_i simplemente generando *t* pares de números aleatorios $b_i, d_i \in Z_r$ y entonces, calculando los valores correspondientes $a_i, c_i \in Z_r$ según las ecuaciones anteriores (1) y (2).

Una vez que *T* ha calculado las claves privadas, las distribuye de manera segura a los firmantes.

2. Verificación de claves

Para verificar que la clave de *T* es correcta, cada firmante, $F_i \in G$, $i = 1, \ldots, t$, solo tiene que comprobar si:

$$\alpha \neq 1 \pmod n,$$

$$\alpha^r = 1 \pmod n.$$

Además, cada firmante debe comprobar si la clave pública conocida corresponde a su clave privada. Para ello, cada firmante solo tiene que verificar si se cumplen estas dos ecuaciones siguientes

$$P = \alpha^{a_i} \cdot \beta^{b_i} \pmod n,$$

$$Q = \alpha^{c_i} \cdot \beta^{d_i} \pmod n.$$

Para realizar la firma digital múltiple, se consideran dos escenarios, dependiendo de si la tercera parte de confianza *T* colabora con el grupo de firmantes o no.

El primer caso, concretamente, cuando la firma múltiple se realiza con la colaboración de *T*, es más rápido, más eficaz y seguro que el segundo.

3a. Firma digital múltiple realizada con la colaboración de *T*

En este protocolo (véase la figura 3), cada uno de los firmantes realiza su firma particular para el resumen del mensaje, *m*, y la envía, de manera segura, a *T*, que se encarga de realizar la firma múltiple. Para ello, en primer lugar debe verificar la validez de todas las firmas recibidas y después sumarlas todas, módulo *r*.

1. Cada firmante $F_i \in G$, $i = 1, \ldots, t$, calcula su firma como sigue:

$$f_i = a_i + c_i \cdot m \pmod r,$$

$$g_i = b_i + d_i \cdot m \pmod r.$$

2. El firmante F_i envía, de manera segura, su firma a *T*.
3. *T* verifica la validez de cada una de las firmas recibidas comprobando si

$$P \cdot Q^m \pmod n = \alpha^{f_i} \cdot \beta^{g_i} \pmod n, \; i = 1, \ldots, t .$$

4. Una vez verificadas todas las firmas, *T* calcula la firma múltiple para el documento *m*, (*f, g*), simplemente sumando todas las firmas parciales:

$$f = \Sigma_{i=1, \ldots, t} f_i \pmod{r},$$

$$g = \Sigma_{i=1, \ldots, t} g_i \pmod{r}.$$

5. Finalmente, *T* publica (*f, g*) como la firma múltiple de *G* para *m*.

3b. Firma digital múltiple realizada sin la colaboración de *T*

Es posible realizar la firma múltiple de *m* sin la colaboración de *T*, pero en este caso el procedimiento no es tan directo como el anterior (véase la figura 4). En este procedimiento cada firmante firma la firma del anterior, de modo que es necesario difundir las firmas parciales entre el grupo de firmantes y establecer un orden en el grupo de firmantes para evitar posibles ataques por un firmante o una conspiración de varios firmantes.

Cuando *T* colaboraba para realizar la firma múltiple, las firmas parciales podían realizarse fuera de línea, es decir, cada firmante podía realizar su firma en cualquier momento sin necesidad de que todos ellos estén conectados al mismo tiempo. Para el presente caso, todos los firmantes deben estar conectados simultáneamente y la firma se realiza en un solo acto. No hay necesidad de esperar al cálculo de la firma de cada firmante.

Sea $G = \{F_1, \ldots, F_t\}$ el grupo de firmantes en un orden fijo arbitrario.

En este caso, en lugar de ser la tercera parte de confianza la que verifica la firma de todos los firmantes, cada firmante verifica por sí mismo la firma del firmante previo. Después realiza su propia firma y la suma a la firma recibida. Más precisamente, el procedimiento es el siguiente:

1) El primer firmante, F_1, realiza su firma para el resumen de un mensaje dado, *m*, calculando

$$f_1 = a_1 + c_1 \cdot m \pmod{r},$$

$$g_1 = b_1 + d_1 \cdot m \pmod{r}$$

y envía (f_1, g_1) al grupo de firmantes.

2) El segundo firmante, F_2:

a) Verifica la firma de F_1 comprobando si

$$P \cdot Q^m = \alpha^{f_1} \cdot \beta^{g_1} \pmod{n}.$$

b) Calcula la firma parcial acumulada para el mensaje, calculando

$$f_2 = f_1 + a_2 + c_2 \cdot m \pmod{r} = a_1 + a_2 + (c_1 + c_2)\, m \pmod{r},$$

$$g_2 = g_1 + b_2 + d_2 \cdot m \pmod{r} = b_1 + b_2 + (d_1 + d_2)\, m \pmod{r}.$$

c) Envía (f_2, g_2), como la firma parcial acumulada al grupo de firmantes.

3) El firmante F_3 recibe la firma parcial (f_2, g_2) y

a) Verifica la firma parcial de F_2 comprobando si

$$P^2 \cdot Q^{2m} = \alpha^{f_2} \cdot \beta^{g_2} \pmod{n}.$$

b) Calcula su propia firma parcial acumulada para el mensaje, calculando

$$f_3 = f_2 + a_3 + c_3 \cdot m \pmod{r} = a_1 + a_2 + a_3 + (c_1 + c_2 + c_3)\, m \pmod{r},$$

$$g_3 = g_2 + b_3 + d_3 \cdot m \pmod{r} = b_1 + b_2 + b_3 + (d_1 + d_2 + d_3)\, m \pmod{r}.$$

c) Envía (f_3, g_3) al grupo de firmantes como su firma parcial.

i) El firmante F_i usa la firma parcial de F_{i-1}, (f_{i-1}, g_{i-1}), y

a) La verifica, comprobando

$$P^{i-1} \cdot Q^{(i-1)m} = \alpha^{f_{i-1}} \cdot \beta^{g_{i-1}} \pmod{n}.$$

b) Calcula su firma parcial acumulada para el mensaje, calculando

$$f_i = f_{i-1} + a_i + c_i \cdot m \pmod{r} = a_1 + \ldots + a_i + (c_1 + \ldots + c_i)\, m \pmod{r},$$

$$g_i = g_{i-1} + b_i + d_i \cdot m \pmod{r} = b_1 + \ldots + b_i + (d_1 + \ldots + d_i)\, m \pmod{r}.$$

c) Envía (f_i, g_i) al grupo de firmantes.

t) El último firmante, F_t

a) Verifica la firma parcial de F_{t-1} comprobando si

$$P^{t-1} \cdot Q^{(t-1)m} = \alpha^{f_{t-1}} \cdot \beta^{g_{t-1}} \pmod{n}.$$

b) Calcula su firma parcial acumulada para el mensaje, calculando $f_t = f_{t-1} + a_t + c_t \cdot m \pmod{r} = a_1 + \ldots + a_t + (c_1 + \ldots + c_t)\, m \pmod{r}$, $g_t = g_{t-1} + b_t + d_t \cdot m \pmod{r} = b_1 + \ldots + b_t + (d_1 + \ldots + d_t)\, m \pmod{r}$.

c) Publica la firma múltiple para m: $(f, g) = (f_t, g_t)$.

La condición que debe satisfacer la firma parcial del firmante F_{i-1} llevada a cabo por el firmante F_i se cumple puesto que:

$$\alpha^{f_{i-1}} \cdot \beta^{g_{i-1}} \pmod{n} = \alpha^{a_1+\ldots+a_{i-1}+(c_1+\ldots+c_{i-1})m} \cdot \beta^{b_1+\ldots+b_{i-1}+(d_1+\ldots+d_{i-1})m} \pmod{n}$$

$$= \alpha^{a_1+\ldots+a_{i-1}} (\alpha^{c_1+\ldots+c_{i-1}})^m \cdot \beta^{b_1+\ldots+b_{i-1}} \left(\beta^{d_1+\ldots+d_{i-1}}\right)^m \pmod{n}$$

$$= \prod_{j=1}^{i-1} \alpha^{a_j} \cdot \beta^{b_j} \left(\alpha^{c_j} \cdot \beta^{d_j}\right)^m \pmod{n}$$

$$= \prod_{j=1}^{i-1} P \cdot Q^m \pmod{n}$$

$$= P^{i-1} \cdot Q^{(i-1)m} \pmod{n}.$$

La verificación de todas las firmas parciales acumuladas realizada por cada uno de los firmantes (salvo por el primero) es obligatoria: ningún firmante debe firmar un mensaje sin haber comprobado la validez de la firma realizada hasta ese momento. Haciendo esto y dado que todos los firmantes son honestos, en la mayoría de los casos, se evitan verificaciones adicionales. En realidad, cada firmante puede actuar como el verificador y puede verificar la firma múltiple usando la firma múltiple y la clave pública común puesto que son conocidas por todos los firmantes.

Si o bien la firma múltiple final o bien cualquiera de las firmas parciales acumuladas no ha satisfecho una o más pruebas de verificación, se sabrá que, al menos, uno de los firmantes ha falsificado su firma o que ha tenido lugar algún tipo de conspiración entre varios firmantes. El motivo por el que todas las firmas parciales acumuladas deben difundirse a todo el grupo es para conocer exactamente dónde ha sucedido la falsificación. En ese momento, el problema será decidir qué firmante es el falsificador. Puesto que todo el grupo conoce las firmas parciales, la única tarea que hay que realizar es restar pares de firmas consecutivas y verificar la firma del firmante correspondiente.

La difusión de las firmas parciales dentro del grupo de firmantes compensa mutuamente las ventajas y desventajas, ya que, por ejemplo, cualquier firma parcial acumulada se verifica solo por un firmante, no por todos. Además, la firma del último firmante no se verifica por ninguno de los firmantes restantes y el primer firmante envía su firma

directamente, sin que haya sido sumada a otra.

Es importante señalar que las firmas múltiples obtenidas a partir de los dos procedimientos explicados, con o sin la colaboración de *T*, coinciden. Por tanto, la verificación de la firma múltiple del primer procedimiento se aplica también al segundo.

