(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 496 740**

(51) Int. CI.:

H04L 9/08 (2006.01)
H04L 9/30 (2006.01)
G09C 1/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **23.04.2010 E 10767167 (9)**

(97) Fecha y número de publicación de la concesión europea: **16.07.2014 EP 2423903**

(54) Título: **Aparato de codificación, aparato de descodificación, método de codificación, método de descodificación, método y programa de seguridad y medio de almacenamiento**

(30) Prioridad:

24.04.2009 JP 2009106021

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
19.09.2014

(73) Titular/es:

NIPPON TELEGRAPH AND TELEPHONE CORPORATION (100.0%)
3-1 Otemachi 2-chome Chiyoda-ku
Tokyo 100-8116, JP

(72) Inventor/es:

FUJISAKI, EIICHIRO y
SUZUKI, KOUTAROU

(74) Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

DESCRIPCIÓN

Aparato de codificación, aparato de descodificación, método de codificación, método de descodificación, método y programa de seguridad y medio de almacenamiento

CAMPO TÉCNICO

La presente invención se refiere a técnicas de seguridad y, más específicamente, a una técnica criptográfica.

ANTECEDENTES DE LA TÉCNICA

Los temas que se han estudiado en el campo de la criptografía incluyen un esquema criptográfico seguro contra ataques de textos codificados seleccionados (Chosen Ciphertext Attacks, CCA). Recientemente, han aumentado los estudios del esquema criptográfico seguro contra CCA, basado en un esquema de codificación (encriptación) basado en la identidad (Identity-Based Encryption, IBE) (consúltese, por ejemplo, la bibliografía no patente 1), que normalmente es seguro contra ataques de textos no codificados seleccionados (Chosen Plaintext Attacks, CPA). En la bibliografía no patente 2, por ejemplo, se propone un método de transformación CHK. En el método de transformación CHK, se usa un esquema de firma de un solo uso ("one-time") para construir el esquema criptográfico seguro contra CCA, basado en el esquema IBE seguro contra CPA. En la bibliografía no patente 3, por ejemplo, se propone un método de transformación BK. En el método de transformación BK, se usan un código de autenticación de mensajes (Message Authentication Code, MAC) y un esquema de compromiso de bit para construir el esquema criptográfico seguro contra CCA, basado en el esquema IBE seguro contra CPA.

En la bibliografía no patente 4, que ha sido publicada después de la fecha de prioridad de la presente solicitud, se propone un esquema de un mecanismo de encapsulación de clave (Key Encapsulation Mechanism, KEM) usando una codificación de predicado con producto interno.

Según la bibliografía no patente 5, la seguridad contra CCA se garantiza partiendo de una codificación simétrica segura contra CCA y un esquema de encapsulación de clave seguro contra CCA.

La bibliografía no patente 6 describe un sistema de codificación de predicado que permite delegar una capacidad de evaluar uno o más predicados en datos codificados en la forma de una capacidad más restrictiva.

BIBLIOGRAFÍA DE LA TÉCNICA ANTERIOR

BIBLIOGRAFÍA NO PATENTE

Bibliografía no patente 1: D. Boneh, M. Franklin, "Identity-Based encryption from the Weil pairing", Crypto 2001, Lecture Notes in Computer Science, vol. 2139, Springer-Verlag, pp 213-229, 2001.

Bibliografía no patente 2: R. Canetti, S. Halevi, J. Katz, "Chosen-Ciphertext Security from Identity-Based Encryption", Proc. Of EUROCRYPT '04, LNCS 3027, pp 207-222, 2004.

Bibliografía no patente 3: D. Boneh, J. Katz, "Improved Efficiency for CCA-Secure Cryptosystems Built Using Identity-Based Encryption", Proc. of CT-RSA '05, LNCS 3376, pp 87-103, 2005.

Bilbiografía no patente 4: T. Okamoto, K. Takashima, "Hierarchical Predictive Encryption for Inner-Products", Proc. Of ASIACRYPT 2009, LNCS 5912, páginas 214-213, 2009.

Bibliografía no patente 5: R. Cramer, V. Shoup, "Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack", SIAM Journal on Computing, Volumen 33, Número 1, páginas 167-226, 2004

Bibliografía no patente 6: E. Shi, B. Waters, "Delegating Capabilities in Predicate Encryption Systems", Proc. Of ICALP 2008, Parte II, páginas 560-578, 2008.

EXPOSICIÓN DE LA INVENCIÓN

PROBLEMAS A RESOLVER POR LA INVENCIÓN

El texto codificado en el método de transformación CHK incluye el texto codificado, una firma de un solo uso del texto codificado, y una clave de verificación de la firma de un solo uso. Por consiguiente, su espacio de texto codificado incluye no sólo el espacio de texto codificado, sino también el espacio de firma de un solo uso y el espacio de la clave de verificación. El texto codificado generado en el método de transformación BK incluye el texto codificado, el MAC y la cadena de compromiso de bit. Por consiguiente, su espacio de texto codificado incluye no sólo el espacio de texto codificado, sino también el espacio de MAC y el espacio de cadena de compromiso de bit. Esto significa que los espacios de texto codificado en el método de transformación CHK y el método de transformación BK incluyen espacios bi-dimensiones adicionales para la seguridad contra CCA. Sin embargo, conforme aumenta el espacio de texto codificado, aumentan la cantidad de cálculos y la cantidad de datos. Por lo tanto, el espacio de texto codificado debería ser pequeño.

En el esquema IBE, la parte que realiza la codificación debe especificar el ID de la parte que realiza la descodificación (desencriptación) antes de proceder a realizar la codificación. Sería conveniente, sin embargo, que la parte que realiza la codificación pueda generar texto codificado sin especificar la parte que realizará la descodificación, y que una parte que satisface unas condiciones determinadas pueda descodificar (desencriptar) el texto codificado.

En vista de lo indicado anteriormente, el esquema criptográfico de la presente invención es muy conveniente y mejora el nivel de seguridad contra CCA sin espacio adicional para la seguridad contra CCA.

MEDIOS PARA RESOLVER LOS PROBLEMAS

El método de codificación según la presente invención incluye al menos las etapas siguientes: primero, se calcula un texto codificado C_1 introduciendo un texto no codificado M y una clave simétrica K a una función de codificación de clave simétrica; se calcula un valor de función de una función H resistente a colisiones introduciendo un valor que contiene un valor correspondiente a la clave simétrica K y un valor correspondiente al primer texto codificado C_1 a la función H; y se calcula un segundo texto codificado $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.b_{n+1} \in G_1^{n+1}$ usando un valor r correspondiente al valor de función de la función H, un valor ρ correspondiente a la clave simétrica K, los elementos v_μ ($\mu = 1, ..., n, n \geq 1$) de un vector n-dimensional $\vec{v} = (v_1, ..., v_n)$, y los vectores base n+1 dimensionales $b_i \in G_1^{n+1}$ ($i = 1, ..., n+1$), compuestos de n+1 elementos de un grupo cíclico G_1.

El procedimiento de descodificación correspondiente al procedimiento de codificación incluye al menos las etapas siguientes: se calcula un valor de función $S^{\sim} = e(C_2, D_1^*) \in G_T$ introduciendo el segundo texto codificado $C_2 \in G_1^{n+1}$ y la información de clave $D_1^* = \alpha.(\sum_{\mu=1}^{n} w_\mu.b_\mu^*) + b_{n+1}^* \in G_2^{n+1}$ a una función e bilineal; y se calcula un valor de función de la función H resistente a colisiones introduciendo un valor correspondiente al valor de función $S^{\sim}$ y el texto codificado C_1 a la función H. Si la información D_1^* de clave es correcta, el valor correspondiente al valor $S^{\sim}$ de función es igual al valor correspondiente a la clave simétrica K, que se introdujo a la función H resistente a colisiones en el procedimiento de codificación. A continuación, se determina si un valor $\rho^{\sim}$, un valor $r^{\sim}$ correspondiente al valor de función de la función H, los elementos v_μ del vector $\vec{v}$ n-dimensional, y los vectores base $b_i \in G_1^{n+1}$ ($i = 1, ..., n+1$) cumplen $C_2 = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$.

El valor r usado en el procedimiento de codificación corresponde al valor de función obtenido al introducir el valor que contiene el valor correspondiente a la clave simétrica K y el valor correspondiente al primer texto codificado C_1 a la función H resistente a colisiones. Por consiguiente, el valor r depende del primer texto codificado C_1 y la clave simétrica K. Incluso si un atacante que no conoce la clave simétrica K puede generar el primer texto codificado C_1 a partir del primer texto C_1' codificado obtenido correctamente, en el que el texto codificado C_1 es diferente del texto codificado C_1', el atacante no puede generar un valor r correcto correspondiente al primer texto codificado C_1 generado. Por lo tanto, la seguridad contra CCA puede ser mejorada mediante la recuperación de un valor $r^{\sim}$ introduciendo el primer texto codificado C_1 y un valor correspondiente al valor de función $S^{\sim}$ a la función H resistente a colisiones y determinando si se cumple o no $C_2 = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$, en el procedimiento de descodificación.

Un valor 2 que corresponde a un valor 1 significa que el valor 2 es igual al valor 1 o el valor 2 depende de al menos el valor 1.

El segundo texto codificado $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$ está formado justo por un espacio n dimensional para un vector n-dimensional $\vec{v} = (v_1, ..., v_n)$, que es el componente básico del segundo texto codificado C_2, y un espacio unidimensional para el valor ρ correspondiente a un valor que contiene la clave simétrica K. Esto significa que el segundo texto codificado C_2 no requiere un espacio usado sólo para mejorar la seguridad contra la CCA.

En un ejemplo del procesamiento de codificación según la presente invención, el primer texto codificado C_1 se calcula introduciendo el texto no codificado M y la clave simétrica K a la función de codificación de clave simétrica; los valores r y ρ se calculan introduciendo la clave simétrica K y el primer texto codificado C_1 a la función H resistente a colisiones; se calcula el valor de función $S = g_T^{\tau.\rho} \in G_T$, en la que τ es una constante diferente a un elemento de identidad aditivo; se calcula un valor OR exclusivo de la clave simétrica K y un valor de función R(S), que es una secuencia binaria obtenida introduciendo el valor de función S a una función R inyectiva, como un tercer texto codificado C_3; y se calcula el segundo texto codificado $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$.

En el procesamiento de descodificación correspondiente al procesamiento de codificación, por ejemplo, se calcula el valor de función $S^{\sim} = e(C_2, D_1^*) \in G_T$ introduciendo el segundo texto codificado $C_2 \in G_1^{n+1}$ y la información de clave D_1^* a la función e bilineal; se calcula un valor OR exclusivo $K^{\sim}$ del tercer texto codificado C_3 y el valor de función $R(S^{\sim})$, que es una secuencia binaria obtenida introduciendo el valor de función $S^{\sim}$ a la función R inyectiva; los valores $r^{\sim}$ y $\rho^{\sim}$ se calculan introduciendo el valor OR exclusivo $K^{\sim}$ y el primer texto codificado C_1 a la función H resistente a colisiones; se calcula un valor de función $C_2^{\sim} = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$; se determina si el valor

de función $C_2^{\sim}$ es igual al segundo texto codificado C_2; cuando la determinación indica que el valor de función $C_2^{\sim}$ es igual al segundo texto codificado C_2, se calcula un valor descodificado $M^{\sim}$ usando el valor OR exclusivo $K^{\sim}$ como una clave simétrica e introduciendo el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a una función de descodificación de clave simétrica.

En otro ejemplo del procedimiento de codificación, se selecciona el valor ρ; se calcula el valor de función $S = g_T^{\tau.\rho} \in G_T$, en la que τ es una constante diferente a un elemento de identidad aditivo; se calcula el primer texto codificado C_1 usando el valor de función R(S) obtenido introduciendo el valor de función S a la función R inyectiva, como la clave simétrica K, e introduciendo la clave simétrica K y el texto no codificado M a la función de codificación de clave simétrica; se calcula el valor r introduciendo el valor de función S y el primer texto codificado C_1 a la función H resistente a colisiones; y se calcula el segundo texto codificado $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$.

En el procesamiento de descodificación correspondiente al procesamiento de codificación, por ejemplo, el valor de función $S^{\sim} = e(C_2, D_1^*) \in G_T$ se calcula introduciendo el segundo texto codificado $C_2 \in G_1^{n+1}$ y la información de clave D_1^* a la función e bilineal; el valor $r^{\sim}$ se calcula introduciendo el valor de función $S^{\sim}$ y el texto codificado C_1 a la función H resistente a colisiones; un valor de función $\lambda = C_2 - r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) \in G_1^{n+1}$ se calcula usando el segundo texto codificado C_2, el valor $r^{\sim}$ y los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$; un valor de función $e(\lambda, D_2^*)$ se calcula introduciendo el valor de función λ, y la información de clave auxiliar $D_2^* = \sum_{\mu=1}^{n} \beta_\mu.b_\mu^*$ a la función e bilineal; se determina si el valor de función $e(\lambda, D_2^*)$ cumple $e(\lambda, D_2^*) = 1 \in G_T$; y si la determinación indica que se cumple $(\lambda, D_2^*) = 1 \in G_T$, el valor descodificado $M^{\sim}$ se calcula usando el valor de función $R(S^{\sim})$ obtenido introduciendo el valor de función $S^{\sim}$ a la función R inyectiva, como la clave simétrica $K^{\sim}$, e introduciendo la clave simétrica $K^{\sim}$ y el primer texto codificado C_1 a la función de descodificación de clave simétrica.

Los ejemplos preferidos de la función R inyectiva y la función H resistente a colisiones son funciones cuasi aleatorias. Esto mejorará el nivel de seguridad.

EFECTOS DE LA INVENCIÓN

Tal como se ha descrito anteriormente, el esquema criptográfico de la presente invención es muy conveniente y proporciona una mejora del nivel de seguridad contra CCA sin espacio adicional para la seguridad contra CCA.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La Figura 1 es una vista que ilustra la estructura de un sistema criptográfico según una primera realización;
La Figura 2 es una vista que ilustra la estructura de un aparato de codificación en la Figura 1;
La Figura 3 es una vista que ilustra la estructura de un aparato de descodificación en la Figura 1;
La Figura 4 es una vista que ilustra la estructura de un aparato de generación de claves en la Figura 1;
La Figura 5 es una vista que ilustra el procesamiento de codificación según la primera realización;
La Figura 6 es una vista que ilustra el procesamiento de generación de información de claves en la primera realización;
La Figura 7 es una vista que ilustra el procesamiento de descodificación según la primera realización;
La Figura 8 es una vista que ilustra la estructura de un sistema criptográfico según una segunda realización;
La Figura 9 es una vista que ilustra la estructura de un aparato de codificación en la Figura 8;
La Figura 10 es una vista que ilustra la estructura de un aparato de descodificación en la Figura 8;
La Figura 11 es una vista que ilustra la estructura de un aparato de generación de claves en la Figura 8;
La Figura 12 es una vista que ilustra el procesamiento de codificación según la segunda realización;
La Figura 13 es una vista que ilustra el procesamiento de generación de información de claves en la segunda realización; y
La Figura 14 es una vista que ilustra el procesamiento de descodificación según la segunda realización.

DESCRIPCIÓN DETALLADA DE LAS REALIZACIONES

Se describirán las realizaciones de la presente invención.

Definiciones

En primer lugar, se describirán los términos y los símbolos que se usarán en las realizaciones.

Matriz: Una matriz representa un arreglo rectangular de elementos de un conjunto en el que se define una operación. No sólo pueden formar la matriz los elementos de un anillo, sino también los elementos de un grupo.

$(\cdot)^T$: $(\cdot)^T$ representa una matriz transpuesta de "·".
$(\cdot)^{-1}$: $(\cdot)^{-1}$ representa una matriz inversa de "·".
^: ^ representa el operador AND lógico.
v: v representa el operador OR lógico.
Z: Z representa un conjunto de números enteros.

k: k representa un parámetro de seguridad (k Є Z, k > 0).

{0, 1}*: {0, 1}* representa una secuencia binaria que tiene una longitud de bits deseada. Un ejemplo de secuencia binaria es una secuencia formada por números enteros 0 y 1. Sin embargo, {0, 1}* no se limita a secuencias formadas por números enteros 0 y 1. {0, 1}* puede ser un campo finito de orden 2 o su campo de extensión.

$\{0, 1\}^{\zeta}$: $\{0, 1\}^{\zeta}$ representa una secuencia binaria que tiene una longitud de bits ζ (ζ Є Z, ζ > 0). Un ejemplo de ello es una secuencia formada por números enteros 0 y 1. Sin embargo, $\{0, 1\}^{\zeta}$ no se limita a secuencias formadas por números enteros 0 y 1. $\{0, 1\}^{\zeta}$ puede ser un campo finito de orden 2 (cuando ζ = 1) o su campo de extensión (cuando ζ > 1).

(+): (+) representa un operador OR exclusivo entre secuencias binarias. Por ejemplo, se cumple lo siguiente: 10110011 (+) 11100001 = 01010010.

F_q: F_q representa un campo finito de orden q, donde q es un número entero igual o mayor que 1. Por ejemplo, el orden q es un número primo o una potencia de un número primo. En otras palabras, el campo F_q finito es un campo de números primos o un campo de extensión sobre el campo de números primos, por ejemplo. Las operaciones en el campo F_q finito de números primos pueden definirse fácilmente como operaciones de módulo con el orden q como módulo, por ejemplo. Las operaciones en el campo F_q de extensión pueden definirse fácilmente como operaciones de módulo con un polinomio irreducible como módulo, por ejemplo. Las estructuras específicas del campo F_q finito se describen, por ejemplo, en la bibliografía de referencia 1, "ISO/IEC 18033-2: Information technology – Security techniques – Encryption algorithms – Part 2: Asymmetric ciphers".

0_F: 0_F representa un elemento identidad aditivo del campo F_q finito.
1_F: 1_F representa un elemento identidad multiplicativo del campo F_q finito.
δ(i, j): δ(i, j) representa una función delta de Kronecker. Cuando i = j, δ(i, j) = 1_F. Cuando i ≠ j, δ(i, j) = 0_F.

E: E representa una curva elíptica sobre el campo F_q finito. E se define como un conjunto que tiene un punto OR específico denominado un punto en el infinito y otros puntos (x, y) de x, y Є F_q que cumplen la siguiente ecuación de Weierstrass sobre coordenadas afines:

$$y^2 + a_1 \cdot x \cdot y + a_3 \cdot y = x^3 + a_2 \cdot x^2 + a_4 \cdot x + a_6 \qquad (1)$$

, en la que a_1, a_2, a_3, a_4, a_6 Є F_q. Puede definirse una operación "+" binaria denominada una adición de curvas elípticas para dos puntos cualesquiera sobre la curva elíptica E, y puede definirse una operación unaria "-" denominada una inversa aditiva para cualquier punto de la curva elíptica E. Es bien sabido que un conjunto finito de puntos racionales sobre la curva elíptica E forma un grupo con respecto a la adición de curvas elípticas. También es bien sabido que puede definirse una operación denominada una multiplicación escalar de curvas elípticas con la adición de curvas elípticas. También es bien conocido un procedimiento de operación específico de las operaciones elípticas, tales como la adición de curvas elípticas, en un ordenador (por ejemplo, véase la bibliografía de referencia 1, la bibliografía de referencia 2, "RFC 5091: Identity-Based Cryptography Standard (IBCS) #1: Supersingular Curve Implementations of the BF and BB1 Cryptosystems", y la bibliografía de referencia 3, Ian F. Blake, Gadiel Seroussi y Nigel P. Smart, "Elliptic Curves in Cryptography", Pearson Education, ISBN 4-89471-431-0).

Un conjunto finito de puntos racionales sobre la curva elíptica E tiene un subgrupo de orden p (p ≥ 1). Por ejemplo, un conjunto finito E[p] de los puntos p-división sobre la curva elíptica E forma un subgrupo de los puntos racionales sobre la curva elíptica, donde #E representa el número de elementos del conjunto finito de los puntos p-división sobre la curva elíptica E y #E es divisible por el número primo p grande. Los puntos p-división sobre la curva elíptica E son puntos A sobre la curva elíptica E que cumplen las multiplicaciones escalares p.A = O de curvas elípticas.

G_1, G_2, G_T: G_1, G_2, G_T representan grupos cíclicos de orden q, respectivamente. Los ejemplos de grupos cíclicos G_1 y G_2 incluyen el conjunto finito E[p] de los puntos p-división sobre la curva E elíptica y sus subgrupos. G_1 puede ser igual a G_2 o G_1 puede ser diferente de G_2. Los ejemplos del grupo cíclico G_T incluyen un conjunto finito que forma un campo de extensión sobre el campo F_q finito. Un ejemplo específico del mismo es un conjunto finito de la raíz p-ésima de 1 sobre la clausura algebraica del campo F_q finito.

En las realizaciones, las operaciones definidas sobre los grupos cíclicos G_1 y G_2 se expresan de manera aditiva, y una operación definida sobre el grupo cíclico G_T se expresa de manera multiplicativa. Más específicamente, χ.Ω Є G_1 para χ Є F_q y Ω Є G_1 significa que la operación definida en el grupo cíclico G_1 es aplicada a Ω Є G_1, χ veces, y

$\Omega_1 + \Omega_2 \in G_1$ para $\Omega_1, \Omega_2 \in G_1$ significa que la operación definida en el grupo cíclico G_1 es aplicada a $\Omega_1 \in G_1$ y $\Omega_2 \in G_1$. De la misma manera, $\chi.\Omega \in G_2$ para $\chi \in F_q$ y $\Omega \in G_2$ significa que la operación definida en el grupo cíclico G_2 se aplica a $\Omega \in G_2$, χ veces, y $\Omega_1 + \Omega_2 \in G_2$ para $\Omega_1, \Omega_2 \in G_2$ significa que la operación definida en el grupo cíclico G_2 se aplica a $\Omega_1 \in G_2$ y $\Omega_2 \in G_2$. Por el contrario, $\Omega^{\chi} \in G_T$ para $\chi \in F_q$ y $\Omega \in G_T$ significa que la operación definida en el grupo cíclico G_T se aplica a $\Omega \in G_T$, χ veces, y $\Omega_1, \Omega_2 \in G_T$ para $\Omega_1, \Omega_2 \in G_T$ significa que la operación definida en el grupo cíclico G_T se aplica a $\Omega_1 \in G_T$ y $\Omega_2 \in G_T$.

G_1^{n+1}: G_1^{n+1} representa un producto directo de (n+1) grupos cíclicos G_1 ($n \geq 1$).
G_2^{n+1}: G_2^{n+1} representa un producto directo de (n+1) grupos cíclicos G_2.
g_1, g_2, g_T: g_1, g_2, g_T representan generadores de los grupos cíclicos G_1, G_2, G_T, respectivamente.
V: V representa un espacio vectorial (n+1)-dimensional formado por el producto directo de los (n+1) grupos cíclicos G_1.
V*: V* representa un espacio vectorial (n+1)-dimensional formado por el producto directo de los (n+1) grupos cíclicos G_2.

e: e representa una función (denominada en adelante "función bilineal") para calcular un mapa bilineal no degenerado que mapea el producto G_1^{n+1} x G_2^{n+1} del producto directo G_1^{n+1} y el producto directo G_2^{n+1} al grupo cíclico G_T. La función e bilineal saca un elemento del grupo cíclico G_T en respuesta a la introducción de (n+1) elementos γ_L (L = 1, ..., n+1) ($n \geq 1$) del grupo cíclico G_1 y (n+1) elementos γ_L^* (L = 1, ..., n+1) del grupo cíclico G_2.

$$e: \quad G_1^{n+1} \times G_2^{n+1} \rightarrow G_T \qquad (2)$$

La función e bilineal satisface las características siguientes:

- Bilinealidad: La siguiente relación se cumple para todos $\Gamma_1 \in G_1^{n+1}$, $\Gamma_2 \in G_2^{n+1}$ y $\nu, \kappa \in F_q$

$$e(\nu \cdot \Gamma_1, \kappa \cdot \Gamma_2) = e(\Gamma_1, \Gamma_2)^{\nu \cdot \kappa} \qquad (3)$$

- No degeneración: Esta función no mapea todo

$$\Gamma_1 \in G_1^{n+1}, \Gamma_2 \in G_2^{n+1} \qquad (4)$$

sobre el elemento identidad del grupo cíclico G_T.

- Computabilidad: Existe un algoritmo para calcular de manera eficiente $e(\Gamma_1, \Gamma 2)$ para todos $\Gamma 1 \in G_1^{n+1}$, $\Gamma_2 \in G_2^{n+1}$.

En las realizaciones, la función e bilineal está formada siguiendo una función bilineal no degenerada que mapea el producto directo G_1 x G_2 de los grupos cíclicos G_1 y G_2 al grupo cíclico G_T.

$$Par: \quad G_1 \times G_2 \rightarrow G_T \qquad (5)$$

La función e bilineal saca un elemento del grupo cíclico G_T en respuesta a un vector (n+1) dimensional (γ_1, ..., γ_{n+1}) de entrada formado por (n+1) elementos γ_L (L = 1, ..., n+1) del grupo cíclico G_1 y un vector (n+1) dimensional (γ_1^*, ..., γ_{n+1}^*) formado por (n+1) elementos γ_L^* (L = 1, ..., n+1) del grupo cíclico G_2.

$$e = \prod_{L=1}^{n+1} Par(\gamma_L, \gamma_L^*) \qquad (6)$$

La función Par bilineal saca un elemento del grupo cíclico G_T en respuesta a un elemento de entrada del grupo cíclico G_1 y un elemento de entrada del grupo cíclico G_2, y satisface las características siguientes:

- Bilinealidad: La siguiente relación se cumple para todos $\Omega_1 \in G_1$, $\Omega_2 \in G_2$, y $\nu, \kappa \in F_q$

$$\mathrm{Par}(\nu\cdot\Omega_1, \kappa\cdot\Omega_2) = \mathrm{Pair}(\Omega_1, \Omega_2)^{\nu\cdot\kappa} \qquad (7)$$

- No degeneración: Esta función no asigna todo

$$\Omega_1 \in G_1, \Omega_2 \in G_2 \qquad (8)$$

sobre el elemento de identidad del grupo cíclico G_T.

- Computabilitidad: Existe un algoritmo para calcular de manera eficiente Par(Ω_1, Ω_2) para todos Ω_1 Є G_1, Ω_2 Є G_2.

Un ejemplo específico de la función Par bilineal es una función para realizar una computación de emparejamiento, tal como el emparejamiento Weil o el emparejamiento Tate. (véase la bibliografía de referencia 4, Alfred. J. Menezes, "Elliptic Curve Public Key Cryptosystems", Kluwer Academic Publishers, ISBN 0-7923-9368-6, pp 61-81, por ejemplo). Dependiendo del tipo de la curva elíptica E, puede usarse una función de emparejamiento modificada e(Ω_1, phi(Ω_2)) (Ω_1 Є G_1, Ω_2 Є G_2) que es una combinación de una función phi predeterminada y la función de computación de emparejamiento, tal como el emparejamiento Tate, puede usarse como la función bilineal Par(véase la bibliografía de referencia 2, por ejemplo). Como el algoritmo para realizar un cálculo de emparejamiento en un ordenador, puede usarse el algoritmo Miller (véase la bibliografía de referencia 5, V. S. Miller, "Short Programs for Functions on Curves", 1986, http://crypto.stanford.edu/miller/miller.pdf) o algún otro algoritmo conocido. Los métodos de formación de un grupo cíclico y una curva elíptica para el cálculo de emparejamiento efectivo son bien conocidos (por ejemplo, véase la bibliografía de referencia 2, descrita anteriormente, la bibliografía de referencia 6, A. Miyaji, M. Nakabayashi y S. Takano, "New Explicit Conditions of Elliptic Curve Traces for FR-Reduction", IEICE Trans. Fundamentals, vol. E84-A, Nº 5, pp 1234-1243, Mayo de 2001, la bibliografía de referencia 7, P. S. L. M. Barreto, B. Lynn, M. Scott, "Constructing Elliptic Curves with Prescribed Embedding Degrees", Proc. SCN '2002, LNCS 2576, pp 259-267, Springer-Verlag. 2003, y la bibliografía de referencia 8, R. Dupont, A. Enge, F. Morain, "Building Curves with Arbitrary Small MOV Degree over Finite Primer Fields", http://eprint.iacr.org/2002/094/).

ai (i = 1, ..., n+1): a_i (i = 1, ..., n+1) representan vectores base (n+1) dimensionales que tienen (n+1) elementos del grupo cíclico G_1 como elementos. Por ejemplo, cada uno de los vectores base a_i es el vector (n+1) dimensional en el que el i-ésimo elemento es κ_1-g_1 Є G_1 y el resto de los elementos son elementos identidad (cada uno de los cuales se expresa de manera aditiva como "0") del grupo cíclico G_1. En ese caso, los elementos de los vectores base (n+1)-dimensionales a_i (i = 1, ..., n+1) pueden enumerarse de la siguiente manera:

$$\begin{aligned} a_1 &= (\kappa_1\cdot g_1, 0, 0, ..., 0) \\ a_2 &= (0, \kappa_1\cdot g_1, 0, ..., 0) \\ &..... \\ a_{n+1} &= (0, 0, 0, ..., \kappa_1\cdot g_1) \end{aligned} \qquad (9)$$

Aquí, κ_1 es una constante que es un elemento del campo F_q finito diferente al elemento de identidad aditivo 0_F. Un ejemplo de κ_1 Є F_q es $\kappa_1 = 1_F$. Los vectores base a_i son bases ortogonales. Cada vector (n+1) dimensional que tiene (n+1) elementos del grupo cíclico G_1 como elementos está expresado mediante una combinación lineal de los vectores base (n+1) dimensionales a_i (i = 1, ..., n+1). Es decir, los vectores base (n+1) dimensionales a_i generan el espacio vectorial V, descrito anteriormente.

a_i* (i = 1, ..., n+1): a_i* (i = 1, ..., n+1) representa vectores base (n+1) dimensionales que tienen (n+1) elementos del grupo cíclico G_2 como elementos. Por ejemplo, cada uno de los vectores base a_i* es el vector (n+1) dimensional en el que el i-ésimo elemento es κ_2-g_2 Є G_2 y el resto de elementos son elementos identidad (cada uno de los cuales se expresa de manera aditiva como "0") del grupo cíclico G_2. En ese caso, los elementos de los vectores base (n+1) dimensionales a_i* (i = 1, ..., n+1) pueden enumerarse de la siguiente manera:

$$a_1^* = (\kappa_2 \cdot g_2, 0, 0, ..., 0)$$
$$a_2^* = (0, \kappa_2 \cdot g_2, 0, ..., 0) \quad (10)$$
$$.....$$
$$a_{n+1}^* = (0, 0, 0, ..., \kappa_2 \cdot g_2)$$

Aquí, κ_2 es una constante que es un elemento del campo F_q finito diferente del elemento identidad aditivo 0_F. Un ejemplo de $\kappa_2 \in F_q$ es $\kappa_2 = 1_F$. Los vectores base a_i^* son bases ortogonales. Cada vector (n+1) dimensional que tiene (n+1) elementos del grupo cíclico G_2 como elementos se expresa mediante una combinación lineal de vectores base (n+1) dimensionales a_i^* (i = 1, ..., n+1). Es decir, los vectores base (n+1) dimensionales a_i^* generan el espacio vectorial V*, descrito anteriormente.

