(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 287 745**

(51) Int. Cl.:
H04L 9/30 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Número de solicitud europea: **04741978 .3**
(86) Fecha de presentación : **08.07.2004**
(87) Número de publicación de la solicitud: **1652336**
(87) Fecha de publicación de la solicitud: **03.05.2006**

(54) Título: **Procedimiento para la aplicación asegurada de un algoritmo de criptografía de tipo RSA y componente correspondiente.**

(30) Prioridad: **31.07.2003 FR 03 09457**

(45) Fecha de publicación de la mención BOPI:
16.12.2007

(45) Fecha de la publicación del folleto de la patente:
16.12.2007

(73) Titular/es: **GEMPLUS**
avenue du Pic de Bertagne
Parc d'Activités de Gémenos
13420 Gémenos, FR

(72) Inventor/es: **Villegas, Karine;**
Joye, Marc y
Chevallier-Mames, Benoit

(74) Agente: **Cañadell Isern, Roberto**

Venta de fascículos: Oficina Española de Patentes y Marcas. Pº de la Castellana, 75 – 28071 Madrid

DESCRIPCIÓN

Procedimiento para la aplicación asegurada de un algoritmo de criptografía de tipo RSA y componente correspondiente.

El invento se refiere a un procedimiento para la aplicación asegurada de un algoritmo de criptografía en un componente electrónico y, más especialmente, para la aplicación asegurada de un algoritmo de criptografía de tipo RSA.

El invento concierne asimismo el componente electrónico correspondiente.

Tales componentes se utilizan principalmente en aplicaciones en donde el acceso a servicios o a datos está controlado severamente.

Poseen una arquitectura dicha informática, es decir programable formada en torno a un microprocesador y memorias, entre las cuales una memoria programa no volátil de tipo EEPROM que contiene uno o varios nombres secretos. Se trata de una arquitectura generalista en condiciones de ejecutar cualquier tipo de algoritmo.

Estos componentes se utilizan en sistemas informáticos, embarcados o no. Se utilizan principalmente en las tarjetas inteligentes, para algunas de sus aplicaciones. Estas aplicaciones son por ejemplo acceso a ciertos bancos de datos, aplicaciones bancarias, aplicaciones de telepeaje, por ejemplo para la televisión, el reposte de gasolina e incluso el paso de peajes de autopistas.

Estos componentes o tarjetas aplican un algoritmo de criptografía para asegurar el cifrado de datos emitidos y/o la decodificación de los datos recibidos, la autentificación o la firma digital de un mensaje.

A partir de este mensaje aplicado a la entrada de la tarjeta por un sistema huésped (servidor, cajero bancario...) y de los números secretos contenidos en la tarjeta, la tarjeta suministra a su vez al sistema huésped este mensaje codificado, autentificado o firmado, lo que permite por ejemplo al sistema huésped autentificar el componente o la tarjeta, intercambiar datos...

Las características de los algoritmos de criptografía pueden conocerse: cálculos efectuados, parámetros usados. El único desconocido es él o los números secretos. Toda la seguridad de estos algoritmos de criptografía radica en este(estos) número(s) secreto(s) contenido(s) en la tarjeta y desconocido(s) del mundo exterior a la tarjeta. Este número secreto no puede deducirse al conocer solamente el mensaje aplicado a la entrada y el mensaje codificado de vuelta.

Ahora bien, se ha podido constatar que ataques externos basados en magnitudes físicas mensurables del exterior del componente cuando éste estaba desarrollando el algoritmo de criptografía, permitían a terceras personas mal intencionadas descubrir el(los) número(s) secreto(s) contenido(s) en esta tarjeta. Estos ataques se llaman ataques de canales ocultos, los ataques SPA, acrónimo anglosajón por Single Powert Análisis basados en una o varias medidas y los ataques DPA, acrónimo anglosajón por Differential Power Análisis basados en análisis estadísticos resultantes de numerosas medidas. El principio de estos ataques de canales ocultos radica por ejemplo en el hecho de que el consumo en corriente del microprocesador que ejecuta instrucciones varía según la instrucción o el dato manipulado.

Existe igualmente un tipo de ataque, dicho “ataque por falta”. En este tipo de ataque, el atacante inyecta cualquier tipo de falta durante el cálculo de un algoritmo criptográfico, con el fin de explotar la presencia de esta falta para extraer una información secreta.

La falta también puede provenir de un error de cálculo debido al material que pone en aplicación el algoritmo criptográfico. No obstante, en uno u otro caso, podemos considerar que se trata de un ataque por falta.

Estos distintos ataques se prevén, principalmente, con los algoritmos de criptografía con llave pública, como por ejemplo el algoritmo RSA (que lleva el nombre de sus autores Rivest, Shamir, Adleman), que es el más utilizado en criptografía en este campo de aplicación, y al que se aplica el presente invento más especialmente.

Seguidamente evocamos brevemente las principales características del sistema criptográfico con llave pública RSA.

La primera realización de esquema de codificación y de firma con llave pública fue puesta a punto en 1977 por Rivest, Shamir y Adleman, quienes inventaron el sistema criptográfico RSA. La seguridad de RSA se basa en la dificultad de factorizar un número mayor que es el producto de dos números primos. Este es el sistema criptográfico con llave pública más utilizado. Puede emplearse como procedimiento de codificación o como procedimiento de firma.

El principio del sistema criptográfico RSA es el siguiente. En primer lugar, consiste en generar el par de llaves ESA.

De este modo, cada usuario crea una llave pública RSA y una lave privada correspondiente, según el siguiente procedimiento en 5 etapas:

1) Generar dos números primos distintos p y q;

2) Calcular n=pq y $\Phi(n) = (p-1)(q-1)$, Φ se llama la función indicadora de Euler;

3) Seleccionar un entero e, $1<e<\Phi(n)$, tal como pgcd $(e,\Phi(n))=1$, de modo aleatorio o a elección del usuario que podría escoger e pequeño, tal como $e=2^{16}+1$ ó e=3 ó e= 17;

4) Calcular el único entero d, $1<d<\Phi(n)$, tal como: e.d=1 módulo $\Phi(n)$; (1)

5) La llave pública es (n,e); la llave privada es d ó (d,p,q).