4. Verificación de la firma múltiple

Sea (*f, g*) una firma digital múltiple para un resumen de un mensaje, *m*, realizada por el siguiente grupo de *t* usuarios:

$$G = \{F_1, \ldots, F_t\}.$$

Para verificar la validez de tal firma el procedimiento es el siguiente, independientemente de si la firma se realizó con o sin la colaboración de la tercera parte de confianza.

La firma múltiple del grupo *G*, con *t* miembros, para *m* es válida si se cumple la siguiente condición

$$P^t \cdot Q^{t \cdot m} = \alpha^f \cdot \beta^g \pmod{n}.$$

En realidad, es suficiente con tener en cuenta que

$$\alpha^f \cdot \beta^g \pmod{n} = \alpha^{a_1+\ldots+a_t+(c_1+\ldots+c_t)m} \cdot \beta^{b_1+\ldots+b_t+(d_1+\ldots+d_t)m} \pmod{n}$$

$$= \alpha^{a_1+\ldots+a_t}\left(\alpha^{c_1+\ldots+c_t}\right)^m \cdot \beta^{b_1+\ldots+b_t}\left(\beta^{d_1+\ldots+d_t}\right)^m \pmod{n}$$

$$= \prod_{j=1}^{t} \alpha^{a_j} \cdot \beta^{b_j}\left(\alpha^{c_j} \cdot \beta^{d_j}\right)^m \pmod{n}$$

$$= \prod_{j=1}^{t} P \cdot Q^m \pmod{n}$$

$$= P^t \cdot Q^{t\,m} \pmod{n}.$$

El diseño global para la arquitectura del procedimiento propuesto en esta invención se muestra en la figura 5.

Ventajas de la invención:

Puede verse que tanto el procedimiento de verificación de la firma múltiple como las verificaciones de las firmas parciales son similares al procedimiento usado en el esquema clásico de ElGamal ([ElG85], [FGHMM04]). Este hecho aumenta la confianza en la seguridad de esta propuesta. No obstante, esta similitud no es una equivalencia, ya que el esquema de esta invención es nuevo y no sigue los mismos patrones que otros esquemas ya publicados.

De hecho, en esta invención, la firma se verifica comprobando si se cumple la siguiente ecuación:

$$\alpha^f \cdot \beta^g \; (= \alpha^f \cdot (\alpha^s)^g \;) \; ¿ = ? \; P \cdot Q^m \pmod{n}.$$

Sin embargo, la verificación de firma en el esquema de ElGamal usa la siguiente ecuación

$$(\alpha^a)^r \cdot (\alpha^h)^s \; ¿ = ? \; \alpha^m \pmod{p}.$$

En ambas expresiones, todos los parámetros los conoce el verificador de la firma; y *m*, en ambas expresiones, es el resumen del mensaje cuya firma debe verificarse.

La similitud de ambas verificaciones reside en el hecho de que en ambas expresiones el objetivo es comprobar una igualdad usando el resumen del mensaje como el exponente de un parámetro público.

Ventajas adicionales del presente esquema son:

1. La longitud de la firma es siempre la misma, independientemente del tamaño del mensaje o del número de firmantes. Además, la firma es corta, ya que el par (*f, g*) son dos elementos del subgrupo S_r *de* Z^*_n.
2. El tiempo de cálculo de todas las operaciones es polinomial.
3. Si la tercera parte de confianza participa en el protocolo, cada firmante calcula su firma por su cuenta; no se espera que los firmantes estén en línea al mismo tiempo. Una vez que *T* ha recibido todas las sumas parciales, determinará la firma múltiple.
4. Además, cuando *T* colabora en el protocolo, no hay obstáculo alguno para añadir nuevos firmantes al grupo original de firmantes en cualquier momento sin tener que realizar todo el protocolo una vez más. La única tarea que los recién llegados tienen que hacer es calcular sus correspondientes firmas y enviarlas a *T*. Después, *T* verificará estas firmas y calculará la nueva firma múltiple sumando las nuevas firmas a la firma múltiple original.
5. Si el protocolo se realiza sin la colaboración de *T*, los firmantes deben estar en línea simultáneamente, de modo que la firma múltiple pueda realizarse en un solo acto. De esta manera no es necesario esperar a que todos los firmantes calculen, en un orden establecido, su firma parcial acumulada.
6. También es posible añadir nuevos firmantes al procedimiento cuando *T* no colabora sin tener que realizar todo el protocolo una vez más. En este caso los nuevos firmantes solo tienen que ordenarse aleatoriamente después del último firmante del grupo original y calcular sus firmas acumuladas usando la última firma conocida.
7. La verificación de la firma múltiple en ambos protocolos es muy simple y eficaz, ya que solo es necesario un cálculo. Este cálculo permite la validación de todas las firmas parciales de todos los miembros del grupo.

Seguridad de la firma múltiple

En caso de usar el protocolo en el que *T* colabora, no existe la posibilidad de que dos firmantes conspiren para generar una firma falsa ya que la firma múltiple la determina la tercera parte de confianza y ésta verifica en el procedimiento que cada una de las firmas parciales corresponde a cada uno de los firmantes.

Si el protocolo se lleva a cabo sin la colaboración de *T*, una conspiración entre dos o más firmantes no es posible porque todos los firmantes participan en el protocolo y cada uno verifica la firma del firmante previo. Además, si cualquiera de los firmantes tuviese alguna sospecha de posibles conspiraciones o firmas falsificadas, podría verificar todas y cada una de las firmas parciales porque todas las firmas parciales acumuladas se difunden a todos los miembros del grupo. De esta manera, pueden detectarse fácilmente tanto la falsificación como al culpable.

Por otro lado, ningún firmante puede determinar el valor secreto *s* elegido por *T* conociendo solo su clave privada y la clave pública común. Debe señalarse que la determinación de *s* a partir de los parámetros α y $\beta = \alpha^s \pmod n$ implica el cálculo de logaritmos discretos en el subgrupo, S_r, de orden *r*, generado por α. Este hecho es imposible ya que *r* se eligió de tal manera que este problema fuese irresoluble en S_r.

Si dos, o más, firmantes, por ejemplo, F_i *y* F_j, conspiran para obtener el valor secreto *s* de *T*, pueden determinar sus correspondientes firmas parciales para un mensaje dado, (f_i, g_i) y (f_j, g_j), respectivamente. Puesto que se cumple la siguiente ecuación

$$P \cdot Q^m \pmod n = \alpha^{f_i} \cdot \beta^{g_i} \pmod n = \alpha^{f_j} \cdot \beta^{g_j} \pmod n, \qquad (3)$$

pueden realizar el siguiente cálculo:

$$\alpha^{f_i} \cdot \beta^{g_i} \pmod n = \alpha^{f_j} \cdot \beta^{g_j} \pmod n,$$

$$\alpha^{f_i + s \cdot g_i} \pmod n = \alpha^{f_j + s \cdot g_j} \pmod n,$$

Sin embargo, puesto que α es un elemento de orden *r*, se cumple la siguiente expresión:

$$f_i - f_j = s\,(g_j - g_i) \pmod r, \qquad (4)$$

es decir,

$$s' = s \pmod r = (f_i - f_j)\,(g_j - g_i)^{-1} \pmod r. \qquad (5)$$

No obstante, después de estos cálculos, los firmantes solo obtendrán el valor de $s' = s \pmod r$ y no el valor real de *s*. Por tanto, este tipo de ataque no afecta a la seguridad del protocolo.

Usando la misma hipótesis, si varios firmantes quieren forzar el sistema y calcular *s* firmando diferentes mensajes, elegidos por ellos mismos o aleatoriamente, intentando obtener más información sobre el protocolo y forzarlo, no

obtendrán más información que la obtenida al firmar un único mensaje.

En realidad, supóngase que dos firmantes, F_i y F_j, eligen dos mensajes diferentes, por ejemplo, m_1 y m_2, y calculan sus correspondientes firmas para cada uno de ellos. Sean (f_i, g_i) y (h_i, k_i) las firmas de F_i para tales mensajes, y sean (f_j, g_j) y (h_j, k_j) las correspondientes firmas de F_j.

En este caso, usando la expresión (3) para cada mensaje, se cumplen las siguientes ecuaciones:

$$P \cdot Q^{m_1} \pmod n = \alpha^{f_i} \cdot \beta^{g_i} \pmod n = \alpha^{f_j} \cdot \beta^{g_j} \pmod n,$$

$$P \cdot Q^{m_2} \pmod n = \alpha^{h_i} \cdot \beta^{k_i} \pmod n = \alpha^{h_j} \cdot \beta^{k_j} \pmod n,$$

donde, tal como sucede en (4), puede obtenerse:

$$f_i - f_j = s\,(g_j - g_i) \pmod r,$$

$$h_i - h_j = s\,(k_j - k_i) \pmod r,$$

por tanto, tal como sucede en (3), el resultado es

$$s' = s \pmod r = (f_i - f_j)\,(g_j - g_i)^{-1} \pmod r = (h_i - h_j)\,(k_j - k_i)^{-1} \pmod r$$

y el valor obtenido para s' es el mismo que el obtenido cuando se usó solo un mensaje.

Aplicaciones de la invención:

La invención puede aplicarse de manera satisfactoria cuando las firmas digitales son obligatorias pero varias entidades o personas deben firmar conjuntamente el mismo documento, mensaje, contrato, etc. Entre otras aplicaciones, pueden mencionarse las siguientes:

- Cualquier proceso que requiera una firma digital en el que esté implicado más de un firmante.
- Firmas digitales en escenarios corporativos para firmar contratos entre empresas o entre una empresa y un cliente.
- Firmas digitales entre empresas o usuarios y el gobierno y las administraciones públicas.
- Firmas digitales usadas para contratos con varios miembros.
- Firmas digitales para acuerdos o actas entre diferentes organizaciones.