Los vectores base a_i y los vectores base a_i^* cumplen la siguiente expresión para un elemento $\tau = \kappa_1.\kappa_2$ del campo F_q finito diferente de 0_F:

$$e(a_i, a_j^*) = g_T^{\tau \cdot \delta(i, j)} \quad (11)$$

Cuando i = j, la siguiente expresión se cumple a partir de las expresiones (6) y (7).

$$e(a_i, a_j^*) = Par(\kappa_1.g_1, \kappa_2.g_2).Par(0, 0). \ldots Par(0, 0)$$
$$= Par(g_1, g_2)^{\kappa 2.\kappa 2}.Par(g_1, g_2)^{0.0}. \ldots .Par(g_1, g_2)^{0.0}$$
$$= Par(g_1, g_2)^{\kappa 1.\kappa 2} = g_T^{\zeta}$$

Cuando i ≠ j, $e(a_i, a_j^*)$ no incluye Par $(\kappa_1.g_1, \kappa_2.g_2)$ y es el producto de Par $(\kappa_1.g_1, 0)$, el Par$(0, \kappa_2.g_2)$ y el Par (0, 0). Además, la siguiente expresión se satisface a partir de la expresión (7).

$$Par(g_1, 0) = Par(0, g_2) = Par(g_1, g_2)^0$$

Por lo tanto, cuando i ≠ j, se cumple la siguiente expresión.

$$e(a_i, a_j^*) = e(g_1, g_2)^0 = gT^0$$

Especialmente cuando $\tau = \kappa_1.\kappa_2 = 1_F$ (por ejemplo, $\kappa_1 = \kappa_2 = 1_F$), se satisface la siguiente expresión.

$$e(a_i, a_j^*) = g_T^{\delta(i, j)} \quad (12)$$

Aquí, $g_T^0 = 1$ es el elemento identidad del grupo cíclico G_T, y $g_T^1 = g_T$ es un generador del grupo cíclico G_T. En ese caso, los vectores base a_i y los vectores base a_i^* son bases ortogonales normales duales, y el espacio vectorial V y el espacio vectorial V* son un espacio vectorial dual en el que puede definirse una aplicación bilineal (espacio vectorial de emparejamiento dual (DPVS)).

A: "A" representa una matriz de (n+1) filas por (n+1) columnas que tiene los vectores base a_i (i = 1, ..., n+1) como elementos.

Cuando los vectores base a_i (i = 1, ..., n+1) se expresan mediante la expresión (9), por ejemplo, la matriz A es la siguiente:

$$A = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_{n+1} \end{pmatrix} = \begin{pmatrix} \kappa_1 \cdot g_1 & 0 & \cdots & 0 \\ 0 & \kappa_1 \cdot g_1 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \kappa_1 \cdot g_1 \end{pmatrix} \qquad (13)$$

A*: "A*" representa una matriz de (n+1) filas por (n+1) columnas que tiene los vectores base a_i* (i = 1, ..., n+1) como elementos.

Cuando los vectores base a_i* (i = 1, ..., n+1) se expresan mediante la expresión (10), por ejemplo, la matriz A* es la siguiente:

$$A^* = \begin{pmatrix} a_1^* \\ a_2^* \\ \vdots \\ a_{n+1}^* \end{pmatrix} = \begin{pmatrix} \kappa_2 \cdot g_2 & 0 & \cdots & 0 \\ 0 & \kappa_2 \cdot g_2 & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & \kappa_2 \cdot g_2 \end{pmatrix} \qquad (14)$$

X: X representa una matriz de (n+1) filas por (n+1) columnas que tiene elementos del campo F_q finito como entradas. La matriz X se usa para la transformación de coordenadas de los vectores base a_i. La matriz X es como sigue:

$$X = \begin{pmatrix} \chi_{1,1} & \chi_{1,2} & \cdots & \chi_{1,n+1} \\ \chi_{2,1} & \chi_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \chi_{n+1,1} & \chi_{n+1,2} & \cdots & \chi_{n+1,n+1} \end{pmatrix} \qquad (15)$$

, en la que cada $\chi_{i,j} \in F_q$ es la entrada en la i-ésima fila y la j-ésima columna (i = 1, ..., n+1, j = 1, .. ., n+1) de la matriz X.

Aquí, cada entrada $\chi_{i,j}$ de la matriz X se denomina coeficiente de transformación.

X*: X* representa la matriz transpuesta de la matriz inversa de la matriz X. $X^* = (X^{-1})^T$. La matriz X* se usa para la transformación de coordenadas de los vectores base a_i*. La matriz X* se expresa como sigue:

$$X^* = \begin{pmatrix} \chi_{1,1}^* & \chi_{1,2}^* & \cdots & \chi_{1,n+1}^* \\ \chi_{2,1}^* & \chi_{2,2}^* & & \vdots \\ \vdots & & \ddots & \vdots \\ \chi_{n+1,1}^* & \chi_{n+1,2}^* & \cdots & \chi_{n+1,n+1}^* \end{pmatrix} \qquad (16)$$

en la que la entrada en la i-ésima fila y la j-ésima columna de la matriz X* es $\chi_{i,j}^* \in F_q$.

Aquí, cada entrada $\chi_{i,j}$* de la matriz X * se denomina coeficiente de transformación.

En ese caso, se cumple $X.(X^*)^T = I$, en la que "I" representa una matriz unidad de (n+1) filas y (n+1) columnas.

En otras palabras, la matriz unidad se expresa como sigue.

$$I = \begin{pmatrix} 1_F & 0_F & \cdots & 0_F \\ 0_F & 1_F & & \vdots \\ \vdots & & \ddots & 0_F \\ 0_F & 0_F & \cdots & 1_F \end{pmatrix} \qquad (17)$$

Se cumple la siguiente expresión.

$$\begin{pmatrix} \chi_{1,1} & \chi_{1,2} & \cdots & \chi_{1,n+1} \\ \chi_{2,1} & \chi_{2,2} & & \vdots \\ \vdots & & \ddots & \vdots \\ \chi_{n+1,1} & \chi_{n+1,2} & \cdots & \chi_{n+1,n+1} \end{pmatrix} \cdot \begin{pmatrix} \chi_{1,1}^* & \chi_{2,1}^* & \cdots & \chi_{n+1,1}^* \\ \chi_{1,2}^* & \chi_{2,2}^* & & \vdots \\ \vdots & & \ddots & \vdots \\ \chi_{1,n+1}^* & \chi_{2,n+1}^* & \cdots & \chi_{n+1,n+1}^* \end{pmatrix} \qquad (18)$$

$$= \begin{pmatrix} 1_F & 0_F & \cdots & 0_F \\ 0_F & 1_F & & \vdots \\ \vdots & & \ddots & 0_F \\ 0_F & 0_F & \cdots & 1_F \end{pmatrix}$$

Aquí, se definirán a continuación vectores (n+1) dimensionales.

$$\vec{\chi_i} = (\chi_{i,1}, \ldots, \chi_{i,n+1}) \qquad (19)$$

$$\vec{\chi_j} = (\chi_{j,1}^*, \ldots, \chi_{j,n+1}^*) \qquad (20)$$

El producto interno de los vectores (n+1) dimensionales $\vec{\chi_i}$ y $\vec{\chi_j}^*$ satisface la siguiente expresión a partir de la expresión (18).

$$\vec{\chi_i} \cdot \vec{\chi_j}^* = \delta(i, j) \qquad (21)$$

b_i: b_i representan vectores base (n+1) dimensionales que tienen (n+1) elementos del grupo cíclico G_1 como elementos. Los vectores base b_i se obtienen mediante transformación de coordenadas de los vectores base a_i (i = 1, ..., n+1) con la matriz X. Es decir, los vectores base b_i se obtienen mediante el cálculo siguiente.

$$b_i = \sum_{j=1}^{n+1} \chi_{i,j} \cdot a_j \qquad (22)$$

Cuando los vectores base a_j (j = 1, ..., n+1) se expresan mediante la expresión (9), cada elemento de los vectores base b_i se muestra a continuación.

$$b_i = (\chi_{i,1}\cdot\kappa_1\cdot g_1,\ \chi_{i,2}\cdot\kappa_1\cdot g_1,\ \ldots,\ \chi_{i,n+1}\cdot\kappa_1\cdot g_1) \quad (23)$$

Cada vector (n+1) dimensional que tiene (n+1) elementos del grupo cíclico G_1 como elementos se expresa mediante una combinación lineal de los vectores básicos (n+1) dimensionales b_i (i = 1, ..., n+1). Es decir, los vectores base (n+1) dimensionales b_i generan el espacio vectorial V, descrito anteriormente.

b_i*: b_i* representa vectores base (n+1) dimensionales que tienen (n+1) elementos del grupo cíclico G_2 como elementos. La vectores base b_i* se obtienen mediante transformación de coordenadas de los vectores base a_i* (i = 1, ..., n+1) con la matriz X*. Es decir, los vectores base b_i* se obtienen mediante el cálculo siguiente.

$$b_i^* = \sum_{j=1}^{n+1} \chi_{i,j}^* \cdot a_j^* \quad (24)$$

Cuando los vectores base a_j (j = 1, ..., n+1) se expresan mediante la expresión (10), cada elemento de los vectores base b_i* se muestran a continuación.

$$b_i^* = (\chi_{i,1}^*\cdot\kappa_2\cdot g_2,\ \chi_{i,2}\cdot\kappa_2\cdot g_2,\ \ldots,\ \chi_{i,n+1}^*\cdot\kappa_2\cdot g_2) \quad (25)$$

Cada vector (n+1) dimensional que tiene (n+1) elementos del grupo cíclico G_2 como elementos se expresa mediante una combinación lineal de vectores base (n+1) dimensionales b_i* (i = 1,. .., n+1). Es decir, los vectores base (n+1) dimensionales b_i* generan el espacio vectorial V*, descrito anteriormente.

Los vectores base b_i y los vectores base b_i* cumplen la siguiente expresión para los elementos $\tau = \kappa_1.\kappa_2$ del campo F_q finito diferente de 0_F:

$$e(b_i, b_j^*) = g_T^{\tau\cdot\delta(i,j)} \quad (26)$$

La siguiente expresión se cumple a partir de las expresiones (6), (21), (23) y (25).

$$\begin{aligned} e(b_i, b_j^*) &= \prod_{L=1}^{n+1} \mathrm{Par}(\chi_{i,L}\cdot\kappa_1\cdot g_1, \chi_{j,L}^*\cdot\kappa_2\cdot g_2) \\ &= \mathrm{Par}(\chi_{i,1}\cdot\kappa_1\cdot g_1, \chi_{j,1}^*\cdot\kappa_2\cdot g_2)\cdot\ldots\cdot(\chi_{i,n}\cdot\kappa_1\cdot g_1, \chi_{j,n}^*\cdot\kappa_2\cdot g_2) \\ &\quad\times \mathrm{Par}(\chi_{j,n+1}\cdot\kappa_1\cdot g_1, \chi_{j,n+1}^*\cdot\kappa_2\cdot g_2) \\ &= \mathrm{Par}(g_1,g_2)^{\kappa_1\cdot\kappa_2\cdot\chi_{i,1}\cdot\chi_{j,1}^*}\cdot\ldots\cdot\mathrm{Par}(g_1,g_2)^{\kappa_1\cdot\kappa_2\cdot\chi_{i,2}\cdot\chi_{j,2}^*} \\ &\quad\times \mathrm{Par}(g_1,g_2)^{\kappa_1\cdot\kappa_2\cdot\chi_{i,n+1}\cdot\chi_{j,n+1}^*} \\ &= \mathrm{Par}(g_1,g_2)^{\kappa_1\cdot\kappa_2(\chi_{i,1}\cdot\chi_{j,1}^*+\chi_{i,2}\cdot\chi_{j,2}^*+\ldots+\chi_{i,n+1}\cdot\chi_{j,n+1}^*)} \\ &= \mathrm{Par}(g_1,g_2)^{\kappa_1\cdot\kappa_2\cdot\vec{\chi_i}\cdot\vec{\chi_j}^*} \\ &= \mathrm{Par}(g_1,g_2)^{\tau\cdot\delta(i,j)} = g_T^{\tau\cdot\delta(i,j)} \end{aligned}$$

Especialmente cuando $\tau = \kappa_1.\ \kappa_2 = 1_F$ (por ejemplo, $\kappa_1 = \kappa_2 = 1_F$), se satisface la expresión siguiente.

$$e(b_i, b_j^*) = g_T^{\delta(i,j)} \quad (27)$$

En ese caso, los vectores base b_i y los vectores base b_i^* son la base ortogonal normal dual de un espacio vectorial de emparejamiento dual (el espacio vectorial V y el espacio vectorial V*).

Siempre y cuando se cumpla la expresión (26), pueden usarse vectores base ai y a_i^* diferentes a los mostrados en las expresiones (9) y (10) como ejemplos, y vectores base b_i y b_i^* diferentes a los mostrados en las expresiones (22) y (24) como ejemplos.

B: B representa una matriz de (n+1) filas por (n+1) columnas que tiene los vectores base b_i (i = 1, ..., n+1) como elementos.

Se cumple B = X·A. Cuando los vectores base b_i se expresan mediante la expresión (23), por ejemplo, la matriz B es la siguiente:

$$B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n+1} \end{pmatrix} = \begin{pmatrix} \chi_{1,1} \cdot \kappa_1 \cdot g_1 & \chi_{1,2} \cdot \kappa_1 \cdot g_1 & \cdots & \chi_{1,n+1} \cdot \kappa_1 \cdot g_1 \\ \chi_{2,1} \cdot \kappa_1 \cdot g_1 & \chi_{2,2} \cdot \kappa_1 \cdot g_1 & & \vdots \\ \vdots & & \ddots & \chi_{n,n+1} \cdot \kappa_1 \cdot g_1 \\ \chi_{n+1,1} \cdot \kappa_1 \cdot g_1 & \cdots & \chi_{n+1,n} \cdot \kappa_1 \cdot g_1 & \chi_{n+1,n+1} \cdot \kappa_1 \cdot g_1 \end{pmatrix} \quad (28)$$

B*: B* representa una matriz de (n+1) filas por (n+1) columnas que tiene los vectores base b_i^* (i = 1, ..., n+1) como elementos.

Se cumple B* = X*·A*. Cuando los vectores base b_i^* (i = 1, ..., n+1) se expresan mediante la expresión (25), por ejemplo, la matriz B* es como se indica a continuación:

$$B^* = \begin{pmatrix} b_1^* \\ b_2^* \\ \vdots \\ b_{n+1}^* \end{pmatrix} = \begin{pmatrix} \chi_{1,1}^* \cdot \kappa_2 \cdot g_2 & \chi_{1,2}^* \cdot \kappa_2 \cdot g_2 & \cdots & \chi_{1,n+1}^* \cdot \kappa_2 \cdot g_2 \\ \chi_{2,1}^* \cdot \kappa_2 \cdot g_2 & \chi_{2,2}^* \cdot \kappa_2 \cdot g_2 & & \vdots \\ \vdots & & \ddots & \chi_{n,n+1}^* \cdot \kappa_2 \cdot g_2 \\ \chi_{n+1,1}^* \cdot \kappa_2 \cdot g_2 & \cdots & \chi_{n+1,n}^* \cdot \kappa_2 \cdot g_2 & \chi_{n+1,n+1}^* \cdot \kappa_2 \cdot g_2 \end{pmatrix} \quad (29)$$

$W^{\rightarrow}$: $W^{\rightarrow}$ representa un vector n-dimensional que tiene elementos del campo F_q finito como elementos.

$$w^{\rightarrow} = (w_1, ..., w_n) \in F_q^n \quad (30)$$

w_μ: w_μ representa el μ-ésimo elemento (μ = 1, ..., n) del vector n-dimensional.

$\vec{v}$: $\vec{v}$ representa un vector n-dimensional que tiene elementos del campo F_q finito como elementos.

$$\vec{v} = (v_1, ..., v_n) \in F_q^n \qquad (31)$$

v_μ: v_μ representa el μ-ésimo elemento (μ = 1, ..., n) del vector n-dimensional.

Función resistente a colisiones: La función resistente a colisiones representa una función h que satisface la condición siguiente con respecto a un parámetro k de seguridad suficientemente más grande, o una función considerada como la función h.

$$Pr[A(h) = (x, y) | h(x) = h(y) \wedge x \neq y] < \varepsilon(k) \qquad (32)$$

Aquí, Pr [·] es la probabilidad de que suceda el evento [·]; A(h) es un algoritmo probabilístico de tiempo polinómico para calcular x e y (x ≠ y) que cumplen h (x) = h(y) para la función h; y Є(k) es una ecuación polinómica del parámetro de seguridad k. Un ejemplo de una función resistente a colisiones es una función hash, tal como la función hash criptográfica descrita en la bibliografía de referencia 1.

Función inyectiva: La función inyectiva representa una función en la que cada elemento de su codominio es mapeada desde cada uno de los elementos de su dominio, o una función considerada como tal. Un ejemplo de la función inyectiva es una función hash, tal como la función de derivación de claves (Key Derivation Function, KDF) descrita en la bibliografía de referencia 1.

Función quasi aleatoria: La función cuasi aleatoria representa una función que pertenece a un subconjunto φ_ζ de un conjunto Φ_ζ, o una función considerada como tal, en la que ningún algoritmo probabilístico de tiempo polinomial puede distinguir el subconjunto φ_ζ del conjunto Φ_ζ; y el conjunto Φ_ζ es un conjunto de todas las funciones que mapean un elemento de un conjunto {0, 1}* a un elemento del conjunto $\{0, 1\}^\zeta$. Un ejemplo de la función cuasi aleatoria es una función hash, tal como la descrita anteriormente.

H_1: H_1 representa una función resistente a colisiones que saca dos elementos (Ψ1, Ψ2) Є F_q x F_q del campo F_q finito en respuesta a la introducción de dos secuencias binarias (ω_1, ω_2) Є $\{0, 1\}^k$ x {0, 1}*.

$$H_1: \quad \{0, 1\}^k \times \{0, 1\}^* \rightarrow F_q \times F_q \qquad (33)$$

Un ejemplo de la función H_1 es una función que saca dos elementos (Ψ1, Ψ2) Є F_q x F_q del campo F_q finito en respuesta a los bits conectados $\omega_1||\omega_2$ de entrada ω_1 y ω_2. Esta función puede incluir una función hash, tal como la función hash criptográfica, una función de conversión secuencia-binaria-a-número-entero (conversión cadena de octetos/entero), y una función de conversión de secuencia-binaria-a-elemento-campo-finito (conversión de cadena de octetos y número entero/campo finito), que se describen en la bibliografía de referencia 1. Preferiblemente, la función H_1 es una función cuasi aleatoria.

H_2: H_2 representa una función resistente a colisiones que saca un elemento Ψ Є F_q del campo F_q finito en respuesta a un elemento de entrada del grupo cíclico G_T y una secuencia binaria de entrada (ξ, ω_2) Є G_T * {0, 1}*.

$$H_2: \quad G_T \times \{0, 1\}^* \rightarrow F_q \qquad (34)$$

Un ejemplo de la función H_2 es una función que acepta un elemento ξ Є G_T del grupo cíclico G_T y una secuencia binaria ω_2 Є {0, 1}*, introduce el elemento ξ Є G_T del grupo cíclico G_T a una función de conversión elemento-campo-finito-a-secuencia-binaria (conversión de cadena de octetos y número entero/campo finito) descrita en la bibliografía de referencia 1 para obtener una secuencia binaria, aplica una función hash, tal como la función hash criptográfica descrita en la bibliografía de referencia 1, a los bits conectados de la secuencia binaria obtenida y la secuencia binaria ω_2 Є {0, 1}*, realiza la función de conversión secuencia-binaria-a-elemento-campo-finito (conversión de cadena de octetos y número entero/campo finito), y saca un elemento Ψ Є F_q del campo F_q finito. Para garantizar la mayor seguridad, es preferible que la función H_2 sea la función cuasi aleatoria.

R: R representa una función inyectiva que saca una secuencia binaria $\Omega \in \{0, 1\}^k$ en respuesta a un elemento de entrada $\xi \in G_T$ del grupo cíclico G_T.

$$R: \quad G_T \rightarrow \{0, 1\}^* \qquad (35)$$

Un ejemplo de la función R inyectiva es una función que recibe un elemento $\xi \in G_T$ del grupo cíclico G_T, realiza cálculos con la función de conversión elemento-campo-finito-a-secuencia-binaria (conversión cadena de octetos y número entero/campo finito) y, a continuación, con una función hash como la KDF (función de derivación de claves) descrita en la bibliografía de referencia 1, y saca una secuencia binaria $\Omega \in \{0, 1\}^k$. Para garantizar la mayor seguridad, es preferible que la función R sea la función resistente a colisiones, y es más preferible que la función R sea la función cuasi aleatoria.

Enc: Enc representa una función de codificación de clave simétrica que indica el procesamiento de codificación de un criptosistema de clave simétrica. Ejemplos de sistemas criptográficos de clave simétrica son Camellia y AES.

$Enc_k(M)$: $Enc_k(M)$ representa un texto codificado obtenido mediante la codificación de un texto no codificado M mediante la función de codificación Enc de clave simétrica con una clave simétrica K.

Dec: Dec representa una función de descodificación de clave simétrica que indica el procesamiento de descodificación del criptosistema de clave simétrica.

$Dec_k(C)$: $Dec_k(C)$ representa un resultado de descodificación obtenido al descodificar un texto codificado C mediante la función de descodificación Dec de clave simétrica con la clave simétrica K.

Codificación de predicado con producto interno
A continuación se describirá el esquema básico de codificación de predicado con producto interno.

Codificación de predicado
En la codificación de predicado (a veces denominada codificación mediante función), un texto codificado puede ser descodificado cuando una combinación de información de atributo e información de predicado hace que se cumpla una fórmula lógica predeterminada. Una de entre la información de atributo y la información de predicado está incluida en el texto codificado y la otra está incluida en la información de clave. La codificación de predicado convencional, por ejemplo, se describe en la bibliografía de referencia 9, Jonathan Katz, Amit Sahai y Brent Waters, "Predicate Encryption Supporting Disjuntions, Polynomial Equations, and Inner Products", uno de los cuatro artículos de Eurocrypt 2008 invitados por el Journal of Criptology.

Codificación de predicado con producto interno
En la codificación de predicado con producto interno, un texto codificado puede ser descodificado cuando el producto interno de la información de atributo y la información de predicado, que son vectores, es cero. En la codificación de predicado con producto interno, un producto interno igual a cero es equivalente a la fórmula lógica de la verdad.

Relación entre la fórmula lógica y el polinomio
En la codificación predicado con producto interno, la fórmula lógica formada por uno o varios OR lógicos y/o uno o varios AND lógicos se expresa mediante un polinomio.

El operador lógico OR $(x = \eta_1) \vee (x = \eta_2)$ de una proposición 1 que indica que x es η_1 y una proposición 2 que indica que x es η_2 se expresa mediante el polinomio siguiente.

$$(x - \eta_1)\cdot(x - \eta_2) \qquad (36)$$

A continuación, las relaciones entre los valores de la verdad y los valores de la función de la expresión (36) se muestran en la tabla siguiente.

Tabla 1

Proposición 1 $(x = \eta_1)$	Proposición 2 $(x = \eta_2)$	OR lógico $(x = \eta_1) \vee (x = \eta_2)$	Valor de función $(x = \eta_1)\cdot(x = \eta_2)$
Verdadero	Verdadero	Verdadero	0
Verdadero	Falso	Verdadero	0
Falso	Verdadero	Verdadero	0
Falso	Falso	Falso	Diferente de 0

Tal como se entiende a partir de la Tabla 1, cuando el OR lógico $(x = \eta_1) \vee (x = \eta_2)$ es verdadero, el valor de la función de la expresión (36) es cero; y cuando el OR lógico $(x = \eta_1) \vee (x = \eta_2)$ es falso, el valor de la función de la expresión (36) es un valor distinto de cero. En otras palabras, el OR lógico $(x = \eta_1) \vee (x = \eta_2)$ de verdadero es equivalente al valor de función de cero en la expresión (36). Por lo tanto, el operador lógico OR puede expresarse mediante la expresión (36).

El AND lógico $(x = \eta_1) \wedge (x = \eta_2)$ de la proposición 1 que indica que x es η_1 y la proposición 2 que indica que x es η_2 se expresa mediante el polinomio siguiente

$$\iota_1\cdot(x - \eta_1) + \iota_2\cdot(x - \eta_2) \qquad (37)$$

en el que ι_1 y ι_2 son números aleatorios. A continuación, las relaciones entre los valores de la verdad y los valores de la función de la expresión (37) se muestran en la tabla siguiente.

Tabla 2

Proposición 1 $(x = \eta_1)$	Proposición 2 $(x = \eta_2)$	AND lógico $(x = \eta_1) \wedge (x = \eta_2)$	Valor de función $\iota_1\cdot(x = \eta_1) + \iota_2\cdot(x = \eta_2)$
Verdadero	Verdadero	Verdadero	0
Verdadero	Falso	Falso	Diferente de 0
Falso	Verdadero	Falso	Diferente de 0
Falso	Falso	Falso	Diferente de 0

Tal como se entiende a partir de la Tabla 2, cuando el AND lógico $(x = \eta_1) \wedge (x = \eta_2)$ es verdadero, el valor de función de la expresión (37) es cero; y cuando el AND lógico $(x = \eta_1) \wedge (x = \eta_2)$ es falso, el valor de función de la expresión (37) es un valor distinto de cero. En otras palabras, un AND lógico $(x = \eta_1) \wedge (x = \eta_2)$ verdadero es equivalente a un valor de función de cero en la expresión (37). Por lo tanto, el AND lógico puede ser expresado mediante la expresión (37).

Tal como se ha descrito anteriormente, usando las expresiones (36) y (37), una fórmula lógica formada por uno varios OR lógicos y/o uno o varios AND lógicos puede ser expresada mediante un polinomio f(x). A continuación, se mostrará un ejemplo.

$$\text{fórmula lógica} \quad : \quad \{(x = \eta_1) \vee (x = \eta_2) \vee (x = \eta_3)\} \wedge (x = \eta_4) \wedge (x = \eta_5)$$

$$\text{Polinomio} : \quad f(x) = \iota_1\cdot\{(x - \eta_1)\cdot(x - \eta_2)\cdot(x - \eta_3)\} + \iota_2\cdot(x - \eta_4) + \iota_3\cdot(x - \eta_5) \qquad (38)$$

En la expresión (36), se usa un elemento indeterminado x para expresar el OR lógico. También puede usarse una pluralidad de elementos indeterminados para expresar un OR lógico. Por ejemplo, cuando se usan dos elementos indeterminados x_0 y x_1, el OR lógico $(x_0 = \eta_0) \vee (x_1 = \eta_1)$ de la proposición 1 que indica que x_0 es η_0 y la proposición 2 que indica que x_1 es η_1 se puede expresar mediante el polinomio siguiente.

$$(x_0 - \eta_0)\cdot(x_1 - \eta_1)$$

Pueden usarse también tres o más elementos indeterminados para expresar un OR lógico mediante un polinomio.

En la expresión (37), se usa un elemento indeterminado x para expresar el AND lógico. Pueden usarse también una pluralidad de elementos indeterminados para expresar un AND lógico. Por ejemplo, el AND lógico ($x_0 = \eta_0$) ^ ($x_1 = \eta_1$) de la proposición 1 que indica que x_0 es η_0 y la proposición 2 que indica que x_1 es η_1 se puede expresar mediante el polinomio siguiente.

$$\iota_0\cdot(x_0 - \eta_0) + \iota_1\cdot(x_1 - \eta_1)$$

Pueden usarse también tres o más elementos indeterminados para expresar un AND lógico mediante un polinomio.