Los enteros e y d se llaman respectivamente exponente público y exponente privado. El entero n se llama el módulo RSA.

Una vez que se hayan definido los parámetros públicos y privados, en vista de x, con $0<x<n$, la operación pública en x que puede ser por ejemplo la codificación del mensaje x consiste en calcular: $y=x^e$ módulo n (2).

En este caso, la operación privada correspondiente es la operación de decodificación del mensaje cifrado y, la cual consiste en calcular: y^d módulo n (3).

La operación pública en x también puede ser la verificación de la firma x, y consiste en calcular: $Y= x^e$ módulo n (2).

La operación privada correspondiente es entonces la generación de una firma x a partir del mensaje previamente codificado y por aplicación de una función de picado μ ("padding" según la terminología anglosajona, y consiste en calcular: y^d módulo n (3).

Con $x= y^d$ módulo n puesto que e.d=1 módulo $\Phi(n)$.

Vamos a presentar otro modo de funcionamiento dicho modo CRT, ya que se basa en el teorema de los restos chinos ("Chinese Remainder Therorem" o CRT en inglés) y cuatro veces más rápido que el del algoritmo RSA estándar. Según este modo CRT, los cálculos módulo n no se efectúan directamente sino que se efectúan en un primer tiempo los cálculos módulo p y módulo q.

Los parámetros públicos son (n, e) pero los privados son en este modo (p, q, d) ó (p, q, d_p, d_q, i_q) con $d_p = d$ módulo (p-1), $d_q = d$ módulo (q-1), e $i_q = q^{-1}$ módulo p.

Por la realización (1), se obtiene: $Ed_p= 1$ módulo (p-1) y $ed_q= 1$ módulo (q-1) (4).

La operación pública se efectúa del mismo modo que para el modo de funcionamiento estándar. En cambio, para la operación privada, se calcula primero: $X_p= y^{dp}$ módulo p y $x_q=y^{dq}$ módulo q.

Seguidamente, por aplicación del teorema de los restos chinos, se obtiene $x=y^d$ módulo n por: $X= CRT(x_p, x_q) = x_q + q[i_q (x_p-x_q)$ módulo p] (5).

Una orientación importante en el campo de la criptografía con llave pública que utiliza el esquema de codificación RSA consiste así pues en asegurar la aplicación de los algoritmos RSA contra los distintos tipos de ataques posibles evocados anteriormente, en particular los ataques de canales ocultos, tales como los ataques DPA y SPA, así como los ataques dichos por falta en donde el atacante, mediante cualquier tipo de método, inyecta una falta durante el cálculo de una operación privada del algoritmo RSA, con el fin de obtener un valor corrompido a partir del cual es posible, en ciertos casos, deducir ciertos datos secretos.

En el estado de la técnica, se han previsto ciertos procedimientos de contramedida para precaver estos diferentes tipos de ataque.

Principalmente, una contramedida posible para precaver los ataques de tipo DPA (y SPA) contra el RSA en modo estándar consiste en hacer aleatorio el cálculo de la operación privada del RSA (firma o decodificación) al introducir en el cálculo un valor aleatorio.

Así pues, un método de contramedida de este tipo consiste en calcular la operación privada en modo estándar (3) $x= y^d$ módulo n de la siguiente forma:

$x= y^{d-r} .y^r$ módulo n, con r que es un número entero aleatorio. No obstante, el inconveniente de este método de contramedida es que el tiempo de cálculo se duplica.

Otro método de contramedida de este tipo para precaver los ataques DPA (y SPA) contra el RSA en modo estándar consiste en calcular la operación privada (3) $x= y^d$ módulo n de la siguiente manera:

$x = y^{(d+r,\Phi(n)}$ módulo n, en donde r es un entero aleatorio. Sin embargo, un inconveniente de este método es que se requiere conocer el valor de $\Phi(n)$, que por lo general lo desconoce el algoritmo de criptografía que aplica la operación privada (firma o decodificación).

Asimismo, se propone una variante de este método, la cual ya no se basa en el conocimiento del valor de $\Phi(n)$, sino en el valor del exponente público e. En efecto, según (1) tenemos: e.d. = 1 módulo $\Phi(n)$, también, existe un entero k tal como: e.d-1= k. $\Phi(n)$.

Por consiguiente, la expresión $x = y^{(d+r,\Phi(n)}$ módulo n puede calcularse en forma de:

$x = y^{(d+r,(ed-1)}$ módulo n, con r un entero aleatorio.

Este método de cálculo de contramedida es equivalente a aquel del que procede, pero sin embargo tiene la ventaja de no requerir el conocimiento del valor de $\Phi(n)$. Necesita menos memoria puesto que no necesita guardar $\Phi(n)$.

No obstante, esta variante de contramedida, para poder aplicarla, necesita conocer el valor del exponente público e. Ahora bien, en numerosas aplicaciones de criptografía, el componente o el dispositivo que aplica la operación privada del algoritmo RSA no dispone siempre del exponente público e, principalmente cuando sólo ejecuta la operación privada. Por tanto, en este contexto, generalmente el exponente público e no se conoce o está indisponible.

Las contramedidas descritas anteriormente están destinadas principalmente a precaver los ataques de tipo DPA. Sin embargo, hacen más difíciles los ataques de tipo SPA en la media en que la ejecución del algoritmo no es determinista.

En lo que se refiere al otro tipo de ataque que fue evocado, dicho ataque por falta, la mejor protección posible para precaverlo consiste en testar, en modo estándar, que el valor x obtenido por aplicación de la operación privada verifica efectivamente la relación $x^e = y$ módulo n de la operación pública. Si no fuera así, no devolveremos el valor y para evitar su utilización a fines de cripto-análisis.

En modo CRT, la protección consiste en verificar por una parte, si efectivamente las relaciones $x^e = y$ módulo p y, por otra parte $x^e = y$ módulo q se han verificado.

En efecto, cuando estas relaciones se han verificado, estamos seguros de que no ha habido errores durante el desarrollo de la operación privada del algoritmo RSA.