Estas aplicaciones son muy útiles en entornos relacionados con estas actividades:

- Gobierno y administración pública (local, regional o nacional). En particular, es muy conocido que el Ministerio de Comercio, Turismo e Industria español ha desarrollado una aplicación, denominada eCoFirma, para realizar y verificar firmas digitales. Además, este proyecto implementa la validación de archivos compuestos por firmas múltiples (http://oficinavirtual.mityc.es/javawebstart/soc_info/ecofirma/index.html). Esta aplicación usa certificados digitales, por tanto usa claves RSA. Como se sabe, estas claves son grandes y hacen que el archivo que contiene la firma aumente a medida que se añaden más firmas. Con nuestra propuesta, la firma múltiple puede realizarse de manera más eficaz y por tanto el procedimiento de validación es más rápido, aunque las claves usadas tengan un origen muy diferente, tal como se mencionó anteriormente.
- Negocios. En este caso, es posible que diferentes empresas que pertenecen a un grupo específico, o que diferentes personas de un comité, usen esta propuesta para firmar de manera múltiple determinados documentos o acuerdos a los que todos se acogen.
- Declaraciones juradas. La mayoría de los documentos legales requieren firmas de todos los miembros implicados (compraventa, hipoteca, divorcios, declaración de herencia, etc.), así como la certificación notarial del documento. Esta propuesta puede usarse como una aplicación fiable y segura para llevar a cabo estas clases de procesos.
- Banca. Al igual que con los procesos legales, los procedimientos bancarios, en muchas situaciones, requieren firmar documentos en los que están implicadas varias partes, incluyendo la propia entidad bancaria. En este caso, también es posible usar esta invención.
- Militar. En entornos militares, es posible que algunas decisiones deban llevarse a cabo siguiendo un orden predeterminado dependiendo del rango de mando. En un determinado momento, cada una de las autoridades implicadas debe firmar una decisión determinada. Cada una de estas firmas podría llevarse a cabo usando la firma múltiple de cada una de las partes implicadas, aunque en este caso debe establecerse un orden específico en todo el proceso.
- Internet. El creciente uso de Internet podría requerir que dos, o más partes, deban llevar a cabo un acuerdo *on-line*. Este compromiso puede realizarse usando nuestra propuesta de procedimiento digital electrónico. De esta

manera, todas las partes implicadas recurren a una tercera parte de confianza para llevar a cabo tal acuerdo.

- Autoridades de certificación. En general, son necesarias autoridades de certificación para emitir certificados digitales fiables, que cumplan con la norma X.509.v3. Este hecho implica el uso de grandes claves (como sucede con las claves RSA) y procedimientos de firma digital que, en caso de usar un software para realizar firmas múltiples, presentan algunos problemas ya mencionados en el presente documento: un gran esfuerzo computacional, un aumento en el tamaño de los archivos a medida que aumenta el número de firmantes, etc. Con nuestra propuesta, estas autoridades de certificación podrían desarrollar sus propias aplicaciones y pasar a ser la tercera parte de confianza necesaria en la invención.

A continuación se describe una posible implementación del procedimiento de firma múltiple en su conjunto, comenzando con la generación de claves hasta la verificación de la firma, pasando por la generación de la firma. En esta implementación se usarán los tamaños de clave actualmente recomendados para evitar posibles ataques. Tales ataques pueden montarse si fuese posible factorizar el módulo n (problema de factorización de enteros), o si fuese posible resolver el problema de logaritmo discreto, ya sea en el subgrupo multiplicativo de números enteros módulo n o en un subgrupo de orden r.

Supóngase que el grupo de firmantes consiste en $t = 5$ usuarios: $G = \{F_1, F_2, F_3, F_4, F_5\}$ y que T es la tercera parte de confianza.

Generación de claves

Tras las etapas mencionadas en las secciones anteriores, T genera su propia clave privada y la clave pública. Para mostrar un ejemplo que puede usarse en aplicaciones prácticas, con garantías de seguridad, se ha generado un número r *con* 192 bits, que hace que el problema de logaritmo discreto sea irresoluble en un subgrupo de orden r. Además, se han generado los números primos p *y* q para tener, aproximadamente, 512 bits cada uno, lo que significa que n tiene aproximadamente 1024 bits. Este tamaño es suficientemente grande para garantizar su seguridad frente a los ataques de factorización durante un tiempo razonable (los dígitos de cada número se han separado en grupos de 10 para facilitar su legibilidad).

Los valores calculados son los siguientes:

$$u_1 = 74 = 2 \cdot 37,$$

$$u_2 = 188 = 2 \cdot 94,$$

r = 4280023136 1972361770 9720134208 9944684948 9050016803 52659163,

p_1 = 3098365935 5115484298 6754567228 9635523537 4615761798 4830977826 9780185513 2773111664 9369523991 95821,

q_1 = 1520450525 2113540207 5164722445 6778025701 0592684151 0042773032 3583726574 4664496833 3500176742 76217,

p = 9813197637 4121676342 6568482314 0310918044 3735395146 3165527063 6735923149 3320377492 6066484931 1740863445 7483957527 4926886593 9416106694 0401463789 3848791487 8903,

q = 1223421923 9653928573 7921720553 5181885769 8744000443 3360186341 6722487366 9987835834 7336009802 3419921079 1174748161 9489598922 7394318308 0189542960 9232557889 57749,

n = 1200568113 3815441777 9200008036 6880591092 5122610186 1101012447 6894157975 4465649221 2596080380 9123496642 1067708087 9742404316 7621962457 4759824948 7705927686 2918453404 3641462423 0810501474 4219629870 8072023173 2433241319 0597257452 8032813795 4341600436 0856910403 2445755423 0802903516 3200102347 0407173383 4836375827 918469347,

$\varphi(n)$ = 1200568113 3815441777 9200008036 6880591092 5122610186 1101012447 6894157975 4465649221 2596080380 9123496642 1067708087 9742404316 7621962457 4759824948 7705927686 2918232930 1953755813 4602443617 5434708573 0497711419 2475273645 1549217844 8350881808 0757606170 2561451002 5022063108 6888205287 5617968754 1429750414 5496514084 214632696,

$\lambda(n)$ = 1402525261 1697779776 1246015561 0377785768 6729617252 9437752774 8325801441 1744936419 7360621133 0844851766 7090293566 1320231059 3956741493 0846169277 4865387347 5469537220 7724643327 9875093536 1891321645 6744284572 2113021501 6365721481 5842212697 6943090932 1704533099 6.

A continuación, *T* determina un elemento g del grupo de los números enteros módulo *n*, Z^*_n cuyo orden es λ(*n*), y entonces calcula el elemento, $\alpha \in Z^*_n$ de orden *r*.

g = 6717842480 6844949028 2650889244 7341097607 4856794135 4583987170 7599944781 2359090515 8314497447 7912737103 9877888876 4046534433 0747123418 0588943781 6284997792 9837297897 4063898679 2643451640 5737585977 8744839425 8032592210 1563318039 7169841482 5698356635 9188997629 5351471211 3905774450 1739771466 4650268641 0946248241 08211637,

α = 8220765701 6677161639 2295645083 3786098404 5377364917 3263164161 5842095460 8347176795 8314383910 6838300087 8332450440 8079227722 7829626715 4492401350 5177874046 3160709427 8939487852 8737556254 2790192185 4409743264 1912537776 6579916297 4553776942 1764040414 7985455200 6834952531 2224875762 4602378529 0612313367 0589148641 54862251.

Entonces, *T* genera un número secreto $s \in S_r$. Para lograr esto, necesita generar un número aleatorio, *z*, en el intervalo [1, *r*] y calcular $s = \alpha^z \pmod n$. Entonces, determina el elemento β:

z = 2562266773 7774597450 4409468429 7834074901 8000412016 9914173,

s = 4753261714 1928251118 2743629876 4391793367 2307333076 1603563912 5002000524 9060497658 0824242118 6704298946 3445895108 4404897626 6689329111 3557924664 5307037629 8989659634 5807828772 4429722874 3556504143 8852477027 0325311761 9225951212 8994160952 8446681790 9094409361 4867349016 1812858607 1648136904 8090599371 8575649900 69295616,

β = 4861935264 2803079954 6555178384 8192834398 3164492398 1652625442 7161959668 1467074572 8858845443 7528030493 1407985056 2750859231 2565714340 0369700824 6078908252 2563853007 1595785428 4460245931 4579818936 4632592036 1993827761 2083044887 0729245762 1484480929 0382280230 6144704709 2769342983 0689033301 5305777214 2718955708 68985223.

T calcula su clave privada, $a_0, b_0, c_0, d_0 \in Z_r$ y la clave pública, (*P, Q*), que compartirán todos los firmantes del grupo *G*:

a_0 = 3796851234 6569680283 3338368391 5365556062 4833209082 53580661,

b_0 = 2591850459 9367460902 5934873820 3760658099 0333592083 17194694,

c_0 = 1518925798 4392236395 1016813551 4526953578 9549887870 12160497,

d_0 = 3795856474 1095936126 3365200938 2597076303 2769773305 51257252,

P = 8319958649 6056447734 7328052772 9933714896 8407194844 8369500960 5176404651 5164701887 3220925662 9019150287 7627255992 0904205949 0852478635 0690085650 7726157268 1634089222 5831822786 0354912705 0697911594 1772574866 2925316237 6241224905 7755828693 7646145188 0473447052 8492011177 6049627590 7768935677 1530034818 3540964671 99575149,

Q = 7646114260 2099407040 5307511791 2284288636 2591565615 1338741603 2204187548 9954259612 1452554504 1316398658 6916773912 5040898369 8517070596 9516952757 9781255613 9956857784 9279960920 3809990561 1580294365 3546145623 3432930343 3600377178 5723369168 1634251649 1476315642 6634673934 9718641568 1411906770 6077428431 1364467438 92098833.

A continuación, *T* difunde los valores (α, β, *n*, *r*).

La siguiente etapa es el cálculo de las claves privadas de los firmantes del grupo G. Para lograr esto, T calcula, en primer lugar, los siguientes valores:

h = 3872871254 9788136665 1132678495 1408621258 8355187161 47094974,

k = 9595184946 6557841042 7041038895 8525680100 0947749031 804377.