Una fórmula lógica que incluye uno más OR lógicos y/o uno o más AND lógicos se expresa con H (H ≥ 1) tipos de elementos indeterminados x_0, ..., X_{H-1} como el polinomio $f(x_0, ..., X_{H-1})$. Se supone que una proposición para cada uno de los elementos indeterminados x_0, …, X_{H-1} es "x_h es η_h", en la que η_h (h = 0, ..., H-1) es una constante determinada para cada proposición. Entonces, en el polinomio $f(x_0, ..., X_{H-1})$ que indica la fórmula lógica, la proposición que indica que un elemento indeterminado X_h es una constante η_h se expresa mediante el polinomio que indica la diferencia entre el elemento indeterminado x_h y la constante η_h; el OR lógico de las proposiciones se expresa mediante el producto de los polinomios que indican las proposiciones; y el AND lógico de las proposiciones o los ORs lógicos de las proposiciones se expresan mediante una combinación lineal de los polinomios que indican las proposiciones o los ORs lógicos de las proposiciones. Por ejemplo, se usan cinco elementos indeterminados x_0, …, x_4 para expresar una fórmula lógica

$$\{(x_0 = \eta_0) \vee (x_1 = \eta_1) \vee (x_2 = \eta_2)\} \wedge (x_3 = \eta_3) \wedge (x_4 = \eta_4)$$

mediante el polinomio siguiente

$$f(x_0, ..., x_4) = \iota_0\cdot\{(x_0 - \eta_0)\cdot(x_1 - \eta_1)\cdot(x_2 - \eta_2)\} + \iota_1\cdot(x_3 - \eta_3) + \iota_2\cdot(x_4 - \eta_4)$$

Relación entre el polinomio y el producto interno

El polinomio $f(x_0, ..., x_{H-1})$ que indica una fórmula lógica puede ser expresado mediante el producto interno de dos vectores n-dimensionales. Más específicamente, el polinomio $f(x_0, ..., x_{H-1})$ es igual al producto interno de un vector

$$\vec{v} = (v_1, ..., v_n)$$

que tiene los elementos indeterminados de los términos del polinomio $f(x_0, ..., x_{H-1})$ como elementos, y un vector

$$\vec{w} = (w_1, ..., w_n)$$

que tiene los coeficientes de los términos del polinomio $f(x_0, ..., x_{H-1})$ como elementos.

$$f(x_0, ..., x_{H-1}) = \vec{w}\cdot\vec{v}$$

En otras palabras, si el polinomio $f(x_0, ..., x_{H-1})$ que indica una fórmula lógica es cero es equivalente a si el producto interno del vector $\vec{v}$ que tiene los elementos indeterminados de los términos del polinomio $f(x_0, ..., x_{H-1})$ como elementos y el vector $\vec{w}$ que tiene los coeficientes de los términos del polinomio $f(x_0, ..., x_{H-1})$ como elementos es

cero .

$$f(x_0, ..., x_{H-1}) = 0 \longleftrightarrow \vec{w} \cdot \vec{v} = 0$$

Por ejemplo, un polinomio $f(x) = \theta_0.x^0 + \theta_1.x + . . . + \theta_{n-1}.x^{n-1}$ expresado con un elemento indeterminado x puede ser expresado mediante el producto interno de dos vectores n-dimensionales de la manera siguiente.

$$\vec{w} = (w_1, ..., w_n) = (\theta_0, ..., \theta_{n-1}) \quad (39)$$

$$\vec{v} = (v_1, ..., v_n) = (x^0, ..., x^{n-1}) \quad (40)$$

$$f(x) = \vec{w} \cdot \vec{v} \quad (41)$$

En otras palabras, si el polinomio f(x) que indica una fórmula lógica es cero es equivalente a si el producto interno en la expresión (41) es cero.

$$f(x) = 0 \longleftrightarrow \vec{w} \cdot \vec{v} = 0 \quad (42)$$

Cuando un vector que tiene los elementos indeterminados de los términos del polinomio $f(x_0, ..., x_{H-1})$ como elementos se expresa mediante

$$\vec{w} = (w_1, ..., w_n)$$

y un vector que tiene los coeficientes de los términos del polinomio $f(x_0, ..., x_{H-1})$ como elementos se expresa mediante

$$\vec{v} = (v_1, ..., v_n)$$

si el polinomio $f(x_0, ..., x_{H-1})$ que indica una fórmula lógica es cero es equivalente a si el producto interno del vector $\vec{w}$ y el vector $\vec{v}$ es cero.

Por ejemplo, cuando las expresiones siguientes se usan en lugar de las expresiones (39) y (40),

$$\vec{w} = (w_1, ..., w_n) = (x^0, ..., x^{n-1}) \quad (43)$$

$$\vec{v} = (v_1, ..., v_n) = (\theta_0, ..., \theta_{n-1}) \quad (44)$$

si el polinomio f(x) que indica una fórmula lógica es cero es equivalente a si el producto interno en la expresión (41) es cero.

En la codificación de predicado con producto interno, uno de los vectores $\vec{v} = (v_0, ..., v_{n-1})$ y $\vec{w} = (w_0, ..., w_{n-1})$ se usa como la información de predicado y el otro se usa como la informacióńn de predicado. Una de entre la información de atributo y la información de predicado está incluida en el texto codificado y la otra está incluida en información de clave. Por ejemplo, un vector n-dimensional $(\theta_0, ..., \theta_{n-1})$ se usa como la información de predicado, otro vector n-dimensional $(x^0, ..., x^{n-1})$ se usa como la información de atributo, una de entre la información de atributo y la información de predicado está incluida en el texto codificado, y la otra está incluida en la información de clave. En la descripción siguiente, se supone que un vector n-dimensional incluido en la información de clave es $\vec{w} = (w_1, ..., w_n)$ y otro vector n-dimensional incluido en el texto codificado es $\vec{v} = (v_1, ..., v_n)$.

Por ejemplo,

Información de predicado: $\vec{w} = (w_1, ..., w_n) = (\theta_0, ..., \theta_{n-1})$
Información de atributo: $\vec{v} = (v_1, ..., v_n) = (x^0, ..., x^{n-1})$
De manera alternativa,
Información de predicado: $\vec{v} = (v_1, ..., v_n) = (\theta_0, ..., \theta_{n-1})$
Información de atributo: $\vec{w} = (w_1, ..., w_n) = (x^0, ..., x^{n-1})$

Esquema básico de codificación de predicado con producto interno
Un ejemplo de un esquema básico de un mecanismo de encapsulación de clave (KEM) se describe en la bibliografía no patente 4, que ha sido publicada después de la fecha de prioridad de la presente solicitud, y se describirá más adelante.

Este esquema incluye Setup(1^k), Genkey(MSK, $\vec{w}$), Enc(PA, $\vec{v}$) y Dec(SKw, C_2).

Configuración de Setup(1^k)
Entrada: parámetro k de seguridad
Salida: Información de clave maestra MSK, parámetro público PK

En un ejemplo de Setup(1^k), el parámetro k de seguridad se usa como n, se seleccionan la matriz A de (n+1) filas por (n+1) columnas que tiene los vectores base (n+1) dimensionales a_i (i = 1, ..., n+1) como elementos, la matriz A* de (n+1) filas por (n+1) columnas que tiene los vectores base a_i* (i = 1, ..., n+1) como elementos, y las matrices X y X * de (n+1) filas por (n+1) columnas usadas para la transformación de coordenadas. A continuación, los vectores base (n+1) dimensionales b_i (i = 1, ..., n+1) se calculan mediante una transformación de coordenadas mediante la expresión (22), y los vectores base (n+1) dimensionales b_i* (i = 1, ..., n+1) se calculan mediante una transformación de coordenadas mediante la expresión (24). A continuación, la matriz B* de (n+1) filas por (n+1) columnas que tiene los vectores base b_i* (i = 1, ..., n+1) como elementos es sacada como la información de clave maestra MSK; y los espacios vectoriales V y V*, la matriz B de (n+1) filas por (n+1) columnas que tienen los vectores base b_i (i = 1, ..., n+1) como elementos, el parámetro k de seguridad, el campo F_q finito, la curva elíptica E, los grupos cíclicos G_1, G_2 y G_T, los generadores g_1, g_2, y g_T, la función e bilineal, etc., son sacados como el parámetro público PK.

Generación de información de clave Genkey(MSK, $\vec{w}$)
Entrada: información de clave maestra MSK, vector $\vec{w}$
Salida: Información de clave D* correspondiente al vector $\vec{w}$

En un ejemplo de Genkey(MSK, $\vec{w}$), se selecciona un elemento $\alpha \in F_q$ desde el campo F_q finito. A continuación, la matriz B*, que es la información de clave maestra MSK, es usada para generar y sacar la información de clave D* correspondiente al vector $\vec{w}$ de la manera siguiente.

$$D^* = \alpha \cdot (\sum_{\mu=1}^{n} w_\mu \cdot b_\mu^*) + b_{n+1}^* \in G_2^{n+1} \qquad (45)$$

Si es difícil resolver un problema logarítmico discreto en el grupo cíclico G_2, es difícil separar y extraer los componentes de $w_\mu \cdot b_\mu$* y b_{n+1}* a partir de la información de clave D*.

Codificación Enc(PA, $\vec{v}$) Entrada: parámetro público PK, vector $\vec{v}$
Salida: Texto codificado C_2, clave simétrica K

En un ejemplo de Enc(PA, $\vec{v}$), se generan la clave simétrica K y un número aleatorio υ_1, que es un elemento del campo F_q finito. A continuación, el parámetro público PK, tal como la matriz B, un elemento υ_2 correspondiente a un valor que incluye la clave simétrica K, en el campo F_q finito, el vector $\vec{v}$, y el número aleatorio υ_1 se usan para generar el texto codificado C_2 de la manera siguiente.

$$C_2 = \upsilon_1 \cdot (\sum_{\mu=1}^{n} v_\mu \cdot b_\mu) + \upsilon_2 \cdot b_{n+1} \in G_1^{n+1} \qquad (46)$$

Se sacan el texto codificado C_2 y la clave simétrica K. Un ejemplo de la clave simétrica K es $g_T^{\tau.u2} \in G_T$, en la que u2 significa u_2. Un ejemplo de τ es 1_F, tal como se ha descrito anteriormente. Si es difícil resolver un problema logarítmico discreto en el grupo cíclico G_1, es difícil separar y extraer los componentes de $v_\mu.b_\mu$ y $u_2.b_{n+1}$ a partir del texto codificado C_2.

Descodificación y compartición de clave Dec(SKW, C_2)

Entrada: Información de clave D_1* correspondiente al vector $\vec{w}$, texto codificado C_2

Salida: clave simétrica K

En un ejemplo de Dec(SKw, C_2), el texto codificado C_2 y la información de clave D_1* son introducidos a la función e bilineal de la expresión (2).

A continuación, a partir de las características de las expresiones (3) y (26), se cumple ecuación siguiente.

$$\begin{aligned} e(C_2, D^*) &= e(\upsilon_1 \cdot (\textstyle\sum_{\mu=1}^{n} v_\mu \cdot b_\mu) + \upsilon_2 \cdot b_{n+1}, \alpha \cdot (\textstyle\sum_{\mu=1}^{n} w_\mu \cdot b_\mu^*) + b_{n+1}^*) \\ &= e(\upsilon_1 \cdot v_1 \cdot b_1, \alpha \cdot w_1 \cdot b_1^*) \cdot \ldots \cdot e(\upsilon_1 \cdot v_n \cdot b_n, \alpha \cdot w_n \cdot b_n^*) \\ &\quad \times e(\upsilon_2 \cdot b_{n+1}, b_{n+1}^*) \\ &= e(b_1, b_1^*)^{\upsilon_1 \cdot v_1 \cdot \alpha \cdot w_1} \cdot \ldots \cdot e(b_n, b_n^*)^{\upsilon_1 \cdot v_n \cdot \alpha \cdot w_n} \cdot e(b_{n+1}, b_{n+1}^*)^{\upsilon_2} \\ &= g_T^{\tau \cdot \upsilon_1 \cdot v_1 \cdot \alpha \cdot w_1} \cdot \ldots \cdot g_T^{\tau \cdot \upsilon_1 \cdot v_n \cdot \alpha \cdot w_n} \cdot g_T^{\tau \cdot \upsilon_2} \\ &= g_T^{\tau \cdot \upsilon_1 \cdot \alpha \cdot \vec{v} \cdot \vec{w}} \cdot g_T^{\tau \cdot \upsilon_2} \end{aligned} \quad (47)$$

Cuando el producto interno $\vec{w} \cdot \vec{v}$ es cero, la expresión (47) puede ser modificada a la forma siguiente.

$$\begin{aligned} e(C_2, D^*) &= g_T^{\tau \cdot \upsilon_1 \cdot \alpha \cdot 0} \cdot g_T^{\tau \cdot \upsilon_2} \\ &= g_T^{\tau \cdot \upsilon_2} \end{aligned} \quad (48)$$

A partir de este resultado, se genera y se saca la clave simétrica K. Un ejemplo de la clave simétrica K es $g_T^{\tau.u2} \in G_T$.

Seguridad contra CCA

El esquema es seguro contra CCA si se cumple la relación siguiente en el supuesto proporcionado a continuación:

$$Pr[bit=bit'] < (1/2) - FNK(k) \quad (49)$$

en la que FNK(k) es una función de k. Si [3] implica [2], el sistema es seguro contra CCA2. Si [2] implica [3], el sistema es seguro contra CCA1. CCA2 es más fuerte que CCA1.

[1] El atacante recibe un parámetro público PK.
[2] El atacante pasa dos grupos de texto no codificado M_0 y M_1, que son dos cadenas de bits, a un oráculo de codificación que tiene el parámetro público PK. El oráculo de codificación selecciona un bit $\in \{0, 1\}$ aleatoriamente, codifica cualquiera de los dos grupos de texto no codificado M_{bit}, y proporciona el texto codificado C al atacante.
[3] El atacante puede pasar un texto codificado $C(\sigma)(C(\sigma) \neq C)$ a un oráculo de descodificación que tiene

información de clave D* y recibe un resultado descodificado del texto codificado C(σ) desde el oráculo de descodificación.
[4] El atacante saca el bit' Є {0, 1}.

Seguridad contra CCA en un esquema básico de codificación de predicado con producto interno
El esquema básico de la codificación de predicado con producto interno descrito anteriormente no es seguro contra CCA. Esto se describirá mostrando un ejemplo sencillo. En el ejemplo sencillo, $\tau = 1_F$; la clave simétrica K es K = MAP (g_T^{u2}); y el texto codificado obtenido mediante la codificación de clave simétrica del texto no codificado M con la clave simétrica K es C_1 = K (+) M. MAP (g_T^{u2}) representa el mapeo de g_T^{u2} Є G_T a una cadena de bits. En este ejemplo, el atacante puede adoptar la estrategia siguiente (denominada en adelante estrategia asumida).

[1] El atacante recibe un parámetro público PK.
[2] El atacante pasa un vector $\vec{v}$ y dos grupos de texto no codificado M_0 y M_1 a un oráculo de codificación que tiene el parámetro público PK. El oráculo de codificación selecciona bit Є {0, 1} aleatoriamente, genera el texto codificado

$$C_1 = K\ (+)\ M_{bit} \qquad (50)$$

obtenido mediante la codificación de cualquiera de los grupos de texto no codificado M_{bit} con la clave simétrica $K = g_T^{u2}$, genera texto codificado

$$C_2 = \upsilon_1 \cdot (\Sigma_{\mu=1}^{n} v_\mu \cdot b_\mu) + \upsilon_2 \cdot b_{n+1} \in G_1^{n+1} \qquad (51)$$

con la clave simétrica $K = g_T^{u2}$, y proporciona el texto codificado (C_1, C_2) al atacante.

[3] El atacante pasa el texto codificado

$$(C_1(+)\Delta M, C_2) \qquad (52)$$

a un oráculo de descodificación que tiene la información de clave D* y recibe los resultados descodificados del texto codificado $C_1(1)$ y $C_1(2)$ desde el oráculo de descodificación.
Aquí, si el bit = 0, entonces C_1 = K (+) M_0, y el resultado descodificado de C_1 (+) ΔM es M_0 (+) ΔM. Si bit = 1, entonces C_1 = K (+) M_1, y el resultado de descodificación de C_1 (+) ΔM es M1 (+) ΔM.
[4] Cuando el resultado descodificado de C_1 (+) ΔM es M_0 (+) ΔM, el atacante saca el bit'= 0. Cuando el resultado descodificado es M_1 (+) ΔM, el atacante saca el bit' = 1.

Entonces, Pr[bit = bit'] = 1, y la expresión (49) no se cumple.

Codificación de predicado con producto interno en las realizaciones
Tal como se ha descrito anteriormente, el esquema básico de la codificación de predicado con producto interno no es seguro contra CCA. Si se usa el procedimiento de transformación CHK o el método de transformación BK para hacer que el esquema básico de codificación de predicado con producto interno sea seguro contra CCA, es necesario un espacio bidimensional usado sólo para mejorar el nivel de seguridad contra CCA en el espacio de texto codificado, y el número de dimensiones de los espacios vectoriales V y V* debe ser n+3. En las realizaciones, la seguridad contra CCA se mejora sin proporcionar un espacio usado sólo para mejorar el nivel de seguridad contra CCA.

Esquema mejorado de codificación de predicado con producto interno
A continuación, se describirá un esquema mejorado de codificación de predicado con producto interno en las realizaciones.

Procesamiento de codificación
Un aparato de codificación para realizar el procesamiento de codificación realiza el procesamiento siguiente.

(ENC-1) Una unidad de codificación de clave simétrica introduce el texto no codificado M y una clave simétrica K, ambos de los cuales son leídos desde el almacenamiento, a una función de codificación de

clave simétrica para calcular el texto codificado C_1 y saca el texto codificado C_1.
(ENC-2) Una unidad de operación de función introduce un valor que contiene un valor correspondiente a la clave simétrica K y un valor correspondiente al texto codificado C_1 a la función H resistente a colisiones para calcular el valor de función de la función H y saca el valor de función de la función H.
(ENC-3) Una unidad de operación vectorial calcula el texto codificado

$$C_2 = r \cdot (\Sigma_{\mu=1}^{n} v_\mu \cdot b_\mu) + \rho \cdot b_{n+1} \in G_1^{n+1} \qquad (53)$$

usando el elemento r del campo F_q finito, correspondiente al valor de función de la función H, el elemento ρ del campo F_q finito, correspondiente a la clave simétrica K, los elementos v_μ ($\mu = 1, ..., n, n \geq 1$) del vector n-dimensional $\vec{v}$ ($v_1, ..., v_n$) compuesto de los elementos del campo F_q finito, y los vectores base n+1 dimensionales $b_i \in G_1^{n+1}$, y saca el texto codificado C_2.

Procesamiento de descodificación

Un aparato de descodificación para realizar el procesamiento de descodificación ejecuta el procesamiento siguiente.

(DEC-1) Un valor que contiene el texto codificado C_1 y el texto codificado $C_2 \in G_1^{n+1}$ es introducido a una unidad de entrada.
(DEC-2) Una unidad de operación de función bilineal introduce el texto codificado $C_2 \in G_1^{n+1}$ y la información de clave

$$D_1^* = \alpha \cdot (\Sigma_{\mu=1}^{n} w_\mu \cdot b_\mu^*) + b_{n+1}^* \qquad (54)$$

a la función e bilineal para calcular el valor de función

$$S^{\sim} = e(C_2, D_1^*) \in G_T \qquad (55)$$

y saca el valor de función $S^{\sim}$.
(DEC-3) Una unidad de operación de función introduce el valor correspondiente al valor de función $S^{\sim}$ y el texto codificado C_1 a la función H resistente a colisiones para calcular el valor de función de la función H y saca el valor de función de la función H. Si la información de clave D_1^* es correcta, el valor correspondiente al valor de función $S^{\sim}$ es igual al valor correspondiente a la clave simétrica K, que es introducido a la función H resistente a colisiones en (ENC-2) del procedimiento de codificación.
(DEC-4) Una unidad de decisión decide si el texto codificado C_2, el elemento $r^{\sim}$ del campo F_q finito, correspondiente al valor de función de la función H, el elemento $\rho^{\sim}$ del campo F_q finito, los elementos v_μ del vector n-dimensional $\vec{v}$, y los vectores base $b_i \in G_1^{n+1}$ ($i = 1, ..., n+1$) cumplen la relación siguiente, y saca a la determinación.

$$C_2 = r^{\sim} \cdot (\Sigma_{\mu=1}^{n} v_\mu \cdot b_\mu) + \rho^{\sim} \cdot b_{n+1} \in G_1^{n+1} \qquad (56)$$

(DEC-5) Si la expresión (55) se cumple, se saca el resultado descodificado del texto codificado C_1. Si la expresión (56) no se cumple, se saca un mensaje de error ($\perp$).

Seguridad contra CCA en un método mejorado de codificación de predicado con producto interno
Se describirá un caso en el que se aplica la estrategia asumida descrita anteriormente al esquema mejorado.

[1] El atacante recibe el parámetro público PK.
[2] El atacante pasa el vector $\vec{v}$ y los dos grupos de texto no codificado M_0 y M_1 al oráculo de codificación que tiene el parámetro público PK. El oráculo de codificación selecciona bit $\in \{0, 1\}$ aleatoriamente, codifica cualquiera de los grupos de texto no codificado M_{bit} con una clave simétrica $K = g_T^{u2}$, y genera el texto codificado

$$C_1 = K\ (+)\ M_{bit} \qquad (57)$$

(ENC-1). El oráculo de codificación introduce adicionalmente el valor que contiene el valor correspondiente a la clave simétrica K y el valor correspondiente al texto codificado C_1 a la función H resistente a colisiones para calcular el valor de función de la función H (ENC-2). A continuación, el oráculo de codificación genera el texto codificado

$$C_2 = r \cdot (\Sigma_{\mu=1}{}^{n} v_\mu \cdot b_\mu) + \rho \cdot b_{n+1} \in G_1{}^{n+1} \qquad (58)$$

de la clave simétrica $K = g_T{}^{v2}$ usando el elemento r del campo F_q finito, correspondiente al valor de función de la función H, el elemento ρ del campo F_q finito, correspondiente a la clave simétrica K, los elementos v_μ ($\mu = 1, ..., n, n \geq 1$) del vector n-dimensional $\vec{v} = (v_1, ..., v_n)$ compuesto de elementos del campo F_q finito, y los vectores base n+1 dimensionales $b_i \in G_1{}^{n+1}$ (ENC-3), y proporciona el texto codificado (C_1, C_2) al atacante.

[3] Debido a que el atacante no conoce la clave simétrica K, el atacante no puede generar $C_2' = r'.(\sum_{\mu=1}{}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1{}^{n+1}$ correspondiente a un elemento r 'del campo F_q finito, correspondiente al valor de función obtenido al introducir el valor que contiene el texto codificado C_1 (+) ΔM y el valor correspondiente a la clave simétrica K a la función H resistente a colisiones. Por lo tanto, el atacante pasa a la siguiente combinación de C_1 (+) ΔM y el texto codificado C_2 correspondiente al texto codificado C_1 a un oráculo de descodificación.

$$(C_1\ (+)\ \Delta M,\ C_2) \qquad (59)$$

El oráculo de descodificación al cual ha sido introducido (DEC-1) el texto codificado de la expresión (59) genera el valor de función $S^{\sim}$ de la expresión (55) (DEC-2), calcula el valor de función de la función H resistente a colisiones introduciendo el texto codificado C_1 (+) ΔM y el valor correspondiente al valor de función $S^{\sim}$ a la función H, y saca el valor de función de la función H (DEC-3). El oráculo de descodificación determina si el segundo texto codificado C_2, un elemento $r^{\sim}{}_{\Delta M}$ del campo F_q finito, correspondiente al valor de función de la función H, el elemento $\rho^{\sim}$ del campo F_q finito, los elementos v_μ del vector n-dimensional $\vec{v}$, y los vectores base $b_i \in G_1{}^{n+1}$ (i = 1, ..., n+1) cumplen la relación siguiente.

$$C_2 = r^{\sim}{}_{\Delta M} \cdot (\Sigma_{\mu=1}{}^{n} v_\mu \cdot b_\mu) + \rho^{\sim} \cdot b_{n+1} \in G_1{}^{n+1} \qquad (60)$$

(DEC-4). Aquí, debido a que $r^{\sim}{}_{\Delta M} \neq r^{\sim}$, la expresión (60) no se cumple, y el oráculo de descodificación saca el mensaje de error (⊥).

[4] Debido a que el atacante no puede obtener el resultado descodificado de C_1 (+) ΔM, el atacante no es capaz de sacar el bit' = 0 cuando el resultado descodificado es M_0 (+) ΔM o de sacar el bit' = 1 cuando el resultado descodificado es M_1 (+) ΔM. Por consiguiente, la estrategia asumida del atacante fracasará.

Primera realización

A continuación se describirá una primera realización del esquema mejorado.

Visión general

A continuación, se proporcionará una visión general de la primera realización.

Procesamiento de codificación

Un aparato de codificación para realizar el procesamiento de codificación realiza el procesamiento siguiente.

(ENC-11) Una unidad de codificación de clave simétrica introduce el texto no codificado M y la clave simétrica K, ambos de los cuales son leídos desde un almacenamiento, a la función de codificación de clave simétrica para calcular el texto codificado C_1 y saca el texto codificado C_1.

(ENC-21) Una unidad de operación de función pone la clave simétrica K y texto codificado C_1 en la función H resistente a colisiones para calcular los elementos r y ρ del campo F_q finito y saca los elementos r y ρ. La clave simétrica K en (ENC-21) corresponde al valor correspondiente a la clave simétrica K indicada

anteriormente en (ENC-2).

(ENC-31) Una unidad de operación de función calcula el valor de función $S = g_T^{\tau \cdot \rho} \in G_T$ usando el elemento ρ y saca el valor de función S.

(ENC-32) Una unidad de operación OR exclusivo calcula el texto codificado C_3 que es el valor OR exclusivo de la clave simétrica K y el valor de función R(S) que es la secuencia binaria obtenida introduciendo el valor de función S a la función R inyectiva, y saca el texto codificado C_3.

(ENC-33) Una unidad de operación vectorial calcula $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$ usando los elementos v_μ (μ = 1, ..., n) del vector n-dimensional $\vec{v}$ compuesto por los elementos del campo F_q finito y los elementos r y ρ, y saca el texto codificado C_2.

Procesamiento de descodificación

Un aparato de descodificación para realizar el procesamiento de descodificación ejecuta el procesamiento siguiente.

(DEC-11) Un valor que contiene el texto codificado C_1, el texto codificado C_2 y el texto codificado C_3 es introducido a una unidad de entrada.

(DEC-21) Una unidad de operación de función bilineal introduce el texto codificado $C_2 \in G_1^{n+1}$ y la información de clave $D_1^* = \alpha.(\sum_{\mu=1}^{n} w_\mu.b_\mu^*) + b_{n+1}^*$ a la función e bilineal para calcular el valor de función $S\tilde{} = E(C_2, D_1^*) \in G_T$ y saca el valor de función $S\tilde{}$.

Si el producto interno cumple $\vec{w} \cdot \vec{v} = 0$, las características de las expresiones (3) y (26) conducen a la expresión siguiente.

$$\begin{aligned} S\tilde{} = e(C_2, D_1^*) &= e(r \cdot (\textstyle\sum_{\mu=1}^{n} v_\mu \cdot b_\mu) + \rho \cdot b_{n+1}, \alpha \cdot (\textstyle\sum_{\mu=1}^{n} w_\mu \cdot b_\mu^*) + b_{n+1}^*) \\ &= e(r \cdot v_1 \cdot b_1, \alpha \cdot w_1 \cdot b_1^*) \cdot \ldots \cdot e(r \cdot v_n \cdot b_n, \alpha \cdot w_n \cdot b_n^*) \\ &\quad \times e(\rho \cdot b_{n+1}, b_{n+1}^*) \\ &= e(b_1, b_1^*)^{r \cdot v_1 \cdot \alpha \cdot w_1} \cdot \ldots \cdot e(b_n, b_n^*)^{r \cdot v_n \cdot \alpha \cdot w_n} \cdot e(b_{n+1}, b_{n+1}^*)^{\rho} \\ &= g_T^{\tau \cdot r \cdot v_1 \cdot \alpha \cdot w_1} \cdot \ldots \cdot g_T^{\tau \cdot r \cdot v_n \cdot \alpha \cdot w_n} \cdot g_T^{\rho} \\ &= g_T^{\tau \cdot r \cdot \alpha \cdot \vec{v} \cdot \vec{w}} \cdot g_T^{\tau \cdot \rho} \\ &= g_T^{\tau \cdot \rho} \\ &= S \end{aligned} \qquad (61)$$

(DEC-31) Una unidad de operación OR exclusivo calcula el valor OR exclusivo $K\tilde{}$ del texto codificado C_3 y el valor de función $R(S\tilde{})$ que es la secuencia binaria obtenida introduciendo el valor de función $S\tilde{}$ a la función R inyectiva, y saca el valor OR exclusivo $K\tilde{}$. Si se cumple la expresión (61), se cumple la relación siguiente.

$$\begin{aligned} K\tilde{} &= R(S\tilde{}) \,(+)\, C_3 \\ &= R(S) \,(+)\, C_3 \\ &= R(S) \,(+)\, R(S) \,(+)\, K \\ &= K \end{aligned} \qquad (62)$$

(Dec-32) Una unidad de operación de función introduce el valor OR exclusivo $K\tilde{}$ y el texto codificado C_1 a la función H resistente a colisiones para calcular los elementos $r\tilde{}$ y $\rho\tilde{}$ del campo F_q finito y saca los elementos $r\tilde{}$ y $\rho\tilde{}$. Los elementos $r\tilde{}$ y $\rho\tilde{}$ en (Dec-31) corresponden al valor que contiene el elemento $r\tilde{}$ indicado anteriormente en (Dec-3).

