(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 693 146**

(51) Int. CI.:

G09C 1/00 (2006.01)
H04L 9/14 (2006.01)
H04L 9/08 (2006.01)
H04L 9/30 (2006.01)
H04L 9/32 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **15.02.2011 PCT/JP2011/053174**

(87) Fecha y número de publicación internacional: **03.11.2011 WO11135895**

(96) Fecha de presentación y número de la solicitud europea: **15.02.2011 E 11774680 (0)**

(97) Fecha y número de publicación de la concesión europea: **01.08.2018 EP 2565862**

(54) Título: **Sistema de procesamiento criptográfico, dispositivo de generación de claves, dispositivo de cifrado, dispositivo de descifrado, sistema de procesamiento de firmas, dispositivo de firma y dispositivo de verificación**

(30) Prioridad:

27.04.2010 JP 2010101657

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
07.12.2018

(73) Titular/es:

MITSUBISHI ELECTRIC CORPORATION (50.0%)
7-3 Marunouchi 2-Chome, Chiyoda-ku
Tokyo 100-8310, JP y
NIPPON TELEGRAPH AND TELEPHONE CORPORATION (50.0%)

(72) Inventor/es:

TAKASHIMA, KATSUYUKI y
OKAMOTO, TATSUAKI

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Sistema de procesamiento criptográfico, dispositivo de generación de claves, dispositivo de cifrado, dispositivo de descifrado, sistema de procesamiento de firmas, dispositivo de firma y dispositivo de verificación

Campo técnico

La presente invención se refiere a un esquema de cifrado funcional (FE). La invención se define en las reivindicaciones independientes.

Antecedentes de la técnica

Las Citas Bibliográficas no de Patente 3 a 6, 10, 12, 13, 15 y 18 describen un esquema de cifrado basado en ID (Identidad) (IBE) que constituye una clase del esquema de cifrado funcional. Las Citas Bibliográficas no de Patente 2, 7, 9, 16, 19, 23 a 26 y 28 describen un esquema de cifrado basado en atributos (ABE) que constituye otra clase del esquema de cifrado funcional.

Lista de referencias

Citas Bibliográficas no de Patente

Cita Bibliográfica no de Patente 1: Beimel, A., Secure schemes for secret sharing and key distribution. Tesis doctoral, Instituto de Tecnología de Israel, Technion, Haifa, Israel, 1996.

Cita Bibliográfica no de Patente 2: Bethencourt, J., Sahai, A., Waters, B.: Ciphertext policy attribute-based encryption. En: Simposio sobre Seguridad y Privacidad del IEEE 2007, páginas 321-34. IEEE Press (2007)

Cita Bibliográfica no de Patente 3: Boneh, D., Boyen, X.: Efficient selective-ID secure identity based encryption without random oracles. En: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, páginas 223-38. Springer Heidelberg (2004)

Cita Bibliográfica no de Patente 4: Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. En: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, páginas 443-59. Springer Heidelberg (2004)

Cita Bibliográfica no de Patente 5: Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. En: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, páginas 440-56. Springer Heidelberg (2005)

Cita Bibliográfica no de Patente 6: Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. En: Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, páginas 213-29. Springer Heidelberg (2001)

Cita Bibliográfica no de Patente 7: Boneh, D., Hamburg, M.: Generalized identity based and broadcast encryption scheme. En: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5350, páginas 455-70. Springer Heidelberg (2008)

Cita Bibliográfica no de Patente 8: Boneh, D., Katz, J., Improved efficiency for CCA-secure cryptosystems built using identity based encryption. RSA-CT 2005, LNCS, Springer Verlag (2005)

Cita Bibliográfica no de Patente 9: Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. En: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, páginas 535-54. Springer Heidelberg (2007)

Cita Bibliográfica no de Patente 10: Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). En: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, páginas 290-07. Springer Heidelberg (2006)

Cita Bibliográfica no de Patente 11: Canetti, R., Halevi S., Katz J., Chosen-ciphertext security from identity-based encryption. EUROCRYPT 2004, LNCS, Springer-Verlag (2004)

Cita Bibliográfica no de Patente 12: Cocks, C.: An identity based encryption scheme based on quadratic residues. En: Honary, B. (ed.) Conf. Int. IMA. LNCS, vol. 2260, páginas 360-63. Springer Heidelberg (2001)

Cita Bibliográfica no de Patente 13: Gentry, C.: Practical identity-based encryption without random oracles. En: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, páginas 445-64. Springer Heidelberg (2006)

Cita Bibliográfica no de Patente 14: Gentry, C., Halevi, S.: Hierarchical identity-based encryption with polynomially many levels. En: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, páginas 437-56. Springer Heidelberg (2009)

Cita Bibliográfica no de Patente 15: Gentry, C., Silverberg, A: Hierarchical ID-based cryptography. En: Zheng, Y. (ed.) ASIACRYPT 2002.LNCS, vol. 2501, páginas 548-66. Springer Heidelberg (2002)

Cita Bibliográfica no de Patente 16: Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. En: Conferencia ACM sobre Seguridad Informática y de Comunicaciones 2006, páginas 89-8, ACM (2006)

Cita Bibliográfica no de Patente 17: Groth, J., Sahai, A.: Efficient non-interactive proof systems for bilinear groups. En: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, páginas 415-32. Springer Heidelberg (2008)

Cita Bibliográfica no de Patente 18: Horwitz, J., Lynn, B.: Towards hierarchical identity-based encryption. En: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, páginas 466-81. Springer Heidelberg (2002)

Cita Bibliográfica no de Patente 19: Katz, J., Sahai, A., Waters, B.: Predicate encryption supporting disjunctions, polynomial equations, and inner products. En: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, páginas 146-62. Springer Heidelberg (2008)

Cita Bibliográfica no de Patente 20: Lewko, A.B., Waters, B.: Fully secure HIBE with short ciphertexts. ePrint, IACR, http://eprint.iacr.org/2009/482

Cita Bibliográfica no de Patente 21: Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. En: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, páginas 57-4. Springer Heidelberg (2008)

Cita Bibliográfica no de Patente 22: Okamoto, T., Takashima, K.: Hierarchical predicate encryption for Inner-Products, En: ASIACRYPT 2009, Springer Heidelberg (2009)

Cita Bibliográfica no de Patente 23: Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. En: Conferencia ACM sobre Seguridad Informática y de Comunicaciones 2007, paginas 195-03, ACM (2007)

Cita Bibliográfica no de Patente 24: Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. En: Conferencia ACM sobre Seguridad Informática y de Comunicaciones 2006, páginas 99-12, ACM, (2006)

Cita Bibliográfica no de Patente 25: Sahai, A., Waters, B.: Fuzzy identity-based encryption. En: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, páginas 457-73. Springer Heidelberg (2005)

Cita Bibliográfica no de Patente 26: Shi, E., Waters, B.: Delegating capability in predicate encryption systems. En: Aceto, L., Damgard, I., Goldberg, L. A., Halldosson, M.M., Ingofsdotir, A., Walukiewicz, I. (eds.) ICALP (2) 2008. LNCS, vol. 5126, páginas 560-78. Springer Heidelberg (2008)

Cita Bibliográfica no de Patente 27: Waters, B.: Efficient identity based encryption without random oracles. Eurocrypt 2005, LNCS Nº 3152, páginas 443-59. Springer Verlag, 2005.

Cita Bibliográfica no de Patente 28: Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. ePrint, IACR, http://eprint.iacr.org/2008/290

Cita Bibliográfica no de Patente 29: Waters, B.: Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions. En: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, páginas 619-36. Springer Heidelberg (2009)

Compendio de la invención

Problema técnico

Es un objeto de la presente invención proporcionar un esquema de cifrado funcional seguro que tiene funciones criptográficas diversificadas.

Solución al problema

Un sistema de procesamiento criptográfico según la presente invención incluye un dispositivo de generación de claves, un dispositivo de cifrado y un dispositivo de descifrado, y que sirve para asegurar un proceso criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más),

en donde el dispositivo de generación de claves incluye

una primera parte de entrada de información que toma como entrada una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), cuya variable ρ(i) es o bien una de una tupla positiva $(t, \vec{v}_i)$ y una tupla negativa $\neg(t, \vec{v}_i)$ de la información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más); y una matriz M predeterminada que tiene L filas y r columnas (r es un número entero de 1 o más); y

una parte de generación de clave de descifrado que genera un elemento k^*_0 y un elemento k^*_i para cada número entero i = 1, ..., L, en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ y un vector $\vec{w}$, cada uno que tiene r partes de elementos, y la matriz M introducida por la primera parte de entrada de información; un valor $s_0 := \vec{w} \cdot \vec{f}$; y un valor predeterminado θ_i (i = 1, ..., L), la parte de generación de clave de descifrado que está configurada

para generar el elemento k^*_0 estableciendo el valor $-s_0$ como coeficiente para el vector base $b^*_{0,p}$ (p es un valor predeterminado) de la base B^*_0 y estableciendo un valor predeterminado κ como coeficiente para un vector base $b^*_{0,q}$ (q es un valor predeterminado diferente del p prescrito), y

para generar un elemento k^*_i para cada número entero i = 1, ..., L, cuando la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$) estableciendo $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b^*_{t,1}$ de la base B^*_t indicado por la información de identificación t de la tupla positiva, y estableciendo $\theta_i v_{i,i'}$ como coeficiente para un vector base $b^*_{t,i'}$ indicado por la información de identificación t y cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$) estableciendo $s_i v_{i,i'}$ como coeficiente para el vector base $b^*_{t,i'}$ indicado por la información de identificación t de la tupla negativa y por cada número entero i' = 1, ..., n_t,

en donde el dispositivo de cifrado incluye

una segunda parte de entrada de información que toma como entrada, un conjunto de atributos Γ que tiene la información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d, y

una parte de generación de datos cifrados que genera un elemento c_0 y un elemento c_t que conciernen a cada información de identificación t incluida t incluida en el conjunto de atributos Γ, en base al conjunto de atributos Γ introducido por la segunda parte de entrada de información, la parte de generación de datos cifrados que está configurada

para generar el elemento c_0 donde un valor aleatorio δ se establece como coeficiente para un vector base $b_{0,p}$ (p es un p prescrito) de la base B_0, y donde un valor predeterminado ζ se establece como coeficiente para un vector base $b_{0,q}$ (q es un q prescrito) de la base B_0, y

para generar el elemento c_t donde $x_{t,i'}$ multiplicado por el valor aleatorio δ se establece como coeficiente para un vector base $b_{t,i'}$ (i' = 1, ..., n_t) de la base B_t para cada información de identificación t incluida en el conjunto de atributos Γ, y

en donde el dispositivo de descifrado incluye

una parte de adquisición de datos que adquiere datos cifrados c que incluyen los elementos c_0 y c_t y el conjunto de atributos Γ, los elementos c_0 y c_t que se generan por la parte de generación de datos cifrados,

una parte de adquisición de clave de descifrado que adquiere una clave de descifrado sk_s que incluye los elementos k^*_0 y k^*_i y el número variable ρ(i), los elementos k^*_0 y k^*_i que se generan por la parte de generación de la clave de descifrado,

una parte de cálculo de coeficiente complementario que, en base al conjunto de atributos Γ incluido en los datos cifrados c adquiridos por la parte de adquisición de datos, y la variable ρ(i) incluida en la clave de descifrado sk_s adquirida por la parte de adquisición de clave de descifrado, especifica, entre los números enteros i = 1, ..., L, un conjunto I de un número entero i para el cual la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$) y con la cual un producto interno de $\vec{v}_i$ de la tupla positiva y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$) y con la cual un producto interno de $\vec{v}_i$ de la tupla negativa y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0; y calcula un coeficiente complementario α_i con el cual un total de $\alpha_i s_i$ para i incluido en el conjunto I especificado llega a ser s_0, y

una parte de operación de emparejamiento que calcula un valor $K = g_T^{\zeta\lambda}$ dirigiendo una operación de emparejamiento indicada en la Fórmula 1 para los elementos c_0 y c_t incluidos en los datos cifrados c y los elementos k^*_0 y k^*_i incluidos en la clave de descifrado sk_s, en base al conjunto I especificado por la parte de cálculo de coeficiente complementario y al coeficiente complementario α_i calculado por la parte de cálculo de coeficiente complementario.

[Fórmula 1]

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_t, k_i^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_t, k_i^*)^{\alpha_i / (\vec{v}_i \cdot \vec{x}_i)}$$

Efectos ventajosos de la invención

El sistema de procesamiento criptográfico según la presente invención implementa un gran número de funciones criptográficas, e implementa cifrado funcional empleando un programa de intervalo y un predicado de producto interno. También este sistema de procesamiento criptográfico es más seguro que un esquema de cifrado funcional convencional.

Breve descripción de los dibujos

La Fig. 1 es un dibujo explicativo de una matriz M^.

La Fig. 2 es un dibujo explicativo de una matriz M_{δ}.

La Fig. 3 es un dibujo explicativo de s_0.

La Fig. 4 es un dibujo explicativo de $s^{\rightarrow T}$.

La Fig.5 es un diagrama de configuración de un sistema de procesamiento criptográfico 10 que ejecuta un esquema de cifrado funcional de política de clave.

La Fig. 6 es un diagrama de bloques de funciones que muestra la función del sistema de procesamiento criptográfico 10 que ejecuta el esquema de cifrado funcional de política de clave.

La Fig. 7 es un diagrama de flujo que muestra el proceso del algoritmo Setup.

La Fig. 8 es un diagrama de flujo que muestra el proceso del algoritmo KeyGen.

La Fig. 9 es un diagrama de flujo que muestra el proceso del algoritmo Enc.

La Fig. 10 es un diagrama de flujo que muestra el proceso del algoritmo Dec.

La Fig. 11 es un diagrama de configuración de un sistema de procesamiento criptográfico 10 que ejecuta el algoritmo de un esquema de cifrado funcional de política de texto cifrado.

La Fig. 12 es un diagrama de bloques de funciones que muestra la función del sistema de procesamiento criptográfico 10 que ejecuta el esquema de cifrado funcional de política de texto cifrado.

La Fig.13 es un diagrama de flujo que muestra el proceso del algoritmo KeyGen.

La Fig. 14 es un diagrama de flujo que muestra el proceso del algoritmo Enc.

La Fig. 15 es un diagrama de flujo que muestra el proceso del algoritmo Dec.

La Fig. 16 es un diagrama de configuración de un sistema de procesamiento de firmas 20.

La Fig. 17 es un diagrama de bloques de funciones que muestra la función del sistema de procesamiento de firmas 20.

La Fig. 18 es un diagrama de flujo que muestra el proceso del algoritmo Setup.

La Fig. 19 es un diagrama de flujo que muestra el proceso del algoritmo KeyGen.

La Fig. 20 es un diagrama de flujo que muestra el proceso del algoritmo Enc.

La Fig. 21 es un diagrama de flujo que muestra el proceso del algoritmo Dec.

La Fig. 22 es un diagrama que muestra un ejemplo de la configuración de hardware de cada uno del dispositivo de generación de claves 100, un dispositivo de cifrado 200, un dispositivo de descifrado 300, un dispositivo de firma 400 y un dispositivo de verificación 500.

Descripción de realizaciones

Se describirán en lo sucesivo realizaciones de la presente invención con referencia a los dibujos anexos.

En la siguiente descripción, un dispositivo de procesamiento es, por ejemplo, una CPU 911 (a ser descrita más tarde). El dispositivo de almacenamiento es, por ejemplo, una ROM 913, una RAM 914 o un disco magnético 920 (cada uno se describirá más tarde). El dispositivo de comunicación es, por ejemplo, una placa de comunicación 915 (a ser descrita más tarde). El dispositivo de entrada es, por ejemplo, un teclado 902 o la placa de comunicación 915 (a ser descrita más tarde). Esto es, el dispositivo de procesamiento, el dispositivo de almacenamiento, el dispositivo de comunicación y el dispositivo de entrada son hardware.

Se explicará la notación en la siguiente descripción.

Cuando A es una variable o distribución aleatoria, la Fórmula 101 indica que y se selecciona aleatoriamente a partir de A según la distribución de A. Esto es, en la Fórmula 101, y es un número aleatorio.

[Fórmula 101]

$$y \xleftarrow{R} A$$

Cuando A es un conjunto, la Fórmula 102 indica que y se selecciona uniformemente a partir de A. Esto es, en la Fórmula 102, y es un número aleatorio uniforme.

[Fórmula 102]

$$y \xleftarrow{U} A$$

La Fórmula 103 indica que y es un conjunto, definido o sustituido por z.

[Fórmula 103]

$y := z$

Cuando a es un valor fijo, la Fórmula 104 indica un evento que una máquina (algoritmo) A emite a en la entrada x.

[Fórmula 104]

$A(x) \rightarrow a$

Por ejemplo,

$A(x) \rightarrow 1$

La Fórmula 105, esto es, F_q, indica un campo finito de orden q.

[Fórmula 105]

$$\mathbb{F}_q$$

Un símbolo vectorial indica una representación vectorial sobre el campo finito F_q. Esto es, se establece la Fórmula 106.

[Fórmula 106]

$$\vec{x}$$

indica

$$(x_1, \ldots, x_n) \in \mathbb{F}_q^n$$

La Fórmula 107 indica el producto interno, indicado por la Fórmula 109, de dos vectores $\vec{x}$ e $\vec{y}$ indicados en la Fórmula 108.

[Fórmula 107]

$$\vec{x} \cdot \vec{v}$$

[Fórmula 108]

$$\vec{x} = (x_1, \ldots, x_n)$$

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

[Fórmula 109]

$$\sum_{i=1}^{n} x_i v_i$$

Obsérvese que X^T indica la transpuesta de la matriz M.

Cuando b_i (i = 1, ..., n) es un elemento de un vector de un espacio V, esto es, cuando se establece la Fórmula 110, la Fórmula 111 indica un subespacio generado por la Fórmula 112.

[Fórmula 110]

$$b_i \in \mathbb{V} \ (i = 1, \ldots, n)$$

[Fórmula 111]

$$\text{intervalo}\left\langle b_1, \ldots, b_n \right\rangle \subseteq \mathbb{V} \ (\text{resp. intervalo}\left\langle \vec{x}_1, \ldots, \vec{x}_n \right\rangle)$$

[Fórmula 112]

$$b_1, \ldots, b_n \ (\text{resp. } \vec{x}_1, \ldots, \vec{x}_n)$$

Obsérvese que para las bases B y B* indicadas en la Fórmula 113, se establece la Fórmula 114.

[Fórmula 113]

$$\mathbb{B} := (b_1, \ldots, b_N),$$

$$\mathbb{B}^* := (b_1^*, \ldots, b_N^*)$$

[Fórmula 114]

$$(x_1, \ldots, x_N)_{\mathbb{B}} := \sum_{i=1}^{N} x_i b_i,$$

$$(y_1, \ldots, y_N)_{\mathbb{B}^*} := \sum_{i=1}^{N} y_i b_i^*$$

En la siguiente descripción, cuando “nt” se indica para incluir un subíndice o un superíndice, nt es n_t. Del mismo modo, cuando “Vt” se indica para incluir un subíndice o un superíndice, Vt es V_t. Del mismo modo, cuando “$\delta_{i,j}$” se indica para incluir un superíndice, δi,j es $\delta_{i,j}$.

Cuando “→” que indica que un vector está unido a un subíndice o superíndice, “→” se une como superíndice al subíndice o superíndice.

Cuando B0 y Bd+1 que representan una base se indica para incluir un subíndice, B0 o Bd+1 representa B_0 o B_{d+1}. Del mismo modo, cuando B*0 y B*d+1 que representan una base se indican para incluir un subíndice, B*0 o B*d+1 representa B^*_0 o B^*_{d+1}.

En la siguiente descripción, un proceso criptográfico incluye un proceso de generación de claves, un proceso de cifrado y un proceso de descifrado, y un proceso de firma incluye un proceso de generación de claves, un proceso de firma y un proceso de verificación.

Realización 1.

Esta realización describe un concepto básico para implementar el “esquema de cifrado funcional a ser descrito en las siguientes realizaciones, y una estructura del cifrado funcional.

En primer lugar, se explicará brevemente el cifrado funcional.

En segundo lugar, se describirá un espacio que tiene una estructura matemática rica llamada "espacios vectoriales de emparejamiento dual (DPVS)” que es un espacio para implementar el cifrado funcional.

En tercer lugar, se describirá un concepto para implementar el cifrado funcional. Aquí, se describirán un “programa de intervalo”, “el producto interno de vectores de atributo, y una estructura de acceso”, y un “esquema de distribución de secreto (esquema de compartición de secreto)”.

En cuarto lugar, se describirá un “esquema de cifrado funcional” según esta realización. En esta realización, se describirá un “esquema de cifrado funcional de política de clave (KP-FE)”. Inicialmente, se describirá la estructura básica del “esquema de cifrado funcional de política de clave”. Posteriormente, se describirá la estructura básica de un “sistema de procesamiento criptográfico 10” que implementa el “esquema de cifrado funcional de política de clave”. Luego, se describirán en detalle un “esquema de cifrado funcional de política de clave” y un “sistema de procesamiento criptográfico 10” según esta realización.

<1. Esquema de cifrado funcional>

El esquema de cifrado funcional es un esquema de cifrado que proporciona relaciones más sofisticadas y flexibles entre una clave de cifrado (ek) y una clave de descifrado (dk).

Según el esquema de cifrado funcional, un atributo x y un atributo y se establecen en una clave de cifrado y una clave de descifrado, respectivamente. Una clave de descifrado $dk_v := (dk, v)$ puede descifrar un texto cifrado, cifrado con una clave de cifrado $ek_x := (ek, x)$ sólo si R(x, v) se mantiene para la relación R.

El esquema de cifrado funcional tiene distintas aplicaciones en las áreas de control de acceso de bases de datos, servicios de correo, distribución de contenidos y similares (véanse las Citas Bibliográficas no de Patente 2, 7, 9, 16, 19, 23 a 26 y 28).

Cuando R es una relación de igualdad, es decir, cuando R(x, v) se mantiene sólo si x = v, el esquema de cifrado funcional es el esquema de cifrado basado en ID.

Como una clase más general de esquema de cifrado funcional que el esquema de cifrado basado en ID, se han propuesto esquemas de cifrado basado en atributos.

Según los esquemas de cifrado basado en atributos, cada atributo que se establece para una clave de cifrado y una clave de descifrado es una tupla de atributos. Por ejemplo, los atributos que se establecen para una clave de cifrado y una clave de descifrado son $X := (x_1, ..., x_d)$ y $V := (v_1, ..., v_d)$, respectivamente.

Las relaciones de igualdad en modo componente para los componentes de atributo (por ejemplo, $\{x_t = v_t\}_{t \in \{1, ..., d\}}$) se introducen en una estructura de acceso S, y R(X, V) se mantiene si y sólo si la entrada se acepta por la estructura de acceso S. Esto es, el texto cifrado, cifrado con la clave de cifrado se puede descifrar con la clave de descifrado.

Si la estructura de acceso S está integrada en la clave de descifrado dk_v, el esquema de cifrado basado en atributos (ABE) se denomina ABE de política de clave (KP-ABE). Si la estructura de acceso S está integrada en un texto cifrado, el esquema de cifrado basado en atributos (ABE) se denomina ABE de política de texto cifrado (CP-ABE).

El cifrado de producto interno (IPE) descrito en la Cita Bibliográfica no de Patente 19 es también una clase de cifrado funcional, donde cada atributo para la clave de cifrado y la clave de descifrado es un vector sobre un campo o anillo. Por ejemplo, $\vec{x} := (x_1, ..., x_n) \in F_q^n$ y $\vec{v} := (v_1, ..., v_n) \in F_q^n$ se establecen para la clave de cifrado y la clave de descifrado. $R(\vec{x}, \vec{v})$ se mantiene si y sólo si $\vec{x} \cdot \vec{y} = 0$.

<2. Espacios vectoriales de emparejamiento dual>

En primer lugar, se describirán grupos de emparejamiento bilineal simétricos.

Los grupos de emparejamiento bilineal simétricos (q, G, G^T, g, e) son una tupla de un primo q, un grupo aditivo cíclico G de orden q y un grupo multiplicativo cíclico G^T de orden q, $g \neq 0 \in G$, y un emparejamiento bilineal no degenerado calculable de polinomio de tiempo $e : G \times G \rightarrow G_T$. El emparejamiento bilineal no degenerado significa $e(sg, tg) = e(g, g)^{st}$, y $e(g, g) \neq 1$.

En la siguiente descripción, permitamos que la Fórmula 115 sea un algoritmo que tome como entrada 1^λ y emita un valor de un parámetro $param_G := (q, G, G_T, g, e)$ de grupos de emparejamiento bilineales con un parámetro de seguridad λ.

[Fórmula 115]

$\mathcal{G}_{bpg}$

Se describirán ahora espacios vectoriales de emparejamiento dual.

Los espacios vectoriales de emparejamiento dual (q, V, G_T, A, e) pueden estar constituidos por un producto directo de grupos de emparejamiento bilineal simétricos ($param_G$:= (q, G, G_T, g, e)). Los espacios vectoriales de emparejamiento dual (q, V, G_T, A, e) son una tupla de un primo q, un espacio vectorial N-dimensional V sobre F_q indicado en la Fórmula 116, un grupo cíclico G_T del orden q, y una base canónica A := (a_1, ..., a_N) del espacio V, y tienen las siguientes operaciones (1) y (2) donde a_i es como se indica en la Fórmula 117.

[Fórmula 116]

$$\mathbb{V} := \overbrace{\mathbb{G} \times \cdots \times \mathbb{G}}^{N}$$

[Fórmula 117]

$$a_i := (\overbrace{0,\ldots,0}^{i-1}, g, \overbrace{0,\ldots,0}^{N-i})$$

Operación (1): Emparejamiento bilineal no degenerado

El emparejamiento en el espacio V se define por la Fórmula 118.

[Fórmula 118]

$$e(x,y) := \prod_{i=1}^{N} e(G_i, H_i) \in \mathbb{G}_T$$

donde

$$(G_1,\ldots,G_N) := x \in \mathbb{V},$$
$$(H_1,\ldots,H_N) := y \in \mathbb{V}$$

Este es bilineal no degenerado, es decir, e(sx, ty) = $e(s, y)^{st}$ y si e(x, y) = 1 para todo y $\in$ V, entonces x = 0. Para todo i y j, $e(a_i, a_j) = e(g, g)^{\delta_{i,j}}$ donde $\delta_{i,j} = 1$ si i = j, y $\delta_{i,j} = 0$ si i ≠ j. También, e(g, g) ≠ 1 $\in$ G_T.

Operación (2): Mapas de distorsión

La transformación lineal $\Phi_{i,j}$ en el espacio V indicada en la Fórmula 119 puede lograr la Fórmula 120.

[Fórmula 119]

$$\phi_{i,j}(a_j) = a_i$$
$$si \quad k \neq j \quad entonces \quad \phi_{i,j}(a_k) = 0$$

[Fórmula 120]

$$\phi_{i,j}(x) := (\overbrace{0,\ldots,0}^{i-1}, g_j, \overbrace{0,\ldots,0}^{N-i})$$

Obsérvese que

$(g_1, \ldots, g_N) := x$

La transformación lineal $\Phi_{i,j}$ se denominará “mapas de distorsión”.

En la siguiente descripción, permitamos que la Fórmula 121 sea un algoritmo que toma como entrada, 1^{λ} ($\lambda \in$ número natural), $N \in$ número natural, y el valor del parámetro $param_G := (q, G, G_T, g, e)$ de grupos de emparejamiento bilineales, y emite el valor de un parámetro $param_v := (q, V, G_T, A, e)$ de espacios vectoriales de emparejamiento dual que tienen un parámetro de seguridad λ, y que forman espacio N-dimensional V.

[Fórmula 121]

$$\mathcal{G}_{\mathrm{dpvs}}$$

Se describirá un caso donde se construyen espacios vectoriales de emparejamiento dual a partir de los grupos de emparejamiento bilineal simétricos descritos anteriormente. Los espacios vectoriales de emparejamiento dual se pueden construir también a partir de grupos de emparejamiento bilineal asimétricos. La siguiente descripción se puede aplicar fácilmente a un caso donde se construyen espacios vectoriales de emparejamiento dual a partir de grupos de emparejamiento bilineal asimétricos.

<3. Concepto para implementar un cifrado funcional>

<3-1. Programa de intervalo>

La Fig. 1 es un dibujo explicativo de una matriz M^.

Permitamos que $\{p_1, \ldots, p_n\}$ sea un conjunto de variables. M^ := (M, ρ) es una matriz etiquetada donde la matriz M es una matriz (L filas x r columnas) sobre F_q, y ρ es una etiqueta de la columna de la matriz M y está relacionada con uno de los literales $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$. Una etiqueta ρ_i (i = 1, ..., L) de cada fila de M está relacionada con uno de los literales, esto es, $\rho : \{1, ..., L\} \rightarrow \{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$.

Para cada secuencia de entrada $\delta \in \{0, 1\}^n$, se define una submatriz M_δ de la matriz M. La matriz M_δ es una submatriz que consiste en aquellas filas de la matriz M, cuyas etiquetas ρ se relacionan con un valor "1" mediante la secuencia de entrada δ. Esto es, la matriz M_δ es una submatriz que consiste en las filas de la matriz M que están relacionadas con p_i con los cuales $\delta_i = 1$ y las filas de la matriz M que están relacionadas con $\neg p_i$ con los cuales $\delta_i = 0$.

La Fig. 2 es un dibujo explicativo de la matriz M_δ. Obsérvese que en la Fig. 2, n = 7, L = 6 y r = 5. Es decir, el conjunto de variables es $\{p_1, \ldots, p_7\}$, y la matriz M es una matriz (6 filas x 5 columnas). En la Fig. 2, suponemos que las etiquetas ρ están relacionadas de manera que ρ_1 corresponde a $\neg p_2$, ρ_2 a p_1, ρ_3 a p_4, ρ_4 a $\neg p_5$, ρ_5 a $\neg p_3$ y ρ_6 a $\neg p_5$.

Supongamos que en una secuencia de entrada $\delta \in \{0,1\}^7$, $\delta_1 = 1$, $\delta_2 = 0$, $\delta_3 = 1$, $\delta_4 = 0$, $\delta_5 = 0$, $\delta_6 = 1$ y $\delta_7 = 1$. En este caso, una submatriz que consiste en las filas de la matriz M que están relacionadas con los literales (p_1, p_3, p_6, p_7, $\neg p_2$, $\neg p_4$, $\neg p_5$) rodeados por líneas discontinuas es la matriz M_δ. Es decir, la submatriz que consiste en la primera fila (M_1), la segunda fila (M_2) y la cuarta fila (M_4) de la matriz M es la matriz M_δ.

En otras palabras, cuando el mapa $\gamma : \{1, ..., L\} \rightarrow \{0, 1\}$ es $[\rho(j) = p_i] \wedge [\delta_i = 1]$ o $[\rho(j) = \neg p_i] \wedge [\delta_i = 0]$, entonces $\gamma(j) = 1$; de otro modo $\gamma(j) = 0$. En este caso, $M_\delta := (M_j)_{\gamma(j) = 1}$. Obsérvese que M_j es la fila de orden j de la matriz M.

Es decir, en la Fig. 2, el mapa $\gamma(j) = 1$ (j = 1, 2, 4), así el mapa $\gamma(j) = 0$ (j = 3, 5, 6). Por lo tanto, $(M_j)_{\gamma(j) = 1}$ es M_1, M_2 y M_4, y la matriz M_δ.

Más específicamente, si la fila de orden i de la matriz M se incluye o no en la matriz M_δ se determina mediante si el valor del mapa $\gamma(j)$ es "0" o "1".

El programa de intervalo M^ acepta una secuencia de entrada δ si y sólo si $1^{\rightarrow} \in$ intervalo $<M_\delta>$, y rechaza la secuencia de entrada δ de otro modo. Esto es, el programa de intervalo M^ acepta la secuencia de entrada δ si y sólo si la combinación lineal de las filas de la matriz M_δ que se obtienen de la matriz M^ mediante la secuencia de entrada δ da $1^{\rightarrow}$. $1^{\rightarrow}$ es un vector de fila que tiene un valor de "1" en cada elemento.

Por ejemplo, en la Fig. 2, el programa de intervalo M^ acepta la secuencia de entrada δ si y sólo si la combinación lineal de las respectivas filas de la matriz M_δ que consiste en la 1ª, 2ª y 4ª filas de la matriz M da $1^{\rightarrow}$. Es decir, si existen α_1, α_2 y α_4 con las cuales $\alpha_1(M_1) + \alpha_2(M_2) + \alpha_4(M_4) = 1^{\rightarrow}$, el programa de intervalo M^ acepta la secuencia de entrada δ.

Un programa de intervalo se llama monótono si las etiquetas ρ se relacionan solamente con los literales positivos $\{p_1, \ldots, p_n\}$. Un programa de intervalo se llama no monótono si las etiquetas ρ se relacionan con los literales $\{p_1, \ldots, p_n, \neg p_1, \ldots, \neg p_n\}$. Supongamos que el programa de intervalo es no monótono. Una estructura de acceso (estructura de acceso no monótona) se constituye usando el programa de intervalo no monótono. En resumen, una estructura de acceso controla el acceso al cifrado, esto es, controla si ha de ser descifrado o no un texto cifrado.

Debido a que el programa de intervalo no es monótono sino no monótono, se amplía la aplicación de los esquemas de cifrado funcional constituidos usando el programa de intervalo. Esto se describirá en detalle más tarde.