Finalmente, *T* genera los valores aleatorios $b_i, d_i \in Z^*_r$ para *i* = 1, …, 5, y determina los correspondientes valores para a_i y c_i. De esta manera, obtiene las 5 claves privadas, que se distribuirán de manera segura a cada uno de los firmantes del grupo:

(a_1, b_1, c_1, d_1) = (3283241757 5870636656 2148739413 6347233920 8366198197 35436378, 1151430213 9289655133 5986558470 5215666832 7514688545 71831717, 3839753842 7757266331 6697562043 9112177778 0738242539 93302939, 2403125912 8894003222 9570075762 6023463547 5329056797 71114669),

(a_2, b_2, c_2, d_2) = (1550178576 4794736629 8920233691 3661760348 7403838961 45128686, 1563760386 7728655548 8657603181 0647710782 2162854443 65985785, 6103724830 5688795750 4070736808 6412512413 3278352318 4958155, 1759919831 6288861490 4411691354 5080836678 8884940065 70680177),
(a_3, b_3, c_3, d_3) = (2791564021 0033574126 8721638070 1589015898 1812242310 5344330, 6677309778 5621420138 9539159254 2346765657 8489201524 0252392, 1221691357 9545259603 5141749286 9077438598 0549008717 28371300, 4095493946 8040799465 7837334005 8468133920 9532706455 00497958),
(a_4, b_4, c_4, d_4) = (2453365078 9361925247 4551356603 8117805834 4499954214 11351773, 6930834099 0325869370 2436425339 9961527746 8907567771 7562893, 2396464971 9220531648 9542977954 4179966718 7496705898 02476593, 3647109711 1114185126 8934154780 7991224353 9039657962 55314516),
(a_5, b_5, c_5, d_5) = (2814663960 7597908778 4976941814 3370387434 4440465417 57439916, 3435450655 2240480352 2565604258 5579327971 4569713507 39009618, 4168673880 1586125932 7609156104 2565600095 2678283105 60380425, 1821609138 4217289729 1499706890 6952954417 7710311415 86835967).

Verificación de claves

Para verificar la clave de *T* y la clave común de todos los firmantes, simplemente es necesario que todos ellos comprueben las ecuaciones

$$\alpha \neq 1 \pmod{n},$$

$$\alpha^{r} = 1 \pmod{n},$$

$$P = \alpha^{a_i} \cdot \beta^{b_i} \pmod{n},$$

$$Q = \alpha^{c_i} \cdot \beta^{d_i} \pmod{n}.$$

lo cual es casi inmediato.

Generación de la firma múltiple con la colaboración de *T*

Supóngase que el mensaje que va a firmarse está almacenado en un archivo, cuyo contenido es el siguiente (es un fragmento de “El Quijote”):

“En esto, descubrieron treinta o cuarenta molinos de viento que hay en aquel campo, y así como don Quijote los vio, dijo a su escudero:

- *La aventura va guiando nuestras cosas mejor de lo que acertáramos a desear; porque ves allí, amigo Sancho Panza, donde se descubrieron treinta, o poco más desaforados gigantes, con quien pienso hacer batalla y quitarles a todos las vidas, con cuyos despojos comenzaremos a enriquecer, que ésta es buena guerra, y es gran servicio de Dios quitar tan mala simiente de sobre la faz de la tierra.*

- *¿ Qué gigantes ? -dijo Sancho Panza.*

- *Aquellos que allí ves -respondió su amo- de los brazos largos, que los suelen tener algunos de casi dos leguas.*

- *Mire vuestra merced -respondió Sancho- que aquellos que allí se parecen no son gigantes, sino molinos de viento, y lo que en ellos parecen brazos son las aspas, que volteadas del viento, hacen andar la piedra del molino.”*

El cálculo de su resumen usando la función SHA-1 proporciona el siguiente valor de 160 bits, siendo sus expresiones hexadecimal y decimal, respectivamente:

m = 7b 30 e0 ac a8 c5 7b 09 0a cb a4 b0 54 38 b7 a1 0c d0 41 f3 = 7032958724 8581323731 3907950135 0552221954 30113779.

Las firmas parciales de cada uno de los 5 firmantes son las siguientes:

(f_1, g_1) = (2072061490 3529681837 0518359522 5737119748 8834793544 63211454, 3424899271 6561622167 2309350509 1899483096 2204248546 1401791),

(f_2, g_2) = (7519865574 6123920960 8321708870 7222592072 2194208225 5244963, 3052287160 9193339307 3261388288 7970269665 4298411495 82863325),

(f_3, g_3) = (3035127013 8473367395 5815130273 0222450081 2709055452 62584542, 3343996361 8263990970 5373048282 9321275620 4171100823 79165477),

(f_4, g_4) = (5045916079 6959772237 4311181991 8729548876 2656407924 8876843, 2222053057 4350855533 1345730712 8312259343 4626990198 61867838),

(f_5, g_5) = (2569158690 2819158389 3813867008 1064366690 7818627298 94259816, 3005227651 1941091704 8712925875 3266438452 3528620588 04923843).

T verifica la firma de cada uno de los firmantes comprobando que cada firma satisface la siguiente ecuación:

$$P \cdot Q^m \pmod{n} = \alpha^{f_i} \cdot \beta^{g_i} \pmod{n},\ i = 1, \ldots, 5.$$

A partir de las firmas anteriores, *T* determina la firma múltiple simplemente sumando todas ellas entre sí, módulo *r*.

(*f, g*) = (3728790875 1858533998 9703774719 7297807179 7475043041 8859292, 3406007886 1460716190 6483759792 8170821493 4745514353 84903948).

Generación de la firma múltiple sin la colaboración de *T*

En caso de que *T* no colabore y suponiendo que el mensaje que va a firmarse es el mismo que antes, el cálculo de cada una de las firmas parciales acumuladas proporciona los siguientes resultados.

La firma de F_1 es:

(f_1, g_1) = (2072061490 3529681837 0518359522 5737119748 8834793544 63211454, 3424899271 6561622167 2309350509 1899483096 2204248546 1401791).

La verificación de la firma de F_1 llevada a cabo por F_2 es correcta:

$P \cdot Q^m \pmod{n}$ = 1170109731 2819367894 2005216580 6494377952 0636567548 9262094548 8344339052 7421432596 8875335205 2402155197 2230378307 2371932777 1738710426 2768241265 6183149313 3478950938 7479706527 9175557453 9751639118 0565551090 7789905395 4819196663 7323682277 9020280066 1028233179 3869192910 1434425082 3384981457 9716434249 5003307128 728028411.

La firma parcial acumulada de F_2 es:

(f_2, g_2) = (2824048047 8142073933 1350530409 645937895 6105421436 718456417, 3394777088 0849501524 0492323339 7160217975 0518836350 44265116).

La verificación de la firma de F_2 llevada a cabo por F_3 es correcta:

$P^2 \cdot Q^{2m} \pmod{n}$ = 7450100059 1091930946 2501858785 0941187993 2537915629 2584859948 2097737510 5853453621 2291364319 8207733759 5107023683 8537744090 0352638596 6504088919 1070451647 0331532795 8444989461 6206800852 4483586001 3000717481 6710298328 4937496631 1635361223 1293962768 8337804042 4332811755 2543722126 3844271970 1015657563 2043360173 11602864.

La firma parcial acumulada de F_3 es:

(f_3, g_3) = (1579151925 4643079557 7445526473 6737144088 4713253016 28381796, 2458750313 7141130723 6145237413 6536808646 5639920370 70771430).

La verificación de la firma de F_3 llevada a cabo por F_4 es correcta:

$P^3 \cdot Q^{3m} \pmod{n}$ = 1082550119 8844366296 9941799029 9216293849 1473245559 8632411906 1624560875 2927058733 8646774165 2669903997 4431524996 7382651963 1695679541 3626062206 6708356175 6861427098 3545735602 5434248163 0193576555 6718878697 5260754944 4570751541 8377058742 3008072805 0382838655 9480158601 5234920504 3658404027 6285123608 4428151260 693636413.

La firma parcial acumulada de F_4 es:

(f_4, g_4) = (2083743533 4339056781 4876644672 8610098976 0978893808 77258639, 4007802349 5196244857 7708339174 9043830411 2168937657 9980105).

La verificación de la firma de F_4 llevada a cabo por F_5 es correcta:

$P^4 \cdot Q^{4m} \pmod{n}$ = 2709868377 7138928608 4769512938 6848964463 7670022794 1393921794 4877807748 7124069046 1975743029 9505734207 6216783854 6104882998 0222474430 8776847832 3556946739 5710850579 9846374769 1955854682 9555230522 0066146680 9121213039 1802059312 0565937216 9390100135 5015555941 6181673016 9769298446 6748808181 1849405818 8552528092 36031649.

La firma parcial acumulada de F_5, que coincide con la firma múltiple de todo el grupo, es:

(f, g) = (f_5, g_5) = (3728790875 1858533998 9703774719 7297807179 7475043041 8859292, 3406007886 1460716190 6483759792 8170821493 4745514353 84903948).

Verificación de la firma múltiple

Para verificar la validez de la firma múltiple anterior si haya colaborado T o no, cualquier verificador que conozca el mensaje, m, el número de firmantes y la clave pública, (P, Q), no tiene más que comprobar la siguiente ecuación

$$P^5 \cdot Q^{5m} = \alpha^f \cdot \beta^g \pmod{n},$$

lo cual es inmediato, ya que ambos valores son

1170109731 2819367894 2005216580 6494377952 0636567548 9262094548 8344339052 7421432596 8875335205 2402155197 2230378307 2371932777 1738710426 2768241265 6183149313 3478950938 7479706527 9175557453 9751639118 0565551090 7789905395 4819196663 7323682277 9020280066 1028233179 3869192910 1434425082 3384981457 9716434249 5003307128 728028411.

Seguridad

En caso de que dos firmantes cualesquiera, por ejemplo, F_1 y F_3, intenten conspirar para calcular el valor secreto, s, de T, unirían sus respectivas firmas, (f_1, g_1) y (f_3, g_3), de modo que se calcularía el siguiente valor

$s' = (f_1 - f_3)(g_3 - g_1)^{-1} \pmod{r}$ = 2495841599 2675813433 1673611873 1109209990 5818069416 83953188,

pero este valor no es el valor real de s:

s = 4753261714 1928251118 2743629876 4391793367 2307333076 1603563912 5002000524 9060497658 0824242118 6704298946 3445895108 4404897626 6689329111 3557924664 5307037629 8989659634 5807828772 4429722874 3556504143 8852477027 0325311761 9225951212 8994160952 8446681790 9094409361 4867349016 1812858607 1648136904 8090599371 8575649900 69295616.