(Dec-41) Una unidad de operación vectorial calcula el valor de función $C_2\tilde{} = r\tilde{}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho\tilde{}.b_{n+1} \in G_1^{n+1}$ usando los elementos v_μ (μ = 1, ..., n) del vector n-dimensional $\vec{v}$ y los elementos $r\tilde{}$ y $\rho\tilde{}$ y saca el valor de función $C_2\tilde{}$.

(Dec-42) Una unidad de comparación determina si el valor de función $C_2^{\sim}$ es igual al texto codificado C_2 y saca la determinación. Esta determinación corresponde a la determinación en (DEC-4).
DEC-51) Si la determinación sacada desde la unidad de comparación indica que el valor de función $C_2^{\sim}$ es igual al texto codificado C_2, una unidad de descodificación de clave simétrica usa el valor OR exclusivo $K^{\sim}$ como la clave simétrica, introduce el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a la función de descodificación de clave simétrica para calcular el valor descodificado $M^{\sim}$, y saca el valor descodificado $M^{\sim}$. Si la determinación sacada desde la unidad de comparación indica que el valor de función $C_2^{\sim}$ no es igual al texto codificado C_2, se saca un mensaje de error ($\perp$).

Seguridad contra CCA en la primera realización
Se describirá un caso en el que se aplica la estrategia asumida descrita anteriormente al esquema de la primera realización. En este ejemplo, $\tau = 1_F$, y el texto codificado obtenido mediante codificación de clave simétrica del texto no codificado M con la clave simétrica K es $C_1 = K (+) M$.

[1] El atacante recibe el parámetro público PK.
[2] El atacante pasa el vector $v^{\rightarrow}$ y dos grupos de texto no codificado M_0 y M_1 al oráculo de codificación que tiene el parámetro público PK. El oráculo de codificación selecciona bit $\in \{0, 1\}$ aleatoriamente, codifica cualquiera de los grupos de texto no codificado M_{bit} con clave simétrica $K = g_T^{u2}$, y genera un texto codificado

$$C_1 = K (+) M_{bit} \qquad (63)$$

(ENC-11). El oráculo de codificación introduce además la clave simétrica K y el texto codificado C_1 a la función H resistente a colisiones para calcular los elementos r y ρ del campo F_q finito (ENC-21) calcula el valor de función $S = g_T^{r,\rho} \in G_T$ (ENC-31) calcula el texto codificado C_3 obteniendo un OR exclusivo del valor R(S) de función y la clave simétrica K (ENC-32), calcula $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$ (ENC-33) y proporciona el texto codificado (C_1, C_2, C_3) al atacante.
[3] Debido a que el atacante no conoce la clave simétrica K, el atacante no puede generar $C_2 '= r'.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$ correspondiente a los elementos r'y ρ' del campo F_q finito, obtenido al introducir la clave simétrica K y el texto codificado C_1 (+) ΔM a la función H resistente a colisiones. Por lo tanto, el atacante pasa la siguiente combinación de C_1 (+) ΔM, el texto codificado C_2 correspondiente al texto codificado C_1, y el texto codificado C_3 a un oráculo de descodificación.

$$(C_1 (+) \Delta M, C_2, C_3) \qquad (64)$$

El oráculo de descodificación al que ha sido introducido el texto codificado de la expresión (64) (DEC-11) genera el valor de función $S^{\sim} = e(C_2, D_1^*) \in G_T$ (DEC-21), calcula el valor OR exclusivo $K^{\sim}$ del texto codificado C_3 y el valor de función $R(S^{\sim})$ que es la secuencia binaria obtenida introduciendo el valor de función $S^{\sim}$ a la función R inyectiva, y calcula los elementos $r^{\sim}_{\Delta M}$ y $\rho^{\sim}_{\Delta M}$ del campo F_q finito introduciendo el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 (+) ΔM a la función H resistente a colisiones (DEC- 32). El oráculo de descodificación calcula el siguiente valor de función $C_2^{\sim}$ usando los elementos v_μ de $v^{\rightarrow}$, el elemento $r^{\sim}_{\Delta M}$, los vectores base $b_i \in G_1^{n+1}$ (i = 1, ..., n+1), y el elemento de $\rho^{\sim}_{\Delta M}$ (DEC-41).

$$C_2^{\sim} = r^{\sim}_{\Delta M} \cdot (\Sigma_{\mu=1}^{n} v_\mu \cdot b_\mu) + \rho^{\sim}_{\Delta M} \cdot b_{n+1} \in G_1^{n+1} \qquad (65)$$

A continuación, el oráculo de descodificación determina si el valor de función $C_2^{\sim}$ es igual al texto codificado $C_2^{\sim}$. Aquí, debido a que $r^{\sim}_{\Delta M} \neq r^{\sim}$ y $\rho^{\sim}_{\Delta M} \neq \rho^{\sim}$, la expresión (65) no cumple esta relación, y el oráculo de descodificación saca un mensaje de error ($\perp$).
[4] Debido a que el atacante no puede obtener el resultado descodificado de C_1 (+) ΔM, el atacante no es capaz de sacar el bit' = 0 cuando el resultado descodificado es M_0 (+) ΔM o sacar el bit' = 1 cuando el resultado es M1 (+) ΔM. Por consiguiente, la estrategia del atacante fracasará.

Detalles
A continuación, se describirá la primera realización. Aunque en este ejemplo se cumple $\tau = 1_F$, esto no limita la presente invención, y τ puede ser un elemento del campo F_q finito diferente de 0_F o 1_F.

Estructura general

La Figura 1 es una vista que ilustra la estructura de un sistema criptográfico 1 de la primera realización.

Tal como se muestra en la Figura 1, el sistema criptográfico 1 de esta realización incluye un aparato 110 de codificación, un aparato 120 de descodificación y un aparato 130 de generación de claves, que están conectados a través de una red 140 de manera que pueden comunicarse entre sí.

Aparato de codificación

La Figura 2 es una vista que ilustra la estructura del aparato 110 de codificación en la Figura 1. Las flechas en la Figura 2 indican el flujo de datos, pero se omiten las flechas para indicar la entrada y salida de datos a y desde una unidad 111 de control y un almacenamiento 113b temporal.

Tal como se muestra en la Figura 2, el aparato 110 de codificación de esta realización incluye la unidad 111 de control, un generador 112a de números aleatorios, una unidad 112b de codificación de clave simétrica, una unidad 112c de operación de función, una unidad 112d de operación de grupo, una unidad 112e de operación OR exclusivo, una unidad 112f de operación vectorial, un almacenamiento113a, el almacenamiento 113b temporal y un transmisor 114.

El aparato 110 de codificación es, por ejemplo, un aparato especial que incluye un programa especial y un ordenador conocido o especializado provisto de una unidad central de procesamiento (CPU), una memoria de acceso aleatorio (RAM), una memoria de sólo lectura (ROM) o similares. La unidad 111 de control, el generador 112a de números aleatorios, la unidad 112b de codificación de clave simétrica, la unidad 112c de operación de función, la unidad 112d de operación de grupo, la unidad 112e de operación OR exclusivo y las unidades 112f de operación vectorial son unidades implementadas por la CPU que ejecuta el programa especial. Al menos algunas de las unidades de procesamiento pueden ser un circuito integrado especializado: Por ejemplo, el generador 112a de números aleatorios puede ser un circuito integrado de generación de números aleatorios conocido. El almacenamiento 113a y el almacenamiento 113b temporal son, por ejemplo, un almacenamiento auxiliar, tal como una RAM, un registro, una memoria caché, un elemento en un chip, o un disco duro, o un área de almacenamiento formada combinando al menos algunos de estos dispositivos. El transmisor 114 es un dispositivo de comunicación, tal como un módem o una tarjeta de red de área local (LAN), por ejemplo.

El aparato 110 de codificación ejecuta el procesamiento bajo el control de la unidad 111 de control. Cada grupo de datos desde cada unidad de procesamiento es almacenado en el almacenamiento 113b temporal o el almacenamiento 113a, los datos almacenados en el almacenamiento 113b temporal o el almacenamiento 113a son leídos, introducidos a una unidad de procesamiento, y usados para su procesamiento, cuando sea necesario, aunque estas descripciones de flujo de datos se simplifican a continuación.

Aparato de descodificación

La Figura 3 es una vista que ilustra la estructura del aparato 120 de descodificación en la Figura 1. Las flechas en la Figura 3 indican el flujo de datos, pero se omiten las flechas para indicar la entrada y la salida de datos a y desde una unidad 121 de control y un almacenamiento 123b temporal.

Tal como se muestra en la Figura 3, el aparato 120 de descodificación de esta realización incluye la unidad 121 de control, una unidad 122a de operación de la función bilineal, una unidad 122b de operación OR exclusivo, una unidad 122c de operación de función, una unidad 122d de decisión, una unidad 122e de descodificación de clave simétrica, un almacenamiento 123a, el almacenamiento 123b temporal, un transmisor 124 y un receptor 125 (correspondiente a la unidad de entrada). La unidad 122d de decisión incluye una unidad 122da de operación vectorial y una unidad 122db de comparación.

El aparato 120 de descodificación es, por ejemplo, un aparato especial que incluye un programa especial y un ordenador conocido o especializado provisto de una CPU, una RAM, una ROM o similar. La unidad 121 de control, la unidad 122a de operación de función bilineal, la unidad 122b de operación OR exclusivo, la unidad 122c de operación de función, la unidad 122d de decisión, y la unidad 122e de descodificación de clave simétrica son unidades de procesamiento implementadas por la CPU que ejecuta el programa especial, por ejemplo. Al menos algunas de las unidades de procesamiento pueden ser un circuito integrado especializado: Por ejemplo, la unidad 122a de operación de función bilineal puede ser un circuito integrado de cálculo de emparejamiento conocido. El almacenamiento 123a y el almacenamiento 123b temporal son, por ejemplo, un almacenamiento auxiliar, tal como una RAM, un registro, una memoria caché, un elemento en un chip, o un disco duro, o una zona de almacenamiento formada por la combinación de al menos algunos de estos dispositivos. El transmisor 124 y el receptor 125 son un dispositivo de comunicación, tal como un módem o una tarjeta de LAN, por ejemplo.

El aparato 120 de descodificación ejecuta el procesamiento bajo el control de la unidad 121 de control. Cada pieza de datos sacada desde cada unidad de procesamiento es almacenada en el almacenamiento 123b temporal o el

almacenamiento 123a, los datos almacenados en el almacenamiento 123b temporal o el almacenamiento 123a son leídos, introducidos a una unidad de procesamiento, y usados para su procesamiento, cuando sea necesario, aunque estas descripciones de flujo de datos se simplifican a continuación.

Aparato de generación de claves

La Figura 4 es una vista que ilustra la estructura del aparato 130 de generación de claves en la Figura 1. Las flechas en la Figura 4 indican el flujo de datos, pero se omiten las flechas para indicar la entrada y salida de datos a y desde una unidad 131 de control y el almacenamiento 133b temporal.

Tal como se muestra en la Figura 4, el aparato 130 de generación de claves de esta realización incluye la unidad 131 de control, un generador 132a de números aleatorios, una unidad 132b de operación vectorial, un almacenamiento 133a, un almacenamiento 133b temporal, un receptor 134 y un transmisor 135.

El aparato 130 de generación de claves es, por ejemplo, un aparato especial que incluye un programa especial y un ordenador conocido o especializado provisto de una CPU, una RAM, una ROM o similar. La unidad 131 de control, el generador 132a de números aleatorios y la unidad 132b de operación vectorial son unidades de procesamiento implementadas por la CPU que ejecuta el programa especial, por ejemplo. Al menos algunas de las unidades de procesamiento pueden ser un circuito integrado especializado: El almacenamiento 133a y el almacenamiento 133b temporal son, por ejemplo, un almacenamiento auxiliar, tal como una RAM, un registro, una memoria caché, un elemento de un circuito integrado, o un disco duro, o una zona de almacenamiento formada combinando al menos algunos de estos dispositivos. El transmisor 134 y el receptor 135 son un dispositivo de comunicación, tal como un módem o una tarjeta LAN, por ejemplo.

El aparato 130 de generación de claves ejecuta el procesamiento bajo el control de la unidad 131 de control. A continuación, se describirá sólo brevemente el flujo de datos. Cada grupo de datos sacados desde cada unidad de procesamiento es almacenado en el almacenamiento 133b temporal o el almacenamiento 133a, los datos almacenados en el almacenamiento 133b temporal o el almacenamiento 133a son leídos, introducidos a una unidad de procesamiento y usados para su procesamiento, cuando sea necesario, aunque estas descripciones de flujo de datos se simplifican a continuación.

Ajustes preparatorios

A continuación, se describirán los ajustes preparatorios para ejecutar el procesamiento en esta realización.

Un parámetro público PK se especifica en el aparato 110 de codificación, el aparato 120 de descodificación y el aparato 130 de generación de claves. El parámetro público PK incluye los espacios vectoriales V y V*, en los que la matriz B de (n+1) filas por (n+1) columnas que tiene los vectores base b_i (i = 1, ..., n+1) como elementos, el parámetro k de seguridad, el campo F_q finito, la curva elíptica E, los grupos cíclicos G_1, G_2 y G_T, los generadores g_1, g_2 y g_T, la función bilineal e y el número entero n mayor o igual que 1. La función H_1 resistente a colisiones mostrada en la expresión (33) y la función R inyectiva mostrada en la expresión (35) se especifican en el aparato 110 de codificación y el aparato 120 de descodificación.

La matriz B* de (n+1) filas por (n+1) columnas, que es la información de clave maestra MSK, es almacenada de manera segura en el almacenamiento 133a del aparato 130 de generación de claves. La matriz B* es información secreta y se mantiene en secreto.

El vector n-dimensional $\vec{v} = (v_1, ..., v_n)$ compuesto de los elementos del campo F_q finito, mostrado en la expresión (31), es almacenado en el almacenamiento 113a del aparato 110 de codificación, y el vector n-dimensional $\vec{v} = (v_1, ..., v_n)$ y el vector n-dimensional $\vec{w} = (w_1, ..., w_n)$, compuesto de los elementos del campo F_q finito, mostrados en la expresión (30), se almacenan en el almacenamiento 123a del aparato 120 de descodificación. Estos vectores cumplen la relación siguiente.

$$\vec{w} \cdot \vec{v} = 0 \qquad (66)$$

La información de predicado y la información de atributo pueden ser $\vec{w} = (w_1, ..., w_n)$ y $\vec{v} = (v_1, ..., v_n)$, respectivamente. De manera alternativa, la información de predicado y la información de atributo pueden ser $\vec{v} = (v_1, ..., v_n)$ y $\vec{w} = (w_1, ..., w_n)$, respectivamente.

El almacenamiento 113a del aparato 110 de codificación almacena el texto no codificado M Є {0, 1}* a codificar (encriptar).

Procesamiento de codificación

La Figura 5 es una vista que ilustra el procesamiento de codificación en la primera realización. A continuación se describirá el procesamiento de codificación en esta realización.

El generador 112a de números aleatorios en el aparato 110 de codificación (Figura 2) genera el número aleatorio de secuencia binario de k bits, y es almacenado en el almacenamiento 113a como la clave simétrica siguiente (etapa S101).

$$K \in \{0,1\}^k \qquad (67)$$

El texto no codificado M y la clave simétrica K leídos desde el almacenamiento son introducidos a la unidad 112b de codificación de clave simétrica. La unidad 112b de codificación de clave simétrica los introduce a la función de codificación de clave simétrica Enc para obtener el siguiente texto codificado C_1, y saca el texto codificado C_1 (etapa S102).

$$C_1 = Enc_k(M) \in \{0,1\}^k \qquad (68)$$

El texto codificado C_1 y la clave simétrica K son introducidos a la unidad 112c de operación de función. La unidad 112c de operación de función introduce la clave simétrica K y el texto codificado C_1 a la función H_1 resistente a colisiones para obtener los siguientes elementos r y ρ del campo F_q finito y saca los elementos r y ρ (etapa S103).

$$(r, \rho) = H_1(K, C_1) \in F_q \times F_q \qquad (69)$$

El elemento ρ del campo F_q finito es introducido a la unidad 112d de operación de grupo. La unidad 112d de operación de grupo calcula el siguiente valor de función S usando el elemento ρ, y saca el valor de función S (etapa S104).

$$S = g_T^{\rho} \in G_T \qquad (70)$$

El valor de función S y la clave simétrica K son introducidos a la unidad 112e de operación OR exclusivo. La unidad 112e de operación OR exclusivo usa la clave simétrica K y el valor de función

$$R(S) \in \{0,1\}^k \qquad (71)$$

que es la secuencia binaria de k bits obtenida introduciendo el valor de función S a la función R inyectiva, para obtener un valor OR exclusivo de los mismos como el texto codificado

$$C_3 = R(S) \,(+)\, K \in \{0,1\}^k \qquad (72)$$

y saca el texto codificado C_3 (etapa S105).

Los elementos r y ρ del campo F_q finito y el vector n-dimensional $\vec{v} = (v_1, ..., v_n)$ son introducidos a la unidad 112f de operación vectorial. La unidad 112f de operación vectorial calcula el texto codificado

$$C_2 = r \cdot (\Sigma_{\mu=1}^{n} v_\mu \cdot b_\mu) + \rho \cdot b_{n+1} \in G_1^{\,n+1} \qquad (73)$$

usando los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$ y los elementos r y ρ, y saca el texto codificado

C_2 (etapa S106).

El texto codificado C_1, el texto codificado C_2 y el texto codificado C_3 son introducidos en el transmisor 114. El transmisor 114 envía un valor que los contiene a través de la red 140 al aparato 120 de descodificación (etapa S107).

Procesamiento de generación de información de claves

El procesamiento de generación de información de clave se realiza especialmente cuando la información de clave D_1* no está almacenada en el almacenamiento 123a del aparato 120 de descodificación. Cuando la información de clave D_1* está almacenada en el almacenamiento 123a del aparato 120 de descodificación, este procesamiento puede ser omitido.

La Figura 6 es una vista que ilustra el procesamiento de generación de información de clave en la primera realización. A continuación se describirá el procesamiento de generación de información de clave en esta realización.

El vector n-dimensional $\vec{w} = (w_1, ..., w_n)$ almacenado en el aparato 120 de descodificación (Figura 3) es leído e introducido al transmisor 124. El transmisor 124 envía el vector n-dimensional $\vec{w} = (w_1, ..., w_n)$ a través de la red 140 al aparato 130 de generación de claves (etapa S111).

El vector n-dimensional $\vec{w} = (w_1, ..., w_n)$ enviado es recibido por el receptor 134 del aparato 130 de generación de claves (Figura 4) (etapa S112). Esto hace que el generador 132a de números aleatorios del aparato 130 de generación de claves genere un número aleatorio α, que es un elemento del campo F_q finito y que saque el número aleatorio α (etapa S113).

$$\alpha \in F_q \qquad (74)$$

El vector n-dimensional $\vec{w} = (w_1, ..., w_n)$, el número aleatorio α y la matriz B* son introducidos a la unidad 132b de operación vectorial. Usando los mismos, la unidad 132b operación vectorial genera la siguiente información de clave D_1*, y saca la información de clave D_1* (etapa S114).

$$D_1^* = \alpha \cdot (\Sigma_{\mu=1}^{n} w_\mu \cdot b_\mu^*) + b_{n+1}^* \in G_1^{n+1} \qquad (75)$$

La información de clave D_1* es introducida al transmisor 135. El transmisor 135 envía la información de clave D_1* a través de la red 140 al aparato 120 de descodificación (etapa S115).

La información clave D_1* enviada es recibida por el receptor 125 del aparato 120 de descodificación (Figura 3) y es almacenada en el almacenamiento 123a (etapa S116).

Procesamiento de descodificación

La Figura 7 es una vista que ilustra el tratamiento de descodificación en la primera realización. A continuación se describirá el procesamiento de descodificación en esta realización.

El receptor 125 del aparato 120 de descodificación (Figura 3) recibe el valor que contiene el texto codificado C_1, el texto codificado C_2 Є G_1^{n+1} y el texto C_3 codificado (etapa S121).

El texto codificado C_2 Є G_1^{n+1} y la información de clave D_1* son introducidos a la unidad 122a de operación de función bilineal. La unidad 122a de operación de función bilineal introduce el texto codificado C_2 Є G_1^{n+1} y la información de clave D_1* a la función e bilineal para calcular el siguiente valor de función $S^{\sim}$ y saca el valor de función $S^{\sim}$ (etapa S122).

$$S^{\sim} = e(C_2, D_1^*) \in G_T \qquad (76)$$

El valor de función $S^{\sim}$ y el texto codificado C_3 son introducidos a la unidad 122b de operación OR exclusivo. La unidad 122b de operación OR exclusivo calcula el siguiente valor OR exclusivo $K^{\sim}$ del texto codificado C_3 y el valor

de función $R(S^{\sim})$ que es la secuencia binaria obtenida introduciendo el valor de función $S^{\sim}$ en la función R inyectiva, y saca el valor OR exclusivo $K^{\sim}$ (etapa S123).

$$K^{\sim} = R(S^{\sim})\,(+)\,C_3 \qquad (77)$$

El valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 se introducen a la unidad 122c operación de función. La unidad 122c de operación de función introduce el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a la función H_1 resistente a colisiones, calcula los siguientes elementos $r^{\sim}$ y $\rho^{\sim}$, y saca los elementos $r^{\sim}$ y $\rho^{\sim}$ (etapa S124).

$$(r^{\sim}, \rho^{\sim}) = H_1(K^{\sim}, C_1) \in F_q \times F_q \qquad (78)$$

Los elementos v_μ (μ = 1, ..., n) del vector n-dimensional $\vec{v}$ y los elementos $r^{\sim}$ y $\rho^{\sim}$ se introducen a la unidad 122da de operación vectorial de la unidad 122d de decisión. Usando los mismos, la unidad 122da de operación vectorial calcula el siguiente valor de función $C_2{}^{\sim}$, y saca el valor de función $C_2{}^{\sim}$ (etapa S125).

$$C_2{}^{\sim} = r^{\sim} \cdot (\Sigma_{\mu=1}{}^{n} v_\mu \cdot b_\mu) + \rho^{\sim} \cdot b_{n+1} \in G_1{}^{n+1} \qquad (79)$$

El valor de función $C_2{}^{\sim}$ y el texto codificado C_2 son introducidos a la unidad 122db de comparación de la unidad 122d de decisión. La unidad 122db de comparación determina si el valor de función $C_2{}^{\sim}$ es igual al texto codificado C_2 y saca la determinación (etapa S126).

Si la determinación sacada desde la unidad 122db de comparación indica que el valor de función $C_2{}^{\sim}$ es igual al texto codificado C_2, el valor OR exclusivo $K^{\sim}$ y el texto codificado son C_1 son introducidos a la unidad 122e de descodificación de clave simétrica, y la unidad 122e de descodificación de clave simétrica introduce el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a la función de descodificación de clave simétrica para obtener el siguiente valor descodificado $M^{\sim}$ y saca el valor descodificado $M^{\sim}$ (etapa S127).

$$M^{\sim} = Dec_{K\sim}(C_1) \qquad (80)$$

Si la determinación sacada desde la unidad 122db de comparación indica que el valor de función $C_2{}^{\sim}$ difiere del texto codificado C_2, la unidad 122e de descodificación de clave simétrica saca un mensaje de error ($\perp$) (etapa S128).

Segunda realización

A continuación se describirá una segunda realización del esquema mejorado descrito anteriormente.

Visión general

En primer lugar, se proporcionará una visión general de la segunda realización.

Procesamiento de codificación

Un aparato de codificación para realizar el procesamiento de codificación realiza el siguiente procesamiento.

(ENC-111) Una unidad de selección selecciona un elemento ρ del campo F_q finito y saca el elemento ρ.

(ENC-112) Una unidad de operación de grupo calcula el valor de función $S = g_T{}^{T,\rho} \in G_T$ usando el elemento ρ y saca el valor de función S.

(ENC-113) Una unidad de codificación de clave simétrica usa, como una clave simétrica K, el valor de función R(S) obtenido introduciendo el valor de función S a la función R inyectiva, introduce la clave simétrica K y en texto no codificado M a la función de codificación de clave simétrica para obtener el texto codificado C_1, y saca el texto codificado C_1.

(ENC-211) Una unidad de operación de función introduce el valor de función S y el texto codificado C_1 a la función H resistente a colisiones para obtener el elemento r del campo F_q finito y saca el elemento r.

(ENC-311) Una unidad de operación vectorial calcula el texto codificado $C_2 = r.(\sum_{\mu=1}{}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1{}^{n+1}$ usando los elementos r y ρ y los elementos v_μ (μ = 1, ..., n) del vector n-dimensional $\vec{v}$ compuesto de los elementos del campo F_q finito, y saca el texto codificado C_2.

Procesamiento de descodificación

En un aparato de descodificación para realizar el procesamiento de descodificación, se ejecuta el siguiente procesamiento.

(DEC-101) Un valor que contiene el texto codificado C_1 y el texto codificado $C_2 \in G_1^{n+1}$ es introducido a una unidad de entrada.

(DEC-201) Una unidad de operación de función bilineal introduce el texto codificado $C_2 \in G_1^{n+1}$ y la información de clave $D_1^* = \alpha.(\sum_{\mu=1}^{n} w_\mu.b_\mu^*) + b_{n+1}^*$ a la función e bilineal para obtener el valor de función $S^\sim = e(C_2, D_1^*) \in G_T$ y saca el valor de función $S^\sim$. Si el producto interno $\vec{w}\cdot\vec{v} = 0$, se cumple $S^\sim = S$ debido a las características de las expresiones (3) y (26).

(DEC-301) Una unidad de operación de función introduce el valor de función $S^\sim$ y el texto codificado C_1 a la función H resistente a colisiones para calcular el elemento $r^\sim$ del campo F_q finito y saca el elemento $r^\sim$. Si Se cumple $S\sim = S$ y si el texto codificado C_1 ha sido generado auténticamente, se cumple $r^\sim = r$.

(DEC-401) Una unidad de operación vectorial calcula el valor de función $\lambda = C_2 - r^\sim.(\sum_{\mu=1}^{n} v_\mu.b_\mu) \in G_1^{n+1}$ usando el texto codificado C_2, el elemento $r^\sim$ y los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$, y saca el valor de función λ. Si se cumple $r^\sim = r$, se cumple $\lambda = \rho.bn_{+1} \in G_1^{n+1}$.

(DEC-402) La unidad de operación de función bilineal calcula el valor de función e (λ, D_2^*) introduciendo el valor de función λ y la información de clave auxiliar $D_2^* = \sum_{\mu=1}^{n} \beta_\mu.b_\mu^*$ a la función bilineal e, y saca el valor de función e(λ, D_2^*). Si se cumple $\lambda = \rho.bn_{+1} \in G_1^{n+1}$, se cumple e(λ, D_2^*) = 1 Є G_T debido a la característica de la expresión (3).

(DEC-403) Una unidad de comparación determina si el valor de función e (λ, D_2^*) satisface e (λ, D_2^*) = 1 Є G_T y saca la determinación. En esta realización, la determinación de si se cumple o no la expresión (56) se realiza determinando si el valor de función e(λ, D_2^*) cumple o no e(λ, D_2^*) = 1 Є G_T. Esta determinación corresponde a la determinación (DEC-4).

(DEC-501) Si la determinación sacada desde la unidad de comparación indica que se cumple e(λ, D_2^*) = 1 Є G_T, una unidad de descodificación de clave simétrica usa, como la clave simétrica $K^\sim$, el valor de función R(S) obtenido introduciendo el valor de función $S^\sim$ a la función R inyectiva, introduce la clave simétrica $K^\sim$ y el primer texto codificado C_1 a la función de descodificación de clave simétrica para obtener el valor descodificado $M^\sim$, y saca el valor descodificado $M^\sim$. Si la determinación sacada desde la unidad de comparación indica que no se cumple e(λ, D_2^*) = 1 Є G_T, se saca un mensaje de error (⊥).

Seguridad contra CCA en la segunda realización

Se describirá un caso en el que la estrategia asumida descrita anteriormente es aplicada al esquema de la segunda realización. En este ejemplo, $\tau = 1_F$, y el texto codificado obtenido mediante codificación de clave simétrica del texto no codificado M con la clave simétrica K es $C_1 = K (+) M$.

[1] Un atacante recibe el parámetro público PK.

[2] El atacante pasa el vector $\vec{v}$ y dos grupos de texto no codificado M_0 y M_1 al oráculo de codificación que tiene el parámetro público PK. El oráculo de codificación selecciona un elemento ρ del campo F_q finito (ENC-111), calcula el valor de función $S = g_T^{\tau,\rho} \in G_T$ (ENC-112), usando un valor de función R(S) como la clave simétrica K (ENC-113), y genera el siguiente texto codificado (ENC-113).

$$C_1 = K\ (+)\ M_{bit} \qquad (81)$$

El oráculo de codificación introduce además el valor de función S y el texto codificado C_1 a la función H resistente a colisiones para obtener el elemento r del campo F_q finito (ENC-211). A continuación, el oráculo de codificación genera el texto codificado $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$ usando los elementos r y ρ y los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$ compuesto de los elementos del campo F_q finito (ENC-311) y proporciona los dos grupos de texto codificado (C_1, C_2) al atacante.