No obstante, un inconveniente que impide la aplicación de tales verificaciones contra los ataques por falta, en modo estándar o en modo CRT, es que estas operaciones de verificación necesitan asimismo conocer previamente el exponente público e. Ahora bien, como ya hemos visto, el componente o el dispositivo que aplica la operación privada del algoritmo RSA, en modo estándar o CRT, no dispone siempre del exponente público e, principalmente cuando sólo ejecuta la operación privada. Por consiguiente, en este contexto generalmente el exponente público e se desconoce o está indisponible.

El documento de la patente FR 2 830 146 (D1) propone con este fin un procedimiento que permite realizar ciertas etapas de un algoritmo de criptografía, y principalmente de tipo RSA en modo estándar o CRT, utilizando un exponente público e que no se conoce *a priori*.

El procedimiento objeto de D1 permite en particular realizar una contramedida, principalmente a los ataques por falta, que ofrece la mejor protección posible tal como se ha evocado más arriba, incluso cuando no se conoce el exponente público e.

Para ello, ya sea (e, d) un par correspondiente de exponentes RSA respectivamente público y privado y ya sea n el módulo RSA. D1 se basa en la siguiente constatación según la cual en un 95% de los casos, el valor del exponente público e se elige entre los valores $2^{16}+1$, 3, 17. El método de D1, que exponemos aquí brevemente como referencia al modo estándar pero que no obstante puede aplicarse al modo CRT, consiste entonces en verificar que e es bien igual a uno de esos valores testando sucesivamente si $e_i.d=1$ módulo $\Phi(n)$, con el e_i Φ E= $\{2^{16}+1, 3, 17\}$, hasta que se verifique la relación.

Cuando la relación se ha verificado para un e_i, entonces sabemos que $e=e_i$. Una vez que se ha determinado el valor del exponente público de este manera, e se memoriza en vista de utilizarlo en cálculos del algoritmo RSA que pone la mira en verificar que no haya errores, debidos a un ataque por falta, durante el desarrollo de una operación privada correspondiente del algoritmo RSA. De este modo, al conocer e, es posible afirmar con una probabilidad igual a 1 que la operación privada relacionada por ejemplo con la generación de una firma s, con $s = \mu(m)^d$ módulo n, $\mu(m)$ es el valor obtenido por la aplicación de una función μ de padding al mensaje m que debe firmarse, se ha efectuado sin error al verificar sencillamente que el valor s obtenido verifica la relación $s^e = \mu(m)$ módulo n de la operación pública correspondiente.

Si no ha podido atribuirse a e ningún valor de e_i, entonces conviene constatar según D1 que los cálculos del algoritmo RSA que emplean el valor e para la seguridad contra los ataques por falta no pueden efectuarse.

Sin embargo, un inconveniente del método propuesto por D1 es que implica ejecutar una pluralidad de cálculos modulares cuanto se testa sucesivamente si la relación $e_i d = 1$ módulo $\Phi(n)$ se ha verificado, para un valor de e_i entre los e_i previstos. Ahora bien, los cálculos modulares son cálculos complejos. Este método resulta ser por tanto penalizador en términos de tiempo de cálculo y recursos de cálculo.

Igualmente, el problema que se plantea es el de compensar los inconvenientes citados anteriormente.

Más particularmente, uno de los objetivos del presente invento consiste en determinar de manera que no sea penalizadora en términos de rapidez y complejidad de cálculo, el valor de un exponente público e entre un conjunto de valores probables predeterminados, cuando no se conoce este valor de e *a priori*, el exponente e se aplica en ciertas etapas de un algoritmo de criptografía de tipo RSA en modo estándar o CRT.

Otra finalidad consiste en poder aplicar, una vez determinado el valor del exponente público e, operaciones de contramedida utilizando el valor del exponente público e, poniendo la ira en precaver por una parte, los ataques dichos ataques por alta y, por otra parte, los ataques dichos de canales ocultos, principalmente de tipo DPA y SPA, susceptibles de realizarse durante la aplicación de una operación privada de un algoritmo de criptografía, en particular de tipo RSA.

Con estos objetivos en vista, el invento se refiere a un procedimiento para la aplicación asegurada de un algoritmo de criptografía con llave pública, dicha llave pública está compuesta de un número entero n, producto de dos grandes números primos p y q, y de un exponente público e, algoritmo que comprende igualmente una llave privada, dicho procedimiento consiste en determinar un conjunto E que comprende un número predeterminado de valores e_i, susceptibles de corresponder al valor del exponente público e, los e_i son números primos, caracterizados porque comprenden las siguientes etapas que consisten en:

a) definir un valor $\varepsilon = \Pi$ ei

$$ei \; \varepsilon \; E$$

tal como ε/e_i, ya sea inferior a $\Phi(n)$ para todo e_i perteneciente a E, Φ es la función indicadora de Euler;

b) aplicar el valor ε en un cálculo predeterminado:

c) para cada uno de los e_i de E, testar si el resultado de dicho cálculo predeterminado es igual a un valor ε/e_i:

- si fuera así, entonces atribuir el valor e_i a e y memorizar e en vista de utilizarlo e cálculos del dicho algoritmo de criptografía;
- en caso contrario, constatar que los cálculos de dicho algoritmo de criptografía que utiliza el valor e no pueden efectuarse.

La ventaja es por tanto clara, solo tenemos una sola multiplicación modular.

En una primera variante, el algoritmo de criptografía se basa en un algoritmo de tipo RSA en modo estándar.

En relación con esta primera variante, el algoritmo de criptografía se basa en un algoritmo de tipo RSA en modo estándar.

En relación con esta primera variante, el cálculo predeterminado de la etapa b) consiste en calcular un valor C:

C = ε.d módulo $\Phi(n)$, d es la llave privada correspondiente del algoritmo RSA, tal como e.d=1 módulo $\Phi(n)$ y Φ es la función indicadora de Euler.

Según una alternativa, el cálculo predeterminado de la etapa b) consiste en calcular un valor C.

C = ε.d módulo λ (n), d es la llave privada correspondiente del algoritmo RSA tal como e.d= 1 módulo λ (n) e λ es la función de Carmichael.