Si cualquier otro par de firmantes intentan llevar a cabo el mismo ataque, obtendrán el mismo valor de s'. Por tanto no se obtendrá ningún beneficio si hubiese más firmantes en la conspiración.

Finalmente, la misma clase de conspiración para firmas de diferentes mensajes tampoco proporcionaría ninguna mejora, porque el valor obtenido al firmar diferentes mensajes es el mismo una vez más.

Funcionamiento e implementación

El esquema propuesto para realizar firmas digitales múltiples se ha implementado como un Notebook de software Maple v.13 en un ordenador con un procesador Intel® Core™2 Quad CPU Q4900 a 2,66 GHz, con el sistema operativo Windows 7 de Microsoft con 64 bits y con una RAM de 4 GB.

El tiempo de cálculo necesario para cada una de las tareas depende, básicamente, de la longitud de las claves y del número de firmantes que participan en la ejecución de la firma múltiple.

Algunos ejemplos de tiempo de cálculo (en segundos) para las diferentes tareas, con diferentes entradas, se muestran en la siguiente tabla:

Tamaño (en bits) de *r* y *n*, respectivamente	**128, 1024**	**192, 1024**	**192, 1024**	**192, 1024**	**256, 2048**
Número de firmantes	5	5	15	50	10
Tiempo para la generación de claves	6,879	5,397	5,413	5,413	132,211

Tiempo para la verificación de claves	0,015	0,031	0,062	0,187	0,187
Tiempo de cálculo de la firma múltiple con la colaboración de una tercera parte de confianza	0,000	0,015	0,031	0,110	0,094
Tiempo de cálculo de la firma múltiple sin la colaboración de una tercera parte de confianza	0,000	0,015	0,047	0,140	0,093
Tiempo para la verificación de la firma múltiple	0,000	0,000	0,000	0,000	0,016

Como puede verse, la mayor parte del tiempo de cálculo se gasta en la fase de generación de claves, lo cual no es un inconveniente ya que una vez generadas las claves, pueden usarse durante un largo periodo de tiempo para firmar diferentes mensajes.

Como comentario final, los tiempos de cálculo anteriores pueden considerarse bastante grandes, ya que se ha usado el paquete computacional Maple v13. Como es de sobra conocido, Maple es un lenguaje interpretado, un hecho que afecta negativamente a los tiempos de ejecución. Para llevar a cabo una optimización de esta invención, es aconsejable desarrollar una implementación de hardware y/o software específica, usando las herramientas de última tecnología más adecuadas.

Un experto en la técnica puede introducir cambios y modificaciones en las realizaciones descritas sin alejarse del alcance de la invención tal como se define en las reivindicaciones adjuntas.

Siglas y abreviaturas

DLP Problema de logaritmo discreto

RSA Criptosistema Rivest-Shamir-Adleman

SDLP Problema de logaritmo discreto en subgrupo

Referencias

[Abo07] S.J. Aboud, Two efficient digital multisignature schemes, Int. J. Soft. Comput. 2 (2007), 113-117.
[AA07] S.J. Aboud y M.A. Al-Fayoumi, A new multisignature scheme using re-encryption technique, J. Applied Sci. 7 (2007), 1813-1817.

[BN06] M. Bellare y G. Neven, Multi-signatures in the plain public-key model and a general forking lemma, Proceedings of the 13th ACM conference on Computer and Communications Security (CCS'06), 390-399, 2006, Alexandria, Virginia, EE.UU.

[DHM05] R. Durán Díaz, L. Hernández Encinas y J. Muñoz Masqué, El criptosistema RSA, RA-MA, Madrid, 2005.

[Boy88] C. Boyd, Some applications of multiple key ciphers`, Lecture Notes in Comput. Sci. 330 (1988), 445-467.

[ElG85] T. ElGamal, A public-key cryptosystem and a signature scheme based on discrete logarithm, IEEE Trans. Inform. Theory 31 (1985), 469-472.

[FSNT09] H. Fujimoto, T. Suzuki, T. Nakayama, A. Takeshita, (NTT DoCoMo, Inc., Tokyo, JP), Multi signature verification system, electronic signature attaching apparatus, data addition apparatus, and electronic signature verification apparatus, patente estadounidense: US 7.627.763 B2, 1 de diciembre de 2009.

[FGHMM04] A. Fúster Sabater, D. de la Guía Martínez, L. Hernández Encinas, F. Montoya Vitini y J. Muñoz Masqué, Técnicas criptográficas de protección de datos, RA-MA, 3ª ed., Madrid, 2004.

[Gen10] C.B. Gentry (NTT DoCoMo, Inc., Tokyo, JP), Signature schemes using bilinear mappings, patente estadounidense: US 7.653.817 B2, 26 de enero de 2010.

[GR10] C.B. Gentry y Z.A. Ramzan (NTT DoCoMo, Inc., Tokyo, JP), Digital signatures including identity-based aggregate signatures, patente estadounidense: US 7.664.957 B2, 16 de febrero de 2010.

[HK89] L. Harn y T. Kiesler, New scheme for digital multisignature, Elect. Lett. 25 (1989), 1002-1003.

[He02] W.H. He, Weakness in some multisignature schemes for specified group of verifiers, Inform. Proc. Lett. 83 (2002), 95-99.

[HCC98] S.J. Hwang, C.Y. Chen y C.C. Chang, An encryption/multisignature scheme with specified receiving

groups, Comput. System Sci. Engrg. 13, 2 (1998), 109-112.
[IN83] K. Itakura y K. Nakamura, A public-key cryptosystem suitable for digital multisignatures, NEC Res. Development 71 (1983), 1-8.

[KH90] T. Kiesler y L. Harn, RSA blocking and multisignature schemes with no bit expansion, Elect. Lett. 26 (1990), 1490-1491.

[KOKS09] Y. Komano, K. Ohta, S. Kawamura, y A. Shimbo (Toshiba Corp., Tokyo, JP), Multisignature method, apparatus, program, and system, patente estadounidense: US 7.496.759 B2, 24 de febrero de 2009.

[LY96] C.S. Laih y S.M. Yen, Multisignature for specified group of verifiers, J. Inform. Sci. Engrg. 12, 1 (1996), 143–152.

[MFM04] D.A. Modiano, L. Friseh, y D. Mouton (France Telecom, París, FR), Electronic group signature method with revocable anonymity, equipment and programs for implementing the method, solicitud de patente estadounidense: US 2004/0260926 A1, 23 de diciembre de 2004.

[LWK05] J. Lv, X. Wang y K. Kim, Security of a multisignature scheme for specified group of verifiers, Appl. Math. Comput. 166 (2005), 58-63.

[HMV04] D. Hankerson, A.J. Menezes y S. Vanstone, Guide to elliptic curve cryptography, Springer-Verlag, Nueva York, 2004.

[Men93] A.J. Menezes. Elliptic curve public key cryptosystems. Kluwer Academic Publishers, Boston, 1993.

[MOV97] A. Menezes, P. van Oorschot y S. Vanstone, Handbook of applied cryptography, CRC Press, Boca Raton, FL, EE.UU., 1997.

[NIST02] National Institute of Standards and Technology, Secure Hash Standard (SHS), Federal Information Processing Standard Publication 180-2, 2002.

[OO91] K. Ohta y T. Okamoto, Multi-signature schemes based on the Fiat-Shamir scheme, Lecture Notes in Comput. Sci. 739 (1991), 139-148.
[OO01] K. Ohta y T. Okamoto (Nippon Telegraph and Telephone Corporation, Tokyo, JP), Method and apparatus for en-bloc verification of plural digital signatures and recording medium with the method recorded thereon, patente estadounidense: US 6.212.637 B1, 3 de abril de 2001.

[Oka88] T. Okamoto, A digital multisignature scheme using bijective public-key cryptosystems, Commun. ACM Trans. Computer Systems 6 (1988), 432-441.

[PPKW97] S. Park, S. Park, K. Kim y D. Won, Two efficient RSA multisignature schemes, Lecture Notes in Comput. Sci. 1334 (1997), 217-222.

[PLL85] S.F. Pon, E.H. Lu y J.Y. Lee, Dynamic reblocking RSA-based multisignatures scheme for computer and communication networks, IEEE Communications Letters 6 (2002), 43-44.

[QX10] H. Qian y S. Xu, Non-interactive multisignatures in the plain public-key model with efficient verification, Inform. Proces. Letters 111 (2010), 82-89.

[RSA78] R.L. Rivest, A. Shamir y L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Comm. ACM 21 (1978), 120-126.

[Shi01] A. Shimbo (Kabushiki Kaisha Toshiba, Kawasaki, JP), Digital signature method using an elliptic curve, a digital signature system, and a program storage medium having the digital signature method stored therein, patente estadounidense: US 6.088.798 A, 27 de marzo de 2001.

[SFH01] F.W. Sudia y P.C. Freund, S.T.F. Huang (CertCo Inc., Nueva York, US), Multi-step digital signature method and system, patente estadounidense: US 6.209.091 B1, 27 de marzo de 2001.

[Sus09] W. Susilo, Short fail-stop signature scheme based on factorization and discrete logarithm assumptions, Theor. Comput. Sci. 410 (2009), 736-744.

[TK02] K. Takaragi, H. Kurumatani (Hitachi, Ltd., Tokyo, JP), Digital signature generating/verifying method and system using public key encryption, patente estadounidense: US 6.341,349 B1, 22 de enero de 2002.

[WP95] M. Waidner y B. Pfitzmann, The dining cryptographers in the disco: Unconditional sender and recipient untraceability with computationally secure serviceability, Lecture Notes in Comput. Sci. 434 (1989), 690.

[WCW96] T.C. Wu, S.L. Chou, y T.S. Wu, Two ID-based multisignature protocols for sequential and broadcasting architectures, Comput. Comm. 19 (1996), 851-856.

[Yen96] S.M. Yen, Cryptanalysis and repair of the multi-verifier signature with verifier specification, Computers & Security 15, 6 (1996), 537-544.