[3] Debido a que el atacante no conoce la clave simétrica K = R(S) y/o el valor de función S (valor correspondiente a la clave simétrica K), el atacante no puede generar $C_2 '= r'.(\sum_{\mu=1}^{n} v_\mu.b_\mu)\ \rho.bn_{+1} \in G_1^{n+1}$ correspondiente al elemento r' del campo F_q finito, obtenido introduciendo el valor de función S y el texto codificado C_1 (+) ΔM a la función H resistente a colisiones. Por lo tanto, el atacante pasa la siguiente combinación de C_1 (+) ΔM y texto codificado C_2 correspondiente al texto codificado C_1, a un oráculo de descodificación.

$$(C_1\ (+)\ \Delta M,\ C_2) \qquad (82)$$

El oráculo de descodificación al cual ha sido introducido el texto codificado de expresión (82) (DEC-101) calcula el valor de función $S^{\sim} = e(C_2, D_1^*) \in G_T$ (DEC-201), calcula el elemento $r^{\sim}_{\Delta M}$ del campo F_q finito introduciendo el valor de función $S^{\sim}$ y el texto codificado C_1 (+) ΔM a la función H resistente a colisiones (DEC-301), y obtiene el valor de función $\lambda_{\Delta M} = C_2 \cdot r^{\sim}_{\Delta M} \cdot (\sum_{\mu=1}^{n} v_\mu . b_\mu) \in G_1^{n+1}$ (DEC-401). Debido a que se cumple $r^{\sim}_{\Delta M} \neq r$, se cumple $\lambda_{\Delta M} = (r\ r^{\sim}_{\Delta M}) . (\sum_{\mu=1}^{n} v_\mu . b_\mu) + \rho . bn\ 1 \neq \rho . bn_{+1}$. El oráculo de descodificación calcula un valor de función $e(\lambda_{\Delta M}, D_2^*)$ (DEC-402) y determina si un valor de función (λ, D_2^*) cumple $e(\lambda, D_2^*) = 1 \in G_T$ (DEC-403). Aquí, debido a que se cumple $\lambda_{\Delta M} \neq \rho . bn_{+1}$, no se cumple $e(\lambda, D_2^*) = 1 \in G_T$ (DEC-403). Por lo tanto, el oráculo de descodificación saca un mensaje de error (⊥).

[4]Debido a que el atacante no puede obtener el resultado descodificado de C_1 (+) ΔM, el atacante no es capaz de sacar el bit' = 0 cuando el resultado descodificado es M_0 (+) ΔM o sacar el bit' = 1 cuando el resultado descodificado es M_1 (+)ΔM. Por consiguiente, la estrategia del atacante fracasará.

Detalles

A continuación, se describirá detalladamente la segunda realización. Aunque en este ejemplo se cumple $\tau = 1_F$, esto no limita la presente invención, y τ puede ser un elemento del campo F_q finito diferente de 0_F o 1_F. Cada componente de la segunda realización idéntico a un componente de la primera realización se describirá usando la misma referencia numérica usada en la primera realización.

Estructura general

La Figura 8 es una vista que ilustra la estructura de un sistema criptográfico de la segunda realización.

Tal como se muestra en la Figura 8, el sistema 2 criptográfico de esta realización incluye un aparato 210 de codificación, un aparato 220 de descodificación y un aparato 230 de generación de claves, que están conectados a través de una red 140 de manera que pueden comunicarse entre sí.

Aparato de codificación

La Figura 9 es una vista que ilustra la estructura del aparato 210 de codificación en la Figura 8. Las flechas en la Figura 9 indican el flujo de datos, pero se omiten las flechas para indicar la entrada y salida de datos a y desde una unidad 111 de control y un almacenamiento 113b temporal.

Tal como se muestra en la Figura 9, el aparato 210 de codificación de esta realización incluye la unidad 111 de control, una unidad 212a de selección, una unidad 112d de operación de grupo, una unidad 212c de operación de función, una unidad 212e de generación de clave simétrica, una unidad 112b de codificación de clave simétrica, una unidad 112f de operación vectorial, un almacenamiento113a, el almacenamiento 113b temporal y un transmisor 114.

El aparato 210 de codificación es, por ejemplo, un aparato especial que incluye un programa especial y un ordenador conocido o especializado provisto de una CPU, una RAM, una ROM o similar. La unidad 111 de control, un generador 212a de números aleatorios (correspondiente a la unidad de selección), la unidad 112d operación de grupo, la unidad 212c operación de función, la unidad 212e generación de clave simétrica, la unidad 112b de codificación de clave simétrica y la unidad 112f de operación vectorial son unidades de procesamiento implementadas por la CPU que ejecuta el programa especial, por ejemplo. Al menos algunas de las unidades de procesamiento pueden ser un circuito integrado especializado: Por ejemplo, el generador 212a de números aleatorios puede ser un circuito integrado de generación de números aleatorios conocido.

El aparato 210 de codificación ejecuta el procesamiento bajo el control de la unidad 111 de control. Cada grupo de datos sacado desde cada unidad de procesamiento es almacenado en el almacenamiento 113b temporal o el almacenamiento 113a, los datos almacenados en el almacenamiento 113b temporal o el almacenamiento 113a son introducidos a una unidad de procesamiento y son usados para su procesamiento, cuando sea necesario, aunque estas descripciones de flujo de datos se simplifican a continuación.

Aparato de descodificación

La Figura 10 es una vista que ilustra la estructura del aparato 220 de descodificación en la Figura 8. Las flechas en la Figura 10 indican el flujo de datos, pero se omiten las flechas para indicar la entrada y la salida de datos a y desde una unidad 121 de control y el almacenamiento 123b temporal.

Tal como se muestra en la Figura 10, el aparato 220 de descodificación en esta realización incluye la unidad 121 de control, una unidad 122a operación de función bilineal, unidades 222b y 222c de operación de función, una unidad 222d de decisión, una unidad 122e de descodificación de clave simétrica, un almacenamiento 123a, el almacenamiento 123b temporal, un transmisor 124 y un receptor 125 (correspondiente a la unidad de entrada). La unidad 222d de decisión incluye una unidad 222da de operación vectorial, una unidad 222db de operación de

función bilineal y una unidad 222dc de comparación.

El aparato 220 de descodificación es, por ejemplo, un aparato especial que incluye un programa especial y un ordenador conocido o especializado provisto de una CPU, una RAM, una ROM o similar. La unidad 121 de control, la unidad 122a de operación de función bilineal, las unidades 222b y 222c de operación de función, la unidad 222d de decisión, y la unidad 122e de descodificación de clave simétrica son unidades de procesamiento implementadas por la CPU que ejecuta el programa especial, por ejemplo. Al menos algunas de las unidades de procesamiento pueden ser un circuito integrado especializado.

El aparato 220 de descodificación ejecuta el procesamiento bajo el control de la unidad 121 de control. Cada grupo de datos sacado desde cada unidad de procesamiento es almacenado en el almacenamiento 123b temporal o el almacenamiento 123a, los datos almacenados en el almacenamiento 123b temporal o el almacenamiento 123a son leídos, introducidos a una unidad de procesamiento, y usados para su procesamiento, cuando sea necesario, aunque estas descripciones de flujo de datos se simplifican a continuación.

Aparato de generación de claves

La Figura 11 es una vista que ilustra la estructura del aparato 230 de generación de claves en la Figura 8. Las flechas en la Figura 11 indican el flujo de datos, pero se omiten las flechas para indicar la entrada y salida de datos a y desde una unidad 131 de control y el almacenamiento 133b temporal.

Tal como se muestra en la Figura 11, el aparato 230 de generación de claves de esta realización incluye la unidad 131 de control, un generador 232a de números aleatorios, las unidades 132b y 232c de operación vectorial, un almacenamiento 133a, el almacenamiento 133b temporal, un receptor 134 y un transmisor 135.

El aparato 230 de generación de claves es, por ejemplo, un aparato especial que incluye un programa especial y un ordenador conocido o especializado provisto de una CPU, una RAM, una ROM o similar. La unidad 131 de control, el generador 232a de números aleatorios y las unidades 132b y 232c de operación vectorial son unidades de procesamiento implementadas por la CPU que ejecuta el programa especial, por ejemplo. Al menos algunas de las unidades de procesamiento pueden ser un circuito integrado especializado.

El aparato 230 de generación de claves ejecuta el procesamiento bajo el control de la unidad 131 de control. Cada grupo de datos sacado desde cada unidad de procesamiento es almacenado en el almacenamiento 133b temporal o el almacenamiento 133a, los datos almacenados en el almacenamiento 133b temporal o el almacenamiento 133a son leídos, introducidos a una unidad de procesamiento, y usados para su procesamiento, cuando sea necesario, aunque estas descripciones de flujo de datos se simplifican a continuación.

Ajustes preparatorios

En el ajuste preparatorio de la segunda realización, el parámetro público PK, como en la primera realización, es especificado en el aparato 210 de codificación, el aparato 220 de descodificación y el aparato 230 de generación de claves. La función H_2 resistente a colisiones mostrada en la expresión (34) es especificada en el aparato 210 de codificación y el aparato 220 de descodificación.

La matriz B* de (n+1) filas por (n+1) columnas, que es la información de clave maestra MSK, es almacenada de manera segura en el almacenamiento 133a del aparato 230 de generación de claves. La matriz B* es información secreta y se mantiene en secreto.

El vector n-dimensional $\vec{v} = (v_1, \ldots v_n)$ compuesto de elementos del campo F_q finito, mostrado en la expresión (31), es almacenado en el almacenamiento 113a del aparato 110 de codificación, y el vector n-dimensional $\vec{v} = (v_1, \ldots, v_n)$ y el vector n-dimensional $\vec{w} = (w_1, \ldots, w_n)$, compuestos de los elementos del campo F_q finito, mostrados en la expresión (30), son almacenados en el almacenamiento 123a del aparato 120 de descodificación. Estos vectores cumplen la expresión (66). La información de predicado y la información de atributo pueden ser $\vec{w} = (w_1, \ldots, w_n)$ y $\vec{v} = (v_1, \ldots, v_n)$, respectivamente. De manera alternativa, la información de predicado y la información de atributo pueden ser $\vec{v} = (v_1, \ldots, v_n)$ y $\vec{w} = (w_1, \ldots, w_n)$, respectivamente.

El almacenamiento 113a del aparato 210 de codificación almacena el texto no codificado $M \in \{0, 1\}^*$ a codificar.

Procesamiento de codificación

La Figura 12 es una vista que ilustra el procesamiento de codificación en la segunda realización. A continuación se describirá el procesamiento de codificación en esta realización.

El generador 212a de números aleatorios del aparato 210 de codificación (Figura 9) selecciona primero un elemento

$$\rho \in F_q \qquad (83)$$

del campo F_q finito aleatoriamente, y saca el elemento ρ (etapa S201).

El elemento ρ es introducido a la unidad 112d operación de grupo. Usando el elemento ρ, la unidad 112d de operación de grupo calcula el siguiente valor de función S, y saca el valor de función S (etapa S202).

$$S = g_T{}^{\rho} \in G_T \qquad (84)$$

El valor de función S es introducido a la unidad 212e de generación de clave simétrica. La unidad 212e de generación de clave simétrica introduce el valor de función S a la función R inyectiva para calcular la siguiente clave simétrica K, y almacena la clave simétrica K en el almacenamiento 113a (etapa S203).

$$K = R(S) \in \{0,1\}^k \qquad (85)$$

La clave simétrica K y el texto no codificado M son introducidos a la unidad 112b de codificación de clave simétrica. La unidad 112b de codificación de clave simétrica introduce la clave simétrica K y el texto no codificado M a una función de codificación de clave simétrica Enc para calcular el siguiente texto codificado C_1, y saca el texto codificado C_1 (etapa S204).

$$C_1 = Enc_k(M) \in \{0,1\}^k \qquad (86)$$

El texto codificado C_1 y el valor de función S son introducidos a la unidad 212c de operación de función. La unidad 212c operación de función introduce el valor de función S y el primer texto codificado C_1 a la función H_2 resistente a colisiones para calcular el siguiente elemento r del campo F_q finito, y saca el elemento r (etapa S205).

$$r = H_2(S, C_1) \in F_q \qquad (87)$$

Los elementos de r y ρ y los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$ son introducidos a la unidad 112f de operación vectorial. La unidad 112f de operación vectorial calcula el siguiente texto codificado C_2 usando los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$ y los elementos r y ρ, y saca el texto codificado C_2 (etapa S206).

$$C_2 = r \cdot (\Sigma_{\mu=1}{}^{n} v_\mu \cdot b_\mu) + \rho \cdot b_{n+1} \in G_1{}^{n+1} \qquad (88)$$

El texto codificado C_1 y el texto codificado C_2 son introducidos al transmisor 114. El transmisor 114 envía un valor que los contiene a través de la red 140 al aparato 120 de descodificación (etapa S207).

Procesamiento de generación de información de clave

El procesamiento de generación de información de clave se lleva a cabo especialmente cuando ni la información de clave D_1* ni la información de clave auxiliar D_2* están almacenadas en el almacenamiento 123a del aparato 220 de descodificación. Cuando la información de clave D_1* y la información de clave auxiliar D_2* están almacenadas en el almacenamiento 123a del aparato 220 de descodificación, este procesamiento puede ser omitido.

La Figura 13 es una vista que ilustra el procesamiento de generación de información de clave en la segunda realización. A continuación se describirá el procesamiento de generación de información de clave en esta realización.

El vector n-dimensional $\vec{w} = (w_1, ..., w_n)$ almacenado en el aparato 220 de descodificación (Figura 10) es leído e

introducido al transmisor 124. El transmisor 124 envía el vector n-dimensional $\vec{w} = (w_1, ..., w_n)$ a través de la red 140 al aparato 230 de generación de claves (etapa S111).

El vector n-dimensional $\vec{w} = (w_1, ..., w_n)$ enviado es recibido por el receptor 134 del aparato 230 de generación de claves (Figura 11) (etapa S112). Esto hace que el generador 232a de números aleatorios y la unidad 132b de operación vectorial del aparato 230 de generación de claves generen la información de clave D_1^* mostrada en la expresión (75) mediante la realización de procesamiento en las etapas S113 y S114 de la primera realización y saquen la información de clave D_1^*.

El generador 232a de números aleatorios genera un número aleatorio

$$\beta_\mu (\mu=1,...,n) \in F_q \qquad (89)$$

de un elemento del campo F_q finito y saca el número aleatorio β_μ (etapa S215).

El número aleatorio β_μ y la matriz B* son introducidos a la unidad 232c de operación vectorial. Usando los mismos, la unidad 232b de operación vectorial genera la siguiente información de clave auxiliar D_2^*, y saca la información de clave auxiliar D_2^* (etapa S216).

$$D_2^* = \Sigma_{\mu=1}^{n} \beta_\mu \cdot b_\mu^* \in G_2^n \qquad (90)$$

La información de clave D_1^* y la información de clave auxiliar D_2^* son introducidas al transmisor 135. El transmisor 135 envía la información de clave D_1^* y la información de clave auxiliar D_2^* a través de la red 140 al aparato 220 de descodificación (etapa S217).

La información de clave D_1^* y la información de clave auxiliar D_2^* enviadas son recibidas por el receptor 125 del aparato 220 de descodificación (Figura 10) y son almacenadas en el almacenamiento 123a (etapa S218).

Procesamiento de descodificación

La Figura 14 es una vista que ilustra el procesamiento de descodificación en la segunda realización. A continuación se describirá el procesamiento de descodificación en esta realización.

El receptor 125 del aparato 220 de descodificación (Figura 10) recibe el valor que contiene el texto codificado C_1 y el texto codificado $C_2 \in G_1^{n+1}$ (etapa S221).

El texto codificado $C_2 \in G_1^{n+1}$ y la información de clave D_1^* son introducidos a la unidad 122a de operación de función bilineal. La unidad 122a de operación de función bilineal introduce el texto codificado $C_2 \in G_1^{n+1}$ y la información de clave D_1^* a una función e bilineal para calcular el valor de función siguiente $\tilde{S}$ y saca el valor de función $\tilde{S}$ (etapa S222).

$$\tilde{S} = e(C_2, D_1^*) \in G_T \qquad (91)$$

El valor de función $\tilde{S}$ y el texto codificado C_1 son introducidos a la unidad 222c operación de función. La unidad 222c de operación de función calcula el siguiente elemento $\tilde{r}$ del campo F_q finito introduciendo el valor de función $\tilde{S}$ y el primer texto codificado C_1 a la función H_2 resistente a colisiones y saca el elemento $\tilde{r}$ (etapa S223).

$$\tilde{r} = H_2(\tilde{S}, C_1) \in F_q \qquad (92)$$

El texto codificado C_2, el elemento $\tilde{r}$ y los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$ son introducidos a la unidad de 222da de operación vectorial de la unidad 222d de decisión. Usando el texto codificado C_2, el elemento $\tilde{r}$ y los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$, la unidad 222da de operación vectorial calcula el siguiente valor de función λ, y saca el valor de función λ (etapa S224).

$$\lambda = C_2 - r^{\sim} \cdot (\Sigma_{\mu=1}^{n} v_\mu \cdot b_\mu) \in G_1^{n+1} \qquad (93)$$

El valor de función λ y la información de clave auxiliar D_2* son introducidos a la unidad 222db de operación de función bilineal de la unidad 222d de decisión. La unidad 222db de operación de función bilineal introduce el valor de función λ y la información de clave auxiliar D_2* a la función e bilineal para calcular el siguiente valor de función (λ, D_2*), y saca el valor de función e (λ, D_2*) (etapa S225).

$$e(\lambda, D_2^*) \in G_T \qquad (94)$$

El valor de función e (λ, D_2*) es introducido a la unidad 222dc de comparación de la unidad 222d de decisión. La unidad 222dc de comparación determina si el valor de función e(λ, D_2*) cumple la relación siguiente, y saca la determinación (etapa S226).

$$e(\lambda, D_2^*) = 1 \in G_T \qquad (95)$$

Si la determinación sacada desde la unidad 222dc de comparación indica que se cumple e(λ, D_2*) = 1 Є G_T, el valor de función $S^{\sim}$ es introducido a la unidad 222b de operación de función. La unidad 222b operación de función introduce el valor de función $S^{\sim}$ a la función R inyectiva para obtener la siguiente clave simétrica $K^{\sim}$, y saca la clave simétrica $K^{\sim}$ (etapa S227).

$$K^{\sim} = R(S^{\sim}) \qquad (96)$$

La clave simétrica $K^{\sim}$ y el texto codificado C_1 son introducidos a la unidad 222b de descodificación de clave simétrica, y la unidad 222b de descodificación de clave simétrica introduce la clave simétrica $K^{\sim}$ y el texto codificado C_1 a la función de descodificación de clave simétrica Dec para calcular el siguiente valor descodificado $M^{\sim}$, y saca el valor descodificado $M^{\sim}$ (etapa S228).

$$M^{\sim} = Dec_K{}^{\sim}(C_1) \qquad (97)$$

Si la determinación sacada desde la unidad 222dc de comparación indica que no se cumple e(λ, D_2*) = 1 Є G_T, la unidad 122e de descodificación de clave simétrica saca un mensaje de error (⊥) (etapa S229) .

Modificaciones, etc.

La presente invención no se limita a las realizaciones descritas anteriormente. Por ejemplo, aunque la clave simétrica K en la etapa S101 de la primera realización es un número aleatorio de secuencia binaria, la clave simétrica K puede ser un valor seleccionado de entre una pluralidad de secuencias binarias predeterminadas según una regla determinada. Aunque α en la etapa S113 de la primera realización es un número aleatorio que es el elemento del campo F_q finito, α puede ser seleccionado de entre los elementos del campo F_q finito según una regla determinada. Los valores generados a partir de los otros números aleatorios pueden ser modificados de manera similar.

Los valores de función obtenidos por la expresión (69) se usan directamente como r y ρ en la primera realización. Los valores de función obtenidos introduciendo los valores de función obtenidos por la expresión (69) a una función adicional pueden ser usados como r y ρ. En ese caso, los valores de función obtenidos introduciendo los valores de función obtenidos por la expresión (78) a la función adicional pueden ser usados como $r^{\sim}$ y $\rho^{\sim}$. El valor de función obtenido por la expresión (87) se usa directamente como r en la segunda realización. Un valor de función obtenido introduciendo el valor de función obtenido por la expresión (87) a una función adicional puede ser usado como r. En ese caso, un valor de función obtenido introduciendo el valor de función obtenido por la expresión (92) a la función adicional se usa como $r^{\sim}$.

En el procedimiento de descodificación en la primera realización y la segunda realización, se saca un mensaje de error cuando no se cumple la condición de determinación en la etapa S126 o S226. Cuando la condición de determinación no se cumple, el aparato de descodificación puede sacar un número aleatorio no relacionado con el resultado descodificado o puede no sacar nada.

Los diversos tipos de procesamiento descritos anteriormente no deben ser ejecutados siempre en el orden en el que se han descrito y puede ser realizados en paralelo o de manera independiente según las capacidades de procesamiento de las unidades que ejecutan el procesamiento o cuando sea necesario. Por ejemplo, el procesamiento en las etapas S105 y S106 en la primera realización puede ser ejecutado en paralelo, y el procesamiento en la etapa S105 puede ser ejecutado después del procesamiento en la etapa S106.

Cada operación definida sobre el campo F_q finito puede ser reemplazada por una operación definida en un anillo Z_q finito de orden q. Un ejemplo de sustitución de cada operación definida sobre el campo F_q finito por una operación definida en el anillo Z_q finito es un método que permite que q no sea un número primo o una potencia de un número primo. Por supuesto, es posible la adición de modificaciones dentro del alcance de la presente invención.

Cuando la estructura descrita anteriormente es implementada por un ordenador, los detalles de procesamiento de las funciones que deberían ser proporcionadas por cada aparato se describen en un programa. Cuando el programa es ejecutado por un ordenador, las funciones de procesamiento descritas anteriormente se implementan en el ordenador.

El programa que contiene los detalles de procesamiento puede ser grabado en un medio de almacenamiento legible por ordenador. El medio de almacenamiento legible por ordenador puede ser cualquier tipo de medio, tal como un dispositivo de almacenamiento magnético, un disco óptico, un medio de almacenamiento magneto-óptico y una memoria basada en semiconductores.

El programa es distribuido mediante la venta, transferencia o alquiler de un medio de grabación portátil, tal como un DVD o un CD-ROM que tiene el programa grabado en el mismo, por ejemplo. El programa puede ser distribuido también almacenando el programa en una unidad de almacenamiento de un equipo servidor y transfiriendo el programa desde el ordenador servidor a otro ordenador a través de la red.

Un ordenador que ejecuta este tipo de programa primero almacena el programa grabado en el medio de grabación portátil o el programa transferido desde el ordenador servidor en su unidad de almacenamiento. A continuación, el ordenador lee el programa almacenado en su unidad de almacenamiento y ejecuta el procesamiento según el programa leído. En una forma diferente de ejecución del programa, el equipo puede leer el programa directamente desde el medio de grabación portátil y puede ejecutar el procesamiento según el programa, o el ordenador puede ejecutar el procesamiento según el programa cada vez que el ordenador recibe el programa transferido desde el servidor ordenador. De manera alternativa, el procesamiento puede ser ejecutado por un servicio denominado proveedor de servicios de aplicaciones (Application Service Provider, ASP), en el que la función de procesamiento es implementada simplemente proporcionando una instrucción de ejecución de programa y obteniendo los resultados sin transferir el programa desde el ordenador servidor al ordenador.

DESCRIPCIÓN DE LOS NÚMEROS DE REFERENCIA

1, 2: Sistemas criptográficos
110, 210: Aparatos de codificación
120, 220: Aparatos de descodificación.

REIVINDICACIONES

1. Un aparato (110, 210) de codificación (encriptación), que comprende:

una unidad (112b) de codificación de clave simétrica adaptado para calcular un texto codificado C_1, introduciendo un de texto no codificado M y una clave simétrica K a una función de codificación de clave simétrica y para sacar el texto codificado C_1;
una unidad (112c, 212c) de operación de función adaptada para calcular un valor de función de una función H resistente a colisiones introduciendo un valor que contiene un valor correspondiente a la clave simétrica K y un valor correspondiente al texto codificado C_1 a la función H; y
una unidad (112f) de operación vectorial adaptada para calcular un segundo texto codificado $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$ usando un valor r correspondiente al valor de función de la función H, un valor ρ correspondiente a la clave simétrica K, elementos v_μ donde $\mu = 1, ..., n$ y $n \geq 1$ de un vector n-dimensional $\vec{v} = (v_1, ..., v_n)$, y vectores base n+1 dimensionales $b_i \in G_1^{n+1}$, donde $i = 1, ..., n+1$, compuestos de n+1 elementos de un grupo cíclico G_1 y para sacar el texto codificado C_2.

2. Aparato (110) de codificación según la reivindicación 1, en el que la clave simétrica K es una secuencia binaria; y

la unidad de operación de función está adaptada para calcular los valores r y ρ introduciendo la clave simétrica K y el texto codificado C_1 a la función H resistente a colisiones;
en el que el aparato (110) de codificación comprende además:
una unidad (112d) de operación de grupo adaptada para calcular un valor de función $S = g_T^{\tau.\rho} \in G_T$ usando el valor ρ, donde g_T es un generador de un grupo cíclico G_T y τ es una constante diferente a un elemento de identidad aditivo; y
una unidad (112e) de operación OR exclusivo adaptada para calcular un tercer texto codificado C_3 y para sacar el tercer texto codificado C_3, en el que el tercer texto codificado C_3 es un OR exclusivo de un valor de función R(S) y la clave simétrica K, en el que el valor de función R(S) es una secuencia binaria obtenida introduciendo el valor de función S a una función R inyectiva

3. Aparato (210) de codificación según la reivindicación 1, que comprende además:

una unidad (212a) de selección adaptada para seleccionar el valor ρ;
una unidad (112d) de operación de grupo adaptada para calcular un valor de función $S = g_T^{\tau.\rho} \in G_T$ usando el valor ρ, donde g_T es un generador de un grupo cíclico G_T y τ es una constante diferente a un elemento de identidad aditivo; y
una unidad (212e) de generación de clave simétrica adaptada para calcular un valor de función R(S) como la clave simétrica K introduciendo el valor de función S a una función R inyectiva, en el que
la unidad (212c) de operación de función está adaptada para calcular el valor r introduciendo el valor de función S y el texto codificado C_1 a la función H resistente a colisiones

4. Aparato (110) de codificación según la reivindicación 2, en el que la función R inyectiva es una función cuasi aleatoria.

5. Aparato (210) de codificación según la reivindicación 3, en el que la función R inyectiva es una función cuasi aleatoria.

6. Aparato (110, 210) de codificación según la reivindicación 1, en el que la función H resistente a colisiones es una función cuasi aleatoria.

7. Aparato (110) de codificación según la reivindicación 2, en el que la función H resistente a colisiones es una función cuasi aleatoria.

8. Aparato (210) de codificación según la reivindicación 3, en el que la función H resistente a colisiones es una función cuasi aleatoria.

9. Aparato (110) de codificación según la reivindicación 2, en el que el grupo cíclico G_1 y el grupo cíclico G_T son conjuntos en los que se define una función e bilineal no degenerada, en el que la función e bilineal no degenerada saca un único elemento del grupo cíclico G_T en respuesta a la introducción de n+1 elementos γ_L, donde $L = 1, ...,$ n+1, del grupo cíclico G_1 y n+1 elementos γ_L^*, donde $L = 1, ..., n+1$, de un grupo cíclico G_2.

10. Aparato (110) de codificación según la reivindicación 9, en el que el los valores r y ρ y la constante τ son elementos de un campo F_q finito; y

el orden de cada uno de los grupos cíclicos G_1 y G_2 es igual al orden q del campo F_q finito, donde $q \geq 1$.

11. Aparato (110) de codificación según la reivindicación 9 o 10, en el que la función e bilineal saca un elemento $e = \Pi_{i=1}^{n+1}$ Par (γ_L, γ_L*) del grupo cíclico G_T en respuesta a la introducción de los n+1 elementos γ_L, donde L = 1, ..., n+1, del grupo cíclico G_1 y los n+1 elementos γ_L*, donde L = 1, ..., n+1, del grupo cíclico G_2, donde Par es una función bilineal no degenerada que traduce una combinación de un único elemento del grupo cíclico G_1 y un único elemento del grupo cíclico G_2 a un único elemento del grupo cíclico G_T.

12. Aparato (210) de codificación según la reivindicación 3, en el que el grupo cíclico G_1 y el grupo cíclico G_T son conjuntos en los que se define una función e bilineal no degenerada, en el que la función e bilineal no degenerada saca un único elemento del grupo cíclico G_T en respuesta a la introducción de n+1 elementos γ_L, donde L = 1, ..., n+1, del grupo cíclico G_1 y n+1 elementos γ_L*, donde L = 1, ..., n+1, de un grupo cíclico G_2.

13. Aparato (210) de codificación según la reivindicación 12, en el que los valores r y ρ y la constante τ son elementos de un campo F_q finito; y
el orden de cada uno de los grupos cíclicos G_1 y G_2 es igual al orden q del campo F_q finito, donde $q \geq 1$.

14. Aparato (210) de codificación según la reivindicación 12 o 13, en el que la función e bilineal saca un elemento $e = \Pi_{i=1}^{n+1}$ Par (γ_L, γ_L*) del grupo cíclico G_T en respuesta a la introducción de los n+1 elementos γ_L, donde L = 1, ..., n+1, del grupo cíclico G_1 y los n+1 elementos γ_L*, donde L = 1, ..., n+1, del grupo cíclico G_2, donde Par es una función bilineal no degenerada que traduce una combinación de un único elemento del grupo cíclico G_1 y un único elemento del grupo cíclico G_2 a un único elemento del grupo cíclico G_T.