En una segunda variante, el algoritmo de criptografía se basa en un algoritmo de tipo RSA en modo CRT.

En relación con esta segunda variante, el cálculo predeterminado de la etapa b) consiste en calcular un valor C:

C = $\varepsilon.d_p$ módulo (p-1), d_p es la llave privada correspondiente al algoritmo RSA tal como $e.d_p$ =1 módulo (p-1).

Según una alternativa, el cálculo predeterminado de la etapa b) consiste en calcular un valor C:

C = $\varepsilon.d_q$ módulo (q-1), d_q es la llave privada correspondiente al algoritmo RSA tal como $e.d_q$ =1 módulo (q-1).

Según otra alternativa, el cálculo predeterminado de la etapa b) consiste en calcular dos valores C_1 y C_2 tales como:

C_1 = $\varepsilon.d_p$ módulo (p-1), d_p es la llave privada correspondiente al algoritmo RSA tal como $e.d_p$ =1 módulo (p-1).

C_2 = $\varepsilon.d_q$ módulo (q-1), d_q es la llave privada correspondiente al algoritmo RSA tal como $e.d_p$ =1 módulo (q-1).

Y en que la etapa de test c) consiste para cada e_i, en testar si C_1 y/o C_2 es igual al valor ε/ e_i:

- si fuera así, entonces atribuir el valor e_i a e y memorizar e en vista de su utilización en cálculos del dicho algoritmo de criptografía;
- en caso contrario, constatar que los cálculos del dicho algoritmo de criptografía que utiliza el valor e no puedan efectuarse.

Según la primera variante y en el caso en que un valor el fuese atribuido a e, los cálculos que utilizan el calor e consisten en:

- elegir un entero aleatorio r;
- calcular un valor d* tal como d*= d+r.(e.d-1);
- realizar una operación privada del algoritmo en la que un valor x se obtiene a partir de un valor y aplicando la relación $x= y^{d*}$ módulo n.

Según la primera variante y en el caso en que un valor el fuese atribuido a e, los cálculos que utilizan el valor e consisten en obtener, al final de una operación privada del algoritmo, un valor x a partir de un valor y, y en verificar si x^e= y módulo n.

Según la segunda variante y en el caso en un valor e_i fuese atribuido a e, los cálculos que utilizan el valor e consisten en obtener, al final de una operación privada del algoritmo, un valor x a partir de un valor y, y en verificar por una parte si x^e=y módulo p, y por otra parte, si x^e=y módulo q.

De preferencia, el conjunto E comprende por lo menos los siguientes valores e_i 3, 17, 2^{16}+1.

El invento concierne igualmente un componente electrónico caracterizado porque comprende medios para aplicar el procedimiento tal como se ha definido anteriormente.

El invento concierne asimismo una tarjeta inteligente que comprende un componente electrónico tal como se define.

El objeto del invento concierne igualmente un procedimiento para la aplicación asegurada de un algoritmo de criptografía con llave pública, dicha llave pública está compuesta de un numero entero n, producto de dos grandes números primos p y q, y de un exponente público e, dicho procedimiento consiste en determinar un conjunto E que comprende un número predeterminado de valores e_i susceptibles de corresponder al valor del exponente público e, los e_i son números primos, caracterizado porque consiste en realizar las siguientes etapas:

a) elegir un valor e_i entre los valores del conjunto E;

b) si $\Phi(p)=\Phi(q)$, testar si el valor e_i elegido verifica la relación: $(1-e_i.d)$ módulo $n< e_i.2^{(\Phi(n)/2)+1}$

o dicha relación simplificada:

$(e_i.d)$ módulo $n< e_i.2^{(\Phi(n)/2)+1}$

con Φ(p), Φ(q) y Φ(n) las funciones dan el número de bits que codifican respectivamente el número p, el número q y el número n;

en caso contrario, en el caso en que p y q están desequilibrados, testar si el valor e_i elegido verifica la relación: (1-e_i.d) módulo n<$e_i 2^{g+1}$ o dicha relación simplificada: (-e_i.d) módulo n< $e_i.2^{g+1}$

con g=max (Φ(p), (Φ(q), si (Φ(p), y (Φ(q), son conocidos o, en caso contrario con g=(Φ(n)/2+t, en donde t designa el factor de desequilibrio o un Terminal en este factor;

c) si la relación de test aplicado en la etapa anterior se verifica, entonces e = e_i, y memorizar e en vista de su utilización en cálculos del dicho algoritmo de criptografía,

- si no fuera así, reiterar las etapas anteriores eligiendo otro valor de e_i en el conjunto E hasta que un valor de e_i pueda atribuirse a e y si no puede atribuirse ningún valor de e_i a e entonces constatar que los cálculos del dicho algoritmo de criptografía que utilizan el valor de e no pueden efectuarse.

El hecho de elegir el orden de los el como aquel de las probabilidades de aparición de los exponentes públicos permite ahorrar tiempo. De este modo, podremos elegir preferiblemente el siguiente orden: e_0-2^{16}+1, e_1=3, e_2=17.

En una variante, tenemos para todos los i, $e_i \leq 2^{16}+1$ y la etapa b) se reemplaza por otra etapa de test que consiste en:

si Φ(p)=Φ(q), testar si el valor e_i elegido verifica la relación: (1-e_i.d) módulo n<$2^{(\Phi(n)/2>+17}$

o dicha relación simplificada: (-e_i.d) módulo n<$2^{(\Phi(n)/2)+17}$

con (Φ(p), (Φ(q), y (Φ(n), las funciones facilitan el número de bits que codifican respectivamente el número p, el número q y el número n;

en caso contrario, en el caso de que p y q estén desequilibrados, testar si el valor e_i elegido verifica la relación: (1-e_i.d) módulo n<2^{g+17} o dicha relación simplificada: (-e_i.d) módulo n<2^{g+17}

con g=max/Φ(p), Φ(q), siΦ(p) y Φ(q) son conocidos o, en el caso contrario, con g=Φ(n)/2+t, donde t designa el factor de desequilibrio o un Terminal en este factor.