<3-2. Producto interno de los vectores de atributo y la estructura de acceso>

El mapa γ(j) descrito anteriormente se calculará usando los productos internos de los vectores de atributos. Esto es, qué fila de la matriz M ha de ser incluida en la matriz M_δ se determinará usando los productos internos de los vectores de atributo.

U_t (t = 1, ..., d y $U_t \subset \{0, 1\}^*$) es un subuniverso y conjunto de atributos. Cada U_t incluye información de identificación (t) del subuniverso y vector n_t dimensional ($\vec{v}$). Esto es, U_t es (t, $\vec{v}$) donde $t \in \{1, ..., d\}$ y $\vec{v} \in F_q^{nt}$.

Permitamos que U_t = (t, $\vec{v}$) sea una variable p del programa de intervalo M^ := (M, ρ), es decir, p := (t, $\vec{v}$). Permitamos que un programa de intervalo M^ := (M, ρ) que tiene la variable (p := (t, $\vec{v}$), (t', $\vec{v'}$), ...) sea una estructura de acceso S.

Es decir, la estructura de acceso S := (M, ρ), y ρ : {1, ..., L} → {(t, $\vec{v}$), (t', $\vec{v'}$), ..., ¬(t, $\vec{v}$), ¬(t', $\vec{v'}$), ...}.

Permitamos que Γ sea un conjunto de atributos, es decir, Γ := {(t, $\vec{x}_t$) | $\vec{x}_t \in F_q^{nt}$, $1 \le t \le d$}.

Cuando Γ se da a la estructura de acceso S, el mapa γ : {1, ..., L} → {0, 1} para el programa de intervalo M^ := (M, ρ) se define como sigue. Para cada número entero i = 1, ..., L, establecemos γ(j) = 1 si [ρ(i) = (t, $\vec{v}_i$)] ^ [(t, $\vec{x}_t$)] ∈ Γ] ^ [$\vec{v}_i \cdot \vec{x}_t$ = 0] o [ρ(i) = ¬(t, $\vec{v}_i$)] ^ [(t, $\vec{x}_t$)] ∈ Γ] ^ [$\vec{v}_i \cdot \vec{x}_t \neq 0$]. Establecemos γ(j) = 0 de otro modo.

Esto es, el mapa γ se calcula en base al producto interno de los vectores de atributo $\vec{v} \cdot$ y $\vec{x}$. Como se ha descrito anteriormente, qué fila de la matriz M ha de ser incluida en la matriz M_δ se determina por el mapa γ. Más específicamente, qué fila de la matriz M ha de ser incluida en la matriz M_δ se determina por el producto interno de los vectores de atributo $\vec{v}$ y $\vec{x}$. La estructura de acceso S := (M, ρ) acepta Γ si y sólo si $1^{\rightarrow} \in$ intervalo $\langle (M_i)_{\gamma(i)=1} \rangle$.

<3-3. Esquema de compartición de secreto>

Se describirá un esquema de compartición de secreto para la estructura de acceso S := (M, ρ).

El esquema de compartición de secreto está permitiendo que la información de secreto sea compartida para presentarla como información compartida sin sentido. Por ejemplo, la información de secreto s se permite que sea compartida entre 10 paquetes para generar 10 partes de información compartida. Cada una de las 10 partes de información compartida no tiene información sobre la información de secreto s. Por lo tanto, incluso cuando se obtiene una cierta parte de información compartida, no se puede obtener información sobre la información de secreto s. Por otra parte, si se obtienen todas de las 10 partes de información compartida, se puede recuperar la información de secreto s.

Otro esquema de compartición de secreto también está disponible según el cual, incluso cuando no se puedan obtener todas de las 10 partes de información compartida, si se pueden obtener una o más, pero no todas, (por ejemplo, 8 partes) de información compartida, entonces se puede recuperar la información de secreto. Un caso como éste donde la información de secreto s se puede recuperar usando 8 de entre las 10 partes de información compartida se llamará 8 de entre 10. Es decir, un caso donde la información de secreto s se puede recuperar usando t de entre n partes de información compartida se llamará t de entre n. Esta t se llamará umbral.

También, otro esquema de compartición de secreto más está disponible según el cual cuando se generan 10 partes de información compartida d_1, …, d_{10}, la información de secreto s se puede recuperar si se dan 8 partes de información compartida d_1, …, d_8, pero no se puede si se dan 8 partes de información compartida d_3, …, d_{10}. Esto es, hay un esquema de compartición de secreto con el que si se puede recuperar o no la información de secreto s se controla no solamente por el número de partes de información compartida obtenido, sino también dependiendo de la combinación de la información compartida.

La Fig. 3 es un dibujo explicativo de s_0. La Fig. 4 es un dibujo explicativo de $s^{\rightarrow T}$.

Permitamos que una matriz M sea una matriz (L filas x r columnas). Permitamos que $\vec{f}$ sea un vector de columna indicado en la Fórmula 122.

[Fórmula 122]

$$\vec{f}^{\mathrm{T}} := (f_1, \ldots f_r)^T \xleftarrow{U} \mathbb{F}_q^r$$

Permitamos que s_0 indicado en la Fórmula 123 sea la información de secreto a ser compartida.

[Fórmula 123]

$$s_0 := \vec{1} \cdot \vec{f}^{\mathrm{T}} := \sum_{k=1}^{r} f_k$$

Permitamos que $s^{\rightarrow T}$ indicado en la Fórmula 124 sea el vector de L partes de información compartida de s_0.

[Fórmula 124]

$$\vec{s}^{\mathrm{T}} := (s_1, \ldots s_L)^{\mathrm{T}} := M \cdot \vec{f}^{\mathrm{T}}$$

Permitamos que la información compartida s_i pertenezca a $\rho(i)$.

Si la estructura de acceso S := (M, ρ) acepta Γ, es decir, $1^{\rightarrow} \in$ intervalo $<(M_i)_{\gamma(i) = 1}>$ para $\gamma : \{1, ..., L\} \rightarrow \{0, 1\}$, entonces existen constantes $\{\alpha_i \in Fq \mid i \in I\}$, de manera que $I \subseteq \{i \in \{1, ..., L\} \mid \gamma(i) = 1\}$.

Esto es obvio a partir de la explicación de la Fig. 2 en que si existen α_1, α_2 y α_4 con las cuales $\alpha_1(M_1) + \alpha_2(M_2) + \alpha_4(M_4) = 1^{\rightarrow}$, el programa de intervalo M^ acepta una secuencia de entrada δ. Esto es, si el programa de intervalo M^ acepta la secuencia de entrada δ cuando existen α_1, α_2 y α_4 con las cuales $\alpha_1(M_1) + \alpha_2(M_2) + \alpha_4(M_4) = 1^{\rightarrow}$, entonces existen α_1, α_2 y α_4 con las cuales $\alpha_1(M_1) + \alpha_2(M_2) + \alpha_4(M_4) = 1^{\rightarrow}$.

Obsérvese la Fórmula 125.

[Fórmula 125]

$$\sum_{i \in I} \alpha_i s_i := s_0$$

Obsérvese que las constantes $\{\alpha_i\}$ se pueden calcular en polinomios de tiempo en el tamaño de la matriz M.

Con el esquema de cifrado funcional según esta y las siguientes realizaciones, se construye una estructura de acceso aplicando el predicado de producto interno y el esquema de compartición de secreto al programa de intervalo, como se ha descrito anteriormente. Por lo tanto, se puede diseñar un control de acceso de manera flexible diseñando la matriz M en el programa de intervalo y la información de atributo x y la información de atributo v (información de predicado) en el predicado de producto interno. Esto es, se puede diseñar un control de acceso con un grado de libertad muy alto. El diseño de la matriz M corresponde al diseño de condiciones tales como el umbral del esquema de compartición de secreto.

Por ejemplo, el esquema de cifrado basado en atributos descrito anteriormente corresponde a un caso, en la estructura de acceso en el esquema de cifrado funcional según esta y las siguientes realizaciones, donde el diseño del predicado del producto interno está limitado a una cierta condición. Es decir, cuando se compara con la estructura de acceso en el esquema de cifrado funcional según esta y las siguientes realizaciones, la estructura de acceso en el esquema de cifrado basado en atributos tiene un grado de libertad menor en el diseño de un control de acceso debido a que carece del grado de libertad en el diseño de la información de atributo x y la información de atributo v (información de predicado) en el predicado del producto interno. Más específicamente, el esquema de cifrado basado en atributos corresponde a un caso donde la información de atributo $\{x^{\rightarrow}t\}t \in \{1, ..., d\}$ y $\{v^{\rightarrow}t\}t \in \{1, ..., d\}$ está limitada a vectores bidimensionales para la relación de igualdad, por ejemplo, $x^{\rightarrow}t := (1, x_t)$ y $v^{\rightarrow}t := (v_t, -1)$.

El esquema de cifrado de predicado del producto interno descrito anteriormente corresponde a un caso, en la estructura de acceso en el esquema de cifrado funcional según esta y las siguientes realizaciones, donde el diseño de la matriz M en el programa de intervalo está limitado a una cierta condición. Es decir, cuando se compara con la estructura de acceso en el esquema de cifrado funcional según esta y las siguientes realizaciones, la estructura de acceso en el esquema de cifrado de predicado del producto interno tiene un grado de libertad menor en el diseño de un control de acceso debido a que carece del grado de libertad en el diseño de la matriz M en el programa de intervalo. Más específicamente, el esquema de cifrado de predicado del producto interno corresponde a un caso donde el esquema de compartición de secreto está limitado a 1 de entre 1 (o d de entre d).

En particular, la estructura de acceso en el esquema de cifrado funcional según esta y las siguientes realizaciones constituye una estructura de acceso no monótona que usa un programa de intervalo no monótono. De esta manera, el grado de libertad en el diseño de un control de acceso mejora.

Más específicamente, dado que el programa de intervalo no monótono incluye un literal negativo (¬p), se puede establecer una condición negativa. Por ejemplo, supongamos que Primera Empresa incluye cuatro departamentos de A, B, C y D. Supongamos que ha de ser realizado el control de acceso que solamente los usuarios que pertenecen a departamentos distintos del departamento B de la Primera Empresa sean capaces de acceder (capaces de descifrado). En este caso, si no se puede establecer una condición negativa, se debe establecer una

condición de que “el usuario pertenezca a cualquiera de los departamentos A, C y D de la Primera Empresa”. Por otra parte, si se puede establecer una condición negativa, se puede establecer una condición en la que “el usuario es empleado de la Primera Empresa y pertenece a un departamento distinto del departamento B”. Esto es, dado que se puede establecer una condición negativa, es posible establecer una condición natural. Aunque el número de departamentos es pequeño en este caso, este esquema es muy eficaz en un caso donde el número de departamentos es grande.

<4 Estructura básica de esquema de cifrado funcional>

<4-1. Estructura básica de esquema de cifrado funcional de política de clave>

Se describirá brevemente la estructura de un esquema de cifrado funcional de política de clave. Obsérvese que política de clave significa que la política está integrada en la clave de descifrado, es decir, que una estructura de acceso está integrada en la clave de descifrado.

El esquema funcional de política de clave consiste en cuatro algoritmos: Setup, KeyGen, Enc y Dec.

(Setup)

Un algoritmo Setup es un algoritmo aleatorizado que toma como entrada un parámetro de seguridad λ y un formato de atributo $\vec{\mu} := (d; n_1, \ldots, n_d)$, y emite los parámetros públicos pk y la clave maestra sk.

(KeyGen)

Un algoritmo KeyGen es un algoritmo aleatorizado que toma como entrada una estructura de acceso $S := (M, \rho)$, los parámetros públicos pk y la clave maestra sk, y emite una clave de descifrado sk_s.

(Enc)

Un algoritmo Enc es un algoritmo aleatorizado que toma como entrada un mensaje m, un conjunto de atributos $\Gamma := \{(t, \vec{x}_t) \mid \vec{x}_t) \in F_q^{nt}, 1 \le t \le d\}$ y los parámetros públicos pk, y emite los datos cifrados c.

(Dec)

Un algoritmo Dec es un algoritmo que toma como entrada los datos cifrados c cifrados bajo el conjunto de atributos Γ, la clave de descifrado sk_s, para la estructura de acceso S, y los parámetros públicos pk, y emite o bien el mensaje m o bien el símbolo distinguido $\perp$.

Un esquema de cifrado funcional de política de clave debería tener la siguiente propiedad: para todas las estructuras de acceso S, el conjunto de atributos Γ, los parámetros públicos pk generados correctamente, la clave maestra sk, y el texto cifrado c indicado en la Fórmula 126, mantiene que $m = Dec(pk, sk_s, c)$ si la estructura de acceso S acepta el conjunto de atributos Γ. Si la estructura de acceso S rechaza el conjunto de atributos Γ, la probabilidad de $m = Dec(pk, sk_s, c)$ es despreciable.

[Fórmula 126]

$$c \xleftarrow{R} \mathrm{Enc}(\mathrm{pk}, m, \Gamma)$$

<4-2. Sistema de procesamiento criptográfico 10>

Se describirá un sistema de procesamiento criptográfico 10 que ejecuta los algoritmos del esquema de cifrado funcional de política de clave descrito anteriormente.

La Fig. 5 es un diagrama de configuración del sistema de procesamiento criptográfico 10 que ejecuta el esquema de cifrado funcional de política de clave.

El sistema de procesamiento criptográfico 10 está dotado con un dispositivo de generación de claves 100, un dispositivo de cifrado 200 y un dispositivo de descifrado 300.

El dispositivo de generación de claves 100 ejecuta el algoritmo Setup tomando como entrada un parámetro de seguridad λ y un formato de atributo $\vec{\mu} := (d; n_1, \ldots, n_d)$, y genera los parámetros públicos pk y una clave maestra sk. El dispositivo de generación de claves 100 hace públicos los parámetros públicos pk generados. El dispositivo de generación de claves 100 también ejecuta el algoritmo KeyGen tomando como entrada una estructura de acceso S, genera una clave de descifrado sk_s, y distribuye la clave de descifrado sk_s al dispositivo de descifrado 300 en secreto.

El dispositivo de cifrado 200 ejecuta el algoritmo Enc tomando como entrada un mensaje m, un conjunto de atributos Γ, y los parámetros públicos pk, y genera los datos cifrados c. El dispositivo de cifrado 200 transmite los datos cifrados c generados al dispositivo de descifrado 300.

El dispositivo de descifrado 300 ejecuta el algoritmo Dec tomando como entrada los parámetros públicos pk, la clave de descifrado sk_s y los datos cifrados c, y emite un mensaje m o símbolo distinguido $\perp$.

<4-3. Esquema de cifrado funcional de política de clave y sistema de procesamiento criptográfico 10 en detalle>

El esquema de cifrado funcional de política de clave, y la función y la operación del sistema de procesamiento criptográfico 10 que ejecuta el esquema de cifrado funcional de política de clave se describirán con referencia a las Fig. 6 a 10.

La Fig. 6 es un diagrama de bloques de funciones que muestra la función del sistema de procesamiento criptográfico 10 que ejecuta el esquema de cifrado funcional de política de clave. El sistema de procesamiento criptográfico 10 está dotado con el dispositivo de generación de claves 100, el dispositivo de cifrado 200 y el dispositivo de descifrado 300, como se ha descrito anteriormente.

Las Fig. 7 y 8 son diagramas de flujo que muestran la operación del dispositivo de generación de claves 100. Obsérvese que la Fig. 7 es un diagrama de flujo que muestra el proceso del algoritmo Setup, y que la Fig. 8 es un diagrama de flujo que muestra el proceso del algoritmo KeyGen. La Fig. 9 es un diagrama de flujo que muestra la operación del dispositivo de cifrado 200 y el proceso del algoritmo Enc. La Fig. 10 es un diagrama de flujo que muestra la operación del dispositivo de descifrado 300 y el proceso del algoritmo Dec.

Supongamos que $x_{t,l} := 1$ en la siguiente descripción.

Se describirá la función y la operación del dispositivo de generación de claves 100. El dispositivo de generación de claves 100 está dotado con una parte de generación de clave maestra 110, una parte de almacenamiento de clave maestra 120, una parte de entrada de información 130 (primera parte de entrada de información), una parte de generación de clave de descifrado 140, y una parte de distribución de clave 150. La parte de generación de clave de descifrado 140 está dotada con una parte de generación de vector f 141, una parte de generación de vector s 142, una parte de generación de número aleatorio 143 y una parte de generación de elemento de clave 144.

El proceso del algoritmo Setup se describirá primero con referencia a la Fig. 7.

(S101: Paso de generación de base ortogonal regular)

La parte de generación de clave maestra 110 calcula la Fórmula 127 con el dispositivo de procesamiento para generar $param_{\mu\rightarrow}$, y las bases B_t y B^*_t para cada número entero t = 0, ..., d.

[Fórmula 127]

(1)

introducir $1^\lambda, \vec{\mu} := (d; n_1 \ldots, n_d)$

(2)

$$\mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{R} \mathcal{G}_{\mathsf{bpg}}(1^\lambda)$$

(3)

$$\psi \xleftarrow{U} \mathbb{F}_q^\times, N_0 := 5, \quad N_t := 4n_t \text{ para } t = 1, \ldots, d$$

Los procesos de (4) a (8) se ejecutan para cada t = 0, …, d.

(4)

$$\mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}})$$

(5)

$$X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{\cup} GL(N_t, \mathbb{F}_q)$$

(6)

$$(\nu_{t,i,j})_{i,j} := \psi \cdot (X_t^{\mathrm{T}})^{-1}$$

(7)

$$b_{t,i} := (\chi_{t,i,1}, \cdots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} a_{t,j},$$

$$\mathbb{B}_t := (b_{t,1}, \ldots, b_{t,N_t})$$

(8)

$$b^*_{t,i} := (\nu_{t,i,1}, \cdots, \nu_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \nu_{t,i,j} a_{t,j},$$

$$\mathbb{B}^*_t := (b^*_{t,1}, \ldots, b^*_{t,N_t})$$

(9)

$$g_T := e(g,g)^{\psi}$$

$$\overrightarrow{\mathrm{param}_{\mu}} := (\{\mathrm{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d}, g_T)$$

Esto es, la parte de generación de clave maestra 110 ejecuta los siguientes procesos.

(1) Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 toma como entrada el parámetro de seguridad $\lambda(1^{\lambda})$ y el formato de atributo $\vec{\mu} := (d; n_1, \ldots, n_d)$, donde d es un número entero de 1 o más, y n_t es un número entero de 1 o más para cada número entero t = 1, ..., d.

(2) Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 ejecuta el algoritmo G_{bpg} tomando como entrada el parámetro de seguridad $\lambda(1^{\lambda})$ introducido en (1), y genera el valor de un parámetro $\mathrm{param}_G := (q, G, G_T, g, e)$ del grupo de emparejamiento bilineal.

(3) Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 genera un número aleatorio ψ, y establece 5 en N_0 y $4n_t$ en N_t para cada número entero t = 1, ..., d.

Posteriormente, la parte de generación de clave maestra 110 ejecuta los procesos de los siguientes (4) a (8) para cada número entero t = 0, ..., d.

(4) Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 ejecuta el algoritmo G_{dpvs} tomando como entrada el parámetro de seguridad $\lambda(1^{\lambda})$ introducido en (1), N_t establecido en (3) y el valor de $\mathrm{param}_G := (q, G, G_T, g, e)$ generado en (2), y genera el valor del parámetro $\mathrm{param}_{vt} := (q, V_T, G_T, A, e)$ de los espacios vectoriales de emparejamiento dual.

(5) Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 toma como entrada N_t establecido en (3), y F_q, y genera la transformación lineal $X_t := (\chi_{t,i,j})_{i,j}$ aleatoriamente. Obsérvese que GL representa Lineal General. Esto es, GL es un grupo lineal general, un conjunto de matrices cuadradas en el que el determinante no es 0, y un grupo con respecto a multiplicación. Obsérvese que $(\chi_{t,i,j})_{i,j}$ significa una matriz: concerniente a los sufijos i y j de la matriz $\chi_{t,i,j}$ donde i, j = 1, ..., n_t.

(6) Con el dispositivo de procesamiento y en base al número aleatorio ψ y la transformación lineal X_t, la parte de generación de clave maestra 110 genera $(v_{t,i,j})_{i,j} := \psi \cdot (X_t^T)^{-1}$. Como lo hace $(\chi_{t,i,j})_{i,j}$, $(v_{t,i,j})_{i,j}$ significa una matriz concerniente a los sufijos i y j de la matriz: $v_{t,i,j}$ donde i, j = 1, ..., n_t.

(7) Con el dispositivo de procesamiento y en base a la transformación lineal X_t generada en (5), la parte de generación de clave maestra 110 genera la base B_t a partir de la base canónica A_t generada en (4).

(8) Con el dispositivo de procesamiento y en base a $(v_{t,i,j})_{i,j}$ generada en (6), la parte de generación de clave maestra 110 genera la base B^*_t a partir de la base canónica A_t generada en (4).

(9) Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 establece e(g, g)* en g_T. La parte de generación de clave maestra 110 también establece $\{param_{vt}\}_{t = 0, ..., d}$ generado en (4), y g_T, en $param_{\mu\rightarrow}$. Obsérvese que $g_T = e(b_{t,i}, b^*_{t,i})$ para cada número entero t = 0, ..., d y cada número entero i = 1, ..., N_t.

En resumen, en (S101), la parte de generación de clave maestra 110 ejecuta el algoritmo G_{ob} indicado en la Fórmula 128, y genera $param_{\mu\rightarrow}$ y las bases B_t y B^*_t para cada número entero t = 0, ..., d.

[Fórmula 128]

$$\mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{\mu} := (d; n_1 \ldots, n_d)) : \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{R} \mathcal{G}_{\mathsf{bpg}}(1^\lambda),$$

$$\psi \xleftarrow{U} \mathbb{F}_q^\times,$$

$$N_0 := 5, \quad N_t := 4n_t \ \text{ para } t = 1, \ldots, d,$$

$$\text{Para } t = 0, \ldots, d, \quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}}),$$

$$X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{U} GL(N_t, \mathbb{F}_q), \quad (\nu_{t,i,j})_{i,j} := \psi \cdot (X_t^{\mathrm{T}})^{-1},$$

$$b_{t,i} := (\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} a_{t,j}, \quad \mathbb{B}_t := (b_{t,1}, \ldots, b_{t,N_t}),$$

$$b^*_{t,i} := (\nu_{t,i,1}, \ldots, \nu_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \nu_{t,i,j} a_{t,j}, \quad \mathbb{B}^*_t := (b^*_{t,1}, \ldots, b^*_{t,N_t}),$$

$$g_T := e(g,g)^\psi, \quad \mathsf{param}^-_{\vec{\mu}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d}, g_T)$$

$$\text{devolver } (\mathsf{param}^-_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d}).$$

(S102: Paso de generación de parámetros públicos)

Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 genera una subbase $B\hat{}_0$ de la base B_t generada en (S101), y una subbase $B\hat{}_t$ para cada número entero t = 1, ..., d, como se indica en la Fórmula 129.

[Fórmula 129]

$$\hat{\mathbb{B}}_0 := (b_{0,1}, b_{0,3}, b_{0,5}),$$

$$\hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,n_t}, b_{t,3n_t+1}, \ldots, b_{t,4n_t})$$

$$\text{para } t = 1, \ldots, d$$

La parte de generación de clave maestra 110 trata las subbases generadas $B\hat{}_0$ y $B\hat{}_t$, el parámetro de seguridad $\lambda(1^\lambda)$ introducido en (S101) y $param_{\mu\rightarrow}$ generado en (S101), como los parámetros públicos pk.

(S103: Paso de generación de clave maestra)

Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 genera una subbase $B\hat{}^*_0$ de la base B^*_t generada en (S101), y una subbase $B\hat{}^*_t$ para cada número entero t = 1, ..., d, como se indica en la Fórmula 130.

[Fórmula 130]

$$\hat{\mathbb{B}}^*_0 := (b^*_{0,1}, b^*_{0,3}, b^*_{0,4})$$

$$\hat{\mathbb{B}}^*_t := (b^*_{t,1}, \ldots, b^*_{t,n_t}, b^*_{t,2n_t+1}, \ldots, b^*_{t,3n_t})$$

$$\text{para } t = 1, \ldots, d,$$

La parte de generación de clave maestra 110 trata las subbases generadas B^{*}_0 y B^{*}_t, como la clave maestra sk,.

(S104: Paso de almacenamiento de clave maestra)

La parte de almacenamiento de clave maestra 120 almacena los parámetros públicos pk incluyendo la subbase B^{*}_t generada en (S102) para cada número entero t = 0, ..., d, en el dispositivo de almacenamiento. La parte de almacenamiento de clave maestra 120 también almacena la clave maestra sk incluyendo la subbase B^{*}_t generada en (S103) para cada número entero t = 0, ..., d, en el dispositivo de almacenamiento.

En resumen, desde (S101) hasta (S103), la parte de generación de clave maestra 110 genera los parámetros públicos pk y la clave maestra sk ejecutando el algoritmo Setup indicado en la Fórmula 131. Entonces, en (S104), la parte de almacenamiento de clave maestra 120 almacena los parámetros públicos pk y la clave maestra sk generados, en el dispositivo de almacenamiento.

Obsérvese que los parámetros públicos se hacen públicos a través de, por ejemplo, una red, de modo que el dispositivo de cifrado 200 y el dispositivo de descifrado 300 pueden adquirirlos.

[Fórmula 131]

$$\mathsf{Setup}(1^{\lambda}, \vec{\mu} := (d; n_1, \ldots, n_d))$$

$$(\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}^*_t\}_{t=0,\ldots,d}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^{\lambda}, \vec{\mu})$$

$$\hat{\mathbb{B}}_0 := (b_{0,1}, b_{0,3}, b_{0,5}), \quad \hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,n_t}, b_{t,3n_t+1}, \ldots, b_{t,4n_t})$$

$$\text{para } t = 1, \ldots, d,$$

$$\hat{\mathbb{B}}^*_0 := (b^*_{0,1}, b^*_{0,3}, b^*_{0,4}), \quad \hat{\mathbb{B}}^*_t := (b^*_{t,1}, \ldots, b^*_{t,n_t}, b^*_{t,2n_t+1}, \ldots, b^*_{t,3n_t})$$

$$\text{para } t = 1, \ldots, d,$$

$$\mathsf{sk} := \{\hat{\mathbb{B}}^*_t\}_{t=0,\ldots,d}, \quad \mathsf{pk} := (1^{\lambda}, \mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t\}_{t=0,\ldots,d}).$$

$$\text{devolver } \mathsf{sk}, \ \mathsf{pk}.$$

El proceso del algoritmo KeyGen se describirá con referencia a la Fig. 8.

(S201: Paso de entrada de información)

Con el dispositivo de entrada, la parte de entrada de información 130 toma como entrada la estructura de acceso S := (M, ρ) descrita anteriormente. La matriz M de la estructura de acceso S se establece dependiendo de la condición del sistema que el usuario desee realizar. También, por ejemplo, la información de atributo del usuario de la clave de descifrado sk_s se establece en ρ de la estructura de acceso S.

(S202: Paso de generación de vector f)

Con el dispositivo de procesamiento, la parte de generación de vector f 141 genera un vector $\vec{f}$ que tiene r partes de elementos, aleatoriamente como se indica en la Fórmula 132.

[Fórmula 132]

$$\vec{f} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$$

(S203: Paso de generación de vector s)

Con el dispositivo de procesamiento, la parte de generación de vector s 142 genera un vector $\vec{s}^T := (s_1, \ldots, s_L)^T$ en base a la matriz M (L filas x r columnas) incluida en la estructura de acceso S introducida en (S201) y el vector $\vec{f}$ generado en (S202), como se indica en la Fórmula 133.

[Fórmula 133]

$$\vec{s}^T := (s_1, \ldots, s_L)^T := M \cdot \vec{f}^T$$

Con el dispositivo de procesamiento, la parte de generación de vector s 142 genera un valor s_0 en base al vector $\vec{f}$ generado en (S202), como se indica en la Fórmula 134.

[Fórmula 134]

$$s_0 := \vec{1} \cdot \vec{f}^T$$

(S204: Paso de generación de números aleatorios)

Con el dispositivo de procesamiento, la parte de generación de números aleatorios 143 genera un número aleatorio η_0, y un número aleatorio θ_i para cada número entero i = 1, ..., L, como se indica en la Fórmula 135.

[Fórmula 135]

$$\eta_0, \theta_i \xleftarrow{U} \mathbb{F}_q (i = 1, \ldots, L)$$

(S205: Paso de generación de elemento de clave)

Con el dispositivo de procesamiento, la parte de generación de elemento de clave 144 genera un elemento k^*_0 de la clave de descifrado sk_s, como se indica en la Fórmula 136.

[Fórmula 136]

$$k_0^* := (-s_0, 0, 1, \eta_0, 0)_{\mathbb{B}_0^*}$$

Como se ha descrito anteriormente, para las bases B y B* indicadas en la Fórmula 113, se establece la Fórmula 114. Por lo tanto, la Fórmula 136 significa establecer $-s_0$ como el coeficiente para un vector base $b^*_{0,1}$ de una base B^*_0, 0 como el coeficiente para un vector base $b^*_{0,2}$, 1 como el coeficiente para un vector base $b^*_{0,3}$, η_0 como el coeficiente para un vector base $b^*_{0,4}$, y 0 como el coeficiente para un vector base $b^*_{0,5}$.

Con el dispositivo de procesamiento, la parte de generación de elemento de clave 144 también genera un elemento k^*_i de la clave de descifrado sk_s para cada número entero i = 1, ..., L, como se indica en la Fórmula 137.

[Fórmula 137]

$$\text{si } \rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q$$

$$k_i^* := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{n_t},}^{n_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*}$$

$$\text{si } \rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$$

$$k_i^* := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t}),}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*}$$

Más específicamente, como lo hace la Fórmula 136, la Fórmula 137 significa, cuando ρ(i) es una tupla positiva (t, $\vec{v}_i$), estableciendo $s_i + \theta_i v_{i,1}$ como el coeficiente para un vector base $b^*_{t,1}$, $\theta_i v_{i,2}$, ..., $\theta_i v_{i,nt}$ como los coeficientes para los vectores base $b^*_{t,2}$, ..., $b^*_{t,nt}$, 0 como los coeficientes para los vectores base $b^*_{t,nt+1}$, ..., $b^*_{t,2nt}$, $\eta_{i,1}$, ..., $\eta_{i,nt}$ como los coeficientes para los vectores base $b^*_{t,2nt+1}$, ..., $b^*_{t,3nt}$, y 0 como los coeficientes para los vectores base $b^*_{t,3nt+1}$, ..., $b^*_{t,4nt}$, de la base B^*_t.

La Fórmula 137 también significa, cuando ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$), estableciendo $s_i v_{i,1}$, …, $s_i v_{i,nt}$ como los coeficientes para los vectores de base $b^*_{t,1}$,, ..., $b^*_{t,nt}$, 0 como los coeficientes para los vectores de base $b^*_{t,nt+1}$, ..., $b^*_{t,2nt}$, $\eta_{i,1}$, …, $\eta_{i,nt}$ como los coeficientes para los vectores de base $b^*_{t,2nt+1}$, ..., $b^*_{t,3nt}$, y 0 como los coeficientes para los vectores base $b^*_{t,3nt+1}$, ..., $b^*_{t,4nt}$, de la base B^*_t.

(S206: Paso de distribución de claves)

Por ejemplo, con el dispositivo de comunicación, la parte de distribución de claves 150 distribuye la clave de descifrado sk_s constituida por, como elementos, la estructura de acceso S introducida en (S201) y k^*_0, k^*_1, ..., k^*_t generados en (S205), en el dispositivo de descifrado 300 en secreto a través de la red. Como cuestión de rutina, la clave de descifrado sk_s se puede distribuir al dispositivo de descifrado 300 por otro método.

Más específicamente, desde (S201) hasta (S205), la parte de entrada de información 130 y la parte de generación de clave de descifrado 140 generan la clave de descifrado sk_s ejecutando el algoritmo KeyGen indicado en la Fórmula 138. Luego, en (S206), la parte de distribución de claves 150 distribuye la clave de descifrado sk_s generada al dispositivo de descifrado 300.

[Fórmula 138]

$$\mathrm{KeyGen}(\mathrm{pk},\mathrm{sk},\mathbb{S}:=(M,\rho))$$

$$\vec{f} \xleftarrow{\cup} \mathbb{F}_q^r,\quad \vec{s}^T := (s_1,\ldots,s_L)^T := M\cdot\vec{f}^T,\quad s_0 := \vec{1}\cdot\vec{f}^T,$$

$$\eta_0,\theta_i \xleftarrow{\cup} \mathbb{F}_q (i=1,\ldots,L),$$

$$k_0^* := (-s_0,0,1,\eta_0,0)_{\mathbb{B}_0^*},$$

$$\text{para } 1\le i\le L$$

$$\text{si } \rho(i)=(t,\vec{v}_i := (v_{i,1},\ldots,v_{i,n_t})\in \mathbb{F}_q{}^{n_t}),\quad \eta_{i,1},\ldots,\eta_{i,n_t} \xleftarrow{\cup} \mathbb{F}_q,$$

$$k_i^* := (\overbrace{s_i+\theta_i v_{i,1},\theta_i v_{i,2},\ldots,\theta_i v_{i,n_t},}^{n_t}\ \overbrace{0^{n_t},}^{n_t}\ \overbrace{\eta_{i,1},\ldots,\eta_{i,n_t},}^{n_t}\ \overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*}$$

$$\text{si } \rho(i)=\neg(t,\vec{v}_i := (v_{i,1},\ldots,v_{i,n_t})\in \mathbb{F}_q{}^{n_t}),\quad \eta_{i,1},\ldots,\eta_{i,n_t} \xleftarrow{\cup} \mathbb{F}_q,$$

$$k_i^* := (\overbrace{s_i(v_{i,1},\ldots,v_{i,n_t}),}^{n_t}\ \overbrace{0^{n_t},}^{n_t}\ \overbrace{\eta_{i,1},\ldots,\eta_{i,n_t},}^{n_t}\ \overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*}$$

$$\mathrm{sk}_{\mathbb{S}} := (\mathbb{S},k_0^*,k_1^*,\ldots,k_L^*).$$

devolver $\mathrm{sk}_{\mathbb{S}}$.