15. Un aparato (120, 220) de descodificación (desencriptación) que comprende:

una unidad (125) de entrada a la cual se introducen un valor que contiene un texto codificado C_1 y un segundo texto codificado $C_2 \in G_1^{n+1}$;
una unidad (122a) de operación de función bilineal adaptada para calcular un valor de función $S^{\sim} = e(C_2, D_1^*) \in G_T$ introduciendo el segundo texto codificado $C_2 \in G_1^{n+1}$ y la información de clave $D_1^* \in G_2^{n+1}$ a una función e bilineal;
una unidad (122c, 222c) de operación de función adaptada para calcular un valor de función de una función H resistente a colisiones introduciendo un valor correspondiente al valor de función $S^{\sim}$ y el texto codificado C_1 a la función H; y
una unidad (122d, 222d) de decisión adaptada para determinar si el segundo texto codificado C_2, un valor $\rho^{\sim}$, un valor $r^{\sim}$ correspondiente al valor de función de la función H, los elementos v_μ ($\mu = 1,. .., n$) de un vector n-dimensional $\vec{v} = (v_1, ..., v_n)$, y los vectores base $b_i \in G_1^{n+1}$, donde i = 1, ..., n+1, cumplen $C_2 = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$ y para sacar un resultado de la determinación, en el que
la función e bilineal es una función bilineal no degenerada que saca un único elemento de un grupo cíclico G_T en respuesta a la introducción de n+1 elementos γ_L, donde L = 1, ..., n+1 y $n \geq 1$, de un grupo cíclico G_1 y n+1 elementos de γ_L*, donde L = 1, ..., n+1, de un grupo cíclico G_2; en el que cada uno de los vectores base $b_i \in G_1^{n+1}$, donde i = 1, ..., n+1, es un vector base n+1 dimensional compuesto de n+1 elementos del grupo cíclico G_1; en el que cada $b_i^* \in G_2^{n+1}$, donde i = 1, ..., n+1, es un vector base n+1 dimensional compuesto de n+1 elementos del grupo cíclico G_2; un valor de función obtenido introduciendo los elementos de los vectores base $b_i \in G_1^{n+1}$, donde i = 1, ..., n+1, y los elementos de los vectores base $b_j \in G_2^{n+1}$, donde j = 1, ..., n+1, a la función e bilineal se expresa como $g_T^{\tau.\delta(i,j)} \in G_T$ usando la función delta de Kronecker δ(i, j); τ es una constante diferente de un elemento de identidad aditivo; g_T es un generador del grupo cíclico G_T; $\vec{w} = (w_1, ..., w_n)$ es un vector n dimensional; w_μ, donde $\mu = 1, ..., n$, son elementos del vector n-dimensional $\vec{w} = (w_1, ..., w_n)$; α es un valor seleccionado; y la información de clave $D_1^* \in G_2^{n+1}$ es $D_1^* = \alpha.(\sum_{\mu=1}^{n} w_\mu.b_\mu^*) + b_{n+1}^* \in G_2^{n+1}$.

16. Aparato (120) de descodificación según la reivindicación 15, en el que el valor que contiene el texto codificado C_1 y el segundo texto codificado $C_2 \in G_1^{n+1}$, introducido a la unidad (125) de entrada, contiene además un tercer texto codificado C_3 que es una secuencia binaria;
el aparato (120) de descodificación comprende además una unidad (122b) de operación OR exclusivo adaptada para calcular un valor OR exclusivo $K^{\sim}$ del tercer texto codificado C_3 y un valor de función $R(S^{\sim})$ que es una secuencia binaria obtenida introduciendo el valor de función $S^{\sim}$ a una función R inyectiva;
la unidad (122c) de operación de función está adaptada para calcular los valores $r^{\sim}$ y $\rho^{\sim}$ introduciendo el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a la función H resistente a colisiones; y
la unidad (122d) de decisión incluye:

una unidad (122da) de operación vectorial adaptada para calcular un valor de función $C_2^{\sim} = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$ usando los elementos v_μ ($\mu = 1, ..., n$) del vector n-dimensional $\vec{v}$ y los valores $r^{\sim}$ y $\rho^{\sim}$; y

una unidad (122db) de comparación adaptada para determinar si el valor de función $C_2\tilde{}$ es igual al segundo texto codificado C_2;

el aparato de descodificación comprende además una unidad (122e) de descodificación de clave simétrica adaptada para usar el valor OR exclusivo $K\tilde{}$ como una clave simétrica, para calcular un valor descodificado $M\tilde{}$ introduciendo el valor OR exclusivo $K\tilde{}$ y el texto codificado C_1 a una función de descodificación de clave simétrica, y para sacar el valor descodificado $M\tilde{}$, cuando la determinación por parte de la unidad de comparación indica que el valor de función $C_2\tilde{}$ es igual al segundo texto codificado C_2.

17. Aparato (220) de descodificación según la reivindicación 15, en el que la unidad (222c) de operación de función está adaptada para calcular el valor $r\tilde{}$ introduciendo el valor de función $S\tilde{}$ y el texto codificado C_1 a la función H resistente a colisiones; y
la unidad (222d) de decisión incluye:

una unidad (222da) de operación vectorial adaptada para calcular un valor de función $\lambda = C_2\text{-}r\tilde{}.\sum_{\mu=1}^{n} v_\mu.b_\mu \in G_1^{n+1}$ usando el segundo texto codificado C_2, el valor $r\tilde{}$ y los elementos v_μ, donde $\mu = 1, ..., n$, del vector n dimensional $\vec{v}$;
una segunda unidad (222db) de operación de función bilineal adaptada para calcular un valor e de función (λ, D_2^*) introduciendo el valor de función λ y la información de clave auxiliar $D_2^* = \sum_{\mu=1}^{n}\beta_\mu.b_\mu^*$ relacionada con un valor β_μ, donde $\mu = 1, ..., n$, a la función e bilineal; y
una unidad (222dc) de comparación adaptada para determinar si el valor de función $e(\lambda, D_2^*)$ cumple $e(\lambda, D_2^*) = 1 \in G_T$;

el aparato (220) de descodificación comprende además una unidad (122e) de descodificación de clave simétrica que está adaptada para usar un valor de función $R(S\tilde{})$ obtenido introduciendo el valor de función $S\tilde{}$ a una función R inyectiva como una clave simétrica $K\tilde{}$ si la determinación por la unidad de comparación indica que se cumple $e(\lambda, D_2^*) = 1 \in G_T$, para calcular un valor descodificado $M\tilde{}$ introduciendo la clave simétrica $K\tilde{}$ y el texto codificado C_1 a una función de descodificación de clave simétrica, y para sacar el valor descodificado $M\tilde{}$.

18. Aparato (120, 220) de descodificación según la reivindicación 16 o 17, en el que la función R inyectiva es una función cuasi aleatoria.

19. Aparato (120, 220) de descodificación según cualquiera de las reivindicaciones 15 a 17, en el que la función H resistente a colisiones es una función cuasi aleatoria.

20. Aparato (120, 220) de descodificación según una cualquiera de las reivindicaciones 15 a 17, en el que el vector base b_i cumple $b_i = \sum_{j=1}^{n+1} \chi_{i,j}.a_j$, donde $\chi_{i,j}$ indica el elemento en la i-ésima fila y la j-ésima columna de una matriz X de (n+1) filas por (n+1) columnas, y $a_i \in G_1^{n+1}$, donde $i = 1, ..., n+1$, indica un vector base n+1 dimensional que tiene $\kappa_1.g_1$ como el i-ésimo elemento y que tiene los elementos identidad del grupo cíclico G_1 como los n elementos restantes, en el que g_1 es un generador del grupo cíclico G_1 y κ_1 es una constante diferente de un elemento de identidad aditivo; y
el vector base b_i^* cumple $b_i^* = \sum_{j=1}^{n+1} \chi_{i,j}^*.a_j^*$ donde $\chi_{i,j}^*$ indica el elemento en la i-ésima fila y la j-ésima columna de una matrix X* de (n+1) filas por (n+1) columnas y $a_i^* \in G_2^{n+1}$, donde $i = 1, ..., n+1$, indica un vector base n+1 dimensional que tiene $\kappa_2.g_2$ como el i-ésimo elemento y que tiene los elementos identidad del grupo cíclico G_2 como los n elementos restantes, en el que g_2 es un generador del grupo cíclico G_2 y κ_2 es una constante diferente al elemento de identidad aditivo.

21. Aparato (120, 220) de descodificación según una cualquiera de las reivindicaciones 15 a 17, en el que los valores $r\tilde{}$ y $\rho\tilde{}$, la constante τ y el valor α son elementos de un campo F_q finito; y
el orden de cada uno de los grupos cíclicos G_1 y G_2 es igual al orden q del campo F_q finito, donde $q \geq 1$.

22. Aparato (120, 220) de descodificación según una cualquiera de las reivindicaciones 15 a 17, en el que la función e bilineal saca un elemento $e=\Pi_{L=1}^{n+1}$ Par (γ_L, γ_L^*) del grupo cíclico G_T en respuesta a la introducción de los n+1 elementos γ_L, donde $L = 1, ..., n+1$, del grupo cíclico G_1 y los n+1 elementos γ_L^*, donde $L = 1, ..., n+1$, del grupo cíclico G_2, donde Par es una función bilineal no degenerada que traduce una combinación de un único elemento del grupo cíclico G_1 y un único elemento del grupo cíclico G_2 a un único elemento del grupo cíclico G_T.

23. Un método de codificación que comprende:

(A) una etapa de cálculo, en unos primeros medios de un aparato (110, 210) de codificación, de un texto codificado C_1, introduciendo un texto no codificado M y una clave simétrica K a una función de codificación de clave simétrica y sacando el texto codificado C_1;

(B) una etapa de cálculo, en unos segundos medios del aparato (110, 210) de codificación, de un primer valor de función de una función H resistente a colisiones introduciendo un valor que contiene un valor correspondiente a la clave simétrica K y un valor correspondiente al texto codificado C_1 a la función H; y
(C) una etapa de cálculo, en unos terceros medios del aparato (110, 210) de codificación, de un segundo texto codificado $C_2 = r.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho.bn_{+1} \in G_1^{n+1}$ usando un valor r correspondiente al primer valor de función de la función H, un valor ρ correspondiente a la clave simétrica K, elementos v_μ, donde μ = 1, ..., n y n 1, de un vector n dimensional $\vec{v} = (v_1, ..., v_n)$, y vectores base n+1 dimensionales $b_i \in G_1^{n+1}$, donde i = 1, ..., n+1, compuestos de n+1 elementos de un grupo cíclico G_1 y sacando el segundo texto codificado C_2;

24. Método de codificación según la reivindicación 23, en el que la clave simétrica K es una secuencia binaria; y
la etapa (B) incluye una etapa de cálculo, en unos cuartos medios del aparato de codificación, de los valores r y ρ introduciendo la clave simétrica K y el texto codificado C_1 a la función H resistente a colisiones;
en el que el método de codificación comprende:

una etapa de cálculo, en unos quintos medios del aparato de codificación, de un valor de función $S = g_T^{\tau,\rho} \in G_T$ usando el valor ρ; en el que g_T es un generador del grupo cíclico G_T y τ es una constante diferente de un elemento de identidad aditivo; t
una etapa de cálculo, en unos sextos medios del aparato de codificación, de un tercer texto codificado C_3 y sacar el tercer texto codificado C_3, en el que el tercer texto codificado C_3 es un valor OR exclusivo de un valor de función R(S) y la clave simétrica K, en el que el tercer texto codificado C_3 es un secuencia binaria obtenida introduciendo el valor de función S a una función R inyectiva.

25. Método de codificación según la reivindicación 23, que comprende además:

una etapa de selección del valor ρ en unos séptimos medios del aparato de codificación;
una etapa de cálculo, en unos octavos medios del aparato de codificación, de un valor de función $S = g_T^{\tau,\rho} \in G_T$ usando el valor ρ, donde g^T es un generador de un grupo cíclico G^T y τ es una constante diferente de un elemento de identidad aditivo; y
una etapa de cálculo, en unos novenos medios del aparato de codificación, de un valor de función R(S) como la clave simétrica K introduciendo el valor de función S a un función R inyectiva en el que
la etapa (B) incluye una etapa de cálculo, en unos décimos medios del aparato de codificación, del valor r introduciendo el valor de función S y el texto codificado C_1 a la función H resistente a colisiones.

26. Método de descodificación que comprende:

(D) una etapa de introducción de un valor que contiene un texto codificado C_1 y un segundo texto codificado $C_2 \in G_1^{n+1}$ a un aparato (120, 220) de descodificación;
(E) una etapa de cálculo, en unos primeros medios del aparato (120, 220) de descodificación, de un valor de función $S^{\sim} = e(C_2, D_1^*) \in G_T$ introduciendo el segundo texto codificado $C_2 \in G_1^{n+1}$ y la información de clave D_1^* a una función e bilineal;
(F) una etapa de cálculo, en unos segundos medios del aparato de descodificación, de un segundo valor de función de la función H resistente a colisiones introduciendo un valor correspondiente al valor de función $S^{\sim}$ y el texto codificado C_1 a la función H; y
(G) una etapa de determinación, en el aparato (120, 220) de descodificación, de si el segundo texto codificado C_2, un valor $r^{\sim}$ correspondiente al segundo valor de función de la función H, un valor $\rho^{\sim}$, los elementos v_μ, (μ = 1, …, n) de un vector n-dimensional $\vec{v} = (v_1, \ldots, V_n)$, y los vectores base $b_i \in G_1^{n+1}$, donde i = 1, ..., n+1, cumplen $C_2 = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$, y sacar un resultado de la determinación, en el que
la función e bilineal es una función bilineal no degenerada que saca un único elemento de un grupo cíclico G_T en respuesta a la introducción de n+1 elementos γ_L, donde L = 1, ..., n+1 y n ≥ 1, de un grupo cíclico G_1 y n+1 elementos γ_L^*, donde L = 1, ..., n+1, de un grupo cíclico G_2; cada uno de los vectores base $b_i \in G_1^{n+1}$, donde i = 1, ..., n+1) es un vector base n+1 dimensional compuesto de n+1 elementos del grupo cíclico G_1; cada uno de los vectores base $b_i^* \in G_2^{n+1}$, donde i = 1, ..., n+1, es un vector base n+1 dimensional compuesto de n+1 elementos del grupo cíclico G_2; un valor de función obtenido introduciendo los elementos de los vectores base $b_i \in G_1^{n+1}$, donde i = 1, ..., n+1, y los elementos de los vectores base $b_j^* \in G_2^{n+1}$, donde j = 1, ..., n+1, a la función e bilineal se expresa como $g_T^{\tau.\delta(i,j)} \in G_T$ usando la función delta de Kronecker δ(i, j); τ es una constante diferente de un elemento de identidad aditivo; g_T es un generador del grupo cíclico G_T; $\vec{w} = (w_1, ..., w_n)$ es un vector n-dimensional; w_μ, donde μ = 1, ..., n, son elementos del vector n-dimensional $\vec{w} = (w_1, ..., w_n)$; α es un valor seleccionado y la información de clave $D_1^* \in G_1^{n+1}$ es $D_1^* = \alpha.(\sum_{\mu=1}^{n} w_\mu.b_\mu^*) + b_{n+1}^* \in G_2^{n+1}$.

27. Método de descodificación según la reivindicación 26, en el que el valor que contiene el texto codificado C_1 y el

segundo texto codificado $C_2 \in G_1^{n+1}$, introducido en la etapa (D), contiene además un tercer texto codificado C_3 que es una secuencia binaria;
en el que el método de descodificación comprende además una etapa de cálculo, en unos cuartos medios del aparato de descodificación, de un valor OR exclusivo $K^{\sim}$ del tercer texto codificado C_3 y un valor de función $R(S^{\sim})$ que es una secuencia binaria obtenida introduciendo el valor de función $S^{\sim}$ a una función R inyectiva;
la etapa (F) incluye una etapa de cálculo, en unos quintos medios del aparato de descodificación, de los valores $r^{\sim}$ y $\rho^{\sim}$ introduciendo el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a la función H resistente a colisiones; y
la etapa (g) incluye:

una etapa (G-1) de cálculo, en unos sextos medios del aparato de descodificación, de un valor de función $C_2^{\sim} = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$ usando los elementos v_μ, donde $\mu = 1, ..., n$, del vector n-dimensional $\vec{v}$ y los valores $r^{\sim}$ y $\rho^{\sim}$; y
una etapa (G-2) de determinación, en unos séptimos medios del aparato de descodificación, de si el valor de función $C_2^{\sim}$ es igual al segundo texto codificado C_2,

en el que el método de codificación comprende además una etapa de uso, en unos octavos medios del aparato de descodificación, del valor el valor OR exclusivo $K^{\sim}$ como una clave simétrica, calcular un valor descodificado $M^{\sim}$ introduciendo el valor OR excusivo $K^{\sim}$ y el texto codificado C_1 a una función de descodificación de clave simétrica, y sacar el valor descodificado $M^{\sim}$, cuando la determinación en la etapa (G-2) indica que el valor de función $C_2^{\sim}$ es igual al segundo texto codificado C_2.

28. Método de descodificación según la reivindicación 26, en el que la etapa (F) incluye una etapa de cálculo, en unos novenos medios del aparato de descodificación, del valor $r^{\sim}$ introduciendo el valor de función $S^{\sim}$ y el texto codificado C_1 a la función H resistente a colisiones; y
la etapa (G) incluye:

(G-1) una etapa de cálculo, en unos décimos medios del aparato de descodificación, de un valor de función $\lambda = C_2\text{-}r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) \in G_1^{n+1}$ usando el segundo texto codificado C_2, el valor $r^{\sim}$ y los elementos v_μ, donde $\mu = 1, ..., n$, del vector n-dimensional $\vec{v}$;
(G-2) una etapa de cálculo, en unos décimo primeros medios del aparato de descodificación, de un valor de función $e(\lambda, D_2^*)$ introduciendo el valor de función λ y la información de clave auxiliar $D_2^* = \sum_{\mu=1}^{n}\beta_\mu.b_\mu^*$ relacionada con un valor β_μ, donde $\mu = 1, ..., n$, a la función e bilineal; y
(G-3) una etapa de determinación, en unos décimo segundos medios del aparato de descodificación, de si el valor de función $e(\lambda, D_2^*)$ cumple $e(\lambda, D_2^*) = 1 \in G_T$;

en el que el método de descodificación comprende además una etapa de usar, en unos décimo terceros medios del aparato de descodificación, de un valor de función $R(S^{\sim})$ obtenido introduciendo el valor de función $S^{\sim}$ a una función R inyectiva, como una clave simétrica, calcular un valor descodificado $M^{\sim}$ introduciendo la clave simétrica $K^{\sim}$ y el texto codificado C_1 a una función de descodificación de clave simétrica, y sacar el valor descodificado $M^{\sim}$, cuando la determinación en la etapa (G-3) indica que se cumple $e(\lambda, D_2^*) = 1 \in G_T$.

29. Un método de seguridad que comprende:

las etapas del método de codificación según la reivindicación 23; y
las etapas del método de descodificación según la reivindicación 26.

30. Método de seguridad según la reivindicación 29, en el que la etapa (B) incluye una etapa de cálculo de los valores r y ρ introduciendo la clave simétrica K y el texto codificado C_1 a la función H resistente a colisiones;
la etapa (C) incluye:

una etapa de cálculo, en el aparato de codificación, de un valor de función $S = g_T^{\tau.\rho} \in G_T$ usando el valor ρ;
una etapa de cálculo, en el aparato de codificación, de un tercer texto codificado C_3 y sacar el tercer texto codificado, en el que el tercer texto codificado C_3 es un valor OR exclusivo de un valor de función R(S) y la clave simétrica K, en el que el valor de función R(S) es una secuencia binaria obtenida introduciendo el valor de función S a una función R inyectiva; y
una etapa de cálculo, en el aparato de codificación, del segundo texto codificado C_2 usando los elementos v_μ, donde $\mu = 1, ..., n$, del vector n dimensional $\vec{v}$ y los valores r y ρ y sacar el segundo texto codificado C_2;

la etapa (D) incluye una etapa de introducción, en el aparato de descodificación, de un valor que contiene el texto codificado C_1, el segundo texto codificado $C_2 \in G_1^{n+1}$, y el tercer texto codificado C_3 al aparato de descodificación;
la etapa (F) incluye:

una etapa de cálculo, en el aparato de descodificación, de un valor OR exclusivo $K^{\sim}$ del tercer texto codificado C_3 y un valor de función $R(S^{\sim})$ que es una secuencia binaria obtenida introduciendo el valor de función $S^{\sim}$ a la función R inyectiva; y
una etapa de cálculo, en el aparato de descodificación, de los valores $r^{\sim}$ y $\rho^{\sim}$ introduciendo el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a la función H resistente a colisiones; y

la etapa (G) incluye:

una etapa de cálculo, en el aparato de descodificación, de un valor de función $C_2^{\sim} = r^{\sim}.(\sum_{\mu=1}^{n} v_\mu.b_\mu) + \rho^{\sim}.b_{n+1} \in G_1^{n+1}$ usando los elementos v_μ, donde $\mu = 1, ..., n$, del vector n dimensional $\vec{v}$ y los valores de $r^{\sim}$ y $\rho^{\sim}$; y
una etapa de determinación, en el aparato de descodificación, de si el valor de función $C_2^{\sim}$ es igual al segundo texto codificado C_2; si el valor de función $C_2^{\sim}$ es igual al segundo texto codificado C_2, usando el valor OR exclusivo $K^{\sim}$ como una clave simétrica, calcular un valor descodificado $M^{\sim}$ introduciendo el valor OR exclusivo $K^{\sim}$ y el texto codificado C_1 a una función de descodificación de clave simétrica y sacar el valor descodificado $M^{\sim}$.

31. Método de seguridad según la reivindicación 29, en el que la etapa (A) incluye:

una etapa de selección del valor ρ en el aparato de codificación;
una etapa de cálculo, en el aparato de codificación, de un valor de función $S = g_T^{\tau.\rho} \in G_T$ usando el valor ρ; y
una etapa de uso, en el aparato de codificación, de un valor de función R(S) obtenido introduciendo el valor de función S a una función R inyectiva, como una clave simétrica K, calcular el texto codificado C_1 introduciendo la clave simétrica K y el texto no codificado M a la función de codificación de clave simétrica, y sacar el texto codificado C_1;
la etapa (B) incluye una etapa de cálculo, en el aparato de codificación, del valor r introduciendo el valor de función S y el texto codificado C_1 a la función H resistente a colisiones;
la etapa (C) incluye una etapa de cálculo, en el aparato de codificación, del segundo texto codificado usando los elementos v_μ, donde $\mu = 1, ..., n$, del vector n dimensional $\vec{v}$ y los valores $r^{\sim}$ y $\rho^{\sim}$ y sacar el segundo texto codificado C_2;
la etapa (F) incluye una etapa de cálculo, en el aparato de descodificación, del valor $r^{\sim}$ introduciendo el valor de función $S^{\sim}$ y el texto codificado C_1 a la función H resistente a colisiones; y
la etapa (G) incluye:
una etapa de cálculo, en el aparato de descodificación, de un valor de función $\lambda = C_2 - r^{\sim}.\sum_{\mu=1}^{n} v_\mu.b_\mu \in G_1^{n+1}$ usando el segundo texto codificado C_2, el valor $r^{\sim}$ y los elementos v_μ, donde $\mu = 1, ..., n$, del vector n-dimensional $\vec{v}$;
una etapa de cálculo, en el aparato de descodificación, de un valor de función e (λ, D_2^*) introduciendo el valor de función λ y la información de clave auxiliar D_2^* a la función e bilineal; y
una etapa de determinación, en el aparato de descodificación, de si el valor de función e(λ, D_2^*) cumple e(λ, D_2^*) = $1 \in G_T$; si se cumple e(λ, D_2^*) = $1 \in G_T$, usando un valor de función $R(S^{\sim})$ obtenido introduciendo del valor de función $S^{\sim}$ a una función R inyectiva, como una clave simétrica $K^{\sim}$, calcular un valor descodificado $M^{\sim}$ introduciendo la clave simétrica $K^{\sim}$ y el texto codificado C_1 a la función de descodificación de clave simétrica y sacar el valor descodificado $M^{\sim}$.

32. Un programa que hace que un ordenador funcione como un aparato de codificación según una cualquiera de las reivindicaciones 1 a 3.

33. Un programa que hace que un ordenador funcione como un aparato de descodificación según una cualquiera de las reivindicaciones 15 a 17.

34. Un medio de grabación legible por ordenador que tiene almacenado en el mismo un programa para hacer que un ordenador funcione como un aparato de codificación según una cualquiera de las reivindicaciones 1 a 3.

35. Un medio de grabación legible por ordenador que tiene almacenado en el mismo un programa para hacer que un ordenador funcione como un aparato de descodificación según una cualquiera de las reivindicaciones 15 a 17.

FIG.1

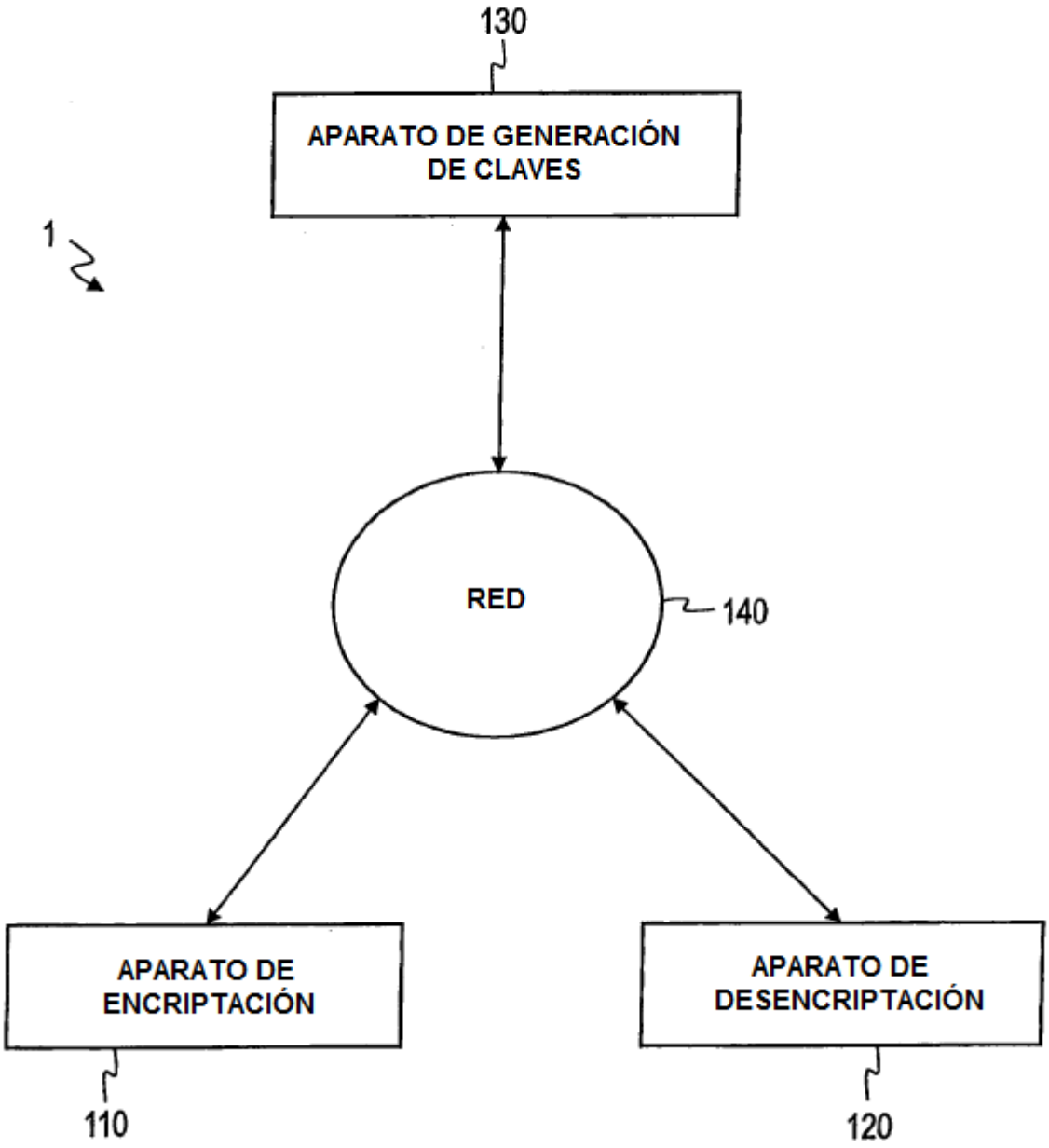

130
APARATO DE GENERACIÓN DE CLAVES
1
RED
140
APARATO DE ENCRIPTACIÓN
APARATO DE DESENCRIPTACIÓN
110
120

FIG.2

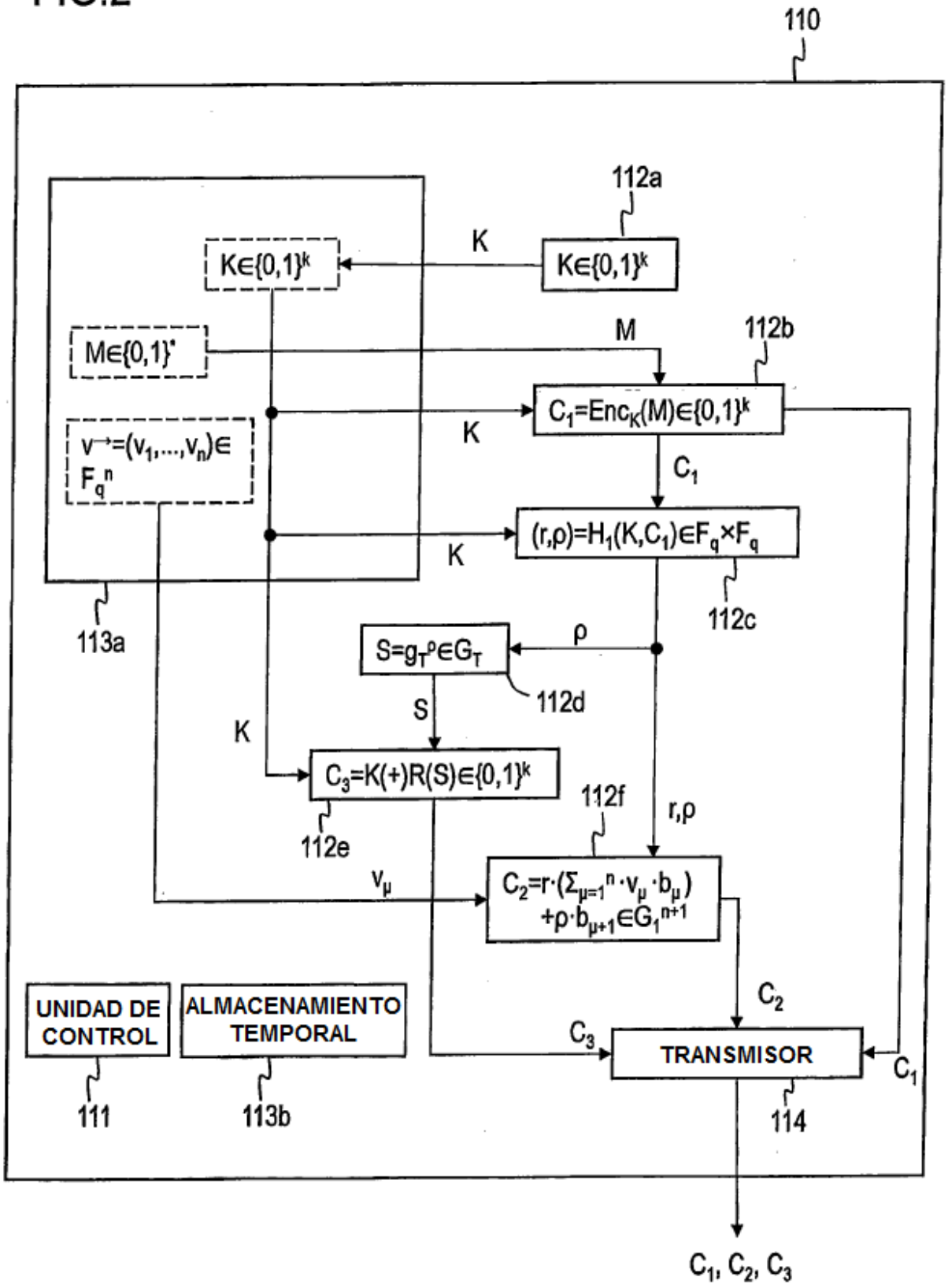

110
112a
K∈{0,1}^k
K
K∈{0,1}^k
M∈{0,1}*
M
112b
C1=Enc_K(M)∈{0,1}^k
K
C1
v→=(v1,...,vn)∈ Fq^n
(r,ρ)=H1(K,C1)∈Fq×Fq
K
112c
113a
S=gT^ρ∈GT
ρ
112d
S
K
C3=K(+)R(S)∈{0,1}^k
112f
r,ρ
112e
vμ
C2=r·(Σμ=1^n vμ·bμ) +ρ·bμ+1∈G1^n+1
UNIDAD DE CONTROL
ALMACENAMIENTO TEMPORAL
C2
C3
TRANSMISOR
C1
111
113b
114
C1, C2, C3