En otra variante, la etapa b) se reemplaza por otra etapa de test que consiste en:

testar si el valor e_i elegido verifica la relación según la cual:

los primeros bits de peso fuerte de e_i.d módulo n son nulos;

o dicha relación simplificada en la que:

los primeros bits de peso fuerte de (-e_i. d) módulo n son nulos.

Preferiblemente, el test se efectúa en los 128 primeros bits de peso fuerte.

Según un modo de realización preferido del invento, el algoritmo de criptografía se basa en un algoritmo de tipo RSA en modo estándar.

Según una característica, un valor e_i que se ha atribuido a e, los cálculos utilizan el valor e que consiste en:

- elegir un entero aleatorio r;
- calcular un valor d* tal como d*= d+r(e.d.-1);
- utilizar una operación privada del algoritmo en el que un valor x se obtiene a partir de un valor y aplicando la relación $x=y^{d^*}$ módulo n.

Según otra característica, en la que se atribuye a e un valor e_i, el procedimiento del invento consiste en obtener, al final de una operación privada del algoritmo, un valor x a partir de un valor y los cálculos que emplean el valor e consisten en verificar si x^e= y módulo n.

De preferencia, el conjunto E comprende por lo menos los siguientes valores e_i 3, 17, 2^{16}+1.

El invento concierne todavía a un componente electrónico caracterizado porque comprende medios para aplicar el procedimiento tal como se acaba de definir.

El invento concierne igualmente una tarjeta inteligente que incluye un componente electrónico tal como se ha definido.

Otras características y ventajas del presente invento se comprenderán mejor con la descripción que hacemos a continuación, a título indicativo y en absoluto limitativo.

El presente invento describe así pues distintas técnicas que permiten validar el valor de un exponente público e que no se conoce *a priori*. Estas técnicas pueden aplicarse en cualquier dispositivo o componente electrónico dotado de medios de cálculos criptográficos adecuados, en particular una tarjeta inteligente.

El objeto del invento está basado en la siguiente constatación: ya sea un conjunto E que comprende al menos los siguientes valores de e: $e_0= 2^{16}+1$; $e_1= 3$ y $e_2= 17$; este ejemplo E de valores cubre aprox. un 95% de los valores de los exponentes públicos corrientemente utilizados en los cálculos de los algoritmos de criptografía de tipo RSA.

La primera técnica propuesta por el presente invento, válida para el modo estándar del algoritmo RSA, consiste de manera general en elegir e_0 y en verificar que $e=e_0$; si $e\neq e_0$ entonces se prueba con e_1; y si $e\neq e_1$ entonces se prueba con que e_2.

Puede ser que para una cierta aplicación correspondiente al 5% de otros casos, e no sea igual ni a e_0, ni a e_1, ni a e_2. También designamos más generalmente el valor de e por e_i. Y el método consiste finalmente en elegir un valor e_i entre los el previstos y en verificar que $e= e_i$.

Más particularmente, la primera técnica para encontrar el valor de e, válida para el modo estándar del algoritmo RSA, se basa en el siguiente razonamiento:

En el modo estándar, el algoritmo privado (que aplica una operación de firma o de decodificación de un mensaje) dispone del valor del módulo n y del exponente privado d.

Así pues, de la expresión (1), se deduce que existe un entero k tal como: $e.d= 1 + k\ \Phi(n)$, ya sea: $1-e.d.=-k\ \Phi(n)= - k.\ (n.p-q+1)$.

Si reducimos los dos lados de la expresión módulo n, obtenemos: $1-e.d.= -k(p+q-1)$ (módulo n).

Observando que siempre tenemos $k<e$ cuando e es relativamente pequeño, la expresión precedente puede escribirse de este modo: (1-e.d) módulo $n= k(p+q-1)$. (6)

El lado izquierdo de la ecuación 6 tiene prácticamente el tamaño del módulo n, mientras que el lado derecho tiene su tamaño definido según la siguiente expresión cuando p y q están equilibrados, es decir que tienen el mismo tamaño que $\Phi(p)=\Phi(q)$: $k.\ (p+q-1)<e.2^{(\Phi(n)/2)+1}$

con $\Phi(n)$, $\Phi(p)$, $\Phi(q)$ las funciones dan el número de bits que codifican respectivamente el número n, el número p y el número q.

Cuando p y q no son del mismo tamaño, la función se denomina $g=\max(\Phi(p), \Phi(q))$, es decir la función da el máximo de longitudes de p y q en el caso en que $(\Phi(p), \Phi(q))$ son conocidos; en caso contrario, se toma $g=\Phi(n)/2+t$, en donde t designa el factor de desequilibrio o un terminal en ese factor en el caso contrario. En ese caso en el que p y q están desequilibrados, la fórmula de la expresión a continuación resulta: $k.\ (p+q-1)<e.2^{1+g}$.

En efecto, como $n=p.q$, si p y q están desequilibrados, entonces tenemos la expresión $p+q<2^{(\Phi(n)/2)+1}$; al contrario si p y q están desequilibrados, entonces: $p+q<2^{1+g}$.

De este modo, para todos los e_i posibles en el conjunto E, si $(\Phi(p), (\Phi(q))$, se testa si el valor e_i elegido verifica la siguiente relación predeterminada: (1-e_i.d) módulo $n<e_i.2^{(\Phi(n)/2)+1}$ (7)

en caso contrario, se testa si el valor e_i elegido verifica la siguiente relación predeterminada: (1-e_i.d) módulo $n< e_i.2^{g+1}$ (7')

si la relación predeterminada de test aplicada se verifica, entonces $e= e_i$ y se memoriza e,

en caso contrario, se elige otro valor de e_i en el conjunto E y se reiteran las etapas anteriores.

En una primera variante, el test puede recobrar el valor de e : (1-e_i. d) módulo $n<e_i.2^{(\Phi(n)/2)+1}$ en el que (1-e_i. d) módulo $n<e_i.2^{g+1}$, según que p y q estén equilibrados o no, puede reemplazarse por el siguiente test: (1-e_i.d) módulo n<B, con $B\geq[\max(e_i)]2^{(\Phi(n)/2)+1}$ en el caso en que $\Phi(p) = \Phi(q)$, y $B\geq[\max(e_i)]2^{g+1}$ en caso contrario.