Se describirá la función y la operación del dispositivo de cifrado 200. El dispositivo de cifrado 200 está dotado con una parte de adquisición de parámetros públicos 210, una parte de entrada de información 220 (segunda parte de entrada de información), una parte de generación de datos cifrados 230 y una parte de transmisión de datos 240 (parte de salida de datos). La parte de generación de datos cifrados 230 está dotada con una parte de generación de números aleatorios 231 y una parte de generación de elementos criptográficos 232.

El proceso del algoritmo Enc se describirá con referencia a la Fig. 9.

(S301: Paso de adquisición de parámetros públicos)

Por ejemplo, con el dispositivo de comunicación, la parte de adquisición de parámetros públicos 210 adquiere los parámetros públicos pk generados por el dispositivo de generación de claves 100, a través de la red.

(S302: Paso de entrada de información)

Con el dispositivo de entrada, la parte de entrada de información 220 toma como entrada el mensaje m a ser transmitido al dispositivo de descifrado 300. Con el dispositivo de entrada, la parte de entrada de información 220 también toma como entrada el conjunto de atributos $\Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,nt} \in F_q{}^{nt})) \{1 \le t \le d\}$. Obsérvese que t no necesita ser todos los números enteros t que caen dentro del intervalo de $1 \le t \le d$, sino que puede ser uno o más de los números enteros t que caen dentro del intervalo de $1 \le t \le d$. También, por ejemplo, la información de atributo de un usuario descifrable se establece en el conjunto de atributos Γ.

(S303: Paso de Generación de Números Aleatorios)

Con el dispositivo de procesamiento, la parte de generación de números aleatorios 231 genera los números aleatorios δ, Φ_0, $\Phi_{t,1}$, ..., $\Phi_{t,nt}$, y ζ, como se indica en la Fórmula 139.

[Fórmula 139]

$$\delta, \phi_0, \phi_{t,1}, \ldots, \phi_{t,n_t}, \zeta \xleftarrow{U} \mathbb{F}_q \text{ de manera que } (t, \vec{x}_t) \in \Gamma$$

(S304: Paso de generación de vector c)

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 232 genera un elemento c_0 de los datos cifrados c como se indica en la Fórmula 140.

[Fórmula 140]

$$c_0 := (\delta, 0, \zeta, 0, \phi_0)_{\mathbb{B}_0}$$

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 232 también genera un elemento c_t de los datos cifrados c para cada número entero t de (t, $x^{\rightarrow}_t$) incluido en Γ, como se indica en la Fórmula 141.

[Fórmula 141]

$$c_t := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{\phi_{t,1}, \ldots, \phi_{t,n_t}}^{n_t})_{\mathbb{B}_t}$$

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 232 también genera un elemento c_{d+1} de los datos cifrados c, como se indica en la Fórmula 142.

[Fórmula 142]

$$c_{d+1} := g_T^{\zeta} m$$

(S305: Paso de transmisión de datos)

Por ejemplo, con el dispositivo de comunicación, la parte de transmisión de datos 240 transmite los datos cifrados c, constituidos por el conjunto de atributos Γ introducidos en (S302) y c_0, c_t, y c_{d+1} generados en (S304), como elementos al dispositivo de descifrado 300 a través de la red. Como cuestión de rutina, los datos cifrados c se pueden transmitir al dispositivo de descifrado 300 por otro método.

Más específicamente, desde (S301) hasta (S304), la parte de adquisición de parámetros públicos 210, la parte de entrada de información 220 y la parte de generación de datos cifrados 230 generan los datos cifrados c ejecutando el algoritmo Enc indicado en la Fórmula 143. En (S305), la parte de transmisión de datos 240 transmite los datos cifrados c generados al dispositivo de descifrado 300.

[Fórmula 143]

$$\mathrm{Enc}(\mathrm{pk}, m, \Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \mid 1 \le t \le d\}(x_{t,1} := 1)$$

$$\delta, \phi_0, \phi_{t,1}, \ldots, \phi_{t,n_t}, \zeta \xleftarrow{\mathrm{U}} \mathbb{F}_q \text{ de manera que } (t, \vec{x}_t) \in \Gamma,$$

$$c_0 := (\delta, 0, \zeta, 0, \phi_0)_{\mathbb{B}_0},$$

$$c_t := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{\phi_{t,1}, \ldots, \phi_{t,n_t}}^{n_t})_{\mathbb{B}_t} \text{ para } (t, \vec{x}_t) \in \Gamma,$$

$$c_{d+1} := g_T^{\zeta} m, \quad c := (\Gamma, c_0, \{c_t\}_{(t,\vec{x}_t)\in\Gamma}, c_{d+1}).$$

devolver c.

Se describirá la función y la operación del dispositivo de descifrado 300. El dispositivo de descifrado 300 está dotado con una parte de adquisición de clave de descifrado 310, una parte de recepción de datos 320 (parte de adquisición de datos), una parte de cálculo de programa de intervalo 330, una parte de cálculo de coeficiente complementario 340, una parte de operación de emparejamiento 350 y una parte de cálculo de información de texto plano 360.

El proceso del algoritmo Dec se describirá con referencia a la Fig. 10.

(S401: Paso de adquisición de clave de descifrado)

Por ejemplo, con el dispositivo de comunicación, la parte de adquisición de clave de descifrado 310 adquiere la clave de descifrado $sk_s := (S, k^*_0, k^*_1, \ldots, k^*_L)$ distribuida desde el dispositivo de generación de claves 100, a través de la red. La parte de adquisición de clave de descifrado 310 también adquiere los parámetros públicos pk generados por el dispositivo de generación de claves 100.

(S402: Paso de recepción de datos)

Por ejemplo, con el dispositivo de comunicación, la parte de recepción de datos 320 recibe los datos cifrados c transmitidos por el dispositivo de cifrado 200, a través de la red.

(S403: Paso de cálculo de programa de intervalo)

Con el dispositivo de procesamiento, la parte de cálculo de programa de intervalo 330 comprueba si la estructura de acceso S incluida en la clave de descifrado sk_s adquirida en (S401) acepta o no Γ incluido en los datos cifrados c recibidos en (S402). El método de comprobación de si la estructura de acceso S acepta o no Γ es el mismo que el descrito en “3. Concepto para implementar cifrado funcional”.

La parte de cálculo de programa de intervalo 330 avanza el proceso a (S404) si la estructura de acceso S acepta Γ (aceptar en S403). Si la estructura de acceso S rechaza Γ (rechazar en S403), la parte de cálculo de programa de intervalo 330 juzga que los datos cifrados c no se pueden descifrar con la clave de descifrado sk_s y finaliza el proceso.

(S404: Paso de cálculo de coeficiente complementario)

Con el dispositivo de procesamiento, la parte de cálculo de coeficiente complementario 340 calcula I y una constante (coeficiente complementario) $\{\alpha_i\}_{i \in I}$, con la cual se establece la Fórmula 144.

[Fórmula 144]

$$s_0 = \sum_{i \in I} \alpha_i s_i, \text{ y } I \subseteq \{i \in \{1, \ldots, L\} \mid [\rho(i) = (t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$
$$\vee [\rho(i) = \neg(t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\}.$$

(S405: Paso de operación de emparejamiento)

La parte de operación de emparejamiento 350 genera una clave de sesión $K = g_T{}^{\zeta}$ calculando la Fórmula 145 con el dispositivo de procesamiento.

[Fórmula 145]

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_t)} e(c_t, k_i^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_t)} e(c_t, k_i^*)^{\alpha_i/(\vec{v}_i \cdot \vec{x}_t)}$$

Calculando la Fórmula 145, se obtiene la clave $K = g_T^{\zeta}$, como se indica en la Fórmula 146.

[Fórmula 146]

$$e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_t)} e(c_t, k_i^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_t)} e(c_t, k_i^*)^{\alpha_i/(\vec{v}_i \cdot \vec{x}_t)}$$
$$= g_T^{-\delta s_0 + \zeta} \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_t)} g_T^{\delta \alpha_i s_i} \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_t)} g_T^{\delta \alpha_i s_i (\vec{v}_i \cdot \vec{x}_t)/(\vec{v}_i \cdot \vec{x}_t)}$$
$$= g_T^{\delta(-s_0 + \sum_{i \in I} \alpha_i s_i) + \zeta} = g_T^{\zeta}.$$

(S406: Paso de cálculo de información de texto plano)

La parte de cálculo de información de texto plano 360 genera un mensaje m' (= m) calculando $m' = c_{d+1}/K$ con el dispositivo de procesamiento. Obsérvese que c_{d+1} es $g_T^{\zeta}m$ como se indica en la Fórmula 142. Dado que K es g_T^{ζ}, el mensaje m se puede obtener calculando $m' = c_{d+1}/K$.

Más específicamente, desde (S401) hasta (S406), el dispositivo de descifrado 300 genera el mensaje m' (= m) ejecutando el algoritmo Dec indicado en la Fórmula 147.

[Fórmula 147]

$$\mathrm{Dec}(\mathrm{pk}, \mathrm{sk}_{\mathbb{S}} := (\mathbb{S}, k_0^*, k_1^*, \ldots, k_L^*),\ c := (\Gamma, c_0, \{c_t\}_{(t,\vec{x}_t) \in \Gamma}, c_{d+1}))$$

Si $\mathbb{S} := (M, \rho)$ acepta $\Gamma := \{(t, \vec{x}_t)\}$, entonces calcular I y $\{\alpha_i\}_{i \in I}$ de manera que

$$s_0 = \sum_{i \in I} \alpha_i s_i, \text{ e } I \subseteq \{i \in \{1, \ldots, L\} \mid [\rho(i) = (t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$
$$\vee [\rho(i) = \neg(t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\}.$$

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_t)} e(c_t, k_i^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_t)} e(c_t, k_i^*)^{\alpha_i/(\vec{v}_i \cdot \vec{x}_t)}$$

$$m' = c_{d+1} / K.$$

devolver m'.

Como se ha descrito anteriormente, el sistema de procesamiento criptográfico 10 implementa el esquema de cifrado (esquema de cifrado funcional) usando la estructura de acceso S construida usando el programa de intervalo, el predicado de producto interno y la compartición de secreto. Por lo tanto, el sistema de procesamiento criptográfico 10 implementa un esquema de cifrado que puede diseñar un control de acceso con un grado de libertad muy alto.

El esquema de cifrado implementado por el sistema de procesamiento criptográfico 10 es muy seguro. Como se ha descrito anteriormente, el esquema de cifrado funcional incluye, como su clase (la clase más limitada), cifrado basado en ID. Incluso cuando se compara con un cifrado basado en ID existente, práctico, el esquema de cifrado implementado por el sistema de procesamiento criptográfico 10 es más seguro en un cierto sentido de significado.

En la descripción anterior, en (3) de (S101), 5 se establece en N_0, y $4n_t$ (= $n_t + n_t + n_t + n_t$) se establece en N_t. Por lo tanto, las bases A_t, B_t y B^*_t son todas $(4n_t+5)$ dimensionales.

Alternativamente, $n_t + n_t + n_t + n_t$ se puede sustituir por $n_t + u_t + w_t + z_t$. Más específicamente, el primer n_t puede permanecer n_t, el segundo n_t se puede cambiar a u_t, el tercer n_t se puede cambiar a w_t, y el cuarto n_t se puede cambiar a z_t. Es decir, $n_t + u_t + w_t + z_t$ se puede establecer en N_t. Obsérvese que n_t, u_t, w_t y z_t pueden ser valores

diferentes donde n_t es un número entero de 1 o más, como se ha descrito anteriormente, y cada uno de u_t, w_t y z_t es un número entero de 0 o más.

En este caso, el algoritmo Setup indicado en la Fórmula 131 se reescribe como se indica en la Fórmula 148. Esto es, se cambian los sufijos de los vectores base de las bases B^$_t$ y B^*$_t$.

[Fórmula 148]

$$\mathsf{Setup}(1^\lambda, \vec{\mu} := (d; n_1, \ldots, n_d))$$

$$(\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d}) \xleftarrow{\mathsf{R}} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{\mu})$$

$$\hat{\mathbb{B}}_0 := (b_{0,1}, b_{0,3}, b_{0,5}), \quad \hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,n_t}, b_{t,n_t+u_t+w_t+1}, \ldots, b_{t,n_t+u_t+w_t+z_t})$$

$$\text{para } t = 1, \ldots, d,$$

$$\hat{\mathbb{B}}_0^* := (b_{0,1}^*, b_{0,3}^*, b_{0,4}^*), \quad \hat{\mathbb{B}}_t^* := (b_{t,1}^*, \ldots, b_{t,n_t}^*, b_{t,n_t+u_t+1}^*, \ldots, b_{t,n_t+u_t+w_t}^*)$$

$$\text{para } t = 1, \ldots, d,$$

$$\mathsf{sk} := \{\hat{\mathbb{B}}_t^*\}_{t=0,\ldots,d}, \quad \mathsf{pk} := (1^\lambda, \mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t\}_{t=0,\ldots,d}).$$

$$\text{devolver } \mathsf{sk}, \ \mathsf{pk}.$$

También, el algoritmo KeyGen indicado en la Fórmula 138 se reescribe como se indica en la Fórmula 149.

[Fórmula 149]

$$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \mathbb{S} := (M, \rho))$$

$$\vec{f} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r, \quad \vec{s}^T := (s_1, \ldots, s_L)^T := M \cdot \vec{f}^T, \quad s_0 := \vec{1} \cdot \vec{f}^T,$$

$$\eta_0, \theta_i \xleftarrow{\mathsf{U}} \mathbb{F}_q (i = 1, \ldots, L),$$

$$k_0^* := (-s_0, 0, 1, \eta_0, 0)_{\mathbb{B}_0^*},$$

$$\text{para } 1 \le i \le L$$

$$\text{si } \rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,w_t} \xleftarrow{\mathsf{U}} \mathbb{F}_q,$$

$$k_i^* := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,w_t}}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*}$$

$$\text{si } \rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,w_t} \xleftarrow{\mathsf{U}} \mathbb{F}_q,$$

$$k_i^* := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,w_t}}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*}$$

$$\mathsf{sk}_{\mathbb{S}} := (\mathbb{S}, k_0^*, k_1^*, \ldots, k_L^*).$$

$$\text{devolver } \mathsf{sk}_{\mathbb{S}}.$$

También, el algoritmo Enc indicado en la Fórmula 143 se reescribe como se indica en la Fórmula 150.

[Fórmula 150]

$$\mathrm{Enc}(\mathrm{pk}, m, \Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \,|\, 1 \le t \le d\}(x_{t,1} := 1)$$

$$\delta, \phi_0, \phi_{t,1}, \ldots, \phi_{t,z_t}, \zeta \xleftarrow{\cup} \mathbb{F}_q \text{ de manera que } (t, \vec{x}_t) \in \Gamma,$$

$$c_0 := (\delta, 0, \zeta, 0, \phi_0)_{\mathbb{B}_0}$$

$$c_t := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\phi_{t,1}, \ldots, \phi_{t,z_t}}^{z_t})_{\mathbb{B}_t} \text{ para } (t, \vec{x}_t) \in \Gamma,$$

$$c_{d+1} := g_T^{\zeta} m, \quad c := (\Gamma, c_0, \{c_t\}_{(t,\vec{x}_t)\in\Gamma}, c_{d+1}).$$

devolver c.

Obsérvese que el algoritmo Dec indicado en la Fórmula 147 permanece sin cambios.

También, no es necesario que N_0 sea 5, sino que puede ser un número entero de 2 o más. Cuando N_0 es 2, las bases B_0 y B^*_0 llegan a ser bidimensionales. En este caso, permitamos que $k^*_0 := (-s_0, 1)_{B^*0}$ en el algoritmo KeyGen y $c_0 := (\delta, \zeta)_{B0}$ en el algoritmo Enc.

En la descripción anterior, $k^*_0 := (-s_0, 0, 1, \eta_0, 0)_{B^*0}$ se establece en el algoritmo KeyGen. Alternativamente, empleando un valor predeterminado κ, se puede establecer $k^*_0 := (-s_0, 0, \kappa, \eta_0, 0)_{B^*0}$. En este caso, dado que $K := g^{\zeta\kappa}{}_T$ se calcula en el algoritmo Dec, $c_{d+1} := g^{\zeta\kappa}{}_T m$ se puede establecer en el algoritmo Enc.

En la descripción anterior, el valor de $v_{i,nt}$ no está particularmente limitado. No obstante, una limitación de $v_{i,nt} := 1$ se puede colocar desde el punto de vista de la prueba de seguridad.

Desde el punto de vista de la prueba de seguridad, $\rho(i)$ para cada número entero $i = 1, ..., L$ se puede limitar a una tupla positiva $(t, \vec{v})$ o una tupla negativa $\neg(t, \vec{v})$ para información de identificación t diferente.

En otras palabras, permitamos que una función $\rho\tilde{}$ sea un mapa de $\{1, ..., L\} \rightarrow \{1, ..., d\}$ con el cual $\rho\tilde{}(i) = t$ se establece cuando $\rho(i) = (t, \vec{v})$ o $\rho(i) = \neg(t, \vec{v})$. En este caso, $\rho\tilde{}$ se puede limitar a inyección. Obsérvese que $\rho(i)$ es $\rho(i)$ en la estructura de acceso $S := (M, \rho(i))$ descrita anteriormente.

Realización 2.

Esta realización describe un “esquema de cifrado funcional de política de texto cifrado (CP-FE)”. El cifrado funcional de política de texto cifrado descrito en esta realización está constituido en base al concepto descrito en la Realización 1.

En esta realización, inicialmente, se describirá la estructura básica del “esquema de cifrado funcional de política de texto cifrado”. Posteriormente, se describirá la estructura básica de un “sistema de procesamiento criptográfico 10” que implementa el “esquema de cifrado funcional de política de texto cifrado”. Luego, se describirán en detalle un “esquema de cifrado funcional de política de texto cifrado” y un “sistema de procesamiento criptográfico 10” según esta realización.

<Estructura básica de esquema de cifrado funcional de política de texto cifrado>

La estructura del esquema de cifrado funcional de política de texto cifrado se describirá brevemente. Obsérvese que la política de texto cifrado significa que la política está integrada en el texto cifrado, es decir, una estructura de acceso está integrada en el texto cifrado.

El esquema funcional de política de texto cifrado consta de cuatro algoritmos: Setup, KeyGen, Enc y Dec, de la misma manera que el esquema de cifrado funcional de política de clave.

(Setup)

Un algoritmo Setup es un algoritmo aleatorizado que toma como entrada un parámetro de seguridad λ, y un formato de atributo $\vec{\mu} := (d; n_1, ..., n_d)$, y emite los parámetros públicos pk y una clave maestra sk.

(KeyGen)

Un algoritmo KeyGen es un algoritmo aleatorizado que toma como entrada un conjunto de atributos $\Gamma := \{(t, \vec{x}_t) \mid \vec{x}_t \in \mathbb{F}_q{}^{nt}, 1 \le t \le d\}$, los parámetros públicos pk y la clave maestra sk, y emite una clave de descifrado sk_Γ.

(Enc)

Un algoritmo Enc es un algoritmo aleatorizado que toma como entrada un mensaje m, una estructura de acceso S := (M, ρ) y los parámetros públicos pk, y emite datos descifrados c.

(Dec)

Un algoritmo Dec es un algoritmo que toma como entrada los datos cifrados c cifrados bajo la estructura de acceso S, la clave de descifrado sk_Γ para el conjunto de atributos Γ y los parámetros públicos pk, y emite o bien el mensaje m o bien el símbolo distinguido $\perp$.

El esquema de cifrado funcional de política de texto cifrado debería tener la siguiente propiedad: para todas las estructuras de acceso S, el conjunto de atributos Γ, los parámetros públicos pk generados correctamente, la clave maestra sk y el texto cifrado c indicado en la Fórmula 151, mantiene que m = Dec(pk, sk_Γ, c) si la estructura de acceso S acepta el conjunto de atributos Γ. Si la estructura de acceso S rechaza el conjunto de atributos Γ, la probabilidad de m = Dec (pk, sk_Γ, c) es despreciable.

[Fórmula 151]

$$c \xleftarrow{R} \mathrm{Enc}(\mathrm{pk}, m, \mathbb{S})$$

<Sistema de procesamiento criptográfico 10>

Se describirá el sistema de procesamiento criptográfico 10 que ejecuta los algoritmos del esquema de cifrado funcional de política de texto cifrado descrito anteriormente.

La Fig. 11 es un diagrama de configuración del sistema de procesamiento criptográfico 10.

El sistema de procesamiento criptográfico 10 está dotado con un dispositivo de generación de claves 100, un dispositivo de cifrado 200 y un dispositivo de descifrado 300.

El dispositivo de generación de claves 100 ejecuta el algoritmo Setup tomando como entrada un parámetro de seguridad λ y el formato de atributo $\vec{\mu} := (d; n_1, ..., n_d)$, y genera los parámetros públicos pk y la clave maestra sk. El dispositivo de generación de claves 100 hace públicos los parámetros públicos pk generados. El dispositivo de generación de claves 100 también ejecuta el algoritmo KeyGen tomando como entrada el conjunto de atributos Γ, genera una clave de descifrado sk_Γ, y distribuye la clave de descifrado sk_Γ al dispositivo de descifrado 300 en secreto.

El dispositivo de cifrado 200 ejecuta el algoritmo Enc tomando como entrada un mensaje m, una estructura de acceso S y los parámetros públicos pk, y genera datos cifrados c. El dispositivo de cifrado 200 transfiere los datos cifrados c generados al dispositivo de descifrado 300.

El dispositivo de descifrado 300 ejecuta el algoritmo Dec tomando como entrada los parámetros públicos pk, la clave de descifrado sk_s y los datos cifrados c, y emite el mensaje m o el símbolo distinguido $\perp$.

<Esquema de cifrado funcional de política de texto cifrado y sistema de procesamiento de criptográfico 10 en detalle>

El esquema de cifrado funcional de política de texto cifrado, y la función y operación del sistema de procesamiento criptográfico 10 que ejecuta el esquema de cifrado funcional de política de texto cifrado se describirán con referencia a las Fig. 12 a 15.

La Fig. 12 es un diagrama de bloques de funciones que muestra la función del sistema de procesamiento criptográfico 10 que ejecuta el esquema de cifrado funcional de política de texto cifrado. El sistema de procesamiento criptográfico 10 está dotado con el dispositivo de generación de claves 100, el dispositivo de cifrado 200 y el dispositivo de descifrado 300, como se ha descrito anteriormente.

La Fig. 13 es un diagrama de flujo que muestra la operación del dispositivo de generación de claves 100 y el proceso del algoritmo Gen. El algoritmo Setup es el mismo que el del esquema de cifrado funcional de política de clave, y por consiguiente se omitirá su descripción detallada. La Fig. 14 es un diagrama de flujo de la operación del dispositivo de cifrado 200 y el proceso del algoritmo Enc. La Fig. 15 es un diagrama de flujo que muestra la operación del dispositivo de descifrado 300 y el proceso del algoritmo Dec.

Supongamos que $x_{t,1} := 1$ en la siguiente descripción.

Se describirá la función y la operación del dispositivo de generación de claves 100. El dispositivo de generación de claves 100 está dotado con una parte de generación de clave maestra 110, una parte de almacenamiento de clave maestra 120, una parte de entrada de información 130 (primera parte de entrada de información), una parte de

generación de clave de descifrado 140 y una parte de distribución de clave 150. La parte de generación de clave de descifrado 140 está dotada con una parte de generación de números aleatorios 143 y una parte de generación de elemento de clave 144.

Como se ha descrito anteriormente, el algoritmo Setup es similar al del esquema de cifrado funcional de política de clave, y es como se indica en la Fórmula 131.

El proceso del algoritmo KeyGen se describirá con referencia a la Fig. 13.

(S501: Paso de entrada de información)

Con un dispositivo de entrada, la parte de entrada de información 130 toma como entrada el conjunto de atributos $\Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n} \in F_q^{nt})) \mid 1 \leq t \leq d\}$. Obsérvese que, por ejemplo, la información de atributo del usuario de la clave de descifrado sk_Γ se establece en el conjunto de atributos Γ.

(S502: Paso de generación de números aleatorios)

Con un dispositivo de procesamiento, la parte de generación de números aleatorios 143 genera los números aleatorios δ, Φ_0, $\Phi_{t,1}$, ..., y $\Phi_{t,nt}$, como se indica en la Fórmula 152.

[Fórmula 152]

$$\delta, \phi_0, \phi_{t,1}, \ldots, \phi_{t,n_t} \xleftarrow{U} \mathbb{F}_q \text{ de manera que } (t, \vec{x}_t) \in \Gamma$$

(S503: Paso de generación de elemento de clave)

Con el dispositivo de procesamiento, la parte de generación de elemento de clave 144 genera un elemento k^*_0 de la clave de descifrado sk_Γ, como se indica en la Fórmula 153.

[Fórmula 153]

$$k_0^* := (\delta, 0, 1, \phi_0, 0)_{\mathbb{B}_0^*}$$

Con el dispositivo de procesamiento, la parte de generación de elemento de clave 144 también genera un elemento k^*_t de la clave de descifrado sk_Γ para cada número entero t de $(t, \vec{x}_t)$ incluido en Γ, como se indica en la Fórmula 154.

[Fórmula 154]

$$k_t^* := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\phi_{t,1}, \ldots, \phi_{t,n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t})_{\mathbb{B}_t^*}$$

(S504: Paso de distribución de clave)

Por ejemplo, con un dispositivo de comunicación, la parte de distribución de clave 150 distribuye la clave de descifrado sk_Γ constituida por el conjunto de atributos Γ introducido en (S501) y k^*_0 y k^*_t generados en (S503) como elementos, al dispositivo de descifrado 300 en secreto a través de la red. Como cuestión de rutina, la clave de descifrado sk_Γ se puede distribuir al dispositivo de descifrado 300 por otro método.

Más específicamente, desde (S501) hasta (S503), la parte de entrada de información 130 y la parte de generación de clave de descifrado 140 generan la clave de descifrado sk_Γ ejecutando el algoritmo KeyGen indicado en la Fórmula 155. Entonces, en (S504), la clave de descifrado sk_Γ generada por la parte de distribución de clave 150 se distribuye al dispositivo de descifrado 300.

[Fórmula 155]

$$\mathsf{KeyGen}(\mathsf{pk},\mathsf{sk},\Gamma := \{(t,\vec{x}_t := (x_{t,1},\ldots,x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \,|\, 1 \le t \le d\}(x_{t,n_t} := 1)$$

$$\delta,\phi_0,\phi_{t,1},\ldots,\phi_{t,n_t} \xleftarrow{\mathsf{U}} \mathbb{F}_q \text{ de manera que } (t,\vec{x}_t) \in \Gamma,$$

$$k_0^* := (\delta,0,1,\phi_0,0)_{\mathbb{B}_0^*},$$

$$k_t^* := (\overbrace{\delta(x_{t,1},\ldots,x_{t,n_t})}^{n_t}, \overbrace{0^{n_t},}^{n_t} \overbrace{\phi_{t,1},\ldots,\phi_{t,n_t}}^{n_t} \overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*} \text{ para } (t,\vec{x}_t) \in \Gamma,$$

$$\mathsf{sk}_\Gamma := (\Gamma,k_0^*,\{k_t^*\}_{(t,\vec{x}_t)\in\Gamma})$$

devolver sk_Γ.

Se describirán la función y la operación del dispositivo de cifrado 200. El dispositivo de cifrado 200 está dotado con una parte de adquisición de parámetros públicos 210, una parte de entrada de información 220 (segunda parte de entrada de información), una parte de generación de datos cifrados 230 y una parte de transmisión de datos 240 (parte de salida de datos). La parte de generación de datos cifrados 230 está dotada con una parte de generación de números aleatorios 231, una parte de generación de elementos criptográficos 232, una parte de generación de vector f 233 y una parte de generación de vector s 234.

El proceso del algoritmo Enc se describirá con referencia a la Fig. 14.

(S601: Paso de adquisición de parámetros públicos)

Por ejemplo, con el dispositivo de comunicación, la parte de adquisición de parámetros públicos 210 adquiere los parámetros públicos pk generados por el dispositivo de generación de claves 100, a través de la red.

(S602: Paso de entrada de información)

Con el dispositivo de entrada, la parte de entrada de información 220 toma como entrada la estructura de acceso S := (M, ρ). La estructura de acceso S se establece dependiendo de la condición del sistema que el usuario desea realizar. También, por ejemplo, la información de atributo del usuario que puede descifrar los datos se establece en ρ de la estructura de acceso S.

Con el dispositivo de entrada, la parte de entrada de información 220 también toma como entrada el mensaje m a ser transmitido al dispositivo de descifrado 300.

(S603: Paso de generación de vector f)

Con el dispositivo de procesamiento, la parte de generación de vector f 233 genera un vector $\vec{f}$ que tiene r partes de elementos, aleatoriamente como se indica en la Fórmula 156.

[Fórmula 156]

$$\vec{f} \xleftarrow{\mathsf{U}} \mathbb{F}_q^r$$

(S604: Paso de generación de vector s)

Con el dispositivo de procesamiento, la parte de generación de vector s 234 genera un vector $\vec{s}^T := (s_1, ..., s_L)^T$ en base a la matriz M (L filas x r columnas) incluida en la estructura de acceso S introducida en (S602) y el vector $\vec{f}$ generado en (S603), como se indica en Fórmula 157.

[Fórmula 157]

$$\vec{s}^T := (s_1,\ldots,s_L)^T := M \cdot \vec{f}^T$$

Con el dispositivo de procesamiento, la parte de generación de vector s 234 también genera un valor s_0 en base al vector $\vec{f}$ generado en (S603), como se indica en la Fórmula 158.

[Fórmula 158]

$$s_0 := \vec{1} \cdot \vec{f}^T$$

(S605: Paso de generación de números aleatorios)

Con el dispositivo de procesamiento, la parte de generación de números aleatorios 231 genera un número aleatorio η_0, un número aleatorio θ_i para cada número entero i = 1, ..., L y un número aleatorio ζ, como se indica en la Fórmula 159.

[Fórmula 159]

$$\eta_0, \theta_i, \zeta \xleftarrow{U} \mathbb{F}_q (i = 1, \ldots, L)$$

(S606: Paso de generación de elementos criptográficos)

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 232 genera un elemento c_0 de los datos cifrados c, como se indica en la Fórmula 160.

[Fórmula 160]

$$c_0 := (-s_0, 0, \zeta, 0, \eta_0)_{\mathbb{B}_0}$$

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 232 también genera un elemento c_i de los datos cifrados c para cada número entero i = 1, ..., L, como se indica en la Fórmula 161.

[Fórmula 161]

$$\text{si } \rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t}}^{n_t})_{\mathbb{B}_t},$$

$$\text{si } \rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t}}^{n_t})_{\mathbb{B}_t}$$

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 232 también genera un elemento c_{d+1} de los datos cifrados c, como se indica en la Fórmula 162.

[Fórmula 162]

$$c_{d+1} := g_T^{\zeta} m$$

(S607: Paso de transmisión de datos)

Por ejemplo, con el dispositivo de comunicación, la parte de transmisión de datos 240 transmite los datos cifrados c, constituidos por la estructura de acceso S introducida en (S602) y c_0, c_1, ..., c_L, y c_{d+1} generados en (S606), como elementos al dispositivo de descifrado 300 a través de la red Como cuestión de rutina, los datos cifrados c se pueden transmitir al dispositivo de descifrado 300 por otro método.

Más específicamente, desde (S601) hasta (S606), la parte de adquisición de parámetros públicos 210, la parte de entrada de información 220 y la parte de generación de datos cifrados 230 generan los datos cifrados c ejecutando el algoritmo Enc indicado en la Fórmula 163. En (S607), la parte de transmisión de datos 240 transmite los datos cifrados c generados al dispositivo de descifrado 300.

[Fórmula 163]

$$\mathrm{Enc}(\mathrm{pk}, m, \mathbb{S} := (M, \rho))$$

$$\vec{f} \xleftarrow{R} \mathbb{F}_q{}^r, \quad \vec{s}^T := (s_1, \ldots, s_L)^T := M \cdot \vec{f}^T, \quad s_0 := \vec{1} \cdot \vec{f}^T,$$

$$\eta_0, \theta_i, \zeta \xleftarrow{U} \mathbb{F}_q (i = 1, \ldots, L),$$

$$c_0 := (-s_0, 0, \zeta, 0, \eta_0)_{\mathbb{B}_0},$$

para $1 \le i \le L$

si $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t}}^{n_t})_{\mathbb{B}_t},$$

si $\rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$

$$c_i := (\overbrace{s_i (v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t}}^{n_t})_{\mathbb{B}_t},$$

$$c_{d+1} := g_T^{\zeta} m, \quad c := (\mathbb{S}, c_0, c_1, \ldots, c_L, c_{d+1}).$$

devolver c.