FIG.3

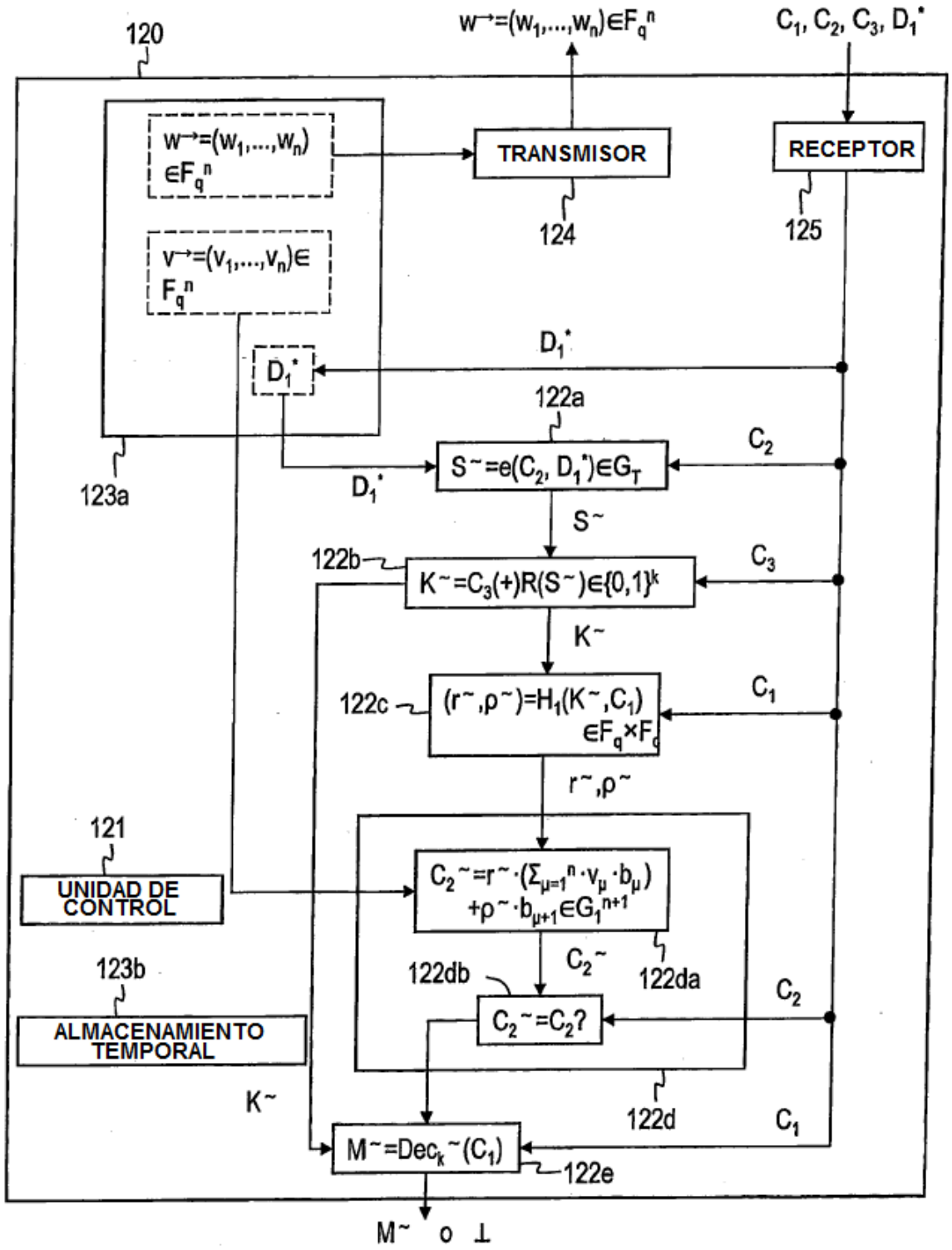

120
$\vec{w}=(w_1,\ldots,w_n)\in F_q^n$
C_1, C_2, C_3, D_1^*
$\vec{w}=(w_1,\ldots,w_n)\in F_q^n$
TRANSMISOR
RECEPTOR
124
125
$\vec{v}=(v_1,\ldots,v_n)\in F_q^n$
D_1^*
D_1^*
122a
C_2
$S^\sim=e(C_2, D_1^*)\in G_T$
123a
D_1^*
$S^\sim$
122b
$K^\sim=C_3(+)R(S^\sim)\in\{0,1\}^k$
C_3
$K^\sim$
122c
$(r^\sim,\rho^\sim)=H_1(K^\sim,C_1)\in F_q\times F_q$
C_1
$r^\sim,\rho^\sim$
121
UNIDAD DE CONTROL
$C_2{}^\sim=r^\sim\cdot(\Sigma_{\mu=1}^n\cdot v_\mu\cdot b_\mu)+\rho^\sim\cdot b_{\mu+1}\in G_1{}^{n+1}$
123b
ALMACENAMIENTO TEMPORAL
$C_2{}^\sim$
122db
122da
C_2
$C_2{}^\sim=C_2?$
$K^\sim$
122d
C_1
$M^\sim=Dec_k{}^\sim(C_1)$
122e
$M^\sim$ o $\perp$

FIG.4

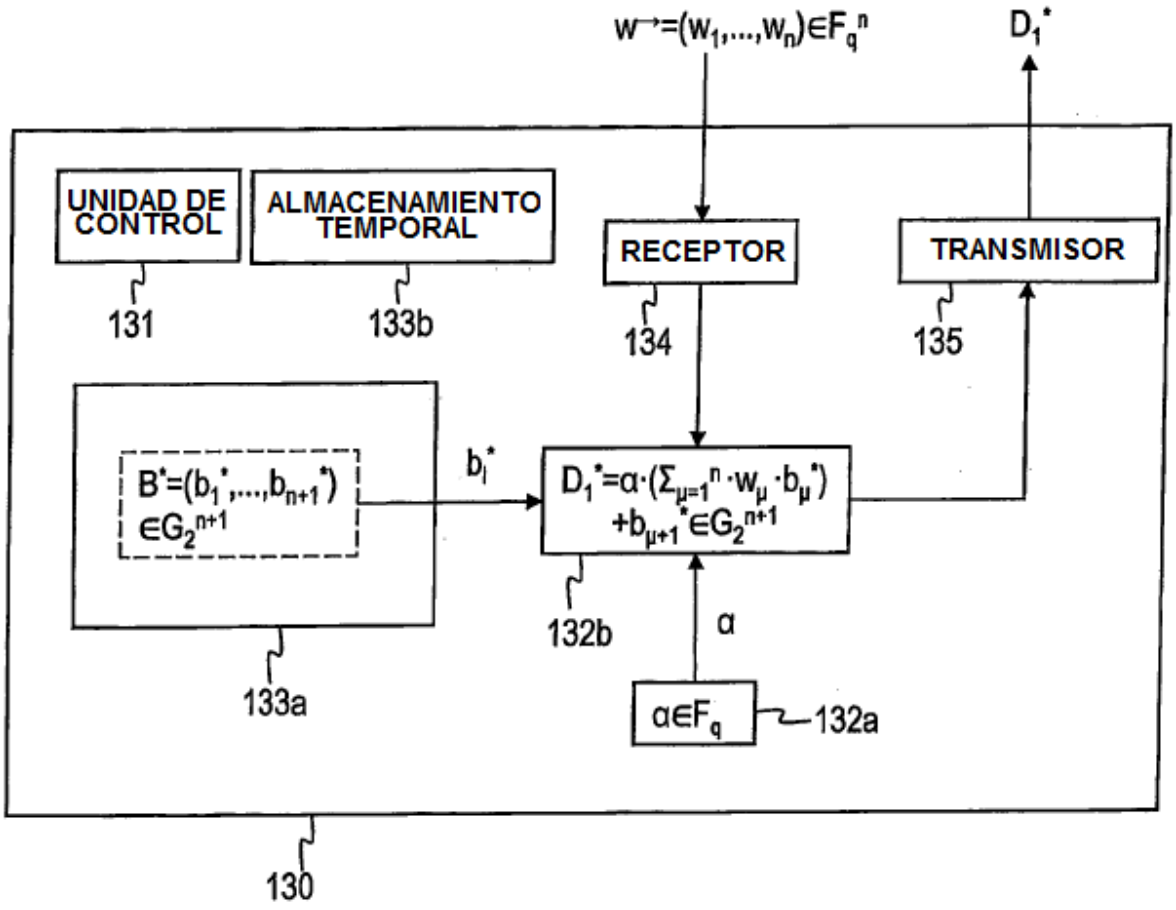

$\vec{w}=(w_1,\ldots,w_n)\in F_q^n$
D_1^*
UNIDAD DE CONTROL
ALMACENAMIENTO TEMPORAL
RECEPTOR
TRANSMISOR
131
133b
134
135
$B^*=(b_1^*,\ldots,b_{n+1}^*)$ $\in G_2^{n+1}$
b_l^*
$D_1^*=\alpha\cdot(\Sigma_{\mu=1}^{n} w_\mu \cdot b_\mu^*)$ $+b_{\mu+1}^*\in G_2^{n+1}$
132b
α
$\alpha\in F_q$
132a
133a
130

FIG.5

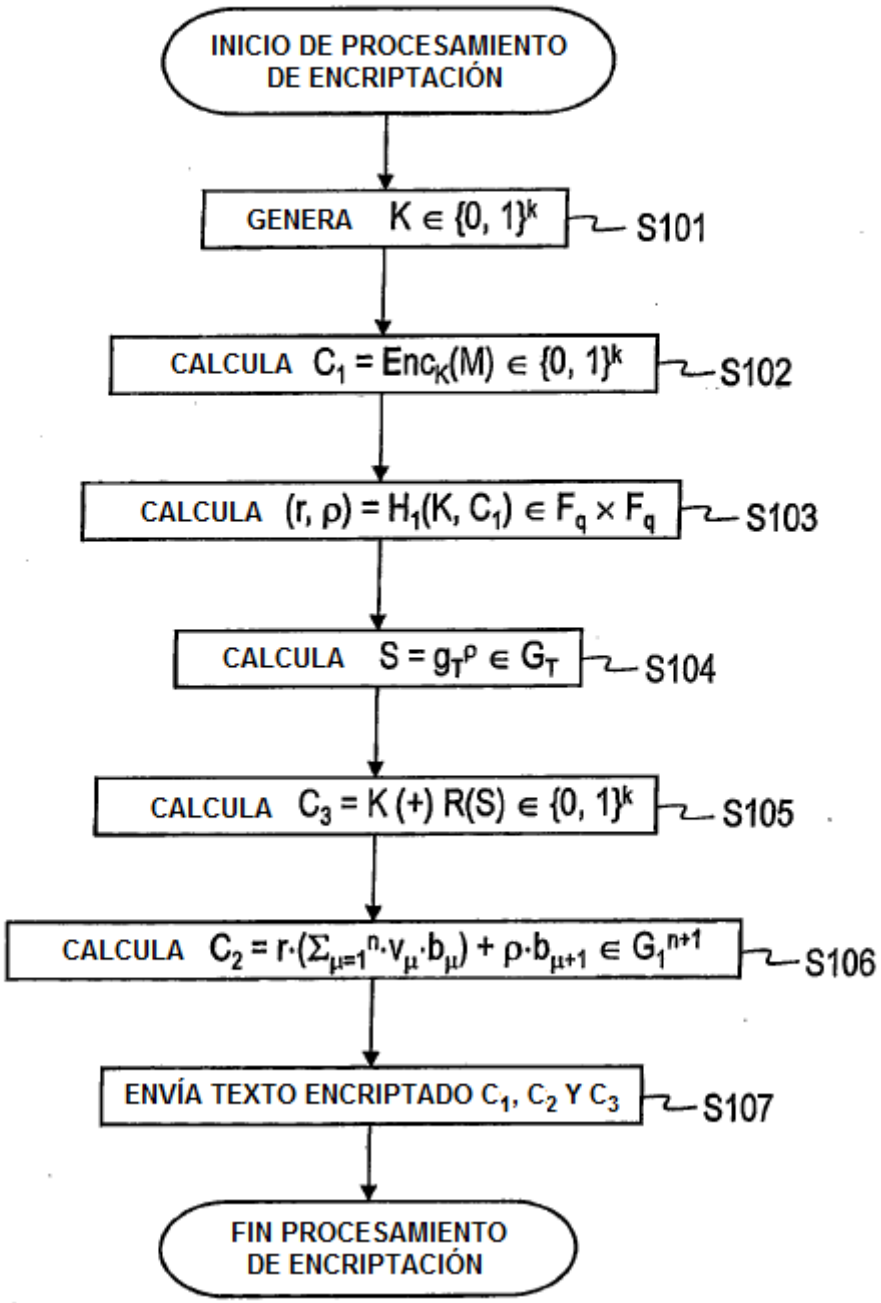

INICIO DE PROCESAMIENTO DE ENCRIPTACIÓN
GENERA $K \in \{0, 1\}^k$
S101
CALCULA $C_1 = Enc_K(M) \in \{0, 1\}^k$
S102
CALCULA $(r, \rho) = H_1(K, C_1) \in F_q \times F_q$
S103
CALCULA $S = g_T^{\rho} \in G_T$
S104
CALCULA $C_3 = K (+) R(S) \in \{0, 1\}^k$
S105
CALCULA $C_2 = r \cdot (\Sigma_{\mu=1}^{n} \cdot v_\mu \cdot b_\mu) + \rho \cdot b_{\mu+1} \in G_1^{n+1}$
S106
ENVÍA TEXTO ENCRIPTADO C_1, C_2 Y C_3
S107
FIN PROCESAMIENTO DE ENCRIPTACIÓN

FIG.6

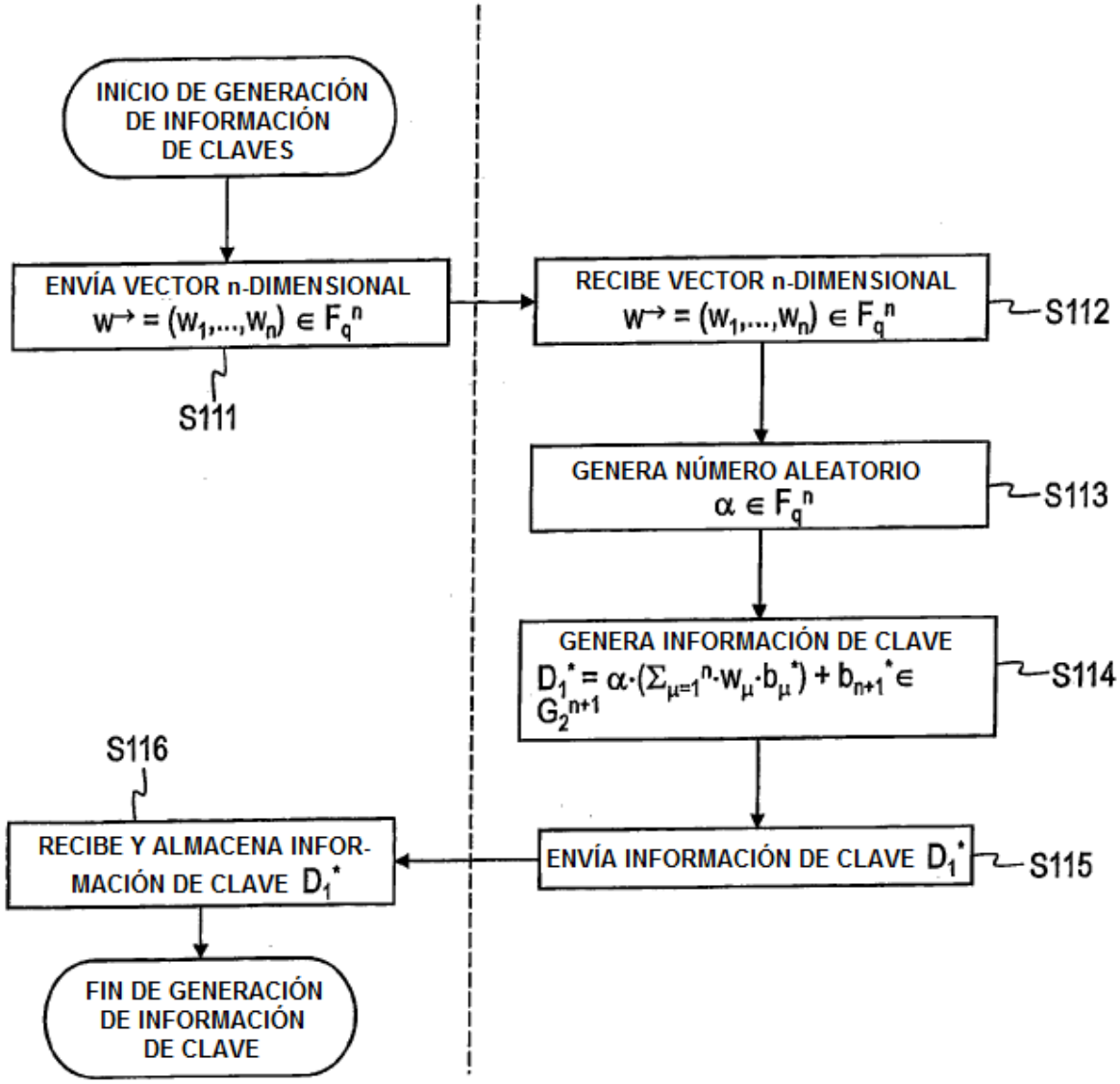

INICIO DE GENERACIÓN DE INFORMACIÓN DE CLAVES
ENVÍA VECTOR n-DIMENSIONAL $\vec{w} = (w_1,...,w_n) \in F_q^n$
S111
RECIBE VECTOR n-DIMENSIONAL $\vec{w} = (w_1,...,w_n) \in F_q^n$
S112
GENERA NÚMERO ALEATORIO $\alpha \in F_q^n$
S113
GENERA INFORMACIÓN DE CLAVE $D_1^* = \alpha \cdot (\Sigma_{\mu=1}^{n} w_\mu \cdot b_\mu^*) + b_{n+1}^* \in G_2^{n+1}$
S114
ENVÍA INFORMACIÓN DE CLAVE D_1^*
S115
S116
RECIBE Y ALMACENA INFORMACIÓN DE CLAVE D_1^*
FIN DE GENERACIÓN DE INFORMACIÓN DE CLAVE

FIG.7

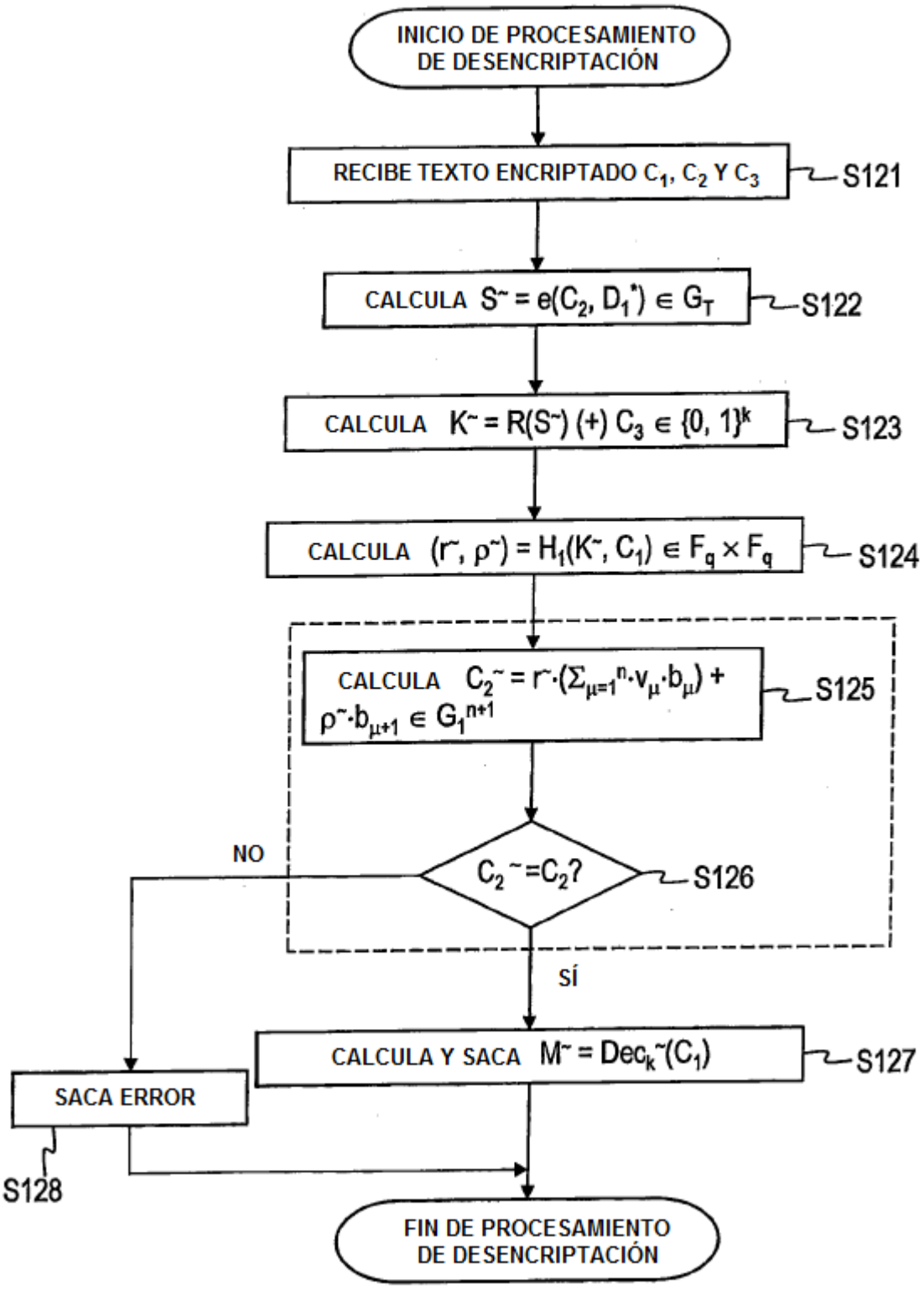

INICIO DE PROCESAMIENTO DE DESENCRIPTACIÓN
RECIBE TEXTO ENCRIPTADO C_1, C_2 Y C_3
S121
CALCULA $S\tilde{} = e(C_2, D_1^*) \in G_T$
S122
CALCULA $K\tilde{} = R(S\tilde{}) (+) C_3 \in \{0, 1\}^k$
S123
CALCULA $(r\tilde{}, \rho\tilde{}) = H_1(K\tilde{}, C_1) \in F_q \times F_q$
S124
CALCULA $C_2\tilde{} = r\tilde{} \cdot (\Sigma_{\mu=1}^{n} \cdot v_\mu \cdot b_\mu) + \rho\tilde{} \cdot b_{\mu+1} \in G_1^{n+1}$
S125
NO
$C_2\tilde{} = C_2$?
S126
SÍ
CALCULA Y SACA $M\tilde{} = Dec_k\tilde{}(C_1)$
S127
SACA ERROR
S128
FIN DE PROCESAMIENTO DE DESENCRIPTACIÓN