En nuestro ejemplo, tenemos $E=\{2^{16}+1, 3, 17\}$. Así pues, para todos los i, tenemos $e_i\leq 2^{16}+1$ y el test precedente puede simplificarse de la siguiente manera que consiste en verificar si:

(1-e_i.d) módulo n<B, con B=$2^{(\Phi(n/2)+17}$ en el caso en que (Φ(p)= Φ(q),

y (1-e_i.d) módulo n<B, con B=2^{g+17} en caso contrario.

En una segunda variante del test, se pueden simplificar el test precedente verificando si los bits más significativos, p. ej. los 128 bits de peso fuerte, de (1-(e_i.d) módulo n son nulos.

Por último, para esta primera técnica, una última simplificación consiste en determinar la relación predeterminada para el test sobre los el comenzando con la siguiente relación: (-e.d) módulo n= k(p+q-1)-1 en vez de la relación (6).

De este modo, a partir de esta simplificación, obtenemos para las relaciones de test (7, 7'), la siguiente simplificación: (e_i.d) módulo n<e_i.$2^{(\Phi(n)/2)+1}$ si Φ(p)=Φ(q), y (e_i.d) módulo n<e_i.2^{g+1} en caso contrario.

Para la primera variante, se obtiene el siguiente test simplificado: (e_i. d) módulo n<B, con B=$2^{(\Phi(n)/2)+17}$ si Φ(p)= Φ (q), y B=2^{g+17} en caso contrario.

Y, para la segunda variante del test, se obtiene el siguiente test simplificado que consiste en verificar si los primeros bits de peso fuerte de (-e_i.d) módulo n son nulos.

Cualquiera que sea la variante aplicada, en su versión simplificada o no, si el test no se ha verificado para un valor de e_i, se elige otro valor para e_i en el conjunto E hasta que se encuentre una correspondencia.

Si para una u otra de las variantes que conciernen la primera técnica expuesta anteriormente, no existe entre los e_i, un valor tal como e=e_i, entonces queda por constatar que los cálculos del algoritmo de criptografía RSA en modo estándar que hacen intervenir e no pueden efectuarse.

En cambio, cuando el valor de e se ha podido encontrar entre los valores e_i del conjunto de valores predeterminados E, por una u otra de las variantes, podemos verificar entonces cada operación privada (3) del algoritmo de criptografía (que consiste en la decodificación de un mensaje o la generación de una firma) cerciorándose de que el valor x obtenido a partir de un valor y por aplicación de la operación privada verifica la relación x^e. = y módulo n. Si no fuera así, el mensaje decodificado o la firma no se reexpide para evitar cualquier criptoanálisis.

Como ya lo hemos visto, una vez que se conoce e, el procedimiento según el invento puede aplicarse igualmente a una contramedida, principalmente contra los ataques de tipo DPA (y SPA), tal y como se ha descrito más arriba e la descripción. Este método consiste en: elegir un entero aleatorio r; calcular un valor d* tal como d*=d+r.(e.d-1); aplicar una operación privada del algoritmo en la que un valor x se obtiene a partir de un valor y aplicando la relación x=y^{d*} módulo n.

Por último, el presente invento concierne una segunda técnica para descubrir el valor del exponente e entre un conjunto E que comprende un conjunto de valores e_i predeterminados. Como ya; lo veremos, esta técnica se aplica tanto en el caso del modo estándar del algoritmo RSA como en el caso del modo CRT.

Esta técnica consiste más particularmente en mejorar el método propuesto en D1. De este modo, se aplican las siguientes etapas:

a) definir un valor ε = Π ei

ei ε E

tal como ε/e_i, ya sea inferior a Φ(n) para todo e_i perteneciente a E, Φ es la función indicadora de Euler;

b) aplicar el valor ε en un cálculo predeterminado;

c) para cada uno de los e_i testar si el resultado de dicho cálculo predeterminado es igual a un valor ε/e_i:

- si fuera así, entonces atribuir el valor e_i a e y memorizar e en vista de utilizarlo en cálculos del dicho algoritmo de criptografía;
- en caso contrario, constatar que los cálculos de dicho algoritmo de criptografía que utiliza el valor e no pueden efectuarse.

En modo estándar, el cálculo predeterminado de la etapa b) consiste en calcular un valor C tal como:

C = ε.d módulo Φ(n), d es la llave privada correspondiente del algoritmo RSA, en modo estándar, tal como e.d=1 módulo Φ(n).

Por ejemplo, ya sea el conjunto E= $\{e_0=3, e_1=17, e_2=2^{16}+1\}$, entonces $\varepsilon = e_0.e_1.e_2=3.17.(2^{16}+1)$.

De este modo, con C= ε.d módulo $\Phi(n)$:

Si $C=17.(2^{16}+1) = \varepsilon/e_0$ entonces $e=e_0=3$;

Si $C=3.(2^{16}+1)= \varepsilon/e_1$ entonces $e=e_1=17$;

Si $C=3.17=\varepsilon/e_2$ entonces $e=e_2=(2^{16}+1)$;

Por mediación de un solo cálculo modular que permite obtener el valor de C, es posible descubrir el valor del exponente e entre un conjunto E, en función del resultado de este cálculo.

Según una alternativa, el cálculo predeterminado de la etapa b) consiste en calcular un valor C tal como:

C = ε.d módulo $\lambda(n)$, d es la llave privada correspondiente del algoritmo RSA en modo estándar pero calculado en esta alternativa módulo la función de Carmicheal en vez de módulo tiene la función indicadora de Euler, tal como: e.d= 1 módulo $\lambda(n)$ y λ es la función de Carmichael.

En caso de que se hubiera podido encontrar y memorizar efectivamente el valor de e, los cálculos del algoritmo de criptografía en modo estándar que aplican el valor de e consisten en precaver los ataques por falta e implementar una contramedida, principalmente contra los ataques de tipo DPA (y SPA), y son idénticos a aquellos descritos en referencia a la primera técnica.