Se describirá la función y la operación del dispositivo de descifrado 300. El dispositivo de descifrado 300 está dotado con una parte de adquisición de clave de descifrado 310, una parte de recepción de datos 320 (parte de adquisición de datos), una parte de cálculo de programa de intervalo 330, una parte de cálculo de coeficiente complementario 340, una parte de operación de emparejamiento 350 y una parte de cálculo de información de texto plano 360.

El proceso del algoritmo Dec se describirá con referencia a la Fig. 15.

(S701: Paso de adquisición de clave de descifrado)

Por ejemplo, con el dispositivo de comunicación, la parte de adquisición de clave de descifrado 310 adquiere la clave de descifrado sk_Γ distribuida desde el dispositivo de generación de claves 100, a través de la red. La parte de adquisición de la clave de descifrado 310 también adquiere los parámetros públicos pk generados por el dispositivo de generación de claves 100.

(S702: Paso de recepción de datos)

Por ejemplo, con el dispositivo de comunicación, la parte de recepción de datos 320 recibe los datos cifrados c transmitidos por el dispositivo de cifrado 200, a través de la red.

(S703: Paso de cálculo de programa de intervalo)

Con el dispositivo de procesamiento, la parte de cálculo del programa de intervalo 330 comprueba si la estructura de acceso S incluida en los datos cifrados c adquiridos en (S702) acepta o no Γ incluido en la clave de descifrado sk_Γ recibida en (S701). El método de comprobación de si la estructura de acceso S acepta o no Γ es el mismo que el descrito en “3. Concepto para implementar cifrado funcional”.

La parte de cálculo de programa de intervalo 330 avanza el proceso a (S704) si la estructura de acceso S acepta Γ (aceptar en S703). Si la estructura de acceso S rechaza Γ (rechazar en S403), la parte de cálculo de programa de intervalo 330 juzga que los datos cifrados c no se pueden descifrar con la clave de descifrado sk_s, y finaliza el proceso.

(S704) hasta (S706) son los mismos que (S404) hasta (406) mostrados en la Fig. 9 de la Realización 1.

Más específicamente, desde (S701) hasta (S706), la parte de adquisición de parámetros públicos 210, la parte de entrada de información 220 y la parte de generación de datos cifrados 230 generan un mensaje m’ (= m) ejecutando el algoritmo Dec indicado en la Fórmula 164.

[Fórmula 164]

$$\mathsf{Dec}(\mathsf{pk},\mathsf{sk}_{\Gamma} := (\Gamma, k_0^*, \{k_t^*\}_{(t,\vec{x}_t)\in\Gamma}),\ c := (\mathbb{S}, c_0, c_1, \ldots, c_L, c_{d+1}))$$

Si $\mathbb{S} := (M, \rho)$ acepta $\Gamma := \{(t, \vec{x}_t)\}$,

entonces calcular I y $\{\alpha_i\}_{i\in I}$ de manera que

$$s_0 = \sum_{i\in I} \alpha_i s_i, \text{ e}$$

$$I \subseteq \{i \in \{1,\ldots,L\} \mid [\rho(i) = (t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$
$$\vee [\rho(i) = \neg(t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\}.$$

$$K := e(c_0, k_0^*) \cdot \prod_{i\in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i} \cdot \prod_{i\in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i/(\vec{v}_i\cdot\vec{x}_t)}$$

$$m' = c_{d+1} / K.$$

devolver m'.

El sistema de procesamiento criptográfico 10 según la Realización 2 implementa un esquema de cifrado que puede diseñar un control de acceso con un grado de libertad muy alto, de la misma manera que el sistema de procesamiento criptográfico 10 según la Realización 1. También, el sistema de procesamiento criptográfico 10 según la Realización 2 es muy seguro, como lo es el sistema de procesamiento criptográfico 10 según la Realización 1.

Como en la Realización 1, $N_t = n_t + n_t + n_t + n_t$ se puede sustituir por $n_t + u_t + w_t + z_t$. Más específicamente, el primer n_t puede permanecer n_t, el segundo n_t se puede cambiar a u_t, el tercer n_t se puede cambiar a w_t y el cuarto n_t se puede cambiar a z_t. Es decir, $n_t + u_t + w_t + z_t$ se pueden establecer en N_t. Obsérvese que n_t, u_t, w_t y z_t pueden ser valores diferentes donde n_t es un número entero de 1 o más, como se ha descrito anteriormente, y cada uno de u_t, w_t y z_t es un número entero de 0 o más.

En este caso, el algoritmo Setup se reescribe como se indica en la Fórmula 148, de la misma manera que en la Realización 1.

También, el algoritmo KeyGen indicado en la Fórmula 155 se reescribe como se indica en la Fórmula 165.

[Fórmula 165]

$$\mathsf{KeyGen}(\mathsf{pk},\mathsf{sk},\Gamma := \{(t,\vec{x}_t := (x_{t,1},\ldots,x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \mid 1 \le t \le d\}(x_{t,n_t} := 1)$$

$$\delta, \phi_0, \phi_{t,1}, \ldots, \phi_{t,w_t} \xleftarrow{\mathsf{U}} \mathbb{F}_q \text{ de manera que } (t,\vec{x}_t) \in \Gamma,$$

$$k_0 := (\delta, 0, 1, \phi_0, 0)_{\mathbb{B}_0^*},$$

$$k_t^* := (\overbrace{\delta(x_{t,1},\ldots,x_{t,n_t})}^{n_t}, \overbrace{0^{u_t},}^{u_t} \overbrace{\phi_{t,1},\ldots,\phi_{t,w_t}}^{w_t} \overbrace{0^{z_t},}^{z_t})_{\mathbb{B}_t^*} \text{ para } (t,\vec{x}_t) \in \Gamma,$$

$$\mathsf{sk}_{\Gamma} := (\Gamma, k_0^*, \{k_t^*\}_{(t,\vec{x}_t)\in\Gamma})$$

devolver sk_{Γ}.

También, el algoritmo Enc indicado en la Fórmula 163 se reescribe como se indica en la Fórmula 166.

[Fórmula 166]

$$\mathrm{Enc}(\mathrm{pk}, m, \mathbb{S} := (M, \rho))$$

$$\vec{f} \xleftarrow{R} \mathbb{F}_q{}^r, \quad \vec{s}^T := (s_1, \ldots, s_L)^T := M \cdot \vec{f}^T, \quad s_0 := \vec{1} \cdot \vec{f}^T,$$

$$\eta_0, \theta_i, \zeta \xleftarrow{U} \mathbb{F}_q \, (i = 1, \ldots, L),$$

$$c_0 := (-s_0, 0, \zeta, 0, \eta_0)_{\mathbb{B}_0},$$

para $1 \le i \le L$

si $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,z_t} \xleftarrow{U} \mathbb{F}_q,$

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t},}^{n_t} \overbrace{0^{u_t},}^{u_t} \overbrace{0^{w_t},}^{w_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,z_t},}^{z_t})_{\mathbb{B}_t},$$

si $\rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,z_t} \xleftarrow{U} \mathbb{F}_q,$

$$c_i := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t}),}^{n_t} \overbrace{0^{u_t},}^{u_t} \overbrace{0^{w_t},}^{w_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,z_t},}^{z_t})_{\mathbb{B}_t},$$

$$c_{d+1} := g_T^{\zeta} m, \quad c := (\mathbb{S}, c_0, c_1, \ldots, c_L, c_{d+1}).$$

devolver c.

Obsérvese que el algoritmo Dec indicado en la Fórmula 164 permanece sin cambios.

También, N_0 no necesita ser 5, sino que puede ser un número entero de 2 o más. Cuando N_0 es 2, las bases B_0 y B^*_0 llegan a ser bidimensionales. En este caso, $k^*_0 := (\delta, 1)_{B^*0}$ se puede establecer en el algoritmo KeyGen, y $c_0 := (-s_0, \zeta)_{B0}$ se puede establecer en el algoritmo Enc.

En la descripción anterior, $k^*_0 := (\delta, 0, 1, \Phi_0, 0)_{B^*0}$ se establece en el algoritmo KeyGen. Alternativamente, empleando un valor predeterminado κ, se puede establecer $k^*_0 := (\delta, 0, \kappa, \Phi_0, 0)_{B^*0}$. En este caso, dado que $K := g_T^{\zeta\kappa}$ se calcula en el algoritmo Dec, $c_{d+1} := g_T^{\zeta\kappa} m$ se puede establecer en el algoritmo Enc.

En la descripción anterior, el valor de $v_{i,nt}$ no está particularmente limitado. No obstante, el valor de $v_{i,nt}$ se puede limitar para satisfacer $v_{i,nt} := 1$ desde el punto de vista de probar la seguridad.

Desde el punto de vista de probar la seguridad, $\rho(i)$ para cada número entero $i = 1, \ldots, L$ se puede limitar a una tupla positiva $(t, \vec{v})$ o tupla negativa $\neg(t, \vec{v})$ para diferente información de identificación t.

En otras palabras, permitamos que una función $\rho^{\sim}$ sea un mapa de $\{1, \ldots, L\} \rightarrow \{1, \ldots, d\}$ con el cual $\rho^{\sim}(i) = t$ cuando se establece $\rho(i) = (t, \vec{v})$ o $\rho(i) = \neg(t, \vec{v})$. En este caso, $\rho^{\sim}$ se puede limitar a inyección. Obsérvese que $\rho(i)$ es $\rho(i)$ en la estructura de acceso $S := (M, \rho(i))$ descrita anteriormente.

La Realización 1 ha descrito el cifrado funcional de política de clave con el que la estructura de acceso S está contenida en la clave de descifrado sk_s y el conjunto de atributos Γ está contenido en los datos cifrados c. También, la Realización 2 ha descrito el cifrado funcional de política de texto cifrado con el que la estructura de acceso S está contenida en los datos cifrados c y el conjunto de atributos Γ está contenido en la clave de descifrado sk_Γ.

Alternativamente, se pueden preparar dos estructuras de acceso S_1 y S_2. Esto es, una estructura de acceso S_1 puede estar contenida en la clave de descifrado, y la otra estructura de acceso S_2 puede estar contenida en los datos cifrados c. Al mismo tiempo, un conjunto de atributos Γ_2 correspondiente a la estructura de acceso S_2 puede estar contenido en la clave de descifrado, y un conjunto de atributos Γ_1 correspondiente a la estructura de acceso S_1 puede estar contenido en los datos descifrados c. Los datos cifrados c pueden ser descifrables con la clave de descifrado si y sólo si la estructura de acceso S_1 acepta el conjunto de atributos Γ_1 y la estructura de acceso S_2 acepta el conjunto de atributos Γ_2.

Esto es, se puede emplear un esquema de cifrado en el que se combinan el cifrado funcional de política de clave y el cifrado funcional de política de texto cifrado.

Realización 3.

En esta realización, se describirá un esquema de firma al que se aplica el “esquema de cifrado funcional de política de texto cifrado” descrito en la Realización 2.

En esta realización, inicialmente, se describirá la estructura básica del “esquema de firma basado en el esquema de cifrado funcional de política de texto cifrado”. Posteriormente, se describirá la estructura básica de un “sistema de procesamiento de firmas 20” que implementa este “esquema de firma”. Entonces, se describirá en detalle un “esquema de firma” y un “sistema de procesamiento de firmas 20” según esta realización.

<Estructura básica de esquema de firma basado en el esquema de cifrado funcional de política de texto cifrado>

El esquema de firma basado en el esquema de cifrado funcional de política de texto cifrado consta de cuatro algoritmos: Setup, KeyGen, Sig y Ver.

(Setup)

Un algoritmo Setup es un algoritmo aleatorizado que toma como entrada un parámetro de seguridad λ y un formato de atributo $\vec{\mu} := (d; n_1, \ldots, n_d)$, y emite los parámetros públicos pk y una clave maestra sk.

(KeyGen)

Un algoritmo KeyGen es un algoritmo aleatorizado que toma como entrada un conjunto de atributos $\Gamma := \{(t, \vec{x}_t) \mid \vec{x}_t \in F_q^{nt}, 1 \leq t \leq d\}$, los parámetros públicos pk y la clave maestra sk, y emite una clave de firma sk_Γ.

(Sig)

Un algoritmo Sig es un algoritmo aleatorizado que toma como entrada un mensaje m, la clave de firma sk_Γ, una estructura de acceso $S := (M, \rho)$, y los parámetros públicos pk, y emite los datos de firma sig.

(Ver)

Un algoritmo Ver es un algoritmo que toma como entrada el mensaje m, la estructura de acceso $S := (M, \rho)$, los datos de firma sig y los parámetros públicos pk, y emite un valor “1” que representa el éxito de la verificación de la firma, o un valor “0” que representa el fallo de verificación de la falta de firma.

<Sistema de procesamiento de firmas 20>

Se describirá el sistema de procesamiento de firmas 20 que ejecuta los algoritmos del proceso de firma descrito anteriormente.

La Fig. 16 es un diagrama de configuración del sistema de procesamiento de firmas 20.

El sistema de procesamiento de firmas 20 está dotado con un dispositivo de generación de claves 100, un dispositivo de firma 400 y un dispositivo de verificación 500.

El dispositivo de generación de claves 100 ejecuta el algoritmo Setup tomando como entrada un parámetro de seguridad λ y un formato de atributo $\vec{\mu} := (d; n_1, \ldots, n_d)$, y genera los parámetros públicos pk y una clave maestra sk. El dispositivo de generación de claves 100 hace públicos los parámetros públicos pk generados. El dispositivo de generación de claves 100 también ejecuta el algoritmo KeyGen tomando como entrada un conjunto de atributos Γ, genera una clave de firma sk_Γ y distribuye la clave de firma sk_Γ al dispositivo de firma 400 en secreto.

El dispositivo de firma 400 ejecuta el algoritmo Sig tomando como entrada un mensaje m, una estructura de acceso S, los parámetros públicos pk y la clave de firma sk_Γ, y genera un vector de firma $\vec{s}^*$. El dispositivo de firma 400 transmite el vector de firma $\vec{s}^*$ generado, el mensaje m y la estructura de acceso S al dispositivo de verificación 500.

El dispositivo de verificación 500 ejecuta el algoritmo Ver tomando como entrada el vector de firma $\vec{s}^*$, el mensaje m, la estructura de acceso S y los parámetros públicos pk, y emite un valor “1” o “0”.

<Esquema de firma y sistema de procesamiento de firmas 20 en detalle>

El esquema de firma, y la función y la operación del sistema de procesamiento de firmas 20 se describirán con referencia a las Fig. 17 a 21.

La Fig. 17 es un diagrama de bloques de funciones que muestra la función del sistema de procesamiento de firmas 20. El sistema de procesamiento de firmas 20 está dotado con el dispositivo de generación de claves 100, el dispositivo de firma 400 y el dispositivo de verificación 500, como se ha descrito anteriormente.

Las Fig. 18 y 19 son diagramas de flujo que muestran la operación del dispositivo de generación de claves 100. La Fig. 18 es un diagrama de flujo que muestra el proceso del algoritmo Setup, y la Fig. 19 es un diagrama de flujo que muestra el proceso del algoritmo KeyGen. La Fig. 20 es un diagrama de flujo que muestra el funcionamiento del

dispositivo de firma 400 y el proceso del algoritmo Sig. La Fig. 21 es un diagrama de flujo que muestra la operación del dispositivo de verificación 500 y el proceso del algoritmo Ver.

Supongamos que $x_{t,1} := 1$ en la siguiente descripción.

Se describirá la función y la operación del dispositivo de generación de claves 100. El dispositivo de generación de claves 100 está dotado con una parte de generación de clave maestra 110, una parte de almacenamiento de clave maestra 120, una parte de entrada de información 130 (primera parte de entrada de información), una parte de generación de clave de firma 160 y una parte de distribución de clave 150. La parte de generación de clave de firma 160 está dotada con una parte de generación de números aleatorios 161, una parte de generación de elemento de clave 162 y una parte de generación de elemento confidencial 163.

El proceso del algoritmo Setup se describirá primero con referencia a la Fig. 18.

(S801: Paso de generación de base ortogonal regular)

La parte de generación de clave maestra 110 calcula la Fórmula 167 con un dispositivo de procesamiento para generar $param_{\mu\rightarrow}$, y las bases B_t y B^*_t para cada número entero t = 0, ..., d+2.

[Fórmula 167]

(1)

introducir $1^\lambda, \vec{\mu} := (d; n_1 \ldots, n_d)$

(2)

$$\text{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{R} \mathcal{G}_{\text{bpg}}(1^\lambda)$$

(3)

$$\psi \xleftarrow{U} \mathbb{F}_q^\times,\ n_0 := n_{d+1} := 1,\ n_{d+2} := 2,\ N_t := 4n_t \text{ para } t = 0, \ldots, d+2$$

Los procesos de (4) a (8) se ejecutan para cada t = 0, …, d+2.

(4)

$$\text{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\text{dpvs}}(1^\lambda, N_t, \text{param}_{\mathbb{G}})$$

(5)

$$X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{U} GL(N_t, \mathbb{F}_q)$$

(6)

$$(\nu_{t,i,j})_{i,j} := \psi \cdot (X_t^{\mathrm{T}})^{-1}$$

(7)

$$b_{t,i} := (\chi_{t,i,1}, \cdots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} a_{t,j},$$

$$\mathbb{B}_t := (b_{t,1}, \ldots, b_{t,N_t})$$

(8)

$$b_{t,i}^* := (\nu_{t,i,1}, \ldots, \nu_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \nu_{t,i,j} a_{t,j},$$

$$\mathbb{B}_t^* := (b_{t,1}^*, \ldots, b_{t,N_t}^*)$$

(9)

$$g_T := e(g,g)^{\psi}$$

$$\mathsf{param}_{\vec{\mu}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+2}, g_T)$$

Se hará una explicación sobre la Fórmula 167, pero solamente para las partes que son diferentes de las de la Fórmula 127 mostrada en (S101) de la Realización 1. Primero, para la Fórmula 167, en (3), 1 se establece en n_0 y n_{d+1}, y 2 se establece en n_{d+2}. Para el número entero t = 0, ..., d+2, 4 n_t se establece en N_t. Los procesos de (4) a (8) se repiten para cada número entero t = 0, ..., d+2.

Más específicamente, en (S801), la parte de generación de clave maestra 110 ejecuta el algoritmo G_{0b} indicado en la Fórmula 168, para generar $\mathsf{param}_{\mu\rightarrow}$, y las bases B_t y B^*_t para cada número entero t = 0, ..., d+2.

[Fórmula 168]

$$\mathcal{G}_{\mathsf{ob}}(1^{\lambda}, \vec{\mu} := (d; n_1 \ldots, n_d)) : \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{R} \mathcal{G}_{\mathsf{bpg}}(1^{\lambda}),$$

$$\psi \xleftarrow{U} \mathbb{F}_q^{\times},$$

$$n_0 := n_{d+1} := 1,\ n_{d+2} := 2,\ N_t := 4n_t \text{ para } t = 0, \ldots, d+2,$$

Para $t = 0, \ldots, d+2$,

$$\mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^{\lambda}, N_t, \mathsf{param}_{\mathbb{G}}),$$

$$X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{U} GL(N_t, \mathbb{F}_q), \quad (\nu_{t,i,j})_{i,j} := \psi \cdot (X_t^{\mathrm{T}})^{-1},$$

$$b_{t,i} := (\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N} \chi_{t,i,j} a_{t,j},\ \mathbb{B}_t := (b_{t,1}, \ldots, b_{t,N_t}),$$

$$b_{t,i}^* := (\nu_{t,i,1}, \ldots, \nu_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \nu_{t,i,j} a_{t,j},\ \mathbb{B}_t^* := (b_{t,1}^*, \ldots, b_{t,N_t}^*),$$

$$g_T := e(g,g)^{\psi},\ \mathsf{param}_{\vec{\mu}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d+2}, g_T)$$

$$\text{devolver } (\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+2}).$$

(S802: Paso de generación de subbase $B\hat{}_t$)

Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 genera una subbase $B\hat{}_0$ para cada número entero t = 1, ..., d+2, como se indica en la Fórmula 169.

[Fórmula 169]

$$\hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,n_t}, b_{t,3n_t+1}, \ldots, b_{t,4n_t})$$

(S803: Paso de generación de subbase $B\hat{}^*_t$)

Con el dispositivo de procesamiento, la parte de generación de clave maestra 110 genera una subbase $B\hat{}^*_t$ para cada número entero t = 1, ..., d+2, como se indica en la Fórmula 170.

[Formula 170]

$$\hat{\mathbb{B}}_t^* := (b_{t,1}^*, \ldots, b_{t,n_t}^*, b_{t,2n_t+1}^*, \ldots, b_{t,3n_t}^*)$$

(S804: Paso de almacenamiento de clave maestra)

La parte de generación de la clave maestra 110 trata la subbase $B^\wedge_t$ para cada número entero t = 0, ..., d+2, la subbase $B^{\wedge *}_0$ para cada número entero t = 1, ..., d+2, el vector base $b^*_{0,3}$, el parámetro de seguridad $\lambda(1^\lambda)$ introducido en (S101), y $param_{\mu\rightarrow}$ generado en (S101), como los parámetros públicos pk.

La parte de generación de clave maestra 110 trata el vector base $b^*_{0,1}$ como la clave maestra sk.

La parte de almacenamiento de clave maestra 120 almacena los parámetros públicos pk y la clave maestra sk, en el dispositivo de almacenamiento.

Más específicamente, desde (S801) hasta (S803), la parte de generación de clave maestra 110 genera los parámetros públicos pk y la clave maestra sk ejecutando el algoritmo Setup indicado en la Fórmula 171. Entonces, en (S804), la parte de almacenamiento de clave maestra 120 almacena los parámetros públicos pk generados y la clave maestra sk, en el dispositivo de almacenamiento.

Obsérvese que los parámetros públicos se hacen públicos a través de, por ejemplo, una red, de modo que el dispositivo de firma 400 y el dispositivo de verificación 500 puedan adquirirlos.

[Fórmula 171]

$$\begin{aligned}
&\mathsf{Setup}(1^\lambda, \vec{\mu} := (d; n_1, \ldots, n_d)) \\
&\quad (\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+2}) \xleftarrow{R} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{\mu}) \\
&\quad n_0 := n_{d+1} := 1,\ n_{d+2} := 2, \\
&\quad \hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,n_t}, b_{t,3n_t+1}, \ldots, b_{t,4n_t}) \ \text{para}\ t = 0, \ldots, d+2, \\
&\quad \hat{\mathbb{B}}_t^* := (b_{t,1}^*, \ldots, b_{t,n_t}^*, b_{t,2n_t+1}^*, \ldots, b_{t,3n_t}^*) \ \text{para}\ t = 1, \ldots, d+2, \\
&\quad \mathsf{sk} := b_{0,1}^*, \\
&\quad \mathsf{pk} := (1^\lambda, \mathsf{param}_{\vec{\mu}}, \{\hat{\mathbb{B}}_t\}_{t=0,\ldots,d+2}, \{\hat{\mathbb{B}}_t^*\}_{t=1,\ldots,d+2}, b_{0,3}^*). \\
&\quad \text{devolver}\ \mathsf{sk},\ \mathsf{pk}.
\end{aligned}$$

El proceso del algoritmo KeyGen se describirá con referencia a la Fig. 19.

(S901: Paso de entrada de información)

Con un dispositivo de entrada, la parte de entrada de información 130 toma como entrada un conjunto de atributos Γ := {(t, $x^\rightarrow_t$:= ($x_{t,1}$, …, $x_{t,nt} \in F_q^{nt}$)) | 1 ≤ t ≤ d}. Por ejemplo, la información de atributo del usuario de la clave de firma sk_Γ se establece en el conjunto de atributos Γ.

(S902: Paso de generación de números aleatorios)

Con el dispositivo de procesamiento, la parte de generación de números aleatorios 161 genera un número aleatorio δ y los números aleatorios Φ_0, $\Phi_{t,\tau}$, $\Phi_{d+2,\tau}$, y $\Phi_{d+3,\tau}$ para cada número entero t = 1, ..., d y cada número entero τ= 1, ..., n_t, como se indica en la Fórmula 172.

[Fórmula 172]

$$\begin{aligned}
&\delta, \phi_0, \phi_{t,\iota}, \phi_{d+2,\iota}, \phi_{d+3,\iota} \xleftarrow{U} \mathbb{F}_q \\
&\quad \text{para}\ t = 1, \ldots, d,\ \iota = 1, \ldots, n_t
\end{aligned}$$

(S903: paso de generación de elemento de clave)

Con el dispositivo de procesamiento, la parte de generación de elemento de clave 162 genera un elemento k^*_0 de la clave de firma sk_Γ, como se indica en la Fórmula 173.

[Fórmula 173]

$$k_0^* := (\delta, 0, \phi_0, 0)_{\mathbb{B}_0^*}$$

Con el dispositivo de procesamiento, la parte de generación de elemento de clave 162 también genera un elemento k^*_t de la clave de firma sk_Γ para cada número entero t de $(t, \vec{x}_t)$ incluido en Γ, como se indica en la Fórmula 174.

[Fórmula 174]

$$k_t^* := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\phi_{t,1}, \ldots, \phi_{t,n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t})_{\mathbb{B}_t^*}$$

También, con el dispositivo de procesamiento, la parte de generación de elemento de clave 162 genera los elementos k^*_{d+2} y k^*_{d+3} de la clave de firma sk_Γ, como se indica en la Fórmula 175.

[Fórmula 175]

$$k_{d+2}^* := (\delta(1,0), 0, 0, \phi_{d+2,1}, \phi_{d+2,2}, 0, 0)_{\mathbb{B}_{d+2}^*}$$

$$k_{d+3}^* := (\delta(0,1), 0, 0, \phi_{d+3,1}, \phi_{d+3,2}, 0, 0)_{\mathbb{B}_{d+2}^*}$$

(S904: Paso de distribución de clave)

Por ejemplo, con el dispositivo de comunicación, la parte de distribución de clave 150 distribuye la clave de firma sk_Γ constituida por, como elementos, el conjunto de atributos Γ introducido en (S901) y k^*_0, k^*_t, k^*_{d+2} y k^*_{d+3} generados en (S903), al dispositivo de firma 400 en secreto a través de la red. Como cuestión de rutina, la clave de descifrado sk_Γ se puede distribuir al dispositivo de firma 400 por otro método.

Más específicamente, desde (S901) hasta (S903), la parte de entrada de información 130 y la parte de generación de clave de firma 160 generan la clave de firma sk_Γ ejecutando el algoritmo KeyGen indicado en la Fórmula 176. Entonces, en (S904), la parte de distribución de clave 150 distribuye la clave de firma generada sk_Γ al dispositivo de firma 400.

[Fórmula 176]

$$\mathrm{KeyGen}(\mathrm{pk},\mathrm{sk},\Gamma := \{(t,\vec{x}_t := (x_{t,1},\ldots,x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \,|\, 1 \le t \le d\})(x_{t,1} := 1)$$

$$\delta,\phi_0,\phi_{t,\iota},\phi_{d+2,\iota},\phi_{d+3,\iota} \xleftarrow{\cup} \mathbb{F}_q$$

para $t = 1,\ldots,d,\ \iota = 1,\ldots,n_t,$

$$k_0^* := (\delta,0,\phi_0,0)_{\mathbb{B}_0^*},$$

$$k_t^* := (\overbrace{\delta(x_{t,1},\ldots,x_{t,n_t})}^{n_t},\ \overbrace{0^{n_t},}^{n_t}\ \overbrace{\phi_{t,1},\ldots,\phi_{t,n_t},}^{n_t}\ \overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*} \text{ para } (t,\vec{x}_t) \in \Gamma,$$

$$k_{d+2}^* := (\delta(1,0),0,0,\phi_{d+2,1},\phi_{d+2,2},0,0)_{\mathbb{B}_{d+2}^*},$$

$$k_{d+3}^* := (\delta(0,1),0,0,\phi_{d+3,1},\phi_{d+3,2},0,0)_{\mathbb{B}_{d+2}^*},$$

$$T := \{0,d+2,d+3\} \cup \{t \,|\, 1 \le t \le d,(t,\vec{x}_t) \in \Gamma\},$$

$$\mathrm{sk}_\Gamma := (\Gamma,\{k_t^*\}_{t\in T})$$

devolver sk_Γ.

Se describirá la función y la operación del dispositivo de firma 400. El dispositivo de firma 400 está dotado con una parte de adquisición de clave de firma 410, una parte de entrada de información 420 (segunda parte de entrada de información), una parte de cálculo de coeficiente complementario 430, una parte de generación de matriz 440, una parte de generación de firma 450 y una parte de transmisión de datos 460 (parte de salida de datos). La parte de generación de firma 450 está dotada con una parte de generación de números aleatorios 451 y una parte de generación de elemento de firma 452.

El proceso del algoritmo Sig se describirá con referencia a la Fig. 20.

(S1001: Paso de adquisición de clave de firma)

Por ejemplo, con el dispositivo de comunicación, la parte de adquisición de clave de firma 410 adquiere la clave de firma $\mathrm{sk}_\Gamma := (\Gamma, k^*_0, k^*_t, k^*_{d+2}, k^*_{d+3})$ distribuida por el dispositivo de generación de claves 100, a través de la red. La parte de adquisición de clave de firma 410 también adquiere los parámetros públicos pk generados por el dispositivo de generación de claves 100.

(S1002: Paso de entrada de información)

Con el dispositivo de entrada, la parte de entrada de información 420 toma como entrada la estructura de acceso S := (M, ρ). Obsérvese que como la estructura de acceso S a ser introducida, se acepta el conjunto de atributos Γ incluido en la clave de firma sk_Γ introducida en (S1001).

Con el dispositivo de procesamiento, la parte de entrada de información 220 toma como entrada el mensaje m al que ha de ser unida la firma.

(S1003: Paso de cálculo de coeficiente complementario)

Con el dispositivo de procesamiento, la parte de cálculo de coeficiente complementario 430 calcula I y una constante (coeficiente complementario) $\{\alpha_i\}_{i\in I}$, con la cual se establece la Fórmula 177.

[Fórmula 177]

$$\sum_{i\in I}\alpha_i M_i = \vec{1},\ \text{e}$$

$$I \subseteq \{i \in \{1,\ldots,L\} \mid [\rho(i) = (t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$

$$\vee\ [\rho(i) = \neg(t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\}$$

Obsérvese que M_i indica la fila de orden i de una matriz M.

(S1004: Paso de adición de fila)

Con el dispositivo de procesamiento, la parte de generación de matriz 440 genera un vector M_{L+1} a ser añadido a la matriz M, y una etiqueta ρ(L+1) del vector M_{L+1}, como se indica en la Fórmula 178.

[Fórmula 178]

$$M_{L+1} := \vec{1} \in \mathbb{F}_q{}^r$$

$$\rho(L+1) := \neg(d+1,\vec{v}_{L+1})$$

Permitamos que $\vec{v}_{L+1}$:= (1).

(S1005: Paso de generación de números aleatorios)

Con el dispositivo de procesamiento, la parte de generación de números aleatorios 451 genera un número aleatorio ξ, como se indica en la Fórmula 179.

[Fórmula 179]

$$\xi \xleftarrow{\text{U}} \mathbb{F}_q$$

(S1006: Paso de generación de elemento de firma)

Con el dispositivo de procesamiento, la parte de generación de elemento de firma 452 genera s^*_0 que es un elemento del vector de firma $\vec{s}^*$, como se indica en la Fórmula 180.

[Fórmula 180]

$$s_0^* := \xi k_0^* + r_0^*$$

Obsérvese que

$$r_0^* \xleftarrow{\text{U}} \text{intervalo}\left\langle b_{0,3}^* \right\rangle$$

Con el dispositivo de procesamiento, la parte de generación de elemento de firma 452 también genera s^*_i que es un elemento del vector de firma $\vec{s}^*$, para cada número entero i = 1, ..., L+1, como se indica en la Fórmula 180.

[Fórmula 181]

$$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot \left(\sum_{l=1}^{n_t} y_{i,l} \cdot b_{t,l}^*\right) + r_i^*$$

Obsérvese que r^*_i se define como sigue.

$$r_i^* \xleftarrow{\text{U}} \text{intervalo}\left\langle b_{t,2n_t+1}^*,\ldots,b_{t,3n_t}^* \right\rangle$$

También, γ_i e $\vec{y}_i := (y_{i,1}, \ldots, y_{i,n})$ se definen como sigue.

$$\text{si } i \in I \wedge \rho(i) = (t, \vec{v}_i),\ \gamma_i := \alpha_i,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\},$$

$$\text{si } i \in I \wedge \rho(i) = \neg(t, \vec{v}_i),\ \gamma_i := \alpha_i / (\vec{v}_i \cdot \vec{x}_t),\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\},$$

$$\text{si } i \notin I \wedge \rho(i) = (t, \vec{v}_i),\ \gamma_i := 0,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\},$$

$$\text{si } i \notin I \wedge \rho(i) = \neg(t, \vec{v}_i),\ \gamma_i := 0,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\}$$

También, β_i se define como sigue.

$$\{\beta_i\} \xleftarrow{U} \{\{\beta_i\} \mid \sum_{i=1}^{L+1} \beta_i M_i = \vec{0}\}$$

Con el dispositivo de procesamiento, la parte de generación de elemento de firma 452 también genera s^*_{L+2} que es un elemento del vector de firma $\vec{s}^*$, como se indica en la Fórmula 182.