[YY05] E.J. Yoon y K.Y. Yoo, Cryptanalysis of Zhang-Xiao's multisignature scheme for specified group of verifiers, Appl. Math. Comput. 170 (2005), 226-229.

[ZW10] Y. Zhang, S. Wang, y X. Wang, Method for protecting security of digital signature documents of multiple verifiers strongly designated by multiple signers, patente china: CN 101651541, 17 de febrero de 2010.

[ZX04] Z. Zhang y G. Xiao, New multisignature scheme for specified group of verifiers, Appl. Math. Comput. 157 (2004), 425-431.

[MS10] Micali Silvio US 5638447 10 de junio de 1997.

[MS11] Micali Silvio US 5610982 11 de marzo de 1997.

[CJ97] Camenisch J. et al, Efficient group signature schemes for large groups, Advances in Cryptology-Crypto 97. Santa Barbara, 17-21 de agosto de 1997.

REIVINDICACIONES

1. Un procedimiento para una firma digital múltiple que comprende:

i) generar, por una tercera parte de confianza *(T)*, un conjunto de parámetros, su propia clave privada y una clave privada para cada firmante o miembro *(F_1, F_2, ..., F_t)* de un grupo de firmantes (G);
ii) generar, por cada uno de dichos firmantes *(F_1, F_2, ..., F_t)*, una firma parcial en un resumen (m) de un documento, o mensaje, (*M*) usando sus claves privadas;
iii) generar una firma múltiple a partir de dichas firmas parciales; y
iv) verificar, por un verificador, dicha firma múltiple;
en el que el procedimiento comprende:

- determinar, por dicha tercera parte de confianza (*T*), una clave pública única y común para todos de dichos firmantes *(F_1, F_2, ..., F_t)* en (G), calculando dos números enteros (P) y (Q), en $\mathbf{Z}_n$,

$$P = \alpha^{a_0} \cdot \beta^{b_0} \pmod{n},$$

$$Q = \alpha^{c_0} \cdot \beta^{d_0} \pmod{n};$$

y
- determinar, por dicha tercera parte de confianza (*T*), claves privadas individuales de los firmantes *(F_1, F_2, ..., F_t)* del grupo de firmantes (*G*), asociadas a dicha determinada clave pública única y común, calculando (a_i, b_i, c_i, d_i), para $i = 1, ..., t$,
en el que:

(a_0, b_0, c_0, d_0) son cuatro números enteros aleatorios que pertenecen a $\mathbf{Z}_r$ que definen la clave privada de la tercera parte de confianza (*T*);
(b_i, d_i), para $i = 1, ..., t$, son t pares de números enteros aleatorios en $\mathbf{Z}_r$, y (a_i, c_i), para $i = 1, ..., t$, son t pares de números enteros en $\mathbf{Z}_r$ que verifican las siguientes condiciones:

$$a_i = (h - s \cdot b_i) \pmod{r},$$

$$c_i = (k - s \cdot d_i) \pmod{r};$$

y h y k son dos números enteros secretos, en $\mathbf{Z}_r$, definidos por

$$h = (a_0 + s \cdot b_0) \pmod{r},$$

$$k = (c_0 + s \cdot d_0) \pmod{r},$$

y

- generar, por dicha tercera parte de confianza (*T*), un conjunto de parámetros (n, r, α, β, p, q, s) de modo que publica n, r, α y β y mantiene p, q y s secretas, donde

$$n = p \cdot q,$$

$p = u_1 \cdot r \cdot p_1 + 1$ y $q = u_2 \cdot r \cdot q_1 + 1$ son dos números primos grandes,
u_1 y u_2 son dos números enteros pares, cuyo máximo común divisor (mcd) verifica

$$\text{mcd}(u_1, u_2) = 2,$$

p_1, q_1, r, son números primos,
α es un elemento reversible en el grupo de los enteros módulo n, $\mathbf{Z}_n$, con orden multiplicativo r, que verifica la condición

$$\text{mcd}(\alpha, (p - 1)(q - 1)) = 1;$$

$\beta = \alpha^s \pmod{n}$ y
s es un número secreto aleatorio en el subgrupo generado por α.

2. Un procedimiento según la reivindicación 1, en el que cada firmante (F_1,…F_t) calcula además, con la necesaria colaboración de la tercera parte de confianza *(T)*, su propia firma, (f_i, g_i), para un *hash* (m) del mensaje (M) que viene

dada por:

$$f_i = a_i + c_i \cdot m \pmod r,$$

$$g_i = b_i + d_i \cdot m \pmod r.$$

y envía además, a cada firmante, dicha propia firma calculada, de una manera segura, a la tercera parte de confianza *(T)*.

3. Un procedimiento de acuerdo con la reivindicación 2, que comprende verificar, por la tercera parte de confianza (*T*), la firma calculada de cada firmante *(F_1, F_2, ..., F_t)* comprobando:

$$P \cdot Q^m \pmod n = \alpha^{f_i} \cdot \beta^{g_i} \pmod n,\ i = 1, \ldots, t.$$

4. Un procedimiento de acuerdo con la reivindicación 3, en el que, tras una verificación válida de la firma calculada de cada firmante *(F_1, F_2, ..., F_t)*, la tercera parte de confianza *(T)* calcula y publica además la firma digital múltiple corta, *(f, g)*, del grupo (*G*) para el *hash (m)* del mensaje (M) que comprende además lo siguiente:

$$f = (f_1 + \cdots + f_t) \pmod r = \Sigma_{i=1,\ldots,t}\ f_i \pmod r,$$

$$g = (g_1 + \cdots + g_t) \pmod r = \Sigma_{i=1,\ldots,t}\ g_i \pmod r.$$

5. Un procedimiento de acuerdo con la reivindicación 1, en el que el primer firmante *(F_1)* además determina, sin la colaboración de la tercera parte de confianza *(T)*, su propia firma parcial agregada (*f_1, g_1*) para el *hash* (*m*) del mensaje (M) donde:

$$f_i = a_1 + c_1 \cdot m \pmod r,$$

$$g_1 = b_1 + d_1 \cdot m \pmod r.$$

y la envía, de una manera segura, al grupo firmantes *(F_1, F_2, ..., F_t)*.

6. Un procedimiento de acuerdo con la reivindicación 5, en el que cada firmante excepto el primero *(F_2, ..., F_t)* verifica además, sin la colaboración de la tercera parte de confianza *(T)*, la firma parcial agregada ((f_{i-1}, g_{i-1}), *i* = 2, ..., *t*) ya calculada por el firmante anterior, comprobando

$$P^{i-1} \cdot Q^{(i-1)m} = \alpha^{f_{i-1}} \cdot \beta^{g_{i-1}} \pmod n,\ i = 2, \ldots, t.$$

7. Un procedimiento de acuerdo con la reivindicación 6, en el que cada firmante excepto el primero *(F_2, ..., F_t)* determina además, sin la colaboración de la tercera parte de confianza *(T)*, su propia firma parcial agregada ((f_{i-1}, g_{i-1}), *i* = 2, ..., *t*) calculando

$$f_i = f_{i-1} + a_i + c_i \cdot m \pmod r = a_1 + \cdots + a_i + (c_1 + \cdots + c_i)\, m \pmod r,\ i = 2, \ldots, t,$$

$$g_i = g_{i-1} + b_i + d_i \cdot m \pmod r = b_1 + \cdots + b_i + (d_1 + \cdots + d_i)\, m \pmod r,\ i = 2, \ldots, t.$$

y envía además dichas firmas parciales agregadas propias, excepto la del último firmante (F_t), de una manera segura, al grupo de firmantes *(F_1, F_2, ..., F_t)*.

8. Un procedimiento de acuerdo con la reivindicación 7, en el que el último firmante (F_t) publica además, sin la colaboración de la tercera parte de confianza *(T)*, su firma parcial agregada como la firma digital múltiple corta, (*f, g*), de todo el grupo de firmantes:

$$(f, g) = (f_t, g_t).$$

9. Un procedimiento de acuerdo con las reivindicaciones 4 u 8, en el que un verificador determina si la firma digital múltiple corta (*f, g*) del grupo (*G*) para el *hash* (*m*) del mensaje (M), se cumple la siguiente expresión:

$$P^f \cdot Q^{f \cdot m} = \alpha^f \cdot \beta^g \pmod n.$$

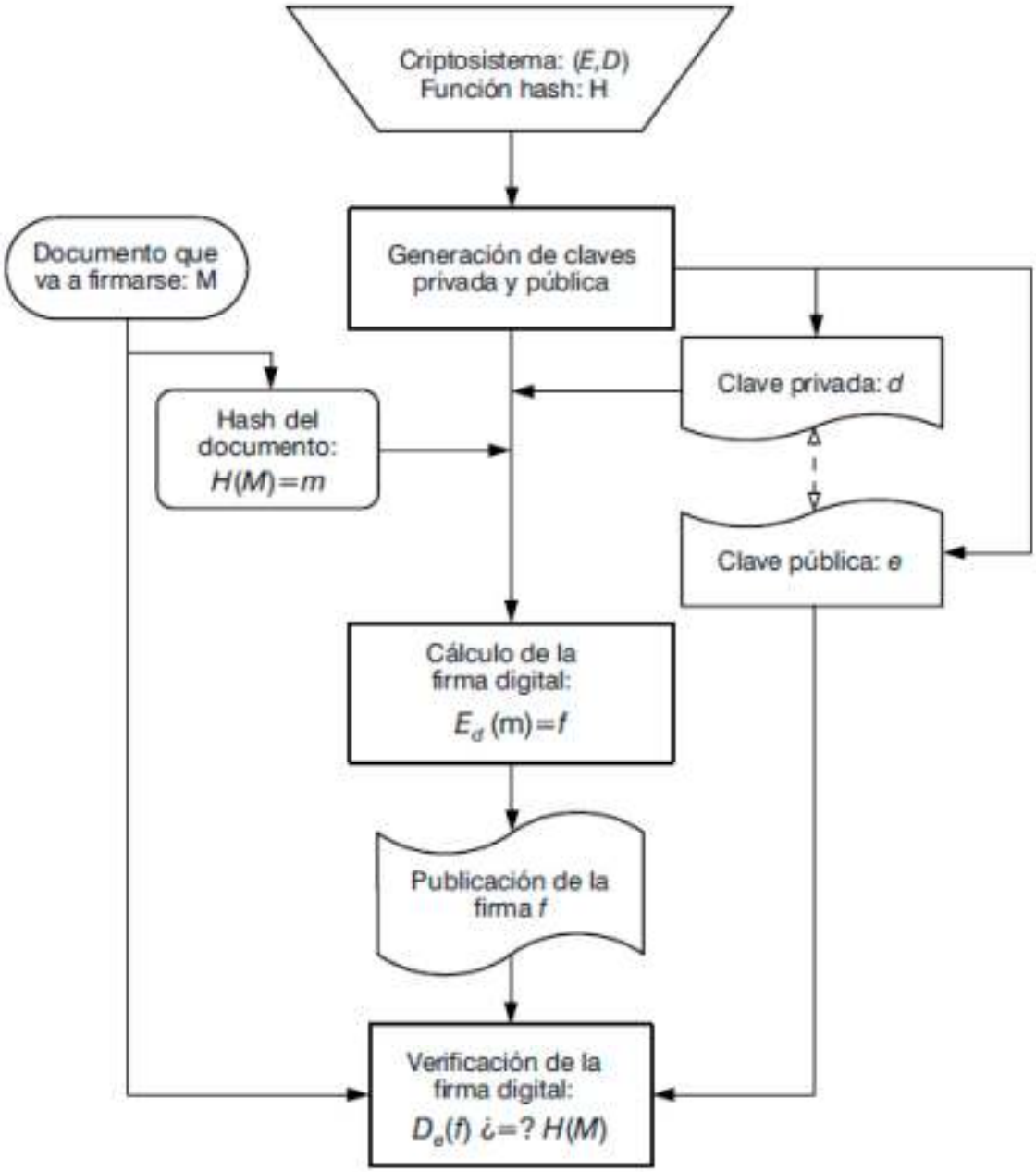
Criptosistema: (E,D)
Función hash: H
Documento que va a firmarse: M
Generación de claves privada y pública
Clave privada: d
Hash del documento: H(M)=m
Clave pública: e
Cálculo de la firma digital: E_d (m)=f
Publicación de la firma f
Verificación de la firma digital: D_e(f) ¿=? H(M)

FIGURA 1

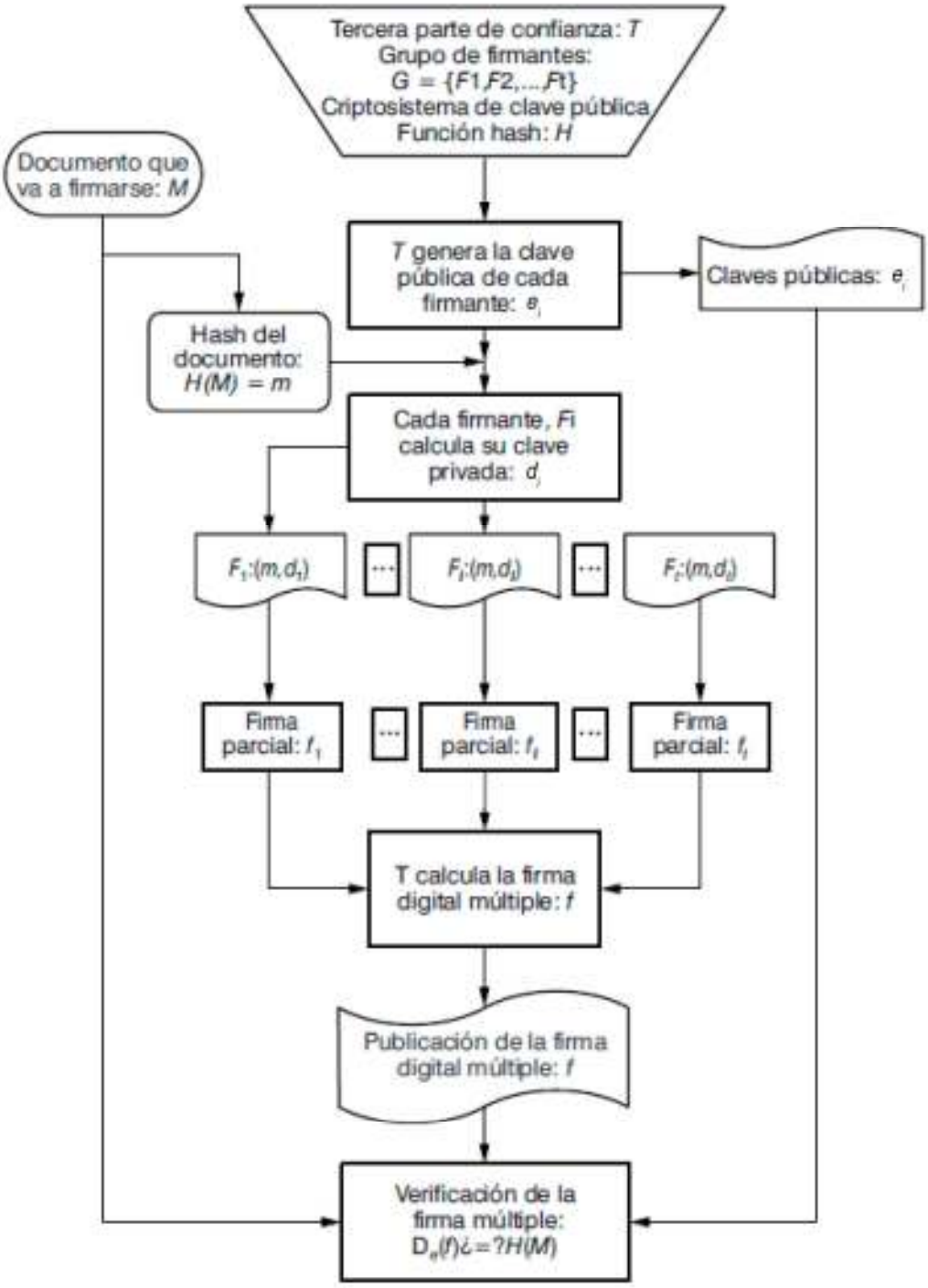
Tercera parte de confianza: T
Grupo de firmantes:
G = {F1,F2,...,Ft}
Criptosistema de clave pública
Función hash: H
Documento que va a firmarse: M
T genera la clave pública de cada firmante: e_i
Claves públicas: e_i
Hash del documento: H(M) = m
Cada firmante, Fi calcula su clave privada: d_i
F1:(m,d1)
Fi:(m,di)
Ft:(m,dt)
Firma parcial: f1
Firma parcial: fi
Firma parcial: ft
T calcula la firma digital múltiple: f
Publicación de la firma digital múltiple: f
Verificación de la firma múltiple:
D_e(f)¿=?H(M)

FIGURA 2

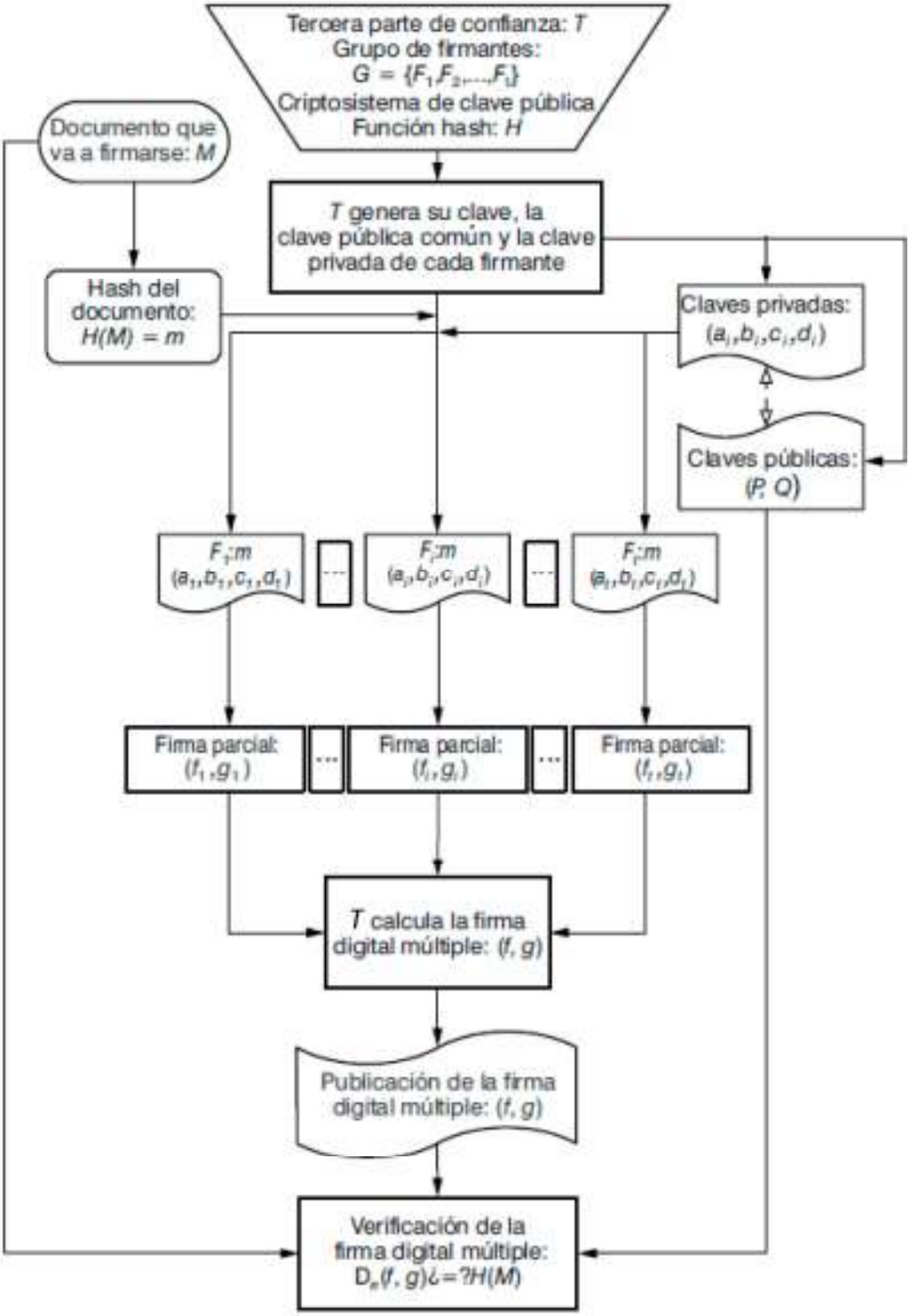
Tercera parte de confianza: T
Grupo de firmantes:
G = {F1,F2,...,Ft}
Criptosistema de clave pública
Función hash: H
Documento que va a firmarse: M
T genera su clave, la clave pública común y la clave privada de cada firmante
Hash del documento: H(M) = m
Claves privadas: (ai,bi,ci,di)
Claves públicas: (P, Q)
F1:m (a1,b1,c1,d1)
Fi:m (ai,bi,ci,di)
Ft:m (at,bt,ct,dt)
Firma parcial: (f1,g1)
Firma parcial: (fi,gi)
Firma parcial: (ft,gt)
T calcula la firma digital múltiple: (f, g)
Publicación de la firma digital múltiple: (f, g)
Verificación de la firma digital múltiple: Dm(f, g)¿=?H(M)