En una variante, cuando el algoritmo RSA aplicado está en modo CRT, el cálculo predeterminado de la etapa b) consiste

C = $\varepsilon.d_p$ módulo (p-1), d_p es la llave privada correspondiente del algoritmo RSA tal como $e.d_p = 1$ módulo(p-1).

O bien, tal como:

C = $\varepsilon.d_q$ módulo (q-1), d_q es la llave privada correspondiente del algoritmo RSA tal como $e.d_q = 1$ módulo(q-1).

O bien ambas, y en emplear el e que se nos da por lo menos en uno de los dos textos.

En caso de que se hubiera podido encontrar y memorizar efectivamente el valor de e, los cálculos del algoritmo de criptografía en modo CRT que aplican el valor de e consisten en precaver los ataques por falta.

Entonces se puede verificar cada operación privada en modo CRT del algoritmo de criptografía (que consiste en la decodificación de un mensaje o la generación de una firma) cerciorándose de que el valor x obtenido a partir de un valor y por aplicación de la operación privada en modo CRT verifica de una parte, la relación $x^e=y$ módulo p y, por otra parte, la relación $x^e=y$ módulo q.

REIVINDICACIONES

1. Procedimiento para la aplicación asegurada de un algoritmo de criptografía con llave pública, dicha llave pública está compuesta de un número entero n, producto de dos grandes números primos p y q, y de un exponente público e, algoritmo que comprende igualmente una llave privada, dicho procedimiento consiste en determinar un conjunto E, que incluye un número predeterminado de números primos e_i susceptibles de corresponder al valor del exponente público e, **caracterizado** porque comprende las siguientes etapas que consisten en:

a) calcular un valor $\varepsilon = \Pi$ ei

ei ε E

tal como ε/e_i, ya sea inferior a $\Phi(n)$ para todo e_i perteneciente a E, Φ es la función indicadora de Euler;

b) aplicar el valor ε en un cálculo predeterminado haciendo intervenir como producto modular el solo producto modular de ε por dicha llave privada del algoritmo;

c) para cada uno de los e_i, testar si el resultado de dicho cálculo predeterminado es igual a un valor ε/e_i:

- si fuera así, entonces atribuir el valor e_i a e y memorizar e en vista de utilizarlo e cálculos del dicho algoritmo de criptografía;

- en caso contrario, constatar que los cálculos de dicho algoritmo de criptografía que utiliza el valor e no pueden efectuarse.

2. Procedimiento según la reivindicación 1, **caracterizado** porque el algoritmo de criptografía está basado en un algoritmo de tipo RSA en modo estándar.

3. Procedimiento según la reivindicación 2, **caracterizado** porque el cálculo predeterminado de la etapa b) consiste en calcular un valor C:

C = ε.d módulo $\Phi(n)$, d es la llave privada correspondiente del algoritmo RSA, tal como e.d=1 módulo $\Phi(n)$ y Φ es la función indicadora de Euler.

4. Procedimiento según la reivindicación 20, **caracterizado** porque el cálculo predeterminado de la etapa b) consiste en calcular un valor C.

C = ε.d módulo λ (n), d es la llave privada correspondiente del algoritmo RSA tal como e.d=1 módulo $\lambda(n)$ e λ es la función de Carmichael.

5. Procedimiento según la reivindicación 1, **caracterizado** porque el algoritmo de criptografía se basa en un algoritmo de tipo RSA en modo CRT.

6. Procedimiento según la reivindicación 5, **caracterizado** porque el cálculo predeterminado de la etapa b) consiste en calcular un valor C:

C = $\varepsilon.d_p$ módulo(p-1), d_p es la clave privada correspondiente al algoritmo RSA tal como $e.d_p$=1 módulo(p-1).

7. Procedimiento según la reivindicación 5, **caracterizado** porque el cálculo predeterminado de la etapa b) consiste en calcular un valor C:

C = $\varepsilon.d_q$ módulo(q-1), d_q es la llave privada correspondiente al algoritmo RSA tal como $e.d_q$=1 módulo(q-1).

8. Procedimiento según la reivindicación 5, **caracterizado** porque el cálculo predeterminado de la etapa b) consiste en calcular dos valores C_1 y C_2 tales como:

C_1 = $\varepsilon.d_p$ módulo(p-1), d_p es la llave privada correspondiente al algoritmo RSA tal como $e.d_p$=1 módulo(p-1).

C_2 = $\varepsilon.d_q$ módulo(q-1), d_q es la clave privada correspondiente al algoritmo RSA tal como $e.d_p$=1 módulo(q-1). y en que la etapa de test c) consiste para cada e_i, en testar si C_1 y/o C_2 es igual al valor ε/e_i:

- si fuera así, entonces atribuir el valor e_i a e y memorizar e en vista de su utilización en cálculos del dicho algoritmo de criptografía;

- en caso contrario, constatar que los cálculos del dicho algoritmo de criptografía que utiliza el valor e no puedan efectuarse.

9. Procedimiento según cualquiera de las reivindicaciones 3 o 4 y según el cual un valor e_i se ha atribuido a e, **caracterizado** porque los cálculos que utilizan el valor e consisten en:

- elegir un entero aleatorio r;
- calcular un valor d* tal como d*= d+r.(e.d-1);
- utilizar una operación privada del algoritmo en la que un valor x se obtiene a partir de un valor y aplicando la relación $x=y^{d*}$ módulo n.

10. Procedimiento según cualquiera de las reivindicaciones 2 a 4 y según el cual un valor se ha atribuido un valor e_i, **caracterizado** porque consiste en obtener, al final de una operación privada del algoritmo, un valor x a partir de un valor y en que los cálculos que emplean el valor e consisten en verificar si $x^e=y$ módulo n.

11. Procedimiento según cualquiera de las reivindicaciones 5 a 8, y según el cual un valor e_i se ha atribuido a e, **caracterizado** porque consiste en obtener, al final de una operación privada del algoritmo, un valor x a partir de un valor y en que los cálculos que utilizan el valor e consisten en verificar por una parte si $x^e=y$ módulo p y, por otra parte si $x^e=y$ módulo q.