[Fórmula 182]

$$s^*_{L+2} := \xi(k^*_{d+2} + m \cdot k^*_{d+3}) + r^*_{L+2}$$

Obsérvese que

$$r^*_{L+2} \xleftarrow{U} \text{intervalo}\left\langle b^*_{d+2,5}, b^*_{d+2,6} \right\rangle$$

(S1007: Paso de transmisión de datos)

Por ejemplo, con el dispositivo de comunicación, la parte de transmisión de datos 460 transmite los datos de firma sig, incluyendo el mensaje m y la estructura de acceso S := (M, ρ) que se introducen en (S1002) y $\vec{s}^*$ generado en (S1007), al dispositivo de verificación 500 a través de la red. Como cuestión de rutina, los datos de firma sig se pueden transmitir al dispositivo de verificación 500 por otro método.

Más específicamente, desde (S1001) hasta (S1006), la parte de adquisición de clave de firma 410, la parte de entrada de información 420, la parte de cálculo de coeficiente complementario 430, la parte de generación de matriz 440 y la parte de generación de firma 450 generan los datos de firma sig ejecutando el algoritmo Sig indicado en la Fórmula 183. En (S1007), la parte de transmisión de datos 460 transmite los datos de firma generados al dispositivo de verificación 500.

[Fórmula 183]

$\mathrm{Sig}(\mathrm{pk}, \mathrm{sk}_{\Gamma}, m, \mathbb{S} := (M, \rho))$

Si $\mathbb{S} := (M, \rho)$ acepta $\Gamma := \{(t, \vec{x}_t)\}$, entonces calcular I y $\{\alpha_i\}_{i \in I}$ de manera que

$$\sum_{i \in I} \alpha_i M_i = \vec{1}, \text{ e}$$

$$I \subseteq \{i \in \{1, \ldots, L\} \mid [\rho(i) = (t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$

$$\vee\ [\rho(i) = \neg(t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\},$$

$$M_{L+1} := \vec{1} \in \mathbb{F}_q{}^r,\ \vec{v}_{L+1} := (1),\ \rho(L+1) := \neg(d+1, \vec{v}_{L+1})$$

$$\xi \xleftarrow{U} \mathbb{F}_q,\ \{\beta_i\} \xleftarrow{U} \{(\beta_1, \ldots, \beta_{L+1}) \mid \textstyle\sum_{i=1}^{L+1} \beta_i M_i = \vec{0}\},$$

$$s_0^* := \xi k_0^* + r_0^*, \text{ donde } r_0^* \xleftarrow{U} \text{intervalo} \langle b_{0,3}^* \rangle,$$

para $1 \le i \le L+1$,

$$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot (\textstyle\sum_{t=1}^{n_t} y_{i,t} \cdot b_{t,t}^*) + r_i^*,$$

donde $r_i^* \xleftarrow{U} \text{intervalo} \langle b_{t,2n_t+1}^*, \ldots, b_{t,3n_t}^* \rangle$,

y $\gamma_i, \vec{y}_i := (y_{i,1}, \ldots, y_{i,n_t})$ se definen como

si $i \in I \wedge \rho(i) = (t, \vec{v}_i)$, $\gamma_i := \alpha_i$, $\vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\}$,

si $i \in I \wedge \rho(i) = \neg(t, \vec{v}_i)$, $\gamma_i := \alpha_i / (\vec{v}_i \cdot \vec{x}_t)$, $\vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\}$,

si $i \notin I \wedge \rho(i) = (t, \vec{v}_i)$, $\gamma_i := 0$, $\vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\}$,

si $i \notin I \wedge \rho(i) = \neg(t, \vec{v}_i)$, $\gamma_i := 0$, $\vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\}$,

$$s_{L+2}^* := \xi(k_{d+2}^* + m \cdot k_{d+3}^*) + r_{L+2}^*, \text{ donde } r_{L+2}^* \xleftarrow{U} \text{intervalo} \langle b_{d+2,5}^*, b_{d+2,6}^* \rangle$$

$$\vec{s}^* := (s_0^*, \ldots, s_{L+2}^*),\ \mathrm{sig} := \{m, \mathbb{S}, \vec{s}^*\}$$

devolver sig.

Se describirá la función y la operación del dispositivo de verificación 500. El dispositivo de verificación 500 está dotado con una parte de adquisición de parámetros públicos 510, una parte de recepción de datos 520 (parte de adquisición de datos), una parte de generación de datos cifrados 530 y una parte de operación de emparejamiento 540. La parte de generación de datos cifrados 530 está dotada con una parte de generación de números aleatorios 531, una parte de generación de elementos criptográficos 532, una parte de generación de vector f 533 y una parte de generación de vector s 534.

El proceso del algoritmo Ver se describirá con referencia a la Fig. 21.

(S1101: Paso de adquisición de parámetros públicos)

Por ejemplo, con el dispositivo de comunicación, la parte de adquisición de parámetros públicos 510 adquiere los parámetros públicos pk generados por el dispositivo de generación de claves 100, a través de la red.

(S1102: Paso de recepción de datos)

Por ejemplo, con el paso de comunicación, la parte de recepción de datos 320 recibe los datos de firma sig transmitidos por el dispositivo de firma 400, a través de la red.

(S1103: Paso de generación de vector f)

Con el dispositivo de procesamiento, la parte de generación de vector f 533 genera aleatoriamente un vector $\vec{f}$ que tiene r partes de elementos, como se indica en la Fórmula 184.

[Fórmula 184]

$$\vec{f} \xleftarrow{R} \mathbb{F}_q{}^r$$

(S1104: Paso de generación de vector s)

Con el dispositivo de procesamiento, la parte de generación de vector s 534 genera un vector $\vec{s}^T := (s_1, ..., s_L)^T$ en base a la matriz M (L filas x r columnas) incluida en la estructura de acceso S recibida en (S1102) y el vector $\vec{f}$ generado en (S1103), como se indica en la Fórmula 185.

[Fórmula 185]

$$\vec{s}^T := (s_1, \ldots, s_L)^T := M \cdot \vec{f}^T$$

Con el dispositivo de procesamiento, la parte de generación de vector s 534 también genera los valores s_0 y s_{L+1} en base al vector $\vec{f}$ generado en (S1103), como se indica en la Fórmula 186.

[Fórmula 186]

$$s_0 := s_{L+1} := \vec{1} \cdot \vec{f}^T$$

(S1105: Paso de generación de números aleatorios)

Con el dispositivo de procesamiento, la parte de generación de números aleatorios 531 genera los números aleatorios η_0, $\eta_{L+2,1}$, $\eta_{L+2,2}$, θ_i y s_{L+2} para cada número entero i = 1, ..., L+2, como se indica en la Fórmula 187.

[Fórmula 187]

$$\eta_0, \eta_{L+2,1}, \eta_{L+2,2}, \theta_i, s_{L+2} \xleftarrow{U} \mathbb{F}_q$$

(S1106: Paso de generación de elementos criptográficos)

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 532 genera un elemento c_0 de los datos cifrados c, como se indica en la Fórmula 188.

[Fórmula 188]

$$c_0 := (-s_0 - s_{L+2}, 0, 0, \eta_0)_{\mathbb{B}_0}$$

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 532 también genera un elemento c_i de los datos cifrados c para cada número entero i = 1, ..., L+1, como se indica en la Fórmula 189.

[Fórmula 189]

$$\text{si } \rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}),\ \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t},}^{n_t})_{\mathbb{B}_t},$$

$$\text{si } \rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}),\ \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t}}^{n_t},)\mathbb{B}_t,$$

Con el dispositivo de procesamiento, la parte de generación de elementos criptográficos 532 también genera un elemento c_{L+2} de los datos cifrados c, como se indica en la Fórmula 190.

[Fórmula 190]

$$c_{L+2} := (s_{L+2} - \theta_{L+2}m, \theta_{L+2}, 0, 0, 0, 0, \eta_{L+2,1}, \eta_{L+2,2})\mathbb{B}_{d+2}$$

(S1107: Paso de operación de emparejamiento)

Con el dispositivo de procesamiento, la parte de operación de emparejamiento 540 calcula la Fórmula 191.

[Fórmula 191]

$$\prod_{i=0}^{L+2} e(c_i, s_i^*)$$

Permitamos que $\rho(L+1) := \neg(d+1, \vec{v}_{L+1} := (1))$.

La parte de operación de emparejamiento 540 emite un valor “1” si el resultado del cálculo de la Fórmula 191 es un valor “1”; y un valor “0” de otro modo. Si el resultado del cálculo de la Fórmula 191 es un valor “1”, esto indica que la firma está verificada; de otro modo, la firma no está verificada.

Si los datos de firma sig son datos correctos, se obtiene un valor “1” calculando la Fórmula 191, como se indica por la Fórmula 192.

[Fórmula 192]

$$\begin{aligned}\prod_{i=0}^{L+2} e(c_i, s_i^*) &= e(c_0, s_0^*) \cdot \prod_{i \in I} e(c_i, k_i^*)^{\xi\gamma_i} \cdot \prod_{i=1}^{L+1} e(c_i, \prod_{t=1}^{n_t} (b_{t,t}^*)^{y_{i,t}})^{\beta_i} \cdot e(c_{L+2}, s_{L+2}^*) \\ &= g_T^{\xi\delta(-s_0 - s_{L+2})} \cdot \prod_{i \in I} g_T^{\xi\delta\alpha_i s_i} \cdot \prod_{i=1}^{L+1} g_T^{\beta_i s_i} \cdot g_T^{\xi\delta s_{L+2}} \\ &= g_T^{\xi\delta(-s_0 - s_{L+2})} \cdot g_T^{\xi\delta s_0} \cdot g_T^{\xi\delta s_{L+2}} = 1\end{aligned}$$

Más específicamente, desde (S1101) hasta (S1107), la parte de adquisición de parámetros públicos 510, la parte de recepción de datos 520, la parte de generación de datos cifrados 530 y la parte de operación de emparejamiento 540 verifican los datos de firma sig ejecutando el algoritmo Ver indicado en la Fórmula 193.

[Fórmula 193]

$$\mathrm{Ver}(\mathrm{pk}, m, \mathbb{S} := (M, \rho), \vec{s}^{\,*})$$

$$\vec{f} \xleftarrow{R} \mathbb{F}_q{}^r, \quad \vec{s}^{\,T} := (s_1, \ldots, s_L)^T := M \cdot \vec{f}^{\,T}, \quad s_0 := s_{L+1} := \vec{1} \cdot \vec{f}^{\,T},$$

$$\eta_0, \eta_{L+2,1}, \eta_{L+2,2}, \theta_i, s_{L+2} \xleftarrow{U} \mathbb{F}_q (i = 1, \ldots, L+2),$$

$$\rho(L+1) := \neg(d+1, \vec{v}_{L+1} := (1))$$

$$c_0 := (-s_0 - s_{L+2}, 0, 0, \eta_0)_{\mathbb{B}_0},$$

para $1 \le i \le L+1$

si $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t})$,

$$\eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t},}^{n_t})_{\mathbb{B}_t},$$

si $\rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t})$,

$$\eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t}),}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t},}^{n_t})_{\mathbb{B}_t},$$

$$c_{L+2} := (s_{L+2} - \theta_{L+2} m, \theta_{L+2}, 0, 0, 0, 0, \eta_{L+2,1}, \eta_{L+2,2})_{\mathbb{B}_{d+2}},$$

devolver 1 si $\prod_{i=0}^{L+2} e(c_i, s_i^*) = 1$, devolver 0 de otro modo.

Como se describió anteriormente, el sistema de procesamiento de firmas 20 implementa el esquema de firma usando la estructura de acceso S construida usando el programa de intervalo, el predicado de producto interno y la compartición de secreto.

En el algoritmo Sig, el elemento del vector de firma $s^{\rightarrow *}$ se genera usando los números aleatorios ξ y r, de modo que se aleatorizan las variables δ y Φ de los elementos k de la clave de firma sk_Γ que es información sobre el generador del elemento del vector de firma $s^{\rightarrow *}$. Si las se aleatorizan las variables δ y Φ de los elementos k de la clave de firma sk_Γ, se puede evitar que estas variables sean leídas desde $s^{\rightarrow *}$ que es un elemento de los datos de firma. Esto es, en base a qué clave de firma sk_Γ se generan los datos de firma, se puede evitar que se lean. Es decir, se puede aumentar la capacidad del enlace.

En la descripción anterior, en (S801), $4n_t$ (= $n_t + n_t + n_t + n_t$) se establece en N_t.

Alternativamente, $n_t + n_t + n_t + n_t$ se puede sustituir por $n_t + u_t + w_t + z_t$. Más específicamente, el primer n_t puede permanecer n_t, el segundo n_t se puede cambiar a u_t, el tercer n_t se puede cambiar a w_t, y el cuarto n_t se puede cambiar a z_t. Es decir, $n_t + u_t + w_t + z_t$ se puede establecer en N_t. Obsérvese que n_t, u_t y z_t pueden ser valores diferentes donde n_t es un número entero de 1 o más, como se ha descrito anteriormente, y cada uno de u_t, w_t y z_t es un número entero de 0 o más.

En este caso, el algoritmo Setup indicado en la Fórmula 171 se reescribe como se indica en la Fórmula 194. Esto es, se cambian los sufijos de los vectores base de las bases $B^_t$ y B^*_t.

[Fórmula 194]

$$\mathsf{Setup}(1^{\lambda}, \vec{\mu} := (d; n_1, \ldots, n_d))$$

$$(\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+2}) \xleftarrow{R} \mathcal{G}_{\mathsf{ob}}(1^{\lambda}, \vec{\mu})$$

$$\hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,n_t}, b_{t,n_t+u_t+w_t+1}, \ldots, b_{t,n_t+u_t+w_t+z_t}) \text{ para } t = 1, \ldots, d+2,$$

$$\hat{\mathbb{B}}_t^* := (b_{t,1}^*, \ldots, b_{t,n_t}^*, b_{t,n_t+u_t+1}^*, \ldots, b_{t,n_t+u_t+w_t}^*) \text{ para } t = 1, \ldots, d+2,$$

$$\hat{\mathbb{B}}_{d+1} := (b_{d+1,1}, b_{d+1,2}, b_{d+1,4}), \quad \hat{\mathbb{B}}_{d+1}^* := (b_{d+1,1}^*, b_{d+1,2}^*, b_{d+1,3}^*),$$

$$\mathsf{sk} := b_{0,1}^*, \quad \mathsf{pk} := (1^{\lambda}, \mathsf{param}_{\vec{\mu}}, \{\hat{\mathbb{B}}_t\}_{t=0,\ldots,d+2}, \{\hat{\mathbb{B}}_t^*\}_{t=1,\ldots,d+2}, b_{0,3}^*).$$

devolver sk, pk.

También, el algoritmo KeyGen indicado en la Fórmula 176 se reescribe como se indica en la Fórmula 195.

[Fórmula 195]

$$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \,|\, 1 \le t \le d\})(x_{t,1} := 1)$$

$$\delta, \phi_0, \phi_{t,\iota}, \phi_{d+2,\iota}, \phi_{d+3,\iota} \xleftarrow{U} \mathbb{F}_q$$

para $t = 1, \ldots, d$, $\iota = 1, \ldots, w_t$,

$$k_0^* := (\delta, 0, \phi_0, 0)_{\mathbb{B}_0^*},$$

$$k_t^* := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\phi_{t,1}, \ldots, \phi_{t,w_t}}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*} \text{ para } (t, \vec{x}_t) \in \Gamma,$$

$$k_{d+2}^* := (\delta(1,0), 0, 0, \phi_{d+2,1}, \phi_{d+2,2}, 0, 0)_{\mathbb{B}_{d+2}^*},$$

$$k_{d+3}^* := (\delta(0,1), 0, 0, \phi_{d+3,1}, \phi_{d+3,2}, 0, 0)_{\mathbb{B}_{d+2}^*},$$

$$T := \{0, d+2, d+3\} \cup \{t \,|\, 1 \le t \le d, (t, \vec{x}_t) \in \Gamma\},$$

$$\mathsf{sk}_{\Gamma} := (\Gamma, \{k_t^*\}_{t \in T})$$

devolver sk_{Γ}.

También, el algoritmo Sig indicado en la Fórmula 183 se reescribe como se indica en la Fórmula 196.

[Fórmula 196]

$$\mathsf{Sig}(\mathsf{pk}, \mathsf{sk}_{\Gamma}, m, \mathbb{S} := (M, \rho))$$

Si $\mathbb{S} := (M, \rho)$ acepta $\Gamma := \{(t, \vec{x}_t)\}$, entonces calcular I y $\{\alpha_i\}_{i \in I}$ de manera que

$\sum_{i \in I} \alpha_i M_i = \vec{1}$, e

$$I \subseteq \{i \in \{1,\ldots,L\} \mid [\rho(i) = (t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$
$$\vee\ [\rho(i) = \neg(t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\},$$

$$M_{L+1} := \vec{1} \in \mathbb{F}_q{}^r,\ \vec{v}_{L+1} := (1),\ \rho(L+1) := \neg(d+1,\vec{v}_{L+1})$$

$$\xi \xleftarrow{U} \mathbb{F}_q,\ \{\beta_i\} \xleftarrow{U} \{(\beta_1,\ldots,\beta_{L+1}) \mid \textstyle\sum_{i=1}^{L+1} \beta_i M_i = \vec{0}\},$$

$$s_0^* := \xi k_0^* + r_0^*,\ \text{donde}\ r_0^* \xleftarrow{U} \text{intervalo}\left\langle b_{0,3}^* \right\rangle,$$

para $1 \le i \le L+1$,

$$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot (\textstyle\sum_{t=1}^{n_t} y_{i,t} \cdot b_{t,t}^*) + r_i^*,$$

donde $r_i^* \xleftarrow{U} \text{intervalo}\left\langle b_{t,n_t+u_t+1}^*,\ldots,b_{t,n_t+u_t+w_t}^* \right\rangle$,

y $\gamma_i,\vec{y}_i := (y_{i,1},\ldots,y_{i,n_t})$ se definen como

si $i \in I \wedge \rho(i) = (t,\vec{v}_i),\ \gamma_i := \alpha_i,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\}$,

si $i \in I \wedge \rho(i) = \neg(t,\vec{v}_i),\ \gamma_i := \alpha_i / (\vec{v}_i \cdot \vec{x}_t),\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\}$,

si $i \notin I \wedge \rho(i) = (t,\vec{v}_i),\ \gamma_i := 0,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\}$,

si $i \notin I \wedge \rho(i) = \neg(t,\vec{v}_i),\ \gamma_i := 0,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\}$,

$$s_{L+2}^* := \xi(k_{d+2}^* + m \cdot k_{d+3}^*) + r_{L+2}^*,\ \text{donde}\ r_{L+2}^* \xleftarrow{U} \text{intervalo}\left\langle b_{d+2,5}^*, b_{d+2,6}^* \right\rangle$$

$$\vec{s}^* := (s_0^*,\ldots,s_{L+2}^*),\ \text{sig} := \{m,\mathbb{S},\vec{s}^*\}$$

devolver sig.

También, el algoritmo Ver indicado en la Fórmula 193 se reescribe como se indica en la Fórmula 197.

[Fórmula 197]

$$\text{Ver}(\text{pk},m,\mathbb{S} := (M,\rho),\vec{s}^*)$$

$$\vec{f} \xleftarrow{R} \mathbb{F}_q{}^r,\ \vec{s}^T := (s_1,\ldots,s_L)^T := M \cdot \vec{f}^T,\ s_0 := s_{L+1} := \vec{1} \cdot \vec{f}^T,$$

$$\eta_0,\eta_{L+1,1},\eta_{L+1,2},\theta_i,s_{L+2} \xleftarrow{U} \mathbb{F}_q\ (i = 1,\ldots,L+2),$$

$$c_0 := (-s_0 - s_{L+1},0,0,\eta_0)_{\mathbb{B}_0},$$

para $1 \le i \le L+1$

si $\rho(i) = (t,\vec{v}_i := (v_{i,1},\ldots,v_{i,n_t}) \in \mathbb{F}_q{}^{n_t})$,

$$\eta_{i,1},\ldots,\eta_{i,z_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,z_t}}^{z_t})_{\mathbb{B}_t},$$

si $\rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t})$,

$$\eta_{i,1}, \ldots, \eta_{i,z_t} \xleftarrow{U} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,z_t}}^{z_t})_{\mathbb{B}_t},$$

$$c_{L+1} := (s_{L+2} - \theta_{L+2} m, \theta_{L+2}, 0, 0, 0, 0, \eta_{L+2,1}, \eta_{L+2,2})_{\mathbb{B}_{d+2}},$$

devolver 1 si $\prod_{i=0}^{L+2} e(c_i, s_i^*) = 1$, devolver 0 de otro modo.

También, cada uno de N_0 y N_{d+1} no necesita ser 4, sino que puede ser un número entero de 1 o más. Cuando N_0 es 1, las bases B_0 y B^*_0 llegan a ser unidimensionales. En este caso, $k^*_{0,j} := (\delta)_{B^*0}$ se puede establecer en el algoritmo KeyGen y $c_0 := (-s_0\text{-}s_{L+2})_{B0}$ se puede establecer en el algoritmo Ver.

También, N_{d+2} no necesita ser 8, sino que puede ser un número entero de 2 o más. Cuando N_{d+2} es 2, las bases B_{d+2} y B^*_{d+2} llegan a ser bidimensionales. En este caso, $k^*_{d+2} := (\delta_j(1,0))_{B^*d+2}$ y $k^*_{d+3} := (\delta_j(0,1))_{B^*d+2}$ se pueden establecer en el algoritmo KeyGen, y $c_0 := (s_{L+2}\text{-}\theta_{L+2}m, \theta_{L+2})_{Bd+2}$ se puede establecer en el algoritmo Ver.

En la descripción anterior, el valor de $v_{i,nt}$ no está particularmente limitado. No obstante, una limitación de $v_{i,nt} := 1$ se puede colocar desde el punto de vista de la prueba de seguridad.

Desde el punto de vista de la prueba de seguridad, ρ(i) para cada número entero i = 1, ..., L puede ser una tupla positiva (t, $v^{\rightarrow}$) o una tupla negativa ¬(t, $v^{\rightarrow}$) para diferente información de identificación t.

En otras palabras, permitamos que una función $\rho^{\sim}$ sea un mapa de {1, ..., L} → {1, ..., d} con el cual $\rho^{\sim}(i) = t$ se establece cuando ρ(i) = (t, $v^{\rightarrow}$) o ρ(i) = ¬(t, $v^{\rightarrow}$). En este caso, $\rho^{\sim}$ puede ser inyección. Obsérvese que ρ(i) es ρ(i) en la estructura de acceso S := (M, ρ(i)) descrita anteriormente.

En la descripción anterior, en el paso de adición de filas (S1004) del algoritmo Sig, la primera fila del vector M_{L+1} se añade, como la fila de orden (L+1), a la matriz M. No obstante, se puede añadir cualquier número de una o más filas a la matriz M. En (S1004), las filas añadidas son $M_{L+1} := 1^{\rightarrow}$. No obstante, las filas a ser añadidas no están limitadas a $1^{\rightarrow}$ sino a otros vectores.

En la descripción anterior, la etiqueta ρ de la fila añadida de orden (L+1) fue ρ(L+1) := ¬(d+1, $v^{\rightarrow}{}_{L+1} := (1)$). No obstante, la etiqueta de la fila a ser añadida no se limita a esto, sino que es suficiente en la medida en que se establece el proceso.

Más específicamente, la etiqueta de la fila se puede establecer de manera que la información en las filas añadidas sea 0 cuando la operación de emparejamiento se realiza en el paso de operación de emparejamiento (S1107) del algoritmo Ver.

Si se añaden dos o más filas, el número de veces de repetición del proceso del algoritmo Sig o del algoritmo Ver necesita ser cambiado según el número de filas añadidas.

Con el fin de mejorar la seguridad del proceso de firma, el algoritmo de firma indicado en la Fórmula 171 se puede cambiar como en la Fórmula 198, y el algoritmo Sig indicado en la Fórmula 183 se puede cambiar como en la Fórmula 199.

[Fórmula 198]

$$\mathsf{Setup}(1^\lambda, \vec{\mu} := (d; n_1, \ldots, n_d))$$

$$(\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d+2}) \xleftarrow{R} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{\mu})$$

$n_0 := n_{d+1} := 1,\ n_{d+2} := 2,$

$\hat{\mathbb{B}}_t := (b_{t,1},\ldots,b_{t,n_t},b_{t,3n_t+1},\ldots,b_{t,4n_t})$ para $t = 0,\ldots,d+2,$

$\hat{\mathbb{B}}_t^* := (b_{t,1}^*,\ldots,b_{t,n_t}^*),\ \widetilde{\mathbb{B}}_t^* := (b_{t,2n_t+1}^*,\ldots,b_{t,3n_t}^*)$ para $t = 1,\ldots,d+2,$

$\sigma,\psi_{t,i,\iota} \xleftarrow{U} \mathbb{F}_q$ para $t = 1,\ldots,d+1,\ i = 1,\ldots,n_t,\ \iota = 1,\ldots,n_t,$

$p_{t,\iota}^* := (\overbrace{0^{\iota-1},\sigma,0^{n_t-\iota}}^{n_t},\overbrace{0^{n_t},}^{n_t}\overbrace{\psi_{t,1,\iota},\ldots,\psi_{t,n_t,\iota},}^{n_t}\overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*}$

para $t = 1,\ldots,d+1,\ \iota = 1,\ldots,n_t,$

$\mathsf{sk} := (b_{0,1}^*,\{\hat{\mathbb{B}}_t^*\}_{t=1,\ldots,d+2}$

$\mathsf{pk} := (1^\lambda,\mathsf{param}_{\vec{\mu}},\hat{\mathbb{B}}_0,b_{0,3},\{\hat{\mathbb{B}}_t,\widetilde{\mathbb{B}}_t^*\}_{t=1,\ldots,d+2},\{p_{t,\iota}^*\}_{t=1,\ldots,d+2;\iota=1,\ldots,n_t}).$

devolver sk, pk.

[Fórmula 199]

$\mathsf{Sig}(\mathsf{pk},\mathsf{sk}_\Gamma,m,\mathbb{S} := (M,\rho))$

Si $\mathbb{S} := (M,\rho)$ acepta $\Gamma := \{(t,\vec{x}_t)\}$, entonces calcular I y $\{\alpha_i\}_{i\in I}$ de manera que

$\sum_{i\in I}\alpha_i M_i = \vec{1},$ e

$I \subseteq \{i \in \{1,\ldots,L\} \mid [\rho(i) = (t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$

$\vee\ [\rho(i) = \neg(t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\},$

$M_{L+1} := \vec{1} \in \mathbb{F}_q{}^r,\ \vec{v}_{L+1} := (1),\ \rho(L+1) := \neg(d+1,\vec{v}_{L+1})$

$\xi \xleftarrow{U} \mathbb{F}_q,\ \{\beta_i\} \xleftarrow{U} \{(\beta_1,\ldots,\beta_{L+1}) \mid \sum_{i=1}^{L+1}\beta_i M_i = \vec{0}\},$

$s_0^* := \xi k_0^* + r_0^*,$ donde $r_0^* \xleftarrow{U}$ intervalo $\langle b_{0,3}^* \rangle,$

para $1 \le i \le L+1,$

$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot (\sum_{\iota=1}^{n_t} y_{i,\iota} \cdot p_{t,\iota}^*) + r_i^*,$

donde $r_i^* \xleftarrow{U}$ intervalo $\langle b_{t,2n_t+1}^*,\ldots,b_{t,3n_t}^* \rangle,$

y $\gamma_i,\vec{y}_i := (y_{i,1},\ldots,y_{i,n_t})$ se definen como

si $i \in I \wedge \rho(i) = (t,\vec{v}_i),\ \gamma_i := \alpha_i,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\},$

si $i \in I \wedge \rho(i) = \neg(t,\vec{v}_i),\ \gamma_i := \alpha_i / (\vec{v}_i \cdot \vec{x}_t),\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\},$

si $i \notin I \wedge \rho(i) = (t,\vec{v}_i),\ \gamma_i := 0,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 0 \wedge y_{i,1} = 1\},$

$$\text{si } i \notin I \wedge \rho(i) = \neg(t, \vec{v}_i),\ \gamma_i := 0,\ \vec{y}_i \xleftarrow{U} \{\vec{y}_i \mid \vec{y}_i \cdot \vec{v}_i = 1\},$$

$$s^*_{L+2} := \xi(k^*_{d+2} + m \cdot k^*_{d+3}) + r^*_{L+2},\ \text{donde } r^*_{L+2} \xleftarrow{U} \text{intervalo} \left\langle b^*_{d+2,5}, b^*_{d+2,6} \right\rangle$$

$$\vec{s}^* := (s^*_0, \ldots, s^*_{L+2}),\ \text{sig} := \{m, \mathbb{S}, \vec{s}^*\}$$

devolver sig.

Más específicamente, el algoritmo Setup se cambia en que se genera el vector confidencial $p^*_{t,\tau}$ en el que se establece un valor aleatorio y el algoritmo Sig se cambia en que el elemento s^*_i del vector de firma $s^{\rightarrow *}$ se genera usando el vector confidencial $p^*_{t,\tau}$. También, el algoritmo Setup se cambia en que la base $B\hat{}^*_t$ está incluida en la clave maestra sk en lugar de en los parámetros públicos pk, y se mantiene en secreto. Esto mejora la seguridad del proceso de firma.

Incluso cuando el algoritmo Setup y el algoritmo Sig se cambian de esta manera, si los datos de firma sig son datos correctos, se puede obtener un valor "1" realizando la operación de emparejamiento en el algoritmo, como se indica en la Fórmula 199.

[Fórmula 200]

$$\prod_{i=0}^{L+2} e(c_i, s^*_i) = e(c_0, s^*_0) \cdot \prod_{i \in I} e(c_i, k^*_i)^{\xi\gamma_i} \cdot \prod_{i=1}^{L+1} e(c_i, \prod_{t=1}^{n_t} (p^*_{t,t})^{y_{i,t}})^{\beta_i} \cdot e(c_{L+2}, s^*_{L+2})$$

$$= g_T^{\xi\delta(-s_0 - s_{L+2})} \cdot \prod_{i \in I} g_T^{\xi\delta\alpha_i s_i} \cdot \prod_{i=1}^{L+1} g_T^{\beta_i s_i} \cdot g_T^{\xi\delta s_{L+2}}$$

$$= g_T^{\xi\delta(-s_0 - s_{L+2})} \cdot g_T^{\xi\delta s_0} \cdot g_T^{\xi\delta s_{L+2}} = 1$$

En el algoritmo Ver, $c_{L+2} := (s_0 - \theta_{L+2}m, \theta_{L+2}, 0, 0, 0, 0, \eta_{L+2,t}, \eta_{L+2,2})$ se puede establecer en el paso de generación de elementos criptográficos (S1106), en lugar de generar c_0. Entonces, en el paso de operación de emparejamiento (S1107), se puede calcular $\Pi_{i=1}^{L+2} e(c_i, s^*_i)$.

Esto es, en la descripción anterior, c_0 y c_{L+2} están vinculados por un número aleatorio s_{L+2}, y s_{L+2} se cancela entre c_0 y c_{L+2} cuando se ejecuta la operación de emparejamiento. Alternativamente, s_{L+2} no necesita ser usado por adelantado, de modo que se simplifica el proceso.

En este caso, k^*_0 no necesita ser generado en el algoritmo KeyGen. Del mismo modo, s^*_0 no necesita ser generado en el algoritmo Sig.

Realización 4.

En la Realización 1, desde el punto de vista de la prueba de seguridad, ρ(i) para cada número entero i = 1, ..., L se describió como una tupla positiva (t, $v^{\rightarrow}$) o una tupla negativa ¬(t, $v^{\rightarrow}$) para diferente información de identificación t. Es decir, $\rho^{\sim}$ puede ser inyección. No obstante, $\rho^{\sim}$ no necesita ser inyección.

En este caso, desde el punto de vista de la prueba de seguridad, el algoritmo Setup, el algoritmo KeyGen y el algoritmo Enc del cifrado funcional de Política de Clave descrito en la Realización 1 se pueden modificar como sigue. Solamente se explicarán las partes modificadas de los algoritmos respectivos del cifrado funcional de Política de Clave descrito en la Realización 1.

En esta realización, permitamos que Φ sea un valor indicado en la Fórmula 201.

[Fórmula 201]

$$\varphi \geq \max_{t=1}^{d} \#\{i \mid \tilde{\rho}(i) = t\}$$

El algoritmo Setup ejecuta Setup(1^{λ}, $\mu'^{\rightarrow}$:= (d; n_1', ..., n_d')) cuando se introduce (1^{λ}, $\mu^{\rightarrow}$:= (d; n_1, ..., n_d). Obsérvese que n_t' := n_t + Φ para cada número entero t = 1, ..., d.

Con respecto al algoritmo KeyGen, el método de generación de k^*_i se modifica como se indica en la Fórmula 202.

[Fórmula 202]

$$\text{si } \rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n'_t} \xleftarrow{U} \mathbb{F}_q$$

$$k_i^* := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}, 0^{\varphi}}^{n'_t}, \overbrace{0^{n'_t}}^{n'_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n'_t}}^{n'_t}, \overbrace{0^{n'_t}}^{n'_t})_{\mathbb{B}_t^*}$$

$$\text{si } \rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n'_t} \xleftarrow{U} \mathbb{F}_q,$$

$$k_i^* := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t}), 0^{\varphi}}^{n'_t}, \overbrace{0^{n'_t}}^{n'_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n'_t}}^{n'_t}, \overbrace{0^{n'_t}}^{n'_t})_{\mathbb{B}_t^*}$$

Con respecto al algoritmo Enc, el método de generación de c_t se modifica como se indica en la Fórmula 203.

[Fórmula 203]

$$c_t := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t}), 0^{\varphi}}^{n'_t}, \overbrace{0^{n'_t}}^{n'_t}, \overbrace{0^{n'_t}}^{n'_t}, \overbrace{\phi_{t,1}, \ldots, \phi_{t,n'_t}}^{n'_t})_{\mathbb{B}_t}$$

La modificación del proceso para un caso donde $\rho^{\sim}$ no es inyección se ha descrito anteriormente solamente con respecto al cifrado funcional de Política de Clave descrito en la Realización 1. El proceso criptográfico y el proceso de firma descritos en otras realizaciones se pueden modificar en base al mismo concepto.

Realización 5.

En las realizaciones anteriores, se ha descrito el método de implementación del proceso criptográfico y del proceso de firma en los espacios vectoriales duales. En la Realización 5, se describirá un método de implementación de un proceso criptográfico y un proceso de firma en grupos aditivos duales.

Más específicamente, en las realizaciones anteriores, el proceso criptográfico se implementa en el grupo cíclico de orden primo q. Cuando un anillo R se expresa usando un número compuesto M como se indica en la Fórmula 204, el proceso criptográfico descrito en las realizaciones anteriores también se puede aplicar a un grupo aditivo que tiene el anillo R como coeficiente.

[Fórmula 204]

$$\mathbb{R} := \mathbb{Z}/M\mathbb{Z}$$

donde

$\mathbb{Z}$: un número entero; y

M: un número compuesto

Por ejemplo, cuando se implementa el cifrado funcional de Política de Clave descrito en la Realización 1 en un grupo aditivo que tiene un anillo R como coeficiente, entonces resultan las Fórmulas 205 a 208.

[Fórmula 205]

$\mathsf{Setup}(1^{\lambda}, \vec{\mu} := (d; n_1, \ldots, n_d))$

$(\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d}) \xleftarrow{R} \mathcal{G}_{\mathsf{ob}}(1^{\lambda}, \vec{\mu})$

$\hat{\mathbb{B}}_0 := (b_{0,1}, b_{0,3}, b_{0,5}), \quad \hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,n_t}, b_{t,3n_t+1}, \ldots, b_{t,4n_t})$

para $t = 1, \ldots, d,$

$\hat{\mathbb{B}}_0^* := (b_{0,1}^*, b_{0,3}^*, b_{0,4}^*), \quad \hat{\mathbb{B}}_t^* := (b_{t,1}^*, \ldots, b_{t,n_t}^*, b_{t,2n_t+1}^*, \ldots, b_{t,3n_t}^*)$

para $t = 1, \ldots, d,$

$\mathsf{sk} := \{\hat{\mathbb{B}}_t^*\}_{t=0,\ldots,d}, \quad \mathsf{pk} := (1^{\lambda}, \mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t\}_{t=0,\ldots,d}).$

devolver sk, pk.

Obsérvese que

$\mathcal{G}_{\mathsf{ob}}(1^{\lambda}, \vec{\mu} := (d; n_1 \ldots, n_d)) : \mathsf{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{R} \mathcal{G}_{\mathsf{bpg}}(1^{\lambda}),$

$\psi \xleftarrow{U} \mathbb{R},$

$N_0 := 5, \quad N_t := 4n_t$ para $t = 1, \ldots, d,$

Para $t = 0, \ldots, d, \quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^{\lambda}, N_t, \mathsf{param}_{\mathbb{G}}),$

$X_t := (\chi_{t,i,j})_{i,j} \xleftarrow{U} GL(N_t, \mathbb{R}), \quad (\upsilon_{t,i,j})_{i,j} := \psi \cdot (X_t^{\mathrm{T}})^{-1},$

$b_{t,i} := (\chi_{t,i,1}, \ldots, \chi_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} a_{t,j}, \quad \mathbb{B}_t := (b_{t,1}, \ldots, b_{t,N_t}),$

$b_{t,i}^* := (\upsilon_{t,i,1}, \ldots, \upsilon_{t,i,N_t})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \upsilon_{t,i,j} a_{t,j}, \quad \mathbb{B}_t^* := (b_{t,1}^*, \ldots, b_{t,N_t}^*),$

$g_T := e(G,G)^{\psi}, \quad \mathsf{param}_{\vec{\mu}} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,\ldots,d}, g_T)$

devolver $(\mathsf{param}_{\vec{\mu}}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,\ldots,d}).$

[Fórmula 206]

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \mathbb{S} := (M, \rho))$

$\vec{f} \xleftarrow{U} \mathbb{R}^r, \quad \vec{s}^T := (s_1, \ldots, s_L)^T := M \cdot \vec{f}^T, \quad s_0 := \vec{1} \cdot \vec{f}^T,$

$\eta_0, \theta_i \xleftarrow{U} \mathbb{R} (i = 1, \ldots, L),$

$k_0^* := (-s_0, 0, 1, \eta_0, 0)_{\mathbb{B}_0^*},$

para $1 \le i \le L$

si $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{R}^{n_t}), \quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{R},$

$$k_i^* := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t} \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t},}^{n_t} \overbrace{0^{n_t},}^{n_t})_{\mathbb{B}_t^*}$$

si $\rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{R}^{n_t}),\quad \eta_{i,1}, \ldots, \eta_{i,n_t} \xleftarrow{U} \mathbb{R},$

$$k_i^* := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t})_{\mathbb{B}_t^*}$$

$\mathsf{sk}_{\mathbb{S}} := (\mathbb{S}, k_0^*, k_1^*, \ldots, k_L^*).$

devolver $\mathsf{sk}_{\mathbb{S}}$.

[Fórmula 207]

$$\mathsf{Enc}(\mathsf{pk}, m, \Gamma := \{(t, \vec{x}_t := (x_{t,1}, \ldots, x_{t,n_t}) \in \mathbb{F}_q{}^{n_t}) \mid 1 \le t \le d\}(x_{t,1} := 1)$$

$\delta, \phi_0, \phi_{t,1}, \ldots, \phi_{t,n_t}, \zeta \xleftarrow{U} \mathbb{R}$ de manera que $(t, \vec{x}_t) \in \Gamma$,

$c_0 := (\delta, 0, \zeta, 0, \phi_0)_{\mathbb{B}_0}$,

$$c_t := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{0^{n_t}}^{n_t}, \overbrace{\phi_{t,1}, \ldots, \phi_{t,n_t}}^{n_t})_{\mathbb{B}_t} \text{ para } (t, \vec{x}_t) \in \Gamma,$$

$c_{d+1} := g_T^{\zeta} m, \quad c := (\Gamma, c_0, \{c_t\}_{(t,\vec{x}_t)\in\Gamma}, c_{d+1}).$

devolver c.

[Fórmula 208]

$$\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_{\mathbb{S}} := (\mathbb{S}, k_0^*, k_1^*, \ldots, k_L^*),\ c := (\Gamma, c_0, \{c_t\}_{(t,\vec{x}_t)\in\Gamma}, c_{d+1}))$$

Si $\mathbb{S} := (M, \rho)$ acepta $\Gamma := \{(t, \vec{x}_t)\}$, entonces calcular I y $\{\alpha_i\}_{i\in I}$ de manera que

$$s_0 = \sum_{i\in I} \alpha_i s_i, \text{ e } I \subseteq \{i \in \{1, \ldots, L\} \mid [\rho(i) = (t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$

$$\vee [\rho(i) = \neg(t, \vec{v}_i) \wedge (t, \vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\}.$$

$$K := e(c_0, k_0^*) \cdot \prod_{i\in I \wedge \rho(i) = (t,\vec{v}_i)} e(c_t, k_i^*)^{\alpha_i} \cdot \prod_{i\in I \wedge \rho(i) = \neg(t,\vec{v}_i)} e(c_t, k_i^*)^{\alpha_i/(\vec{v}_i \cdot \vec{x}_t)}$$

$m' = c_{d+1} / K.$

devolver m'.

El método de implementación del proceso criptográfico y del proceso de firma en un grupo aditivo que tiene un anillo R como coeficiente se ha indicado solamente para el cifrado funcional de Política de Clave descrito en la Realización 1. Como principio, si el proceso descrito como un campo F_q en las realizaciones anteriores se sustituye por un anillo R, también se pueden implementar otros procesos criptográficos y procesos de firma descritos en las realizaciones anteriores en un grupo aditivo que tiene un anillo R como coeficiente.

El algoritmo Setup en las realizaciones anteriores se puede ejecutar solamente una vez en la configuración del sistema de procesamiento criptográfico 10 o del sistema de procesamiento de firmas 20, y no necesita ser ejecutado cada vez que ha de ser generada una clave de descifrado. En la descripción anterior, el algoritmo Setup y el algoritmo KeyGen se ejecutan por el dispositivo de generación de claves 100. Alternativamente, el algoritmo Setup y el algoritmo KeyGen se pueden ejecutar por diferentes dispositivos respectivamente.

En las realizaciones anteriores, el programa de intervalo M^ acepta la secuencia de entrada δ si y sólo si la combinación lineal de las filas de la matriz M_δ obtenida a partir de la matriz ^ por la secuencia de entrada δ da $1^{\rightarrow}$. Alternativamente, el programa de intervalo M^ puede aceptar la secuencia de entrada δ solamente si se obtiene otro vector $v^{\rightarrow}$ en lugar de $1^{\rightarrow}$.

En este caso, en el algoritmo KeyGen, $s_0 := v^{\rightarrow} \cdot f^{\rightarrow T}$ se puede establecer en lugar de $s_0 := 1^{\rightarrow} \cdot f^{\rightarrow T}$. Del mismo modo, cuando se calcula α_i en el algoritmo Sig, se puede calcular α_i con el cual $\Sigma_{i \in I} \alpha_i M_i = v^{\rightarrow}$.

El proceso criptográfico en la descripción anterior se puede adoptar para delegación de autoridad. Delegación de autoridad significa que una persona que tiene una clave de descifrado genera una clave de descifrado de nivel más bajo que tiene una autoridad más débil que la clave de descifrado suya propia. Una autoridad más débil significa que están limitados los datos cifrados que la clave de descifrado puede descifrar.

Por ejemplo, en la primera capa jerárquica (orden más alto), se usan las bases B_t y B^*_t para t = 1, en la segunda capa jerárquica, se usan las bases B_t y B^*_t para t = 1, 2, ..., en la capa jerárquica de orden k, se usan las bases B_t y B^*_t para t = 1, ..., k. A medida que aumentan las bases B_t y B^*_t a ser usadas, se establece un número mayor de partes de información de atributo. Por consiguiente, la autoridad de la clave de descifrado se limita más.

Se describirá la configuración de hardware del sistema de procesamiento criptográfico 10 (el dispositivo de generación de claves 100, el dispositivo de cifrado 200 y el dispositivo de descifrado 300) y del sistema de procesamiento de firmas 20 (el dispositivo de generación de claves 100, el dispositivo de firma 400 y el dispositivo de verificación 500) en las realizaciones anteriores.

La Fig. 22 es una ilustración que muestra un ejemplo de la configuración de hardware de cada uno del dispositivo de generación de claves 100, el dispositivo de cifrado 200, el dispositivo de descifrado 300, el dispositivo de firma 400 y el dispositivo de verificación 500.

Como se muestra en la Fig. 22, cada uno del dispositivo de generación de claves 100, dispositivo de cifrado 200, dispositivo de descifrado 300, dispositivo de firma 400 y dispositivo de verificación 500 incluye una CPU 911 (también conocida como Unidad Central de Procesamiento, dispositivo de procesamiento central, dispositivo de procesamiento, dispositivo de cálculo, microprocesador, microordenador o procesador) que ejecuta programas. La CPU 911 está conectada a la ROM 913, la RAM 914, un LCD 901 (Visualizador de Cristal Líquido), el teclado 902 (K/B), la placa de comunicación 915 y el dispositivo de disco magnético 920 a través de un bus 912, y controla estos dispositivos de hardware. En lugar del dispositivo de disco magnético 920 (dispositivo de disco fijo), se puede conectar un dispositivo de almacenamiento tal como un dispositivo de disco óptico o un dispositivo de lectura/escritura de tarjeta de memoria. El dispositivo de disco magnético 920 está conectado a través de una interfaz predeterminada de disco fijo.

La ROM 913 y el dispositivo de disco magnético 920 son ejemplos de una memoria no volátil. La RAM 914 es un ejemplo de una memoria volátil. La ROM 913, la RAM 914 y el dispositivo de disco magnético 920 son ejemplos del dispositivo de almacenamiento (memoria). El teclado 902 y la placa de comunicación 915 son ejemplos de un dispositivo de entrada. La placa de comunicación 915 es un ejemplo de un dispositivo de comunicación (interfaz de red). Además, el LCD 901 es un ejemplo de un dispositivo de visualización.

El dispositivo de disco magnético 920, la ROM 913 o similar almacena un sistema operativo (OS) 921, un sistema de ventanas 922, programas 923 y archivos 924. La CPU 911, el sistema operativo 921 y el sistema de ventanas 922 ejecutan cada programa de los programas 923.

Los programas 923 almacenan software y programas que ejecutan las funciones descritas como la “parte de generación de clave maestra 110”, la “parte de almacenamiento de clave maestra 120”, la “parte de entrada de información 130”, la “parte de generación de clave de descifrado 140”, la “parte de distribución de clave 150”, la “parte de adquisición de parámetros públicos 210”, la “parte de entrada de información 220”, la “parte de generación de datos cifrados 230”, la “parte de transmisión de datos 240”, la “parte de adquisición de clave de descifrado 310”, la “parte de recepción de datos 320”, la “parte de cálculo de programa de intervalo 330”, la “parte de cálculo de coeficiente complementario 340”, la “parte de operación de emparejamiento 350”, la “parte de cálculo de información de texto plano 360”, la “parte de adquisición de clave de firma 410”, la “parte de entrada de información 420”, la “parte de cálculo de coeficiente complementario 430”, la “parte de generación de firma 450”, la “parte de transmisión de datos 460”, la “parte de adquisición de parámetros públicos 510”, la “parte de recepción de datos 520”, la “parte de generación de datos cifrados 530”, la “parte de operación de emparejamiento 540”, y similares en la descripción anterior. Los programas 923 almacenan otros programas también. Los programas se leen y se ejecutan por la CPU 911.

Los archivos 924 almacenan información, datos, valores de señal, valores de variables y parámetros tales como los “parámetros públicos pk”, la “clave maestra sk”, los “datos cifrados c”, la “clave de descifrado sk_s”, la “clave de descifrado sk_Γ”, la “estructura de acceso S”, el “conjunto de atributos Γ”, el “mensaje m”, los “datos de firma sig” y similares de la explicación anterior, como los elementos de un “archivo” y una “base de datos”. El “archivo” y la “base de datos” se almacenan en un medio de grabación tal como un disco o memoria. La información, los datos, los valores de señal, los valores de variables y los parámetros almacenados en el medio de grabación tal como el disco

o la memoria se leen de la memoria principal o la memoria caché por la CPU 911 a través de un circuito de lectura/escritura, y se usan para las operaciones de la CPU 911 tales como extracción, examen, búsqueda, comparación, computación, cálculo, proceso, salida, impresión y visualización. La información, los datos, los valores de señal, los valores de variables y los parámetros se almacenan temporalmente en la memoria principal, la memoria caché, la memoria de almacenador temporal o similares durante las operaciones de la CPU 1911, que incluyen extracción, examen, búsqueda, comparación, computación, cálculo, proceso, salida, impresión y visualización.

Las flechas de los diagramas de flujo en la explicación anterior indican principalmente la entrada/salida de datos y señales. Los valores de datos y señales se almacenan en la memoria de la RAM 914, el medio de grabación, tal como un disco óptico, o en un chip de IC. Los datos y las señales se transmiten en línea a través de un medio de transmisión tal como el bus 912, líneas de señal o cables, u ondas eléctricas.

La “parte” en la explicación anterior puede ser un “circuito”, “dispositivo”, “equipo”, “medio” o “función”; o un “paso”, “procedimiento” o “proceso”. El “dispositivo” puede ser un “circuito”, “equipo”, “medio” o “función”; o un “paso”, “procedimiento” o “proceso”. El “proceso” puede ser un “paso”. Esto es, la “parte” se puede implementar como microprograma almacenado en la ROM 913. Alternativamente, la “parte” se puede implementar mediante solamente software, mediante solamente hardware tal como un elemento, un dispositivo, un sustrato, o una línea de cableado; mediante una combinación de software y hardware; o además mediante una combinación de software, hardware y microprograma. El microprograma y el software se almacenan, como programas, en el medio de grabación tal como la ROM 913. El programa se lee por la CPU 911 y se ejecuta por la CPU 911. Esto es, el programa hace que el ordenador funcione como una “parte” descrita anteriormente. Alternativamente, el programa hace que el ordenador o similar ejecute el procedimiento y el método de la “parte” descrita anteriormente.

Lista de signos de referencia

10: sistema de procesamiento criptográfico; 20: sistema de procesamiento de firmas; 100: dispositivo de generación de claves; 110: parte de generación de clave maestra; 120: parte de almacenamiento de clave maestra; 130: parte de entrada de información; 140: parte de generación de clave de descifrado; 141: parte de generación de vector f; 142: parte de generación de vector s; 143: parte de generación de números aleatorios; 144: parte de generación de elemento de clave; 145: parte de generación de elemento confidencial; 150: parte de distribución de clave; 160: parte de generación de clave de firma; 161: parte de generación de números aleatorios; 162: parte de generación de elemento de clave; 200: dispositivo de cifrado; 210: parte de adquisición de parámetro público; 220: parte de entrada de información; 230: parte de generación de datos cifrados; 231: parte de generación de números aleatorios; 232: parte de generación de elementos criptográficos; 233: parte de generación de vector f; 234: parte de generación de vector s; 240: parte de transmisión de datos; 300: dispositivo de descifrado; 310: parte de adquisición de clave de descifrado; 320: parte de recepción de datos; 330: parte de cálculo del programa de intervalo; 340: parte de cálculo de coeficiente complementario; 350: parte de operación de emparejamiento; 360: parte de cálculo de información de texto plano; 400: dispositivo de firma; 410: parte de adquisición de clave de firma; 420: parte de entrada de información; 430: parte de cálculo de coeficiente complementario; 440: parte de generación de matriz; 450: parte de generación de firma; 451: parte de generación de números aleatorios; 452: parte de generación de elemento de firma; 460: parte de transmisión de datos; 500: dispositivo de verificación; 510: parte de adquisición de parámetros públicos; 520: parte de recepción de datos; 530: parte de generación de datos cifrados; 540: parte de operación de emparejamiento; 531: parte de generación de números aleatorios; 532: parte de generación de elementos criptográficos; 533: parte de generación de vector f; 534: parte de generación de vector s.

REIVINDICACIONES

1. Un sistema de procesamiento criptográfico (10) que comprende un dispositivo de generación de claves (100), un dispositivo de cifrado (200) y un dispositivo de descifrado (300), y que sirve para ejecutar un proceso criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más),

en donde el dispositivo de generación de claves (100) incluye

una primera parte de entrada de información (130) que toma como entrada, una variable ρ(i) para cada número entero i = 1, …, L (L es un número entero de 1 o más), cuya variable ρ(i) es una cualquiera de una tupla positiva (t, $\vec{v}_i$) y una tupla negativa ¬(t, $\vec{v}_i$) de la información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,\,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más), y una matriz M predeterminada que tiene L filas y r columnas (r es un número entero de 1 o más); y

una parte de generación de clave de descifrado (140) que genera un elemento k^*_0 y un elemento k^*_i para cada número entero i = 1, ..., L, en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ y un vector $\vec{w}$, cada uno que tiene r partes de elementos, y la matriz M introducida por la primera parte de entrada de información; un valor $s_0 := \vec{w} \cdot \vec{f}$; y un valor predeterminado θ_i (i = 1, ..., L), la parte de generación de clave de descifrado (140) que está configurada

para generar el elemento k^*_0 estableciendo el valor $-s_0$ como coeficiente para un vector base $b^*_{0,p}$ (p es un valor predeterminado) de la base B^*_0 y estableciendo un valor predeterminado κ como coeficiente para un vector base $b^*_{0,q}$ (q es un valor predeterminado diferente del p prescrito), y

para generar el elemento k^*_i para cada número entero i = 1, ..., L, cuando la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$) estableciendo $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b^*_{t,1}$ de la base B^*_t indicado por la información de identificación t de la tupla positiva, y estableciendo $\theta_i v_{i,i'}$ como coeficiente para un vector base $b^*_{t,i'}$ indicado por la información de identificación t y cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$), estableciendo $s_i v_{i,i'}$ como coeficiente para el vector base $b^*_{t,i'}$ indicado por la información de identificación t de la tupla negativa y por cada número entero i' = 1, ..., n_t,

en donde el dispositivo de cifrado (200) incluye

una segunda parte de entrada de información (220) que toma como entrada, un conjunto de atributos Γ que tiene la información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d, y

una parte de generación de datos cifrados (230) que genera un elemento c_0, y un elemento c_t que conciernen a cada información de identificación t incluida en el conjunto de atributos Γ, en base al conjunto de atributos Γ introducido por la segunda parte de entrada de información, la parte de generación de datos cifrados que está configurada

para generar el elemento c_0 donde un valor aleatorio δ se establece como coeficiente para un vector base $b_{0,p}$ (p es un p prescrito) de la base B_0, y donde un valor predeterminado ζ se establece como coeficiente para un vector base $b_{0,q}$ (q es un q prescrito) de la base B_0, y

para generar el elemento c_t donde $x_{t,i'}$ multiplicado por el valor aleatorio δ se establece como coeficiente para un vector base $b_{t,i'}$ (i' = 1, ..., n_t) de la base B_t para cada información de identificación t incluida en el conjunto de atributos Γ, y

en donde el dispositivo de descifrado (300) incluye

una parte de adquisición de datos (320) que adquiere datos cifrados c que incluyen los elementos c_0 y c_t y el conjunto de atributos Γ, los elementos c_0 y c_t que se generan por la parte de generación de datos cifrados (230),

una parte de adquisición de clave de descifrado (310) que adquiere una clave de descifrado sk_s que incluye los elementos k^*_0 y k^*_i y el número variable ρ(i), los elementos k^*_0 y k^*_i que se generan por la parte de generación de clave de descifrado (140),

una parte de cálculo de coeficiente complementario (340) que, en base al conjunto de atributos Γ incluido en los datos cifrados c adquiridos por la parte de adquisición de datos, y la variable ρ(i) incluida en la clave de descifrado sk_s adquirida por la parte de adquisición de clave de descifrado, especifica, entre los números enteros i = 1, ..., L, un conjunto I de un número entero i para el cual la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$) y con la cual un producto interno de $\vec{v}_i$ de la tupla positiva y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$) y con la cual un producto interno de $\vec{v}_i$ de la tupla negativa y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0; y calcula un coeficiente complementario α_i con el cual un total de $\alpha_i s_i$ para i incluido en el conjunto I especificado llega a ser s_0, y

una parte de operación de emparejamiento (350) que calcula un valor $K = g_T^{\zeta\lambda}$ dirigiendo una operación de emparejamiento indicada en la Fórmula 1 para los elementos c_0 y c_t incluidos en los datos cifrados c y los elementos k^*_0 y k^*_i incluidos en la clave de descifrado sk_s, en base al conjunto I especificado por la parte de cálculo de coeficiente complementario (340) y al coeficiente complementario α_i calculado por la parte de cálculo de coeficiente complementario (340).

[Fórmula 1]

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_t, k_i^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_t, k_i^*)^{\alpha_i / (\vec{v}_i \cdot \vec{x}_i)}$$

2. El sistema de procesamiento criptográfico (10) según la reivindicación 1, que ejecuta el proceso criptográfico usando la base B_0 que tiene al menos un vector base $b_{0,i}$ (i = 1, ..., 5), la base B_t (t = 1, …, d) que tiene al menos un vector base $b_{t,i}$ (i = 1, ..., n_t, …, n_t+u_t, …, $n_t+u_t+w_t$, …, $n_t+u_t+w_t+z_t$) (u_t, w_t y z_t son cada uno un número entero de 1 o más), la base B^*_0 que tiene al menos un vector base $b^*_{0,i}$ (i = 1, ..., 5), y la base B^*_t (t = 1, …, d) que tiene al menos un vector base $b^*_{t,i}$ (i = 1, ..., n_t, …, n_t+u_t, …, $n_t+u_t+w_t$, …, $n_t+u_t+w_t+z_t$),

en donde, la parte de generación de clave de descifrado (140) del dispositivo de generación de claves (100) genera el elemento k^*_0 indicado en la Fórmula 2 en base a un valor aleatorio η_0 y el valor predeterminado κ, genera el elemento k^*_i indicado en la Fórmula 3 en base al valor aleatorio θ_i (i = 1, …, L) y un valor aleatorio $\eta_{i,i'}$ (i = 1, …, L, i' = 1, …, w_t) cuando la variable $\rho(i)$ es una tupla positiva (t, $\vec{v}_i$), y genera el elemento k^*_i indicado en la Fórmula 4 en base al valor aleatorio $\eta_{i,i'}$ (i = 1, …, L, i' = 1, …, w_t) cuando la variable $\rho(i)$ es una tupla negativa $\neg$(t, $\vec{v}_i$), y

en donde la parte de generación de datos cifrados (230) del dispositivo de cifrado (200) genera el elemento c_0 indicado en la Fórmula 5 en base al valor aleatorio δ, un valor aleatorio Φ_0, y un valor predeterminado ζ, y genera el elemento c_t indicado la Fórmula 6 en base al valor aleatorio δ, y un valor aleatorio $\Phi_{t,i}$ (i = 1, ..., z_t).

[Fórmula 2]

$$k_0^* := (-s_0, 0, \kappa, \eta_0, 0)_{\mathbb{B}_0^*}$$

[Fórmula 3]

$$k_i^* := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,w_t}}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*}$$

[Fórmula 4]

$$k_i^* := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,w_t}}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*}$$

[Fórmula 5]

$$c_0 := (\delta, 0, \zeta, 0, \phi_0)_{\mathbb{B}_0}$$

[Fórmula 6]

$$c_t := (\overbrace{\delta(x_{t,1}, \ldots, x_{t,n_t})}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\phi_{t,1}, \ldots, \phi_{t,z_t}}^{z_t})_{\mathbb{B}_t}$$

3. Un dispositivo de generación de claves (100) que genera una clave de descifrado sk_s, en un sistema de procesamiento criptográfico (10) que ejecuta un proceso criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más), el dispositivo de generación de claves (100) que comprende:

una primera parte de entrada de información (130) que toma como entrada, una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), que es una cualquiera de una tupla positiva $(t, \vec{v}_i)$ y una tupla negativa $\neg(t, \vec{v}_i)$ de información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más), y una matriz predeterminada M que tiene L filas y r columnas (r es un número entero de 1 o más);

una parte de generación de clave de descifrado (140) que genera un elemento k^*_0, y un elemento k^*_i para cada número entero i = 1 ..., L, en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ y a un vector $\vec{w}$, cada uno que tiene r partes de elementos, y la matriz M introducida por la primera parte de entrada de información; un valor $s_0 := \vec{w} \cdot \vec{f}$; y un valor predeterminado θ_i (i = 1, ..., L), la parte de generación de clave de descifrado (140) que se configura

para generar el elemento k^*_0 estableciendo un valor $-s_0$ como coeficiente para un vector base $b^*_{0,p}$ (p es un valor predeterminado) de la base B^*_0 y estableciendo un valor predeterminado κ como coeficiente para el vector base $b^*_{0,q}$ (q es un valor predeterminado diferente del p prescrito), y

para generar el elemento k^*_i para cada número entero i = 1, ..., L, cuando la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, estableciendo $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b^*_{t,1}$ de la base B^*_t indicado por la información de identificación t de la tupla positiva, y estableciendo $\theta_i v_{i,i'}$ como coeficiente para un vector base $b^*_{t,i'}$ indicado por la información de identificación t y cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$ estableciendo $s_i v_{i,i'}$ como coeficiente para el vector base $b^*_{t,i'}$ indicado por la información de identificación t de la tupla negativa y para cada número entero i' = 1, ..., n_t; y

una parte de distribución de clave (150) que distribuye datos que incluyen la variable ρ(i) introducida por la primera parte de entrada de información (130) y los elementos k^*_0 y k^*_i generados por la parte de generación de clave de descifrado (140), como la clave de descifrado sk_s.

4. Un dispositivo de cifrado (200) que genera datos cifrados c en un sistema de procesamiento criptográfico (10) que ejecuta un proceso de cifrado criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (1 es un número entero de 1 o más), el dispositivo de cifrado (200) que comprende:

una segunda parte de entrada de información (220) que toma como entrada, un conjunto de atributos Γ que tiene la información de identificación t y el vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d;

una parte de generación de datos cifrados (230) que genera un elemento c_0, y un elemento c_t concerniente a cada información de identificación t incluida en el conjunto de atributos Γ, en base al conjunto de atributos Γ introducido por la segunda parte de entrada de información, la parte de generación de datos cifrados que está configurada

para generar el elemento c_0 donde se establece un valor aleatorio δ como coeficiente para un vector base $b_{0,p}$ (p es un p prescrito) de una base B_0, y donde un valor predeterminado ζ se establece como coeficiente para un vector base $b_{0,q}$ (q es un q prescrito) de una base B_0, y

para generar el elemento c_t donde $x_{t,i'}$ multiplicado por el valor aleatorio δ se establece como coeficiente para un vector base $b_{t,i'}$ (i' = 1, ..., n_t) de la base B_t para cada información de identificación t incluida en el conjunto de atributos; y

una parte de salida de datos (240) que emite el conjunto de atributos Γ introducido por la segunda parte de entrada de información (220) y los elementos c_0 y c_t generados por la parte de generación de datos cifrados (230), como los datos cifrados c.

5. Un dispositivo de descifrado (300) que descifra los datos cifrados c con una clave de descifrado sk_s en un sistema de procesamiento criptográfico (10) que ejecuta un proceso criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más), el dispositivo de descifrado (300) que comprende:

una parte de adquisición de datos (320) que adquiere

un elemento c_0 y un elemento c_t (cada número entero incluido en el conjunto de atributos Γ) generados en base a un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d, junto con el conjunto de atributos Γ, como los datos cifrados c,

el elemento c_0 que se establece con un valor aleatorio δ como un coeficiente para un vector base $b_{0,p}$ (p es un valor predeterminado) de una base B_0 y que se establece con un valor predeterminado ζ como coeficiente para un vector base $b_{0,q}$ (q es un valor predeterminado diferente del p prescrito) de la base B_0,

el elemento c_t que se establece con $x_{t,i'}$ multiplicado por el valor aleatorio δ, como coeficiente para un vector base $b_{t,i'}$ ($i' = 1, ..., n_t$) de la base B_t, para cada información de identificación t incluida en el conjunto de atributos;

una parte de adquisición de clave de descifrado (310) que adquiere

un elemento k^*_0 y un elemento k^*_i (cada número entero i = 1 ..., L) generado en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ y un vector $\vec{w}$ cada uno que tiene r (r es un entero de 1 o más) partes de elementos, y una matriz M predeterminada que tiene L filas y r columnas; un valor $s_0 := \vec{w} \cdot \vec{f}$; un valor predeterminado θ_i (i = 1, ..., L); y una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), junto con la variable ρ(i), la variable ρ(i) que es una cualquiera de una tupla positiva $(t, \vec{v}_i)$, y una tupla negativa $\neg(t, \vec{v}_i)$, de la información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ ($i' = 1, ..., n_t$, n_t es un número entero de 1 o más), como la clave de descifrado sk_s,

el elemento k^*_0 que se establece con el valor $-s_0$ como coeficiente para un vector base $b^*_{0,p}$ (p es un p prescrito) de una base B^*_0 y que se establece con un valor predeterminado κ como coeficiente para el vector base $b^*_{0,q}$ (q es un q prescrito) de la base B^*_0, y

el elemento k^*_i para cada número entero i = 1, ..., L, cuando la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, que se establece con $s_i + \theta_i v_{i,1}$ como coeficiente para un vector base $b^*_{t,1}$ de la base B^*_t indicado por la información de identificación t de la tupla positiva, y que se establece con $\theta_i v_{i,i'}$ como coeficiente de un vector base $b^*_{t,i'}$ indicado por la información de identificación t y por cada número entero $i' = 2, ..., n_t$, y cuando la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, que se establece con $s_i v_{i,i'}$ como coeficiente para el vector base $b^*_{t,i'}$ indicado por la información de identificación t de la tupla negativa y por cada número entero $i' = 1, ..., n_t$;

una parte de cálculo de coeficiente complementario (340) que, en base al conjunto de atributos Γ incluido en los datos cifrados c adquiridos por la parte de adquisición de datos, y la variable ρ(i) incluida en la clave de descifrado sk_s adquirida por la parte de adquisición de clave de descifrado, especifica, entre los números enteros i = 1, ..., L, un conjunto I de un número entero i para el cual la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, y con la cual un producto interno de $\vec{v}_i$ de la tupla positiva y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$ y con la cual un producto interno de $\vec{v}_i$ de la tupla negativa y $\vec{x}_t$ incluido en el Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0; y calcula un coeficiente complementario α_i con el cual un total de $\alpha_i s_i$ para i incluido en el conjunto I especificado llega a ser s_0; y

una parte de operación de emparejamiento (350) que calcula un valor $K = g_T^{\zeta\chi}$ dirigiendo una operación de emparejamiento indicada en la Fórmula 7 para los elementos c_0 y c_t incluidos en los datos cifrados c y los elementos k^*_0 y k^*_i incluidos en la clave de descifrado sk_s, en base al conjunto I especificado por la parte de cálculo de coeficiente complementario y al coeficiente complementario α_i calculado por la parte de cálculo de coeficiente complementario (340).

[Fórmula 7]

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i) = (t, \vec{v}_i)} e(c_t, k_i^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i) = \neg(t, \vec{v}_i)} e(c_t, k_i^*)^{\alpha_i / (\vec{v}_i \cdot \vec{x}_i)}$$

6. Un sistema de procesamiento criptográfico (10) que comprende un dispositivo de generación de claves (100), un dispositivo de cifrado (200) y un dispositivo de descifrado (300), y que sirve para ejecutar un proceso criptográfico usando una base B_t y base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más),

en donde el dispositivo de generación de claves (100) incluye

una primera parte de entrada de información (130) que toma como entrada, un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ ($i' = 1, ..., n_t$ donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d, y

una parte de generación de clave de descifrado (140) que genera un elemento k^*_0, y un elemento k^*_t que conciernen a cada información de identificación t incluida en el conjunto de atributos Γ, en base al conjunto de atributos Γ introducido por la primera parte de entrada de información, la parte de generación de clave de descifrado (140) que está configurada

para generar el elemento k^*_0 donde un valor aleatorio δ se establece como coeficiente para un vector base $b^*_{0,p}$ (p es un valor predeterminado) de una base B^*_0, y donde un valor κ predeterminado se establece como coeficiente para un vector base $b^*_{0,q}$ (q es un valor predeterminado diferente de un p prescrito), y

para generar el elemento k^*_t donde $x_{t,i'}$ multiplicado por el valor aleatorio δ se establece como coeficiente para un vector base $b^*_{t,i'}$ (i '= 1, ..., n_t) de la base B^*_t, para cada información de identificación t incluida en el conjunto de atributos Γ,

en donde el dispositivo de cifrado (200) incluye

una segunda parte de entrada de información (220) que toma como entrada, una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), que es una cualquiera de una tupla positiva (t, $\vec{v}_i$) y una tupla negativa ¬(t, $\vec{v}_i$), de la información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más), y una matriz M predeterminada que tiene L filas y r columnas (r es un número entero de 1 o más), y

una parte de generación de datos cifrados (230) que genera un elemento c_0 y un elemento c_i para cada número entero i = 1 ..., L, en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ y un vector $\vec{w}$ cada uno que tiene r partes de elementos, y la matriz M introducida por la segunda parte de entrada de información; un valor $s_0 := \vec{w} \cdot \vec{f}$; y un valor predeterminado θ_i (i = 1, ..., L); la parte de generación de datos cifrados (230) que está configurada

para generar el elemento c_0 estableciendo el valor $-s_0$ como un coeficiente para un vector base $b_{0,p}$ (p es un p prescrito) de la base B_0 y estableciendo un valor predeterminado ζ como coeficiente para el vector base $b_{0,q}$ (q es un q prescrito), y

para generar el elemento c_i para cada número entero i = 1, ..., L, cuando la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$), estableciendo $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b_{t,1}$ de la base B_t indicada por la información de identificación t de la tupla positiva, y estableciendo $\theta_i v_{i,i'}$ como coeficiente de un vector base $b_{t,i'}$ indicado por la información de identificación t y cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$) estableciendo $s_i v_{i,i'}$ como coeficiente para el vector base $b_{t,i'}$ indicado por la información de identificación t de la tupla negativa y por cada número entero i' = 1, ..., n_t, y

en donde el dispositivo de descifrado (300) incluye

una parte de adquisición de datos (320) que adquiere datos cifrados c que incluyen los elementos c_0 y c_t y la variable ρ(i), los elementos c_0 y c_t que se generan por la parte de generación de datos cifrados (230),

una parte de adquisición de clave de descifrado (310) que adquiere la clave de descifrado sk_Γ que incluye los elementos k^*_0 y k^*_i y el conjunto de atributos Γ, los elementos k^*_0 y k^*_i que se generan por la parte de generación de clave de descifrado (140),

una parte de cálculo de coeficiente complementario (340) que, en base a la variable ρ(i) que concierne a cada número entero i = 1, ..., L incluido en los datos cifrados c adquiridos por la parte de adquisición de datos (320) y el conjunto de atributos Γ incluido en la clave de descifrado sk_Γ adquirida por la parte de adquisición de clave de descifrado, especifica, entre los números enteros i = 1, ..., L, un conjunto I de un número entero i para el cual la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$) y con el cual un producto interno de $\vec{v}_i$ de la tupla positiva y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$) y con el cual un producto interno de $\vec{v}_i$ de la tupla negativa y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0; y calcula un coeficiente complementario α_i con el cual un total de $\alpha_i s_i$ para i incluido en el conjunto I especificado llega a ser s_0; y

una parte de operación de emparejamiento (350) que calcula un valor $K = g_T^{\zeta\chi}$ dirigiendo una operación de emparejamiento indicada en la Fórmula 8 para los elementos c_0 y c_t incluidos en los datos cifrados c y los elementos k^*_0 y k^*_i incluidos en la clave de descifrado sk_Γ, en base al conjunto I especificado por la parte de cálculo de coeficiente complementario (340) y el coeficiente complementario α_i calculado por la parte de cálculo de coeficiente complementario (340).

[Fórmula 8]

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i/(\vec{v}_i \cdot \vec{x}_t)}$$

7. El sistema de procesamiento criptográfico según la reivindicación 6, que ejecuta el proceso criptográfico usando la base B_0 que tiene al menos un vector base $b_{0,i}$ (i = 1, ..., 5), la base B_t (t = 1, ..., d) que tiene al menos un vector base $b_{t,i}$, (i = 1, ..., n_t, ..., n_t+u_t, ..., $n_t+u_t+w_t$, ..., $n_t+u_t+ w_t+z_t$) (u_t, w_t y z_t son cada uno un número entero de 1 o más), la base B^*_0 que tiene al menos un vector base $b^*_{0,i}$ (i = 1, ..., 5), y la base B^*_t (t = 1, ..., d) que tiene al menos un vector base $b^*_{t,i}$ (i = 1, ..., n_t, ..., n_t+u_t, ..., $n_t+u_t+w_t$, ..., $n_t+u_t+ w_t+z_t$),

en donde la parte de generación de clave de descifrado (140) del dispositivo de generación de claves (100) genera el elemento k^*_0 indicado en la Fórmula 9 en base al valor aleatorio δ, un valor aleatorio Φ_0 y el valor predeterminado κ, y genera el elemento k^*_t indicado en la Fórmula 10 en base a los valores aleatorios δ y $\Phi_{t,i}$ (i = 1, ..., w_t), y

en donde la parte de generación de datos cifrados (230) del dispositivo de cifrado (200) genera el elemento c_0 indicado en la Fórmula 11 en base a un valor aleatorio η_0 y un valor predeterminado ζ, genera un elemento c_i indicado en la Fórmula 12 en base a un valor aleatorio θ_i (i = 1, ..., L) y un valor aleatorio $\eta_{i,i'}$ (i = 1, ..., L, i '= 1, ..., z_t) cuando la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$), y genera un elemento c_i indicado en la Fórmula 13 en base al valor aleatorio $\eta_{i,i'}$ (i = 1, ..., L, i '= 1, ..., z_t) cuando la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$).

[Fórmula 9]

$$k_0^* := (\delta, 0, \kappa, \phi_0, 0)_{\mathbb{B}_0^*}$$

[Fórmula 10]

$$k_t^* := (\overbrace{\delta(x_{t,1},\ldots,x_{t,n_t})}^{n_t}, \overbrace{0^{u_t},}^{u_t} \overbrace{\phi_{t,1},\ldots,\phi_{t,w_t}}^{w_t} \overbrace{0^{z_t},}^{z_t})_{\mathbb{B}_t^*}$$

[Fórmula 11]

$$c_0 := (-s_0, 0, \zeta, 0, \eta_0)_{\mathbb{B}_0}$$

[Fórmula 12]

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2},\ldots,\theta_i v_{i,n_t},}^{n_t} \overbrace{0^{u_t},}^{u_t} \overbrace{0^{w_t},}^{w_t} \overbrace{\eta_{i,1},\ldots,\eta_{i,z_t},}^{z_t})_{\mathbb{B}_t}$$

[Fórmula 13]

$$c_i := (\overbrace{s_i(v_{i,1},\ldots,v_{i,n_t}),}^{n_t} \overbrace{0^{u_t},}^{u_t} \overbrace{0^{w_t},}^{w_t} \overbrace{\eta_{i,1},\ldots,\eta_{i,z_t},}^{z_t})_{\mathbb{B}_t}$$

8. Un dispositivo de generación de claves (100) que genera una clave de descifrado sk_Γ en un sistema de procesamiento criptográfico (10) que ejecuta un proceso criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más), el dispositivo de generación de claves (100) que comprende:

una primera parte de entrada de información (130) que toma como entrada, un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d;

una parte de generación de clave de descifrado (140) que genera un elemento k^*_0 y un elemento k^*_t que conciernen a cada información de identificación t incluida en el conjunto de atributos Γ, en base al conjunto de atributos Γ introducido por la primera parte de entrada de información, la parte de generación de clave de descifrado (140) que está configurada

para generar el elemento k^*_0 donde un valor aleatorio δ se establece como coeficiente para un vector base $b^*_{0,p}$ (p es un valor predeterminado) de una base B^*_0, y donde un valor predeterminado κ se establece como coeficiente para un vector base $b^*_{0,q}$ (q es un valor predeterminado diferente de un p prescrito), y

para generar el elemento k^*_t donde $x_{t,i'}$ multiplicado por el valor aleatorio δ se establece como coeficiente para un vector base $b^*_{t,i'}$ (i' = 1, ..., n_t) de la base B^*_t, para cada información de identificación t incluida en el conjunto de atributos Γ; y

una parte de distribución de claves (150) que distribuye datos que incluyen el conjunto de atributos Γ introducido por la primera parte de entrada de información (130) y los elementos k^*_0 y k^*_i generados por la parte de generación de clave de descifrado (140), como la clave de descifrado sk_s.

9. Un dispositivo de cifrado (200) que genera datos cifrados c en un sistema de procesamiento criptográfico (10) que ejecuta un proceso criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más), el dispositivo de cifrado (200) que comprende:

una segunda parte de entrada de información (220) que toma como entrada, una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), que es una cualquiera de una tupla positiva $(t, \vec{v}_i)$ y una tupla negativa $\neg(t, \vec{v}_i)$ de información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más), y una matriz M predeterminada que tiene L filas y r columnas (r es un número entero de 1 o más);

una parte de generación de datos cifrados (230) que genera un elemento c_0, y un elemento c_i para cada número entero i = 1 ..., L, en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ y un vector $\vec{w}$ cada uno que tiene r partes de elementos, y la matriz M introducida por la segunda parte de entrada de información; un valor $s_0 := \vec{w} \cdot \vec{f}$; y un valor predeterminado θ_i (i = 1, ..., L), la parte de generación de datos cifrados (230) que está configurada

para generar el elemento c_0 estableciendo un valor $-s_0$ como coeficiente para un vector base $b^*_{0,p}$ (p es un p prescrito) de una base B_0 y estableciendo un valor predeterminado ζ como coeficiente para un vector base $b_{0,q}$ (q es un q prescrito), y

para generar el elemento c_i para cada número entero i = 1, ..., L, cuando la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, estableciendo $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b_{t,1}$ de la base B_t indicada por la información de identificación t de la tupla positiva, y estableciendo $\theta_i v_{i,i'}$ como coeficiente de un vector base $b_{t,i'}$ indicado por la información de identificación t y cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, estableciendo $s_i v_{i,i'}$ como coeficiente para el vector base $b_{t,i'}$ indicado por la información de identificación t de la tupla negativa y por cada número entero i' = 1, ..., n_t; y

una parte de salida de datos (240) que emite la variable ρ(i) introducida por la segunda parte de entrada de información (220) y los elementos c_0 y c_t generados por la parte de generación de datos cifrados (230), como los datos cifrados c.

10. Un dispositivo de descifrado (300) que descifra los datos cifrados c con una clave de descifrado sk_Γ en un sistema de procesamiento criptográfico (10) que ejecuta un proceso criptográfico usando una base B_t y una base B^*_t para cada número entero t = 0, ..., d (d es un número entero de 1 o más), el dispositivo de descifrado que comprende:

una parte de adquisición de datos (320) que adquiere

un elemento c_0 y un elemento c_i (para cada número entero i = 1 ..., L) generado en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ y un vector $\vec{w}$ cada uno que tiene r (r es un número entero de 1 o más) partes de elementos, y la matriz M predeterminada que tiene L filas y r columnas; un valor $s_0 := \vec{w} \cdot \vec{f}$; un valor predeterminado θ_i (i = 1, ..., L); y una variable ρ(i) para cada número entero i = 1, …, L (L es un número entero de 1 o más), junto con la variable ρ(i), la variable ρ(i) que es una cualquiera de una tupla positiva $(t, \vec{v}_i)$ y una tupla negativa $\neg(t, \vec{v}_i)$ de información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ (i' = 1, ..., n_t, n_t es un número entero de 1 o más), como los datos cifrados c,

el elemento c_0 que se establece con un valor $-s_0$ como coeficiente para un vector base $b_{0,p}$ (p es un valor predeterminado) de una base B_0 y que se establece con un valor predeterminado ζ como coeficiente para un vector base $b_{0,q}$ (q es un valor predeterminado diferente del p prescrito) de la base B_0, y

el elemento c_i para cada número entero i = 1, ..., L, cuando la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, que se establece con $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b_{t,1}$ de la base B_t indicada por la información de identificación t de la tupla positiva, y que se establece con $\theta_i v_{i,i'}$ como coeficiente para un vector base $b_{t,i'}$ indicado por la información de identificación t y por cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$ que se establece con $s_i v_{i,i'}$ como coeficiente para el vector base $b_{t,i'}$ indicado por la información de identificación t de la tupla negativa y por cada número entero i' = 1, ..., n_t;

una parte de adquisición de clave de descifrado (310) que adquiere

un elemento k^*_0 generado en base a un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d, y un elemento k^*_t (t es cada número entero incluido en el conjunto de atributos Γ), como la clave de descifrado sk_Γ junto con el conjunto de atributos Γ,

el elemento k^*_0 que se establece con un valor aleatorio δ como coeficiente para un vector base $b^*_{0,p}$ (p es un p prescrito) de una base B^*_0 y que se establece con un valor predeterminado κ como coeficiente para un vector base $b^*_{0,q}$ (q es un q prescrito) de la base B^*_0,

el elemento k^*_t que se establece con $x_{t,i'}$ multiplicado por el valor aleatorio δ, como coeficiente para un vector base $b^*_{t,i'}$ ($i' = 1, ..., n_t$) de una base B^*_t, para cada información de identificación t incluida en el conjunto de atributos Γ;

una parte de cálculo de coeficiente complementario (340) que, en base a la variable ρ(i) incluida en los datos cifrados c adquiridos por la parte de adquisición de datos (320), y el conjunto de atributos Γ incluido en la clave de descifrado sk_s adquirida por la parte de adquisición de clave de descifrado (310), especifica, entre los números enteros i = 1, ..., L, un conjunto I de un número entero i para el cual la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$ y con la cual un producto interno de $\vec{v}_i$ de la tupla positiva y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$ y con la cual un producto interno de $\vec{v}_i$ de la tupla negativa y $\vec{x}_t$ incluido en el Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0; y calcula un coeficiente complementario α_i con el cual un total de $\alpha_i s_i$ para i incluido en el conjunto I especificado llega a ser s_0; y

una parte de operación de emparejamiento (350) que calcula un valor $K = g_T^{\zeta\chi}$ dirigiendo una operación de emparejamiento indicada en la Fórmula 14 para los elementos c_0 y c_t incluidos en los datos cifrados c y los elementos k^*_0 y k^*_i incluidos en la clave de descifrado sk_s, en base al conjunto I especificado por la parte de cálculo de coeficiente complementario (340) y el coeficiente complementario α_i calculado por la parte de cálculo de coeficiente complementario (340).

[Fórmula 14]

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i} \cdot \prod_{i \in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i/(\vec{v}_i \cdot \vec{x}_t)}$$

11. Un sistema de procesamiento de firmas (20) que comprende un dispositivo de generación de claves (100), un dispositivo de firma (400) y un dispositivo de verificación (500), y que sirve para ejecutar un proceso de firma usando una base B_t y una base B^*_t para cada número entero t = 1, ..., d, d+2 (d es un número entero de 1 o más),

en donde el dispositivo de generación de claves (100) incluye

una primera parte de entrada de información (130) que toma como entrada, un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ ($i' = 1, ..., n_t$ donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d, y

una parte de generación de clave de firma (160) que genera un elemento k^*_t que concierne a cada información de identificación t incluida en el conjunto de atributos Γ, y un elemento k^*_{d+2} y un elemento k^*_{d+3} en base al conjunto de atributos Γ introducido por la primera parte de entrada de información, la parte de generación de clave de firma (160) que está configurada

para generar el elemento k^*_t donde $x_{t,i'}$ multiplicado por un valor aleatorio δ se establece como coeficiente para un vector base $b^*_{t,i'}$ ($i' = 1, ..., n_t$) de la base B^*_t, para cada información de identificación t incluida en el conjunto de atributos Γ,

para generar el elemento k^*_{d+2} donde el valor aleatorio δ se establece como coeficiente para un vector base $b^*_{d+2,p'}$ (p' es un valor predeterminado) de una base B^*_{d+2}, y

para generar el elemento k^*_{d+3} donde el valor aleatorio δ se establece como coeficiente para un vector base $b^*_{d+2,q'}$ (q' es un valor predeterminado diferente de un p' prescrito) de la base B^*_{d+2},

en donde el dispositivo de firma (400) incluye

una segunda parte de entrada de información (420) que toma como entrada, una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), que es una cualquiera de una tupla positiva $(t, \vec{v}_i)$ y una tupla negativa $\neg(t, \vec{v}_i)$ de información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ ($i' = 1, ..., n_t$ donde n_t es un número entero de 1 o más), una matriz M predeterminada que tiene L filas y r columnas (r es un número entero de 1 o más), y un mensaje m,

una parte de adquisición de clave de firma (410) que adquiere los elementos k^*_t, k^*_{d+2} y k^*_{d+3} generados por la parte de generación de clave de firma (160), y el conjunto de atributos Γ, como clave de firma sk_Γ,

una parte de cálculo de coeficiente complementario (430) que, en base a la variable ρ(i) para cada número entero i = 1, ..., L y el conjunto de atributos Γ incluido en la clave de firma sk_Γ adquirida por la parte de adquisición de clave de

firma (410), especifica, entre los números enteros i = 1, ..., L, un conjunto I de un número entero i para el cual la variable ρ(i) es una tupla positiva (t, $v^{\rightarrow}_i$) y con la cual un producto interno de $v^{\rightarrow}_i$ de la tupla positiva y $x^{\rightarrow}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa ¬(t, $v^{\rightarrow}_i$) y con la cual un producto interno de $v^{\rightarrow}_i$ de la tupla negativa y $x^{\rightarrow}_t$ incluido en Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0; y calcula un coeficiente complementario α_i con el cual un total de $\alpha_i M_i$ para i incluido en el conjunto I especificado, en base a M_i que es un elemento en una fila de orden i de la matriz M introducida por la segunda parte de entrada de información (420), llega a ser un vector predeterminado $w^{\rightarrow}$, y

una parte de generación de firma (450) que, en base a la variable ρ(i), el conjunto de atributos Γ, el conjunto I especificado por la parte de cálculo de coeficiente complementario (430), y el coeficiente complementario α_i calculado por la parte de cálculo de coeficiente complementario (430), genera s^*_i para cada número entero i = 1, ..., L que representa cada número de fila de la matriz M, y s^*_x, como se indica en la Fórmula 15, donde

β_i es un valor con el que cada elemento llega a ser 0 cuando $\beta_i M_i$ se suma para cada número entero i que representa cada número de fila de la matriz M, y

para cada número entero i que representa cada número de fila de la matriz M,

cuando está establecido $i \in I$ y la variable ρ(i) es una tupla positiva (t, $v^{\rightarrow}_i$), permitamos un valor $\gamma_i := \alpha_i$, y permitamos que un vector $y^{\rightarrow}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $y^{\rightarrow}_i$ y $v^{\rightarrow}_i$ es 0 e $y_{i,1} = 1$,

cuando está establecido $i \in I$ y la variable ρ(i) es una tupla negativa ¬(t, $v^{\rightarrow}_i$), permitamos un valor $\gamma_i := \alpha_i/(v^{\rightarrow}_i \cdot x^{\rightarrow}_t)$, y permitamos que un vector $y^{\rightarrow}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $y^{\rightarrow}_i$ y $v^{\rightarrow}_i$ es 1,

cuando no está establecido $i \in I$ y la variable ρ(i) es una tupla positiva (t, $v^{\rightarrow}_i$), permitamos un valor $\gamma_i := 0$, y permitamos que un vector $y^{\rightarrow}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $y^{\rightarrow}_i$ y $v^{\rightarrow}_i$ es 0 e $y_{i,1} = 1$, y

cuando no está establecido $i \in I$ y la variable ρ(i) es una tupla negativa ¬(t, $v^{\rightarrow}_i$), permitamos un valor $\gamma_i := 0$, y permitamos que un vector $y^{\rightarrow}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $y^{\rightarrow}_i$ y $v^{\rightarrow}_i$ es 1, y

en donde el dispositivo de verificación (500) incluye

una parte de adquisición de datos (520) que adquiere datos de firma sig que incluyen s^*_i y s^*_x generados por la parte de generación de firma (450), el mensaje m, la variable ρ(i), y la matriz M,

una parte de generación de datos cifrados (530) que genera un elemento c_i y un elemento c_x, el elemento c_i que se genera para cada número entero i que representa un número de fila de la matriz M, en base a un vector de columna $s^{\rightarrow T} := (s_1, ..., s_L)^T := M \cdot f^{\rightarrow T}$ generado en base a un vector $f^{\rightarrow}$ que tiene r partes de elementos, el vector $w^{\rightarrow}$, y la matriz M adquirida por la parte de adquisición de datos (520); un valor $s_0 := w^{\rightarrow} \cdot f^{\rightarrow}$; y un valor predeterminado θ_i (i es cada número de fila de la matriz M, y X), la parte de generación de datos cifrados (530) que está configurada

para generar el elemento c_i para cada número entero i = 1 que representa un número de fila de la matriz M, cuando la variable ρ(i) es una tupla positiva (t, $v^{\rightarrow}_i$), estableciendo $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b_{t,1}$ de la base B_t indicada por la información de identificación t de la tupla positiva, y estableciendo $\theta_i v_{i,i'}$ como coeficiente para un vector base $b_{t,i'}$ indicado por la información de identificación t y cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa ¬(t, $v^{\rightarrow}_i$), estableciendo $s_i v_{i,i'}$ como coeficiente para el vector base $b_{t,i'}$ indicado por la información de identificación t de la tupla negativa y cada número entero i' = 1, ..., n_t, y

para generar el elemento c_x estableciendo $s_0+\theta_x m$ en un vector base $b_{d+2,p'}$ (p' es una p' prescrita) de un vector base B_{d+2} y estableciendo θ_x en un vector base $b_{d+2,q'}$ (q' es una q' prescrita) del vector base B_{d+2}, y

una parte de operación de emparejamiento (540) que verifica la validez de los datos de firma sig realizando una operación de emparejamiento indicada en la Fórmula 16 para cada número entero i que indica X y un número de fila de la matriz M.

[Fórmula 15]

$$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot (\sum_{l=1}^{n_t} y_{i,l} \cdot b_{t,l}^*) + r_i^*$$

$$s_X^* := \xi(k_{d+2}^* + m \cdot k_{d+3}^*) + r_X^*$$

donde

ξ, r^*_i y r^*_x son valores predeterminados.

[Fórmula 16]

$$\prod e(c_i, s_i^*)$$

12. El sistema de procesamiento de firmas (20) según la reivindicación 11, que ejecuta el proceso de firma usando una base B_0 que tiene al menos un vector base $b_{0,i}$ (i = 1, ..., 4), una base B_t (t = 1, ..., d) que tiene al menos un vector base $b_{t,i}$ (i = 1, ..., n_t, ..., n_t+u_t, ..., $n_t+u_t+w_t$, ..., $n_t+u_t+w_t+z_t$) (u_t, w_t y z_t son cada uno un número entero de 1 o más), una base B_{d+1} que tiene al menos un vector base $b_{d+1,i}$ (i = 1, ..., 4), una base B_{d+2} que tiene al menos un vector base $b_{d+2,i}$ (i = 1, ..., 8), una base B^*_0 que tiene al menos un vector base $b^*_{0,i}$ (i = 1, ..., 4), una base B^*_t (t = 1, ..., d) que tiene al menos un vector base $b^*_{t,i}$ (i = 1, ..., n_t, ..., n_t+u_t, ..., $n_t+u_t+w_t$, ..., $n_t+u_t+w_t+z_t$), una base B^*_{d+1} que tiene al menos un vector base $b^*_{d+1,i}$ (i = 1, ..., 4), y la base B^*_{d+2} que tiene al menos un vector base $b^*_{d+2,i}$ (i = 1, ..., 8),

en donde la parte de generación de clave de firma (160) del dispositivo de generación de claves (100) genera un elemento k^*_0, un elemento k^*_t, un elemento k^*_{d+2} y un elemento k^*_{d+3}, para cada información de identificación t incluida en el conjunto de atributos Γ y cada número entero τ = 1, …, w_t, en base a valores aleatorios δ, Φ_0, $\Phi_{t,\tau}$, $\Phi_{d+2,1}$, $\Phi_{d+2,2}$, $\Phi_{d+3,1}$ y $\Phi_{d+3,2}$, como se indica en la Fórmula 17,

en donde el dispositivo de firma (400) incluye además

una parte de generación de matriz (440) que añade un vector de fila predeterminado M_{L+1} a una fila de orden (L+1) de la matriz M,

en donde la parte de generación de firma (450) del dispositivo de firma (400) genera s^*_i para cada número entero i = 1, ..., L+1, y s^*_0 y s^*_{L+2} como s^*_x, en base a los elementos k^*_0, k^*_t,. k^*_{d+2} y k^*_{d+3}, y el valor aleatorio ξ, como se indica en Fórmula 18,

en donde la parte de generación de datos cifrados (530) del dispositivo de verificación (500) genera un elemento c_i, y los elementos c_0 y c_{L+2} como un elemento c_x, para cada número entero i = 1, ..., L+1 y cada número entero i' = 1, ..., z_t, en base a valores aleatorios θ_i, $\eta_{i,i'}$, $\eta_{L+2,1}$ y $\eta_{L+2,2}$, como se indica en la Fórmula 19, y

en donde la parte de operación de emparejamiento (540) realiza la operación de emparejamiento para cada número entero i = 0, .., L+2.

[Fórmula 17]

$$k_0^* := (\delta, 0, \phi_0, 0)_{\mathbb{B}_0^*}$$

$$k_t^* := (\overbrace{\delta(x_{t,1},\ldots,x_{t,n_t})}^{n_t}, \overbrace{0^{u_t},}^{u_t} \overbrace{\phi_{t,1},\ldots,\phi_{t,w_t},}^{w_t} \overbrace{0^{z_t},}^{z_t})_{\mathbb{B}_t^*}$$

$$k_{d+2}^* := (\delta(1,0), 0, 0, \phi_{d+2,1}, \phi_{d+2,2}, 0, 0)\mathbb{B}_{d+2}^*$$

$$k_{d+3}^* := (\delta(0,1), 0, 0, \phi_{d+3,1}, \phi_{d+3,2}, 0, 0)\mathbb{B}_{d+2}^*$$

[Fórmula 18]

$$s_0^* := \xi k_0^* + r_0^*$$

$$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot (\sum_{l=1}^{n_t} y_{i,l} \cdot b_{t,l}^*) + r_i^*$$

$$s_{L+2}^* := \xi(k_{d+2}^* + m \cdot k_{d+3}^*) + r_{L+2}^*$$

donde

ξ, r^*_0, r^*_i y r^*_{L+2} son valores predeterminados.

[Fórmula 19]

$$c_0 := (-s_0 - s_{L+2}, 0, 0, \eta_0)_{\mathbb{B}_0}$$

$$\text{si } \rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}),\ \eta_{i,1}, \ldots, \eta_{i,z_t} \xleftarrow{\cup} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,z_t}}^{z_t})_{\mathbb{B}_t},$$

$$\text{si } \rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q{}^{n_t}),\ \eta_{i,1}, \ldots, \eta_{i,z_t} \xleftarrow{\cup} \mathbb{F}_q,$$

$$c_i := (\overbrace{s_i(v_{i,1}, \ldots, v_{i,n_t})}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\eta_{i,1}, \ldots, \eta_{i,z_t}}^{z_t})_{\mathbb{B}_t},$$

$$c_{L+2} := (s_{L+2} - \theta_{L+2} m, \theta_{L+2}, 0, 0, 0, 0, \eta_{L+2,1}, \eta_{L+2,2})_{\mathbb{B}_{d+2}}$$

13. Un dispositivo de generación de claves (100) que genera una clave de firma sk_Γ en un sistema de procesamiento de firmas (20) que ejecuta un proceso de firma usando una base B_t y una base B^*_t para cada número entero t = 1, ..., d, d+2 (d es un número entero de 1 o más),

el dispositivo de generación de claves (100) que comprende:

una primera parte de entrada de información (130) que toma como entrada, un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) para al menos un número entero t = 1, ..., d;

una parte de generación de clave de firma (160) que genera un elemento k^*_t, un elemento k^*_{d+2} y un elemento k^*_{d+3} en base al conjunto de atributos Γ introducido por la primera parte de entrada de información, el elemento k^*_t que concierne a cada información de identificación t incluida en el conjunto de atributos Γ, la parte de generación de clave de firma (160) que está configurada

para generar el elemento k^*_t donde $x_{t,i'}$ multiplicado por el valor aleatorio δ se establece como coeficiente para un vector base $b^*_{t,i'}$ (i' = 1, ..., n_t) de una base B^*_t, para cada información de identificación t incluida en el conjunto de atributos Γ,

para generar el elemento k^*_{t+2} donde el valor aleatorio δ se establece como coeficiente para un vector base $b^*_{d+2,p'}$ (p' es un valor predeterminado) de una base B^*_{d+2}, y

para generar el elemento k^*_{d+3} donde el valor aleatorio δ se establece como coeficiente para un vector base $b^*_{d+2,q'}$ (q' es un valor predeterminado diferente de un p' prescrito) de la base B^*_{d+2}, y

una parte de distribución de claves (150) que distribuye datos que incluyen el conjunto de atributos Γ introducido por la parte de entrada de información (130) y los elementos k^*_t, k^*_{d+2} y k^*_{d+3}, como la clave de firma sk_Γ.

14. Un dispositivo de firma (400) que genera datos de firma sig en un sistema de procesamiento de firmas (20) que ejecuta un proceso de firma usando una base B_t y una base B^*_t para cada número entero t = 1, ..., d, d + 2 (d es un número entero de 1 o más), el dispositivo de firma (400) que comprende:

una segunda parte de entrada de información (420) que toma como entrada, una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), que es una cualquiera de una tupla positiva $(t, \vec{v}_i)$ y una tupla negativa $\neg(t, \vec{v}_i)$ de información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más), una matriz M predeterminada que tiene L filas y r columnas (r es un número entero de 1 o más), y un mensaje m;

una parte de adquisición de clave de firma (410) que adquiere un elemento k^*_t, un elemento k^*_{d+2} y un elemento k^*_{d+3} generado para al menos un número entero t = 1, ..., d, en base a un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) (t es cada

información de identificación incluida en el conjunto de atributos Γ), junto con el conjunto de atributos Γ, como una clave de firma sk_Γ, la parte de adquisición de clave de firma (410) que está configurada

para generar el elemento k^*_t que se establece con $x_{t,i'}$ multiplicado por el valor aleatorio δ como coeficiente para un vector base $b^*_{t,i'}$ ($i' = 1, ..., n_t$) de una base B^*_t, para cada información de identificación t incluida en el conjunto de atributos Γ,

para generar el elemento k^*_{d+2} que se establece con el valor aleatorio δ como coeficiente para un vector base $b^*_{d+2,p'}$ (p' es un valor predeterminado) de una base B^*_{d+2}, y

para generar el elemento k^*_{d+3} que se establece con el valor aleatorio δ como coeficiente para un vector base $b^*_{d+2,q'}$ (q' es un valor predeterminado diferente de un p' prescrito) de la base B^*_{d+2};

una parte de cálculo de coeficiente complementario (430) que, en base a la variable ρ(i) para cada número entero i = 1, ..., L y el conjunto de atributos Γ incluido en la clave de firma sk_Γ adquirida por la parte de adquisición de clave de firma (410), especifica, entre los números enteros i = 1, ..., L, un conjunto I de un número entero i para el cual la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$) y con el cual un producto interno de $\vec{v}_i$ de la tupla positiva y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$) y con el cual un producto interno de $\vec{v}_i$ de la tupla negativa y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0; y calcula un coeficiente complementario α_i con el cual un total de $\alpha_i M_i$ para i incluido en el conjunto I especificado, en base a M_i que es un elemento en una fila de orden i de la matriz M introducida por la segunda parte de entrada de información (420), llega a ser un vector $\vec{w}$ predeterminado;

una parte de generación de firma (450) que, en base a la variable ρ(i), el conjunto de atributos Γ, el conjunto I especificado por la parte de cálculo de coeficiente complementario (430), y el coeficiente complementario α_i calculado por la parte de cálculo de coeficiente complementario (430), genera s^*_i para cada número entero i que representa cada número de fila de la matriz M, y s^*_x, como se indica en la Fórmula 20, donde

β_i es un valor con el que cada elemento llega a ser 0 cuando $\beta_i M_i$ se suma para cada número entero i que representa cada número de fila de la matriz M, y

para cada número entero i que representa cada número de fila de la matriz M,

cuando está establecido $i \in I$ y la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$), permitamos un valor $\gamma_i := \alpha_i$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 0 e $y_{i,1} = 1$,

cuando está establecido $i \in I$ y la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$), permitamos un valor $\gamma_i := \alpha_i/(\vec{v}_i \cdot \vec{x}_t)$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 1,

cuando no está establecido $i \in I$ y la variable ρ(i) es una tupla positiva (t, $\vec{v}_i$), permitamos un valor $\gamma_i := 0$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 0 e $y_{i,1} = 1$, y

cuando no está establecido $i \in I$ y la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$), permitamos un valor $\gamma_i := 0$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ ($\tau := 1, ..., n_t$) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 1, y

una parte de salida de datos (460) que emite datos que incluyen la variable ρ(i), la matriz M y el mensaje m que se introducen por la parte de entrada de información (420), y s^*_0, s^*_i y s^*_x generados por la parte de generación de firma (450), como los datos de firma sig.