FIG.8

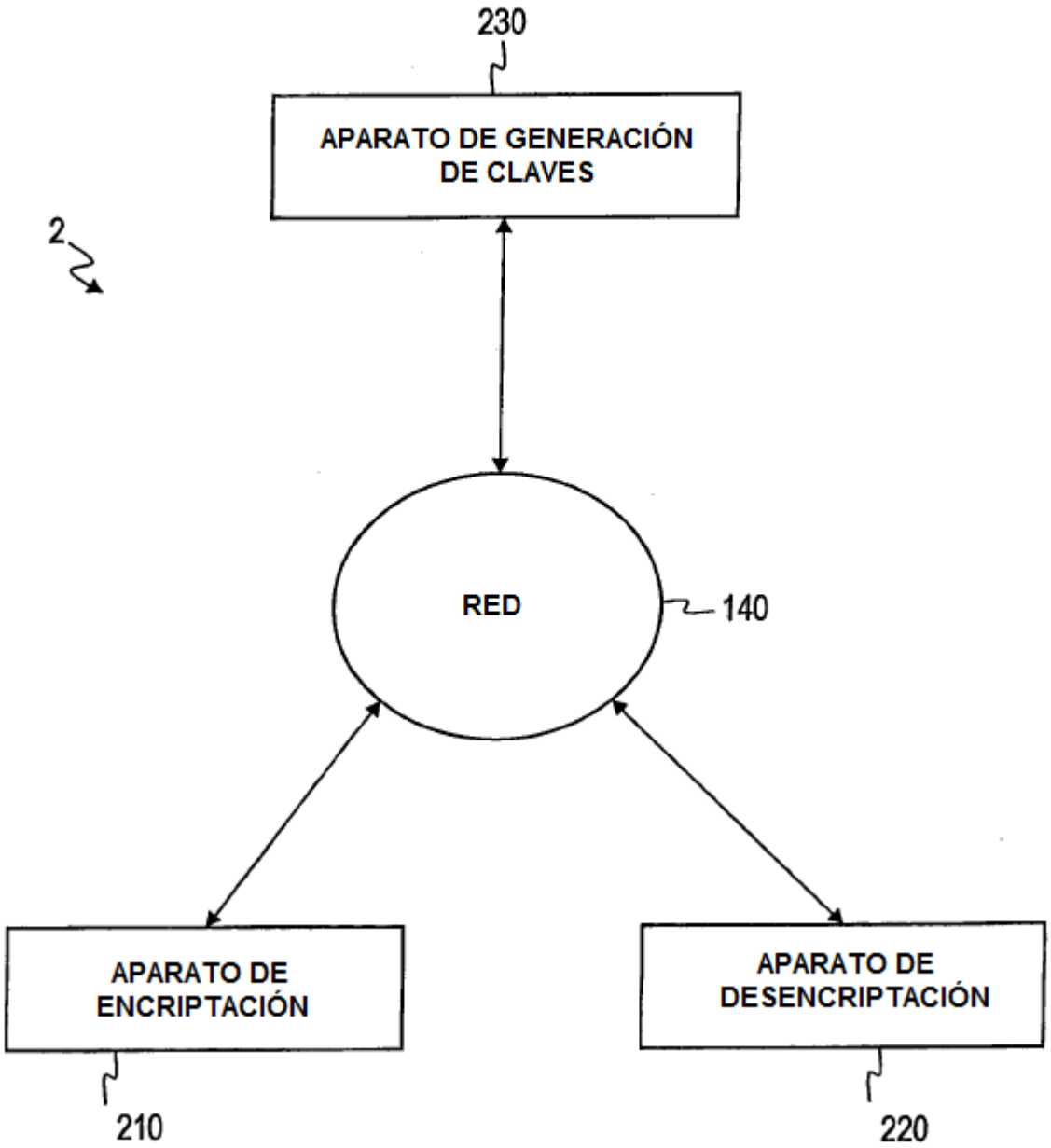

230
APARATO DE GENERACIÓN DE CLAVES
2
RED
140
APARATO DE ENCRIPTACIÓN
APARATO DE DESENCRIPTACIÓN
210
220

FIG.9

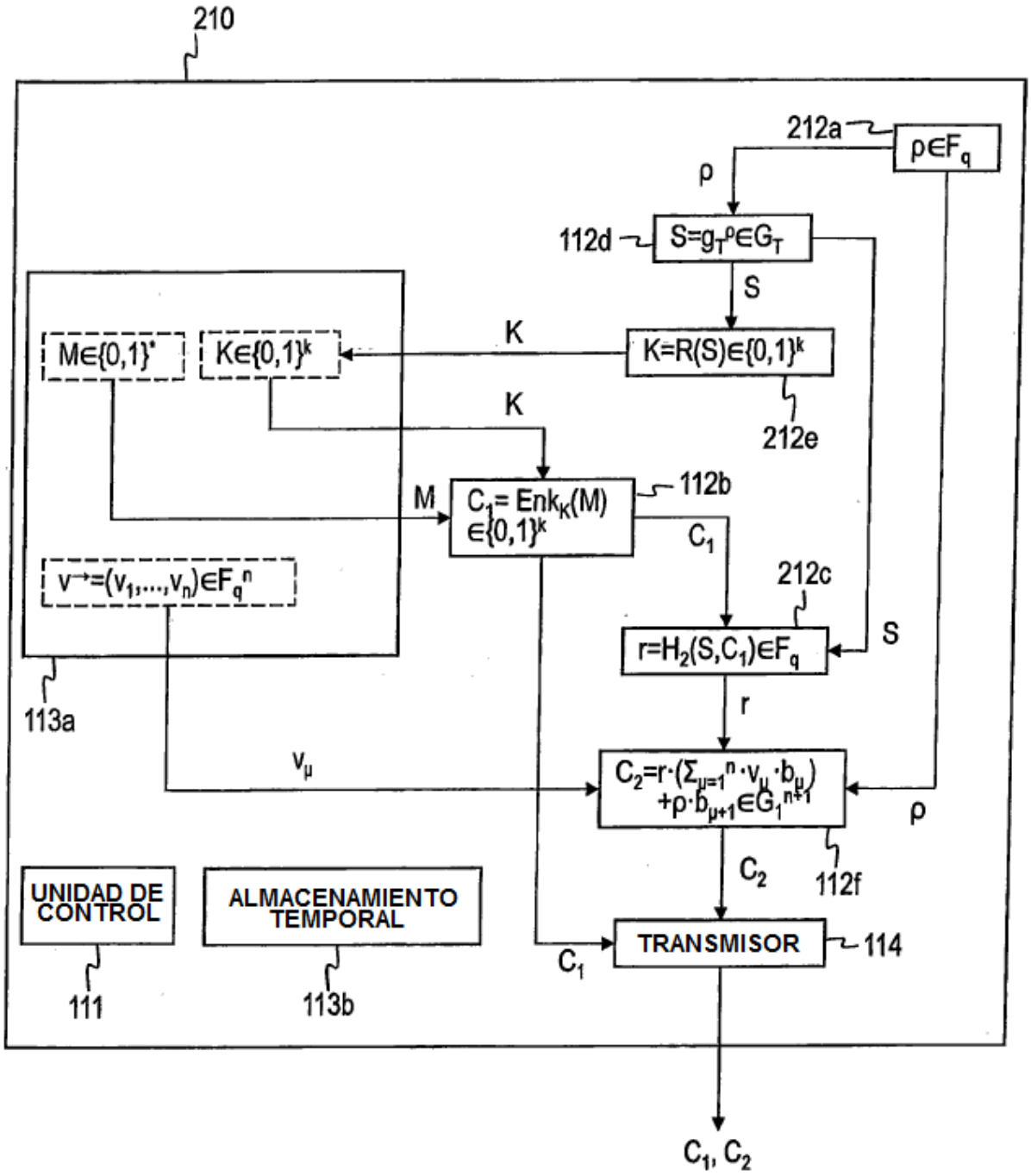

210
212a
ρ∈F_q
ρ
112d
S=g_T^ρ∈G_T
S
K
M∈{0,1}*
K∈{0,1}^k
K=R(S)∈{0,1}^k
212e
K
M
C_1= Enk_K(M) ∈{0,1}^k
112b
C_1
212c
v→=(v_1,...,v_n)∈F_q^n
r=H_2(S,C_1)∈F_q
S
113a
r
v_μ
C_2=r·(Σ_{μ=1}^n·v_μ·b_μ) +ρ·b_{μ+1}∈G_1^{n+1}
ρ
C_2
112f
UNIDAD DE CONTROL
ALMACENAMIENTO TEMPORAL
TRANSMISOR
114
C_1
111
113b
C_1, C_2

FIG.10

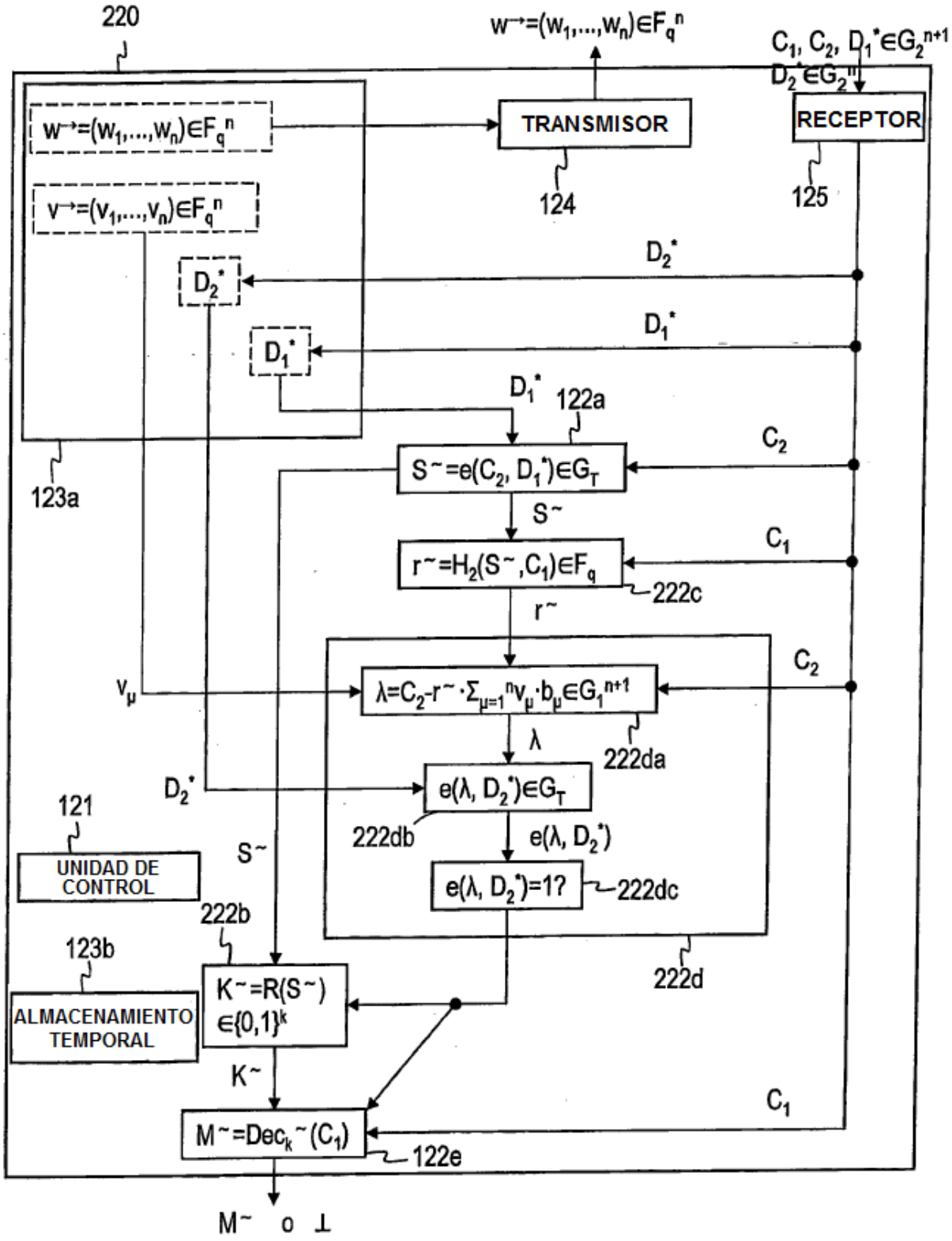

220
$\vec{w}=(w_1,\ldots,w_n)\in F_q^n$
$C_1, C_2, D_1^*\in G_2^{n+1}$
$D_2^*\in G_2^n$
TRANSMISOR
RECEPTOR
124
125
$\vec{w}=(w_1,\ldots,w_n)\in F_q^n$
$\vec{v}=(v_1,\ldots,v_n)\in F_q^n$
D_2^*
D_1^*
122a
123a
C_2
$S\sim=e(C_2, D_1^*)\in G_T$
$S\sim$
C_1
$r\sim=H_2(S\sim, C_1)\in F_q$
222c
$r\sim$
v_μ
$\lambda=C_2-r\sim\cdot\Sigma_{\mu=1}^{n} v_\mu\cdot b_\mu\in G_1^{n+1}$
λ
222da
$e(\lambda, D_2^*)\in G_T$
121
UNIDAD DE CONTROL
222db
$e(\lambda, D_2^*)$
$e(\lambda, D_2^*)=1?$
222dc
222b
222d
123b
ALMACENAMIENTO TEMPORAL
$K\sim=R(S\sim)\in\{0,1\}^k$
$K\sim$
$M\sim=Dec_k\sim(C_1)$
122e
$M\sim$ o $\perp$

FIG.11

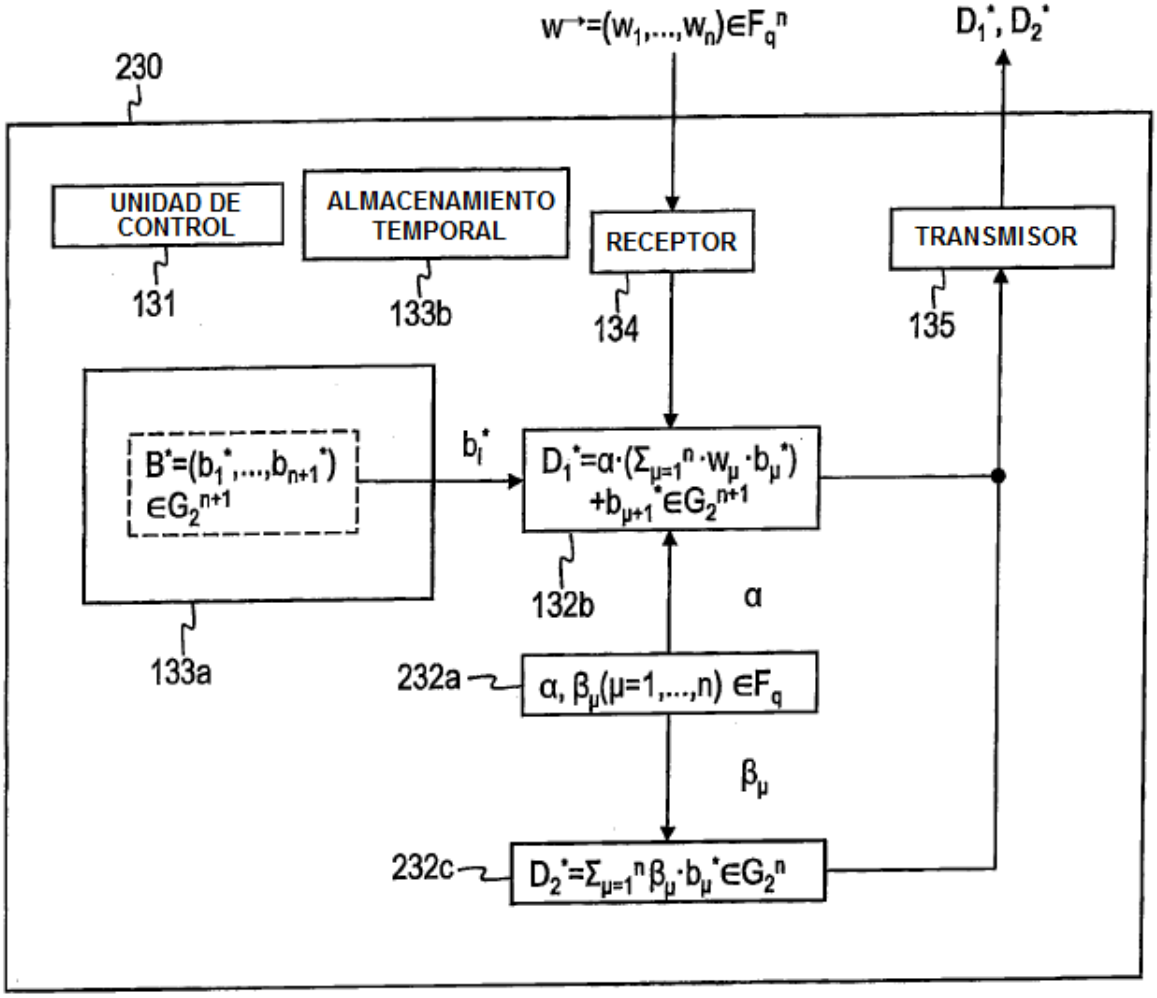

$\vec{w}=(w_1,\ldots,w_n)\in F_q^n$
D_1^*, D_2^*
230
UNIDAD DE CONTROL
131
ALMACENAMIENTO TEMPORAL
133b
RECEPTOR
134
TRANSMISOR
135
$B^*=(b_1^*,\ldots,b_{n+1}^*)$ $\in G_2^{n+1}$
133a
b_l^*
$D_1^*=\alpha\cdot(\Sigma_{\mu=1}^{n} w_\mu \cdot b_\mu^*)$ $+b_{\mu+1}^* \in G_2^{n+1}$
132b
α
232a
$\alpha, \beta_\mu(\mu=1,\ldots,n) \in F_q$
β_μ
232c
$D_2^*=\Sigma_{\mu=1}^{n} \beta_\mu \cdot b_\mu^* \in G_2^n$

FIG.12

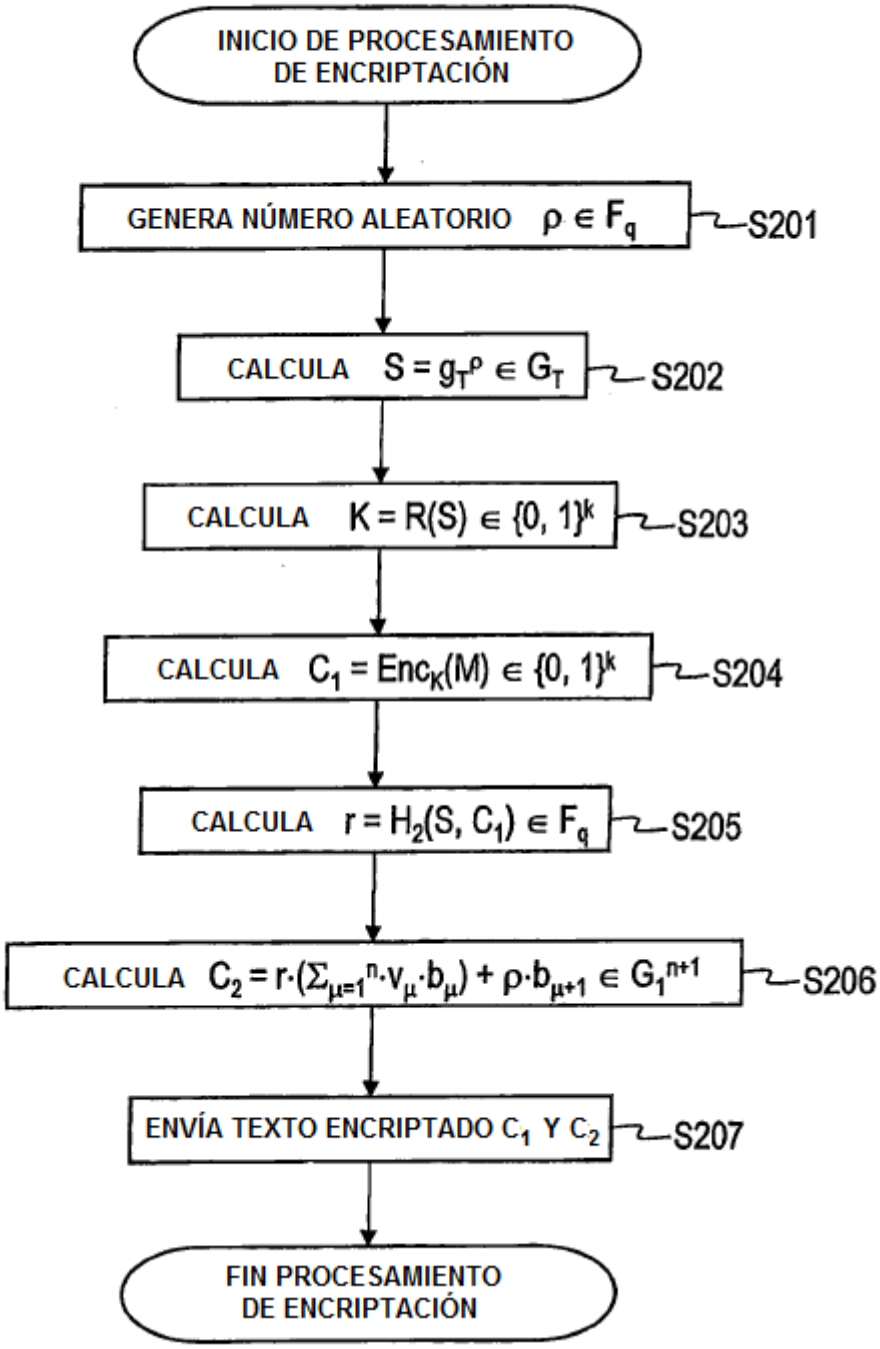

INICIO DE PROCESAMIENTO DE ENCRIPTACIÓN
GENERA NÚMERO ALEATORIO $\rho \in F_q$
S201
CALCULA $S = g_T^{\rho} \in G_T$
S202
CALCULA $K = R(S) \in \{0, 1\}^k$
S203
CALCULA $C_1 = Enc_K(M) \in \{0, 1\}^k$
S204
CALCULA $r = H_2(S, C_1) \in F_q$
S205
CALCULA $C_2 = r \cdot (\Sigma_{\mu=1}^{n} \cdot v_\mu \cdot b_\mu) + \rho \cdot b_{\mu+1} \in G_1^{n+1}$
S206
ENVÍA TEXTO ENCRIPTADO C_1 Y C_2
S207
FIN PROCESAMIENTO DE ENCRIPTACIÓN

FIG.13

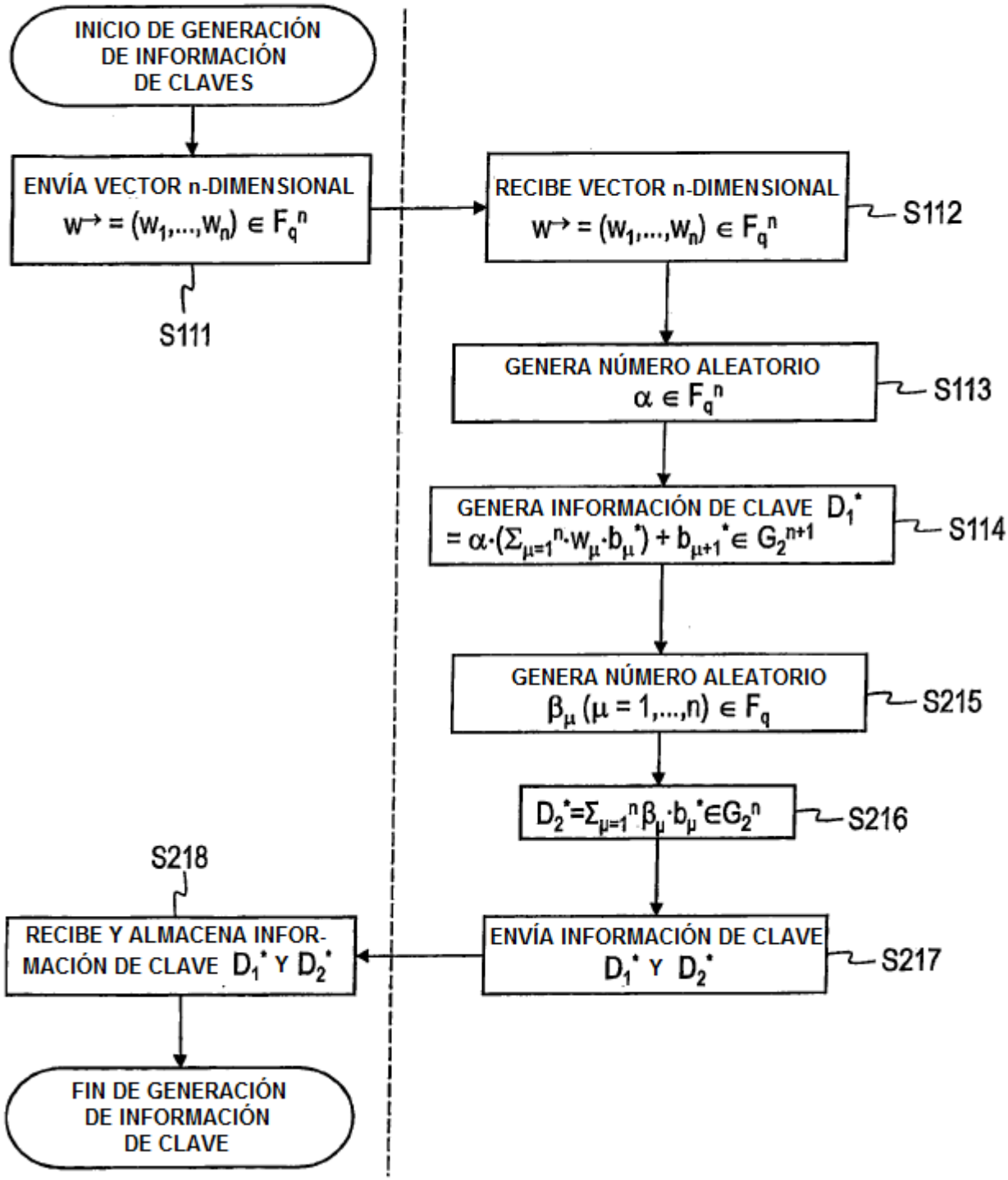
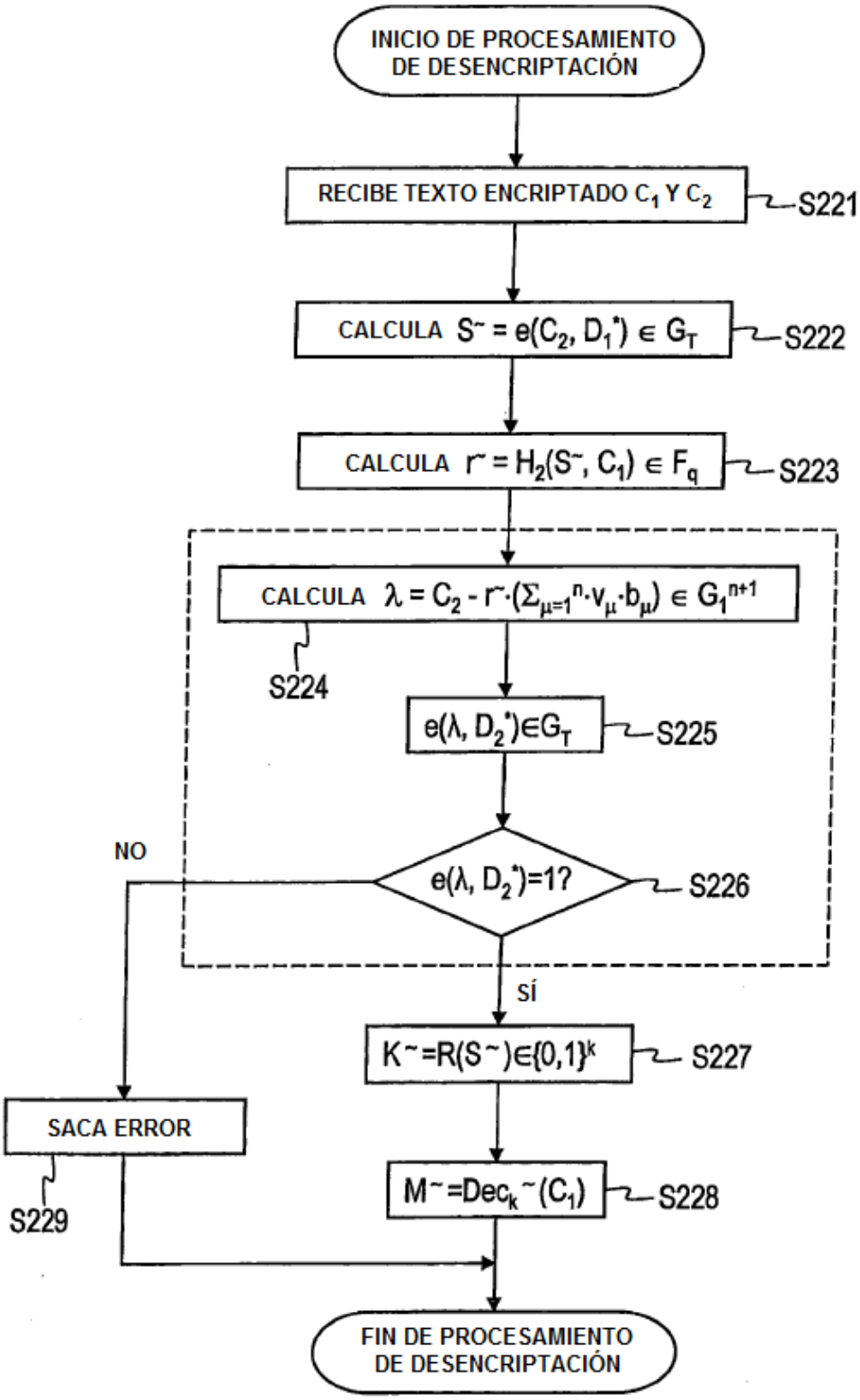

INICIO DE GENERACIÓN DE INFORMACIÓN DE CLAVES
ENVÍA VECTOR n-DIMENSIONAL $\vec{w} = (w_1,...,w_n) \in F_q^n$
S111
RECIBE VECTOR n-DIMENSIONAL $\vec{w} = (w_1,...,w_n) \in F_q^n$
S112
GENERA NÚMERO ALEATORIO $\alpha \in F_q^n$
S113
GENERA INFORMACIÓN DE CLAVE $D_1^* = \alpha \cdot (\Sigma_{\mu=1}^{n} w_\mu \cdot b_\mu^*) + b_{\mu+1}^* \in G_2^{n+1}$
S114
GENERA NÚMERO ALEATORIO $\beta_\mu (\mu = 1,...,n) \in F_q$
S215
$D_2^* = \Sigma_{\mu=1}^{n} \beta_\mu \cdot b_\mu^* \in G_2^n$
S216
ENVÍA INFORMACIÓN DE CLAVE D_1^* Y D_2^*
S217
S218
RECIBE Y ALMACENA INFORMACIÓN DE CLAVE D_1^* Y D_2^*
FIN DE GENERACIÓN DE INFORMACIÓN DE CLAVE

FIG.14


INICIO DE PROCESAMIENTO DE DESENCRIPTACIÓN
RECIBE TEXTO ENCRIPTADO C_1 Y C_2
S221
CALCULA $S\tilde{} = e(C_2, D_1^*) \in G_T$
S222
CALCULA $r\tilde{} = H_2(S\tilde{}, C_1) \in F_q$
S223
CALCULA $\lambda = C_2 - r\tilde{}\cdot(\Sigma_{\mu=1}^{n}\cdot v_\mu \cdot b_\mu) \in G_1^{n+1}$
S224
$e(\lambda, D_2^*) \in G_T$
S225
NO
$e(\lambda, D_2^*)=1?$
S226
SÍ
$K\tilde{}=R(S\tilde{})\in\{0,1\}^k$
S227
SACA ERROR
S229
$M\tilde{}=Dec_k\tilde{}(C_1)$
S228
FIN DE PROCESAMIENTO DE DESENCRIPTACIÓN