12. Procedimiento según cualquiera de las reivindicaciones anteriores, **caracterizado** porque el conjunto E comprende por lo menos los siguientes valores e_i 3, 17, $2^{16}+1$.

13. Componente electrónico **caracterizado** porque comprende medios para la aplicación del procedimiento según cualquiera de las reivindicaciones anteriores.

14. Tarjeta inteligente que comprende un componente electrónico según la reivindicación 13.

15. Procedimiento para la aplicación asegurada de un de un algoritmo de criptografía con llave pública, dicha llave pública está compuesta de un numero entero n, producto de dos grandes números primos p y q, y de un exponente público e, dicho procedimiento consiste en determinar un conjunto E que comprende un número predeterminado de números primos el susceptibles de corresponder al valor del exponente público e, **caracterizado** porque consiste en realizar las siguientes etapas:

a) elegir un valor e_i entre los valores del conjunto E;

b) si $\delta(p)=\delta(q)$, con $\delta(n)$, $\delta(p)$, $\delta(q)$, funciones dando el número de bits que codifican respectivamente el número n, el número p y el número q, testar si el valor e_i elegido verifica la relación: $(1-e_i.d)$ módulo $n<e_i.2^{(\delta(n)/2)+1}$, o dicha relación simplificada: $(e_i.d)$ módulo $n<e_i.2^{(\delta(n)/2)+1}$

con $\delta(p)$, $\delta(q)$ y $\delta(n)$ las funciones dan el número de bits que codifican respectivamente el número p, el número q y el número n;

c) si la relación de test aplicado en la etapa anterior se verifica, entonces $e=e_i$, y memorizar e en vista de su utilización en cálculos del dicho algoritmo de criptografía,

- si no fuera así, reiterar las etapas anteriores eligiendo otro valor de e_i en el conjunto E hasta que un valor de e_i pueda atribuirse a e y si no puede atribuirse ningún valor de e_i a e entonces constatar que los cálculos del dicho algoritmo de criptografía que utilizan el valor de e no pueden efectuarse.

16. Procedimiento para la aplicación asegurada de un algoritmo de criptografía con llave pública según la reivindicación 15, **caracterizado** porque consiste en realizar la etapa b de la siguiente manera cuando $\delta(p)\#\delta(q)$, caso en que p y q están desequilibrados, etapa que consiste en testar si el valor e_i elegido verifica la relación: $(1-e_i.d)$ módulo $n<e_i.2^{g+1}$, O dicha relación simplificada: $(-e_i.d)$ módulo $n<e_i.2^{g+1}$

con $g=\max/\delta(p)$, $\delta(q)$, si $\delta(p)$ y $\delta(q)$ son conocidos o, en el caso contrario, con $g=\delta(n)/2+t$, donde t designa el factor de desequilibrio o un Terminal en este factor.

17. Procedimiento según la reivindicación 15 o 16, **caracterizado** porque todos los i, $e_i \leq 2^{16}+1$ y porque en la etapa b) se reemplaza por otra etapa de test que consiste en:

si $\delta(p)= \delta(q)$, testar si el valor e_i elegido verifica la relación: $(1-e_i.d)$ módulo $n<2^{(\delta(n)/2)+17}$, o dicha relación simplificada: $(-e_i.d)$ módulo $n<2^{(\delta(n)/2)+17}$

con $\delta(p)$, $\delta(q)$, y $\delta(n)$,las funciones facilitan el número de bits que codifican respectivamente el número p, el número q y el número n;

en caso contrario, en el caso de que p y q estén desequilibrados, testar si el valor e_i elegido verifica la relación: (1-e_i.d) módulo n<2^{g+17}, o dicha relación simplificada: (-e_i.d) módulo n<$^{2g+17}$

con g=max/δ(p), δ(q), si δ(p) y δ(q) son conocidos o, en el caso contrario, con g=δ(n)/2+t, donde t designa el factor de desequilibrio o un Terminal en este factor.

18. Procedimiento según la reivindicación 15 o 16, **caracterizado** porque la etapa b) se reemplaza por otra etapa de test que consiste en:

testar si el valor e_i elegido verifica la relación según la cual:

los primeros bits de peso fuerte de (1-e_i.d) módulo n son nulos;

o dicha relación simplificada en la que:

los primeros bits de peso fuerte de (-e_i.d) módulo n son nulos.

19. Procedimiento según la reivindicación 18, **caracterizado** porque el testo se efectúa en los 128 primeros bits de peso fuerte.

20. Procedimiento según cualquiera de las reivindicaciones 15 a 19, **caracterizado** porque el algoritmo de criptografía se basa e un algoritmo de tipo RSA en modo estándar.

21. Procedimiento según cualquiera de las reivindicaciones 15 a 20, y según el cual un valor e_i se ha atribuido a e, **caracterizado** porque los cálculos que utilizan el valor e consisten en:

- elegir un entero aleatorio r;
- calcular un valor d* tal como d*= d+r(e.d.-1);
- utilizar una operación privada del algoritmo en el que un valor x se obtiene a partir de un valor y aplicando la relación $x=y^{d*}$ módulo n.

22. Procedimiento según cualquiera de las reivindicaciones 15 a 20 y según el cual un valor e_i se ha atribuido a e, **caracterizado** porque consiste en obtener, al final de una operación privada del algoritmo, un valor x a partir de un valor y porque los cálculos que utilizan el valor e consisten en verificar si x^e= y módulo n.

23. Procedimiento según cualquiera de las reivindicaciones 15 a 22 **caracterizado** porque el conjunto E comprende por los menos los siguientes valores e_i 3, 17, 2^{16}+1.

24. Procedimiento según la reivindicación 23, **caracterizado** porque la elección preferencial de los valores e_i entre los valores del conjunto E se efectúa según el orden siguiente: 2^{16}+1, 3, 17.

25. Componente electrónico **caracterizado** porque comprende medios para la aplicación del procedimiento según cualquiera de las reivindicaciones 15 a 24.

26. Tarjeta inteligente que comprende un componente electrónico según la reivindicación 25.