[Fórmula 20]

$$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot (\sum_{l=1}^{n_t} y_{i,l} \cdot b_{t,l}^*) + r_i^*$$

$$s_X^* := \xi(k_{d+2}^* + m \cdot k_{d+3}^*) + r_X^*$$

donde

ξ, r^*_i y r^*_X son valores predeterminados.

15. Un dispositivo de verificación (500) que verifica los datos de firma sig en un sistema de procesamiento de firmas (20) que ejecuta un proceso de firma usando una base B_t y una base B^*_t para cada número entero t = 1, ..., d, d+2 (d es un número entero de 1 o más), el dispositivo de verificación (500) que comprende:

una parte de adquisición de datos (520) que adquiere los datos de firma sig que incluyen s^*_i, s^*_x, un mensaje m, un número variable ρ(i) y una matriz M, de entre: los elementos k^*_t, k^*_{d+2}, k^*_{d+3}; un conjunto I y un coeficiente complementario α_i; y S^*_i y S^*_x;

en donde los elementos k^*_t, k^*_{d+2} y k^*_{d+3} se generan para al menos un número entero t = 1, ..., d, en base a un conjunto de atributos Γ que tiene información de identificación t y un vector de atributo $\vec{x}_t := (x_{t,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más) (t es cada información de identificación incluida en el conjunto de atributos Γ),

el elemento k^*_t que se establece con $x_{t,i'}$ multiplicado por el valor aleatorio δ, como coeficiente para un vector base $b^*_{t,i'}$ (i' = 1, ..., n_t) de una base B^*_t, para cada información de identificación t incluida en el conjunto de atributos Γ,

el elemento k^*_{d+2} que se establece con el valor aleatorio δ, como coeficiente para un vector base $b^*_{d+2,p'}$ (p' es un valor predeterminado) de una base B^*_{d+2}, y

el elemento k^*_{d+3} que se establece con el valor aleatorio δ, como coeficiente para un vector base $b^*_{d+2,q'}$ (q' es un valor predeterminado diferente de un p' prescrito) de la base B^*_{d+2},

en donde el conjunto I y el coeficiente complementario α_i se calculan en base a una variable ρ(i) para cada número entero i = 1, ..., L (L es un número entero de 1 o más), que es una cualquiera de una tupla positiva $(t, \vec{v}_i)$ y una tupla negativa $\neg(t, \vec{v}_i)$ de información de identificación t (t es cualquier número entero de t = 1, ..., d) y un vector de atributo $\vec{v}_i := (v_{i,i'})$ (i' = 1, ..., n_t donde n_t es un número entero de 1 o más), y en base al conjunto de atributos Γ,

el conjunto I que es un conjunto de, entre los números enteros i = 1, ..., L, un número entero i para el cual la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$ y con el cual un producto interno de $\vec{v}_i$ de la tupla positiva y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla positiva llega a ser 0, y un número entero i para el cual la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$ y con el cual un producto interno de $\vec{v}_i$ de la tupla negativa y $\vec{x}_t$ incluido en Γ indicado por la información de identificación t de la tupla negativa no llega a ser 0,

el coeficiente complementario α_i que es de manera que con el cual un total de $\alpha_i M_i$ para i incluido en el conjunto I, en base a M_i que es un elemento en una fila de orden i de la matriz M introducida, llega a ser un vector $\vec{w}$ predeterminado, y

en donde s^*_i (cada número entero de i = 1, ..., L) y s^*_x se generan en base a la variable ρ(i), el conjunto de atributos Γ, el conjunto I y el coeficiente complementario α_i, como se indica en la Fórmula 21, donde

β_i es un valor con el que cada elemento llega a ser 0 cuando $\beta_i M_i$ se suma para cada número entero i que representa cada número de fila de la matriz M, y

para cada número entero i que representa cada número de fila de la matriz M,

cuando está establecido $i \in I$ y la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, permitamos un valor $\gamma_i := \alpha_i$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ (τ := 1, ..., n_t) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 0 e $y_{i,1} = 1$,

cuando está establecido $i \in I$ y la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, permitamos un valor $\gamma_i := \alpha_i/(\vec{v}_i \cdot \vec{x}_t)$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ (τ := 1, ..., n_t) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 1,

cuando no está establecido $i \in I$ y la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, permitamos un valor $\gamma_i := 0$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ (τ := 1, ..., n_t) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 0 e $y_{i,1} = 1$, y

cuando no está establecido $i \in I$ y la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, permitamos un valor $\gamma_i := 0$, y permitamos que un vector $\vec{y}_i := (y_{i,\tau})$ (τ := 1, ..., n_t) sea un vector con el cual un producto interno de $\vec{y}_i$ y $\vec{v}_i$ es 1;

una parte de generación de datos cifrados (530) que genera un elemento c_i y un elemento c_x, el elemento c_i que se genera para cada número entero i = 1, ..., L, en base a un vector de columna $\vec{s}^T := (s_1, ..., s_L)^T := M \cdot \vec{f}^T$ generado en base a un vector $\vec{f}$ que tiene r partes de elementos, el vector $\vec{w}$ y la matriz M introducida por la parte de adquisición de datos (520); un valor $s_0 := \vec{w} \cdot \vec{f}$; y un valor predeterminado θ_i (i es cada número de fila de la matriz M, y X), la parte de generación de datos cifrados (530) que está configurada

para generar el elemento c_i, para cada número entero i = 1 que representa un número de fila de la matriz M, cuando la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, estableciendo $s_i+\theta_i v_{i,1}$ como coeficiente para un vector base $b_{t,1}$ de la base B_t indicada por la información de identificación t de la tupla positiva, y estableciendo $\theta_i v_{i,i'}$ como coeficiente para un vector base $b_{t,i'}$ indicado por la información de identificación t y cada número entero i' = 2, ..., n_t, y cuando la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, estableciendo $s_i v_{i,i'}$ como coeficiente para el vector base $b_{t,i'}$ indicado por la información de identificación t de la tupla negativa y cada número entero i' = 1, ..., n_t, y

para generar el elemento c_x estableciendo $s_0-\theta_x m$ en un vector base $b_{d+2,p'}$ (p' es una p' prescrita) de un vector base B_{d+2} y estableciendo θ_x en un vector base $b_{d+2,q'}$ (q' es una q' prescrita) del vector base B_{d+2}; y

una parte de operación de emparejamiento (540) que verifica la validez de los datos de firma sig realizando una operación de emparejamiento indicada en la Fórmula 22 para cada número entero i que indica X y un número de fila de la matriz M.

[Fórmula 21]

$$s_i^* := \gamma_i \cdot \xi k_t^* + \beta_i \cdot (\sum_{l=1}^{n_t} y_{i,l} \cdot b_{t,l}^*) + r_i^*$$

$$s_X^* := \xi(k_{d+2}^* + m \cdot k_{d+3}^*) + r_X^*$$

donde

ξ, r^*_i y r^*_X son valores predeterminados.

[Fórmula 22]

$$\prod e(c_i, s_i^*)$$

Fig. 1

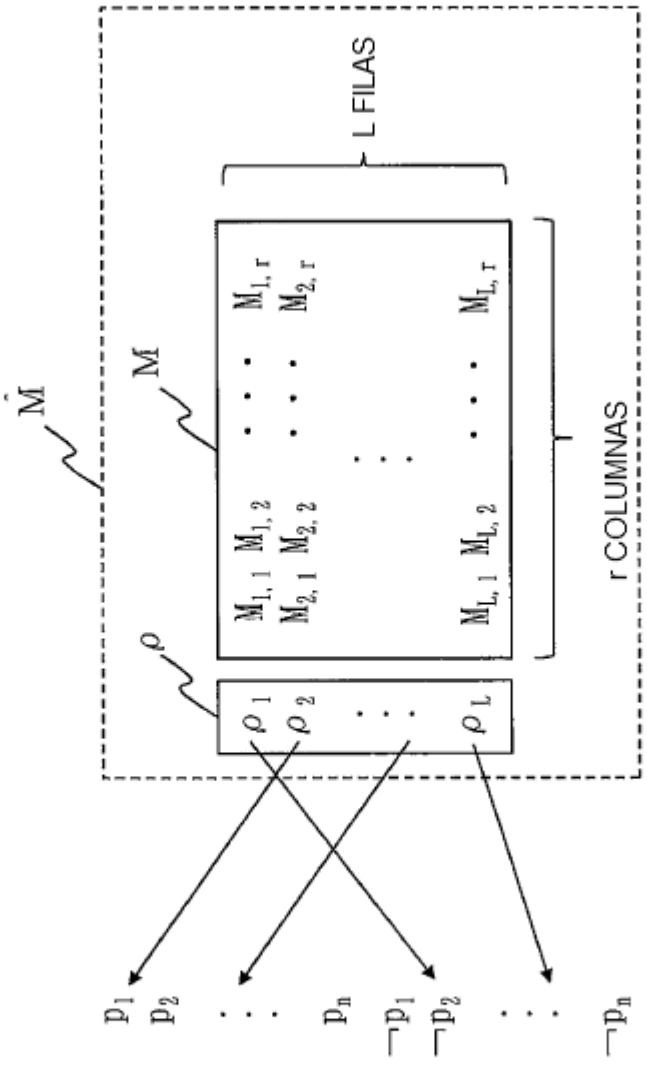

M̃
M
L FILAS
r COLUMNAS
ρ
ρ 1
ρ 2
ρ L
M1, 1 M1, 2 M1, r
M2, 1 M2, 2 M2, r
ML, 1 ML, 2 ML, r
p1
p2
pn
¬p1
¬p2
¬pn

Fig. 2

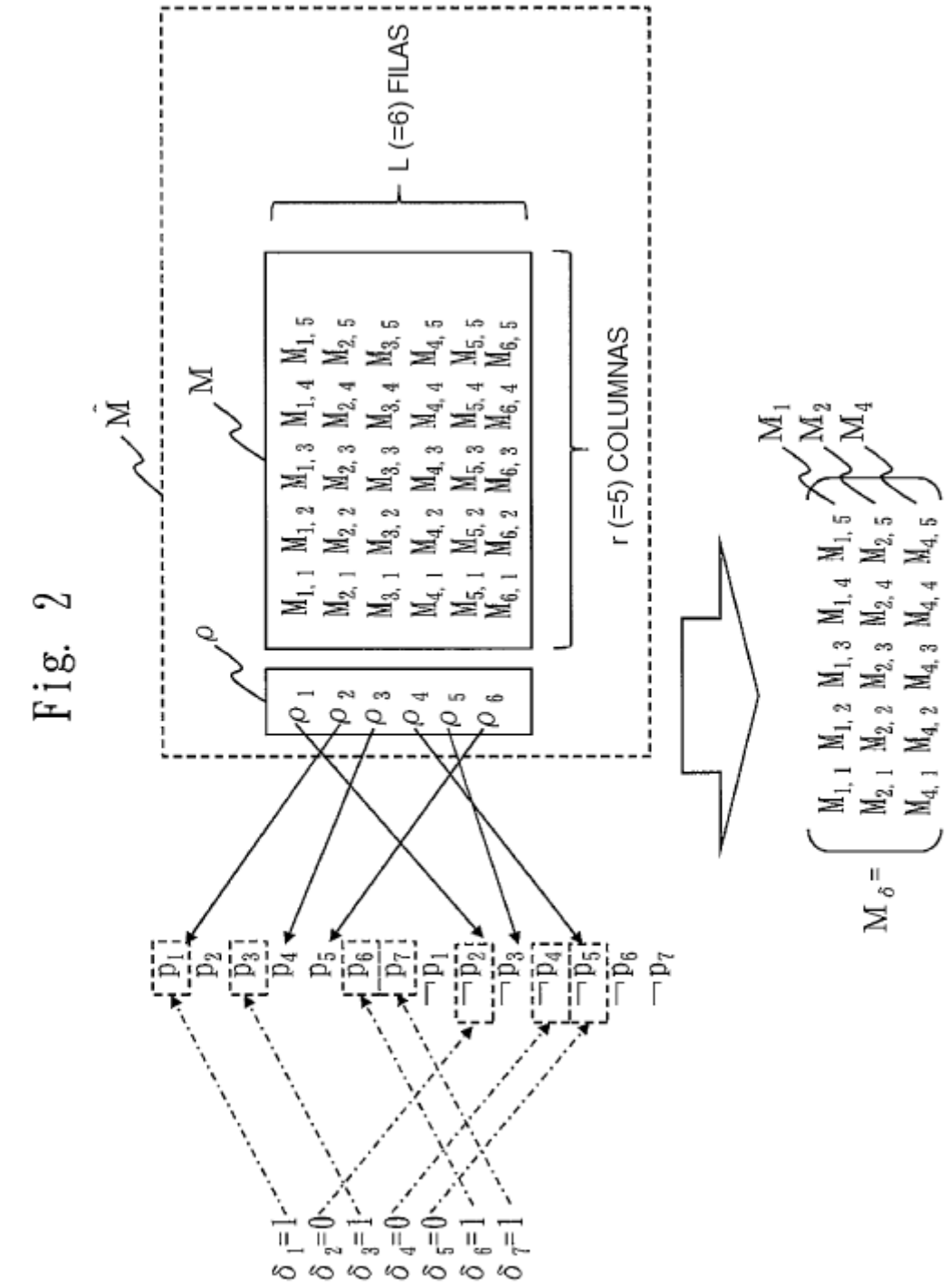


Fig. 3

$$s_0 = \begin{bmatrix} \overbrace{1, \ldots, 1}^{\text{r COLUMNAS}} \end{bmatrix} \begin{pmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_r \end{pmatrix} = \sum_{k=1}^{r} f_k$$

Fig. 4


$\vec{s} = \vec{T}$
$\begin{pmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,r} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,r} \\ & & \vdots & \\ M_{L,1} & M_{L,2} & \cdots & M_{L,r} \end{pmatrix}$
$\begin{pmatrix} f_1 \\ \cdot \\ \cdot \\ \cdot \\ f_r \end{pmatrix}$
$=$
$\begin{pmatrix} s_1 \\ \cdot \\ \cdot \\ \cdot \\ s_r \end{pmatrix}$

Fig. 5

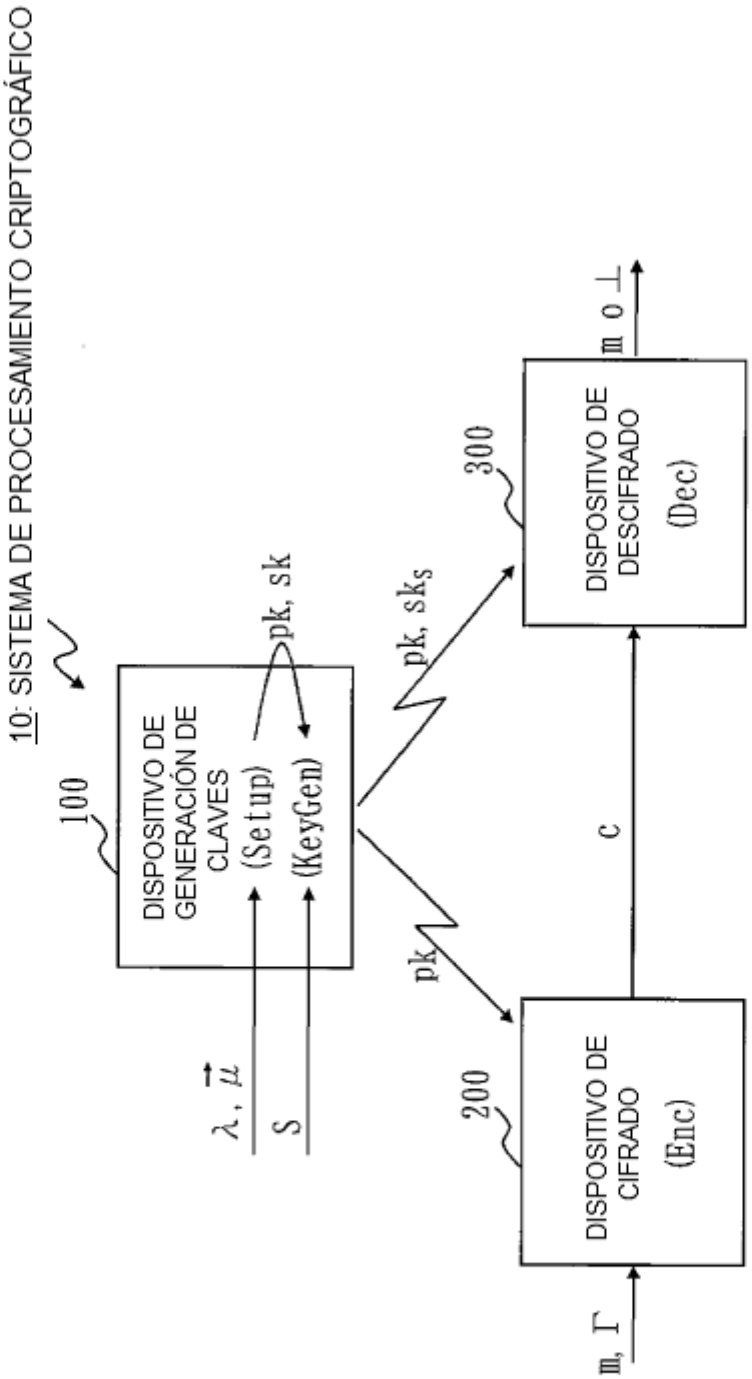

10: SISTEMA DE PROCESAMIENTO CRIPTOGRÁFICO
100
DISPOSITIVO DE GENERACIÓN DE CLAVES
(Setup)
(KeyGen)
pk, sk
λ, μ
S
pk
pk, sk_S
200
DISPOSITIVO DE CIFRADO
(Enc)
c
300
DISPOSITIVO DE DESCIFRADO
(Dec)
m o ⊥
m, Γ

Fig. 6

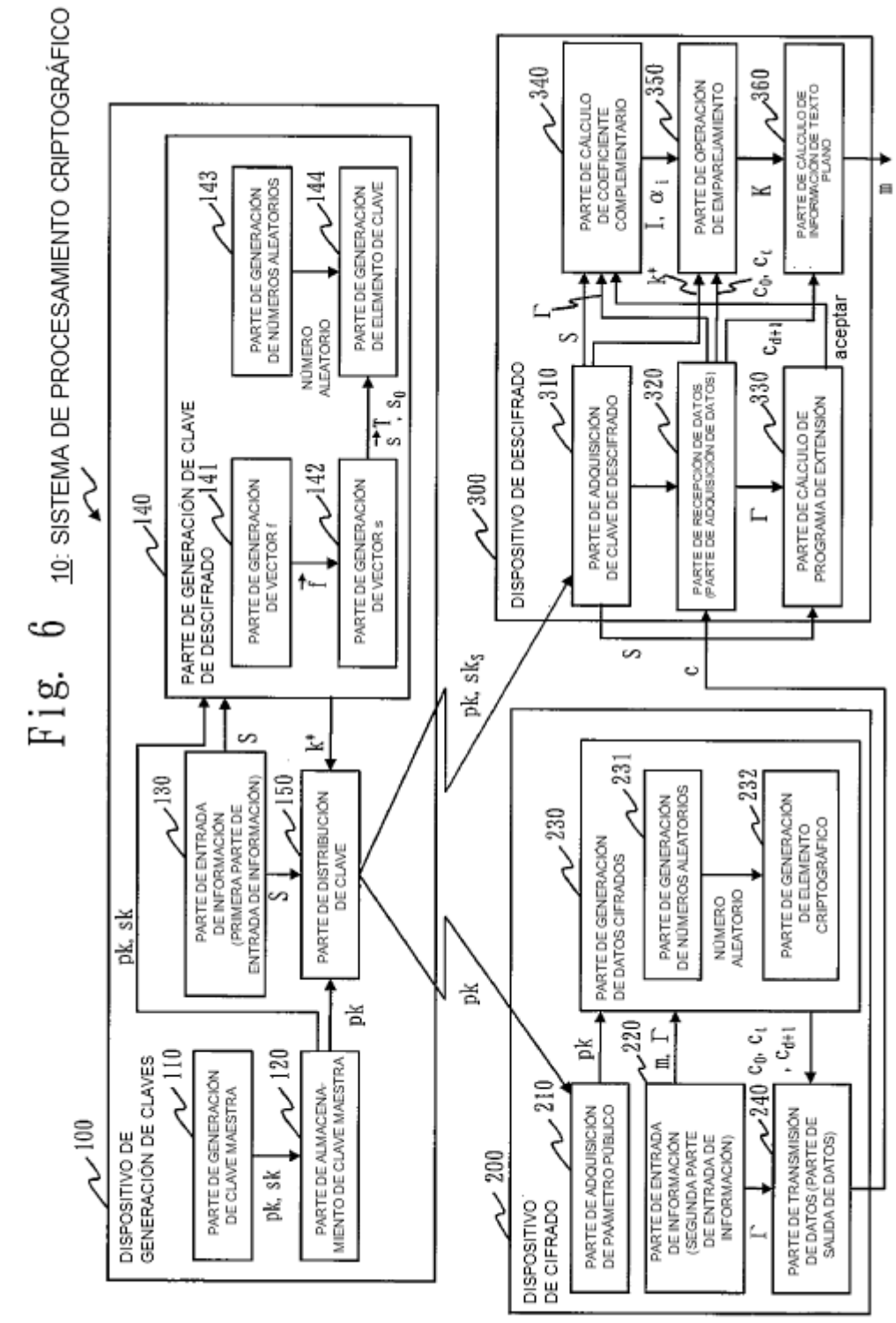

10: SISTEMA DE PROCESAMIENTO CRIPTOGRÁFICO
100
DISPOSITIVO DE GENERACIÓN DE CLAVES
110
PARTE DE GENERACIÓN DE CLAVE MAESTRA
120
PARTE DE ALMACENAMIENTO DE CLAVE MAESTRA
pk, sk
pk
130
PARTE DE ENTRADA DE INFORMACIÓN (PRIMERA PARTE DE ENTRADA DE INFORMACIÓN)
S
150
PARTE DE DISTRIBUCIÓN DE CLAVE
k*
140
PARTE DE GENERACIÓN DE CLAVE DE DESCIFRADO
141
PARTE DE GENERACIÓN DE VECTOR f
142
PARTE DE GENERACIÓN DE VECTOR s
143
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
NÚMERO ALEATORIO
144
PARTE DE GENERACIÓN DE ELEMENTO DE CLAVE
s*T, s0
pk, sks
300
DISPOSITIVO DE DESCIFRADO
310
PARTE DE ADQUISICIÓN DE CLAVE DE DESCIFRADO
320
PARTE DE RECEPCIÓN DE DATOS (PARTE DE ADQUISICIÓN DE DATOS)
330
PARTE DE CÁLCULO DE PROGRAMA DE EXTENSIÓN
340
PARTE DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
I, αi
350
PARTE DE OPERACIÓN DE EMPAREJAMIENTO
K
360
PARTE DE CÁLCULO DE INFORMACIÓN DE TEXTO PLANO
m
Γ
S
c
k*
cd+1
aceptar
200
DISPOSITIVO DE CIFRADO
210
PARTE DE ADQUISICIÓN DE PARÁMETRO PÚBLICO
220
PARTE DE ENTRADA DE INFORMACIÓN (SEGUNDA PARTE DE ENTRADA DE INFORMACIÓN)
m, Γ
230
PARTE DE GENERACIÓN DE DATOS CIFRADOS
231
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
NÚMERO ALEATORIO
232
PARTE DE GENERACIÓN DE ELEMENTO CRIPTOGRÁFICO
240
PARTE DE TRANSMISIÓN DE DATOS (PARTE DE SALIDA DE DATOS)
c0, ci, cd+1

Fig. 7

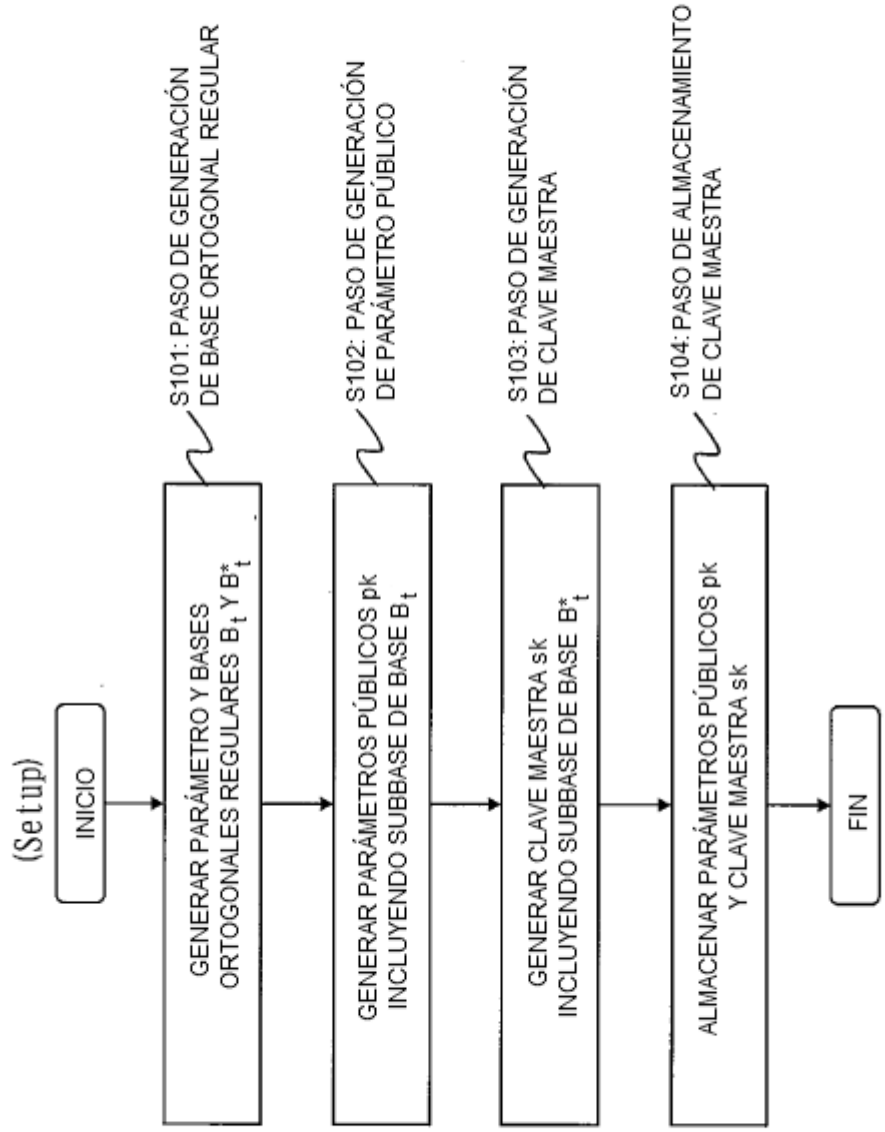

(Setup)
INICIO
GENERAR PARÁMETRO Y BASES ORTOGONALES REGULARES B_t Y B^*_t
S101: PASO DE GENERACIÓN DE BASE ORTOGONAL REGULAR
GENERAR PARÁMETROS PÚBLICOS pk INCLUYENDO SUBBASE DE BASE B_t
S102: PASO DE GENERACIÓN DE PARÁMETRO PÚBLICO
GENERAR CLAVE MAESTRA sk INCLUYENDO SUBBASE DE BASE B^*_t
S103: PASO DE GENERACIÓN DE CLAVE MAESTRA
ALMACENAR PARÁMETROS PÚBLICOS pk Y CLAVE MAESTRA sk
S104: PASO DE ALMACENAMIENTO DE CLAVE MAESTRA
FIN

Fig. 8

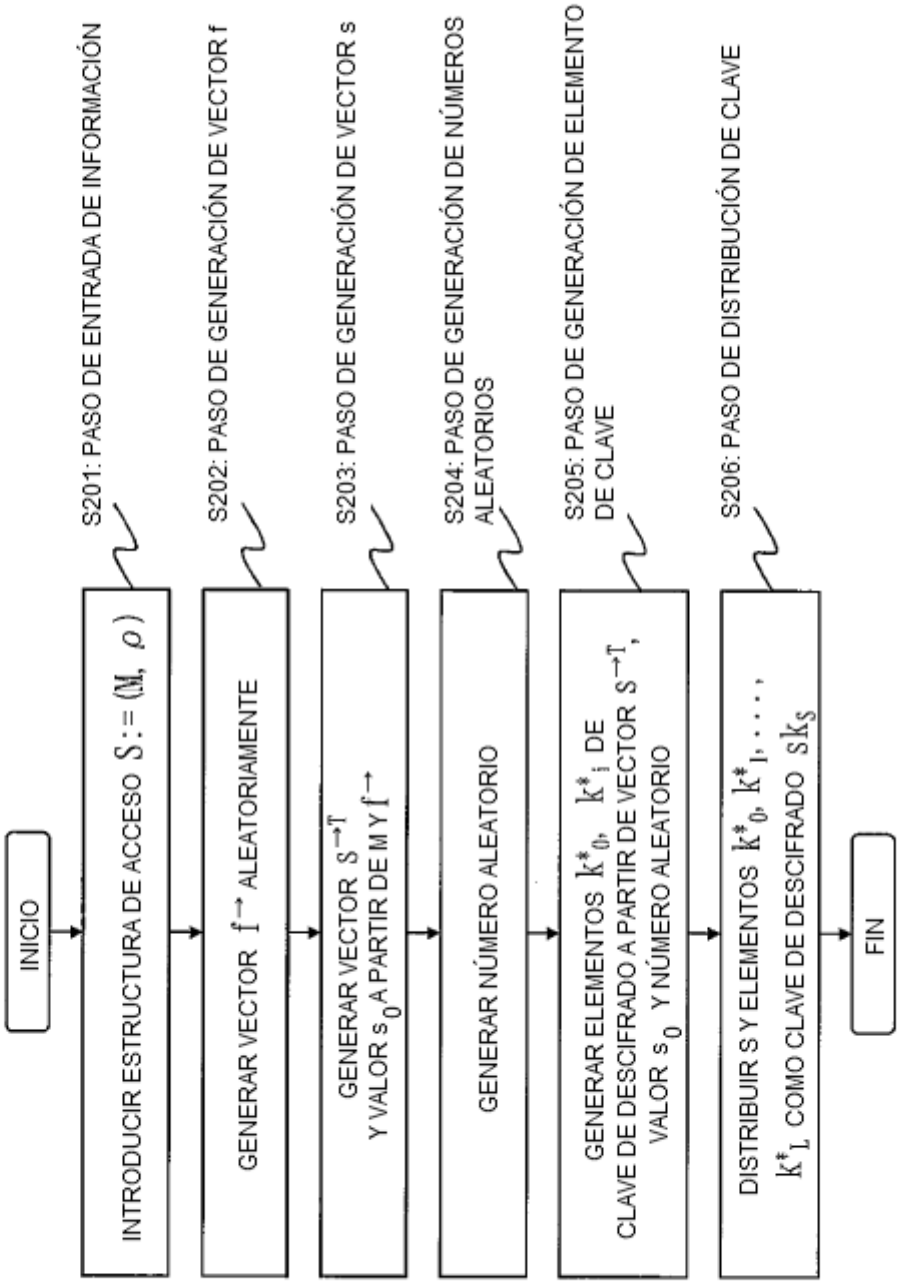

(KeyGen)
INICIO
S201: PASO DE ENTRADA DE INFORMACIÓN
INTRODUCIR ESTRUCTURA DE ACCESO S := (M, ρ)
S202: PASO DE GENERACIÓN DE VECTOR f
GENERAR VECTOR f→ ALEATORIAMENTE
S203: PASO DE GENERACIÓN DE VECTOR s
GENERAR VECTOR s→T
Y VALOR s0 A PARTIR DE M Y f→
S204: PASO DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMERO ALEATORIO
S205: PASO DE GENERACIÓN DE ELEMENTO DE CLAVE
GENERAR ELEMENTOS k*0, k*i DE
CLAVE DE DESCIFRADO A PARTIR DE VECTOR s→T,
VALOR s0 Y NÚMERO ALEATORIO
S206: PASO DE DISTRIBUCIÓN DE CLAVE
DISTRIBUIR S Y ELEMENTOS k*0, k*1, . . . ,
k*L COMO CLAVE DE DESCIFRADO sks
FIN

Fig. 9