FIGURA 3

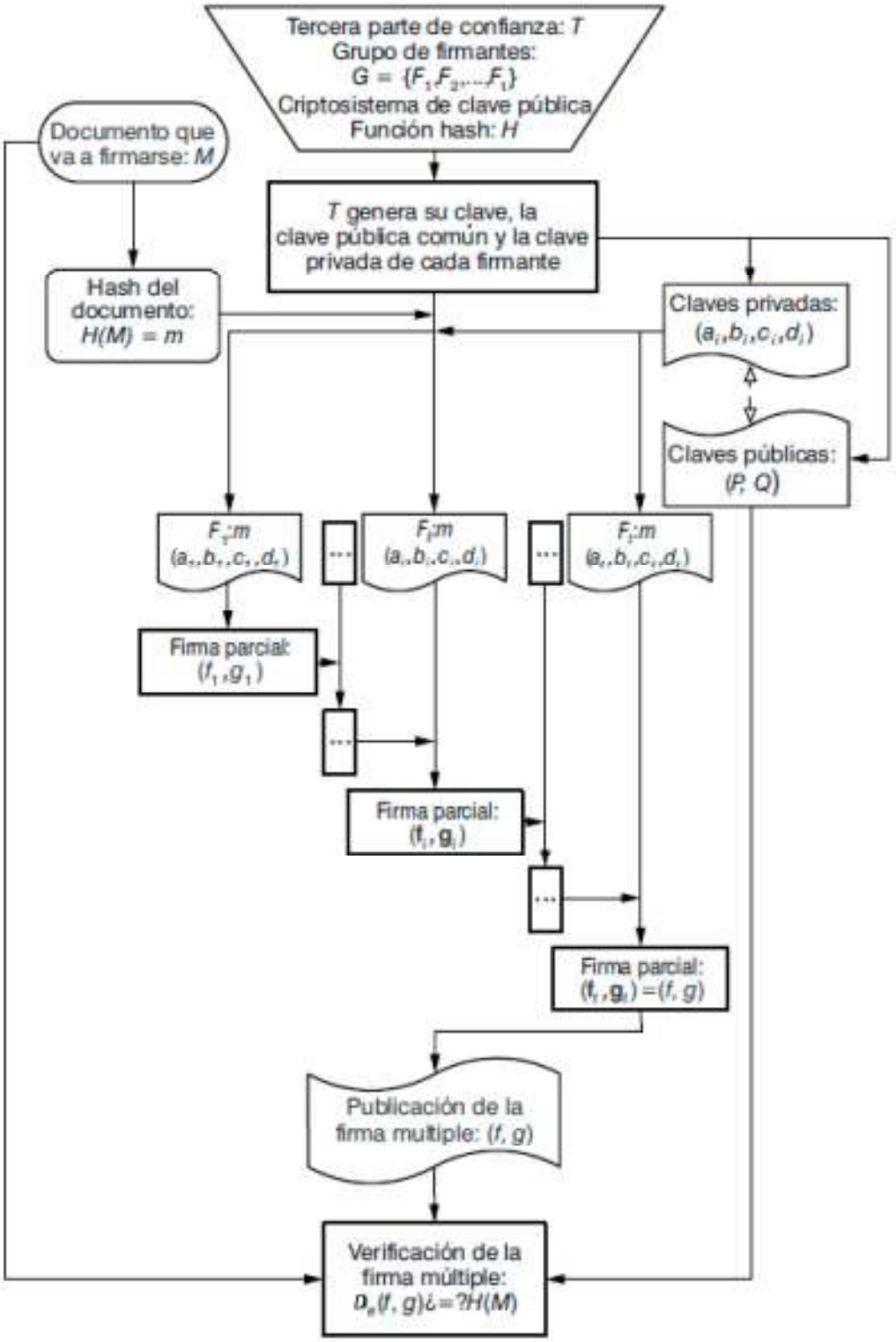
Tercera parte de confianza: T
Grupo de firmantes:
G = {F1,F2,...Ft}
Criptosistema de clave pública
Función hash: H
Documento que va a firmarse: M
T genera su clave, la clave pública común y la clave privada de cada firmante
Hash del documento: H(M) = m
Claves privadas: (ai,bi,ci,di)
Claves públicas: (P, Q)
F1:m (a1,b1,c1,d1)
Fi:m (ai,bi,ci,di)
Ft:m (at,bt,ct,dt)
Firma parcial: (f1,g1)
Firma parcial: (fi,gi)
Firma parcial: (ft,gt) = (f, g)
Publicación de la firma multiple: (f, g)
Verificación de la firma múltiple: Dp(f, g)¿=?H(M)

FIGURA 4

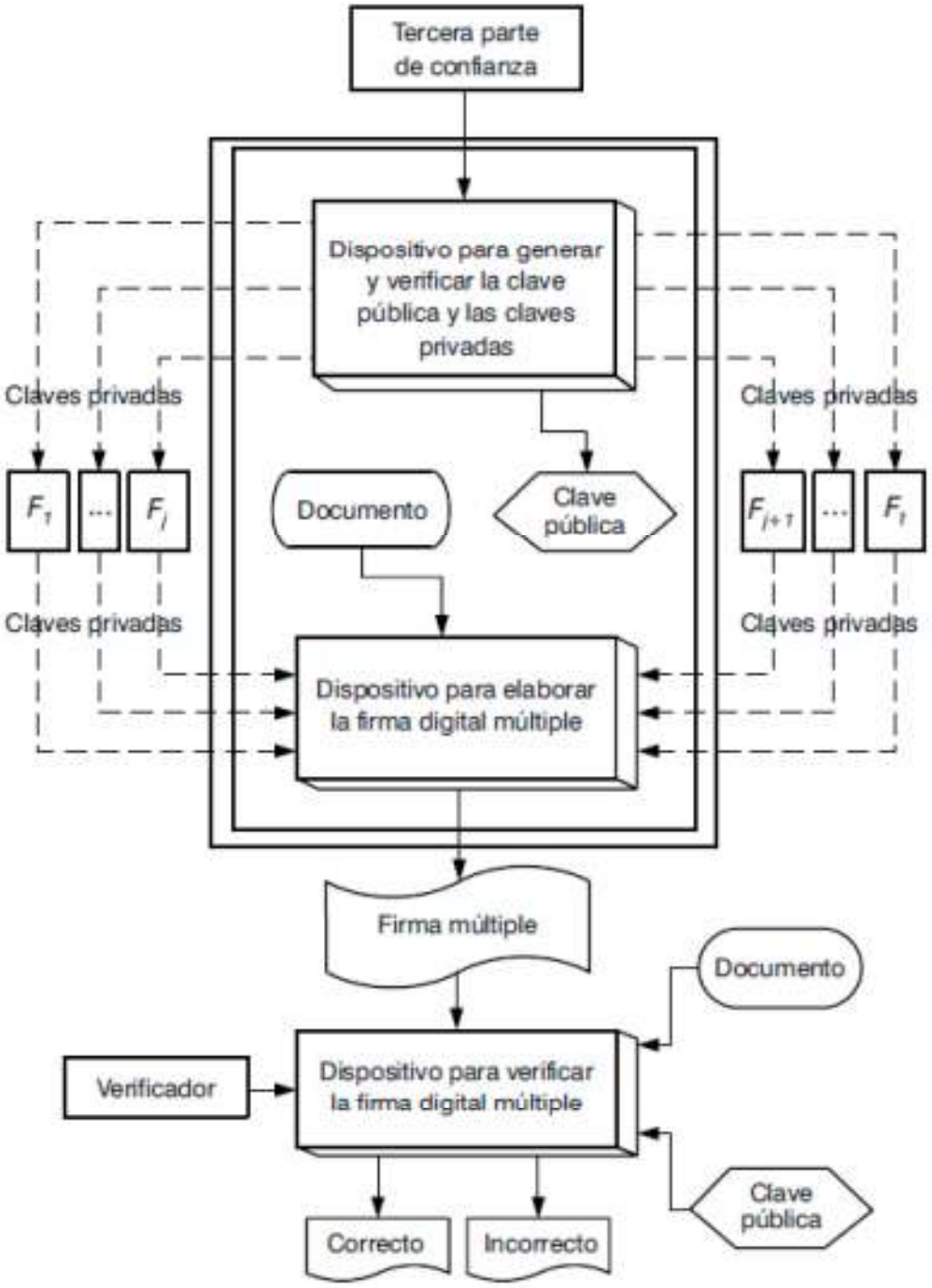
Tercera parte de confianza
Dispositivo para generar y verificar la clave pública y las claves privadas
Claves privadas
Claves privadas
F_1
...
F_l
Documento
Clave pública
F_{l+1}
...
F_t
Claves privadas
Claves privadas
Dispositivo para elaborar la firma digital múltiple
Firma múltiple
Documento
Verificador
Dispositivo para verificar la firma digital múltiple
Clave pública
Correcto
Incorrecto

FIGURA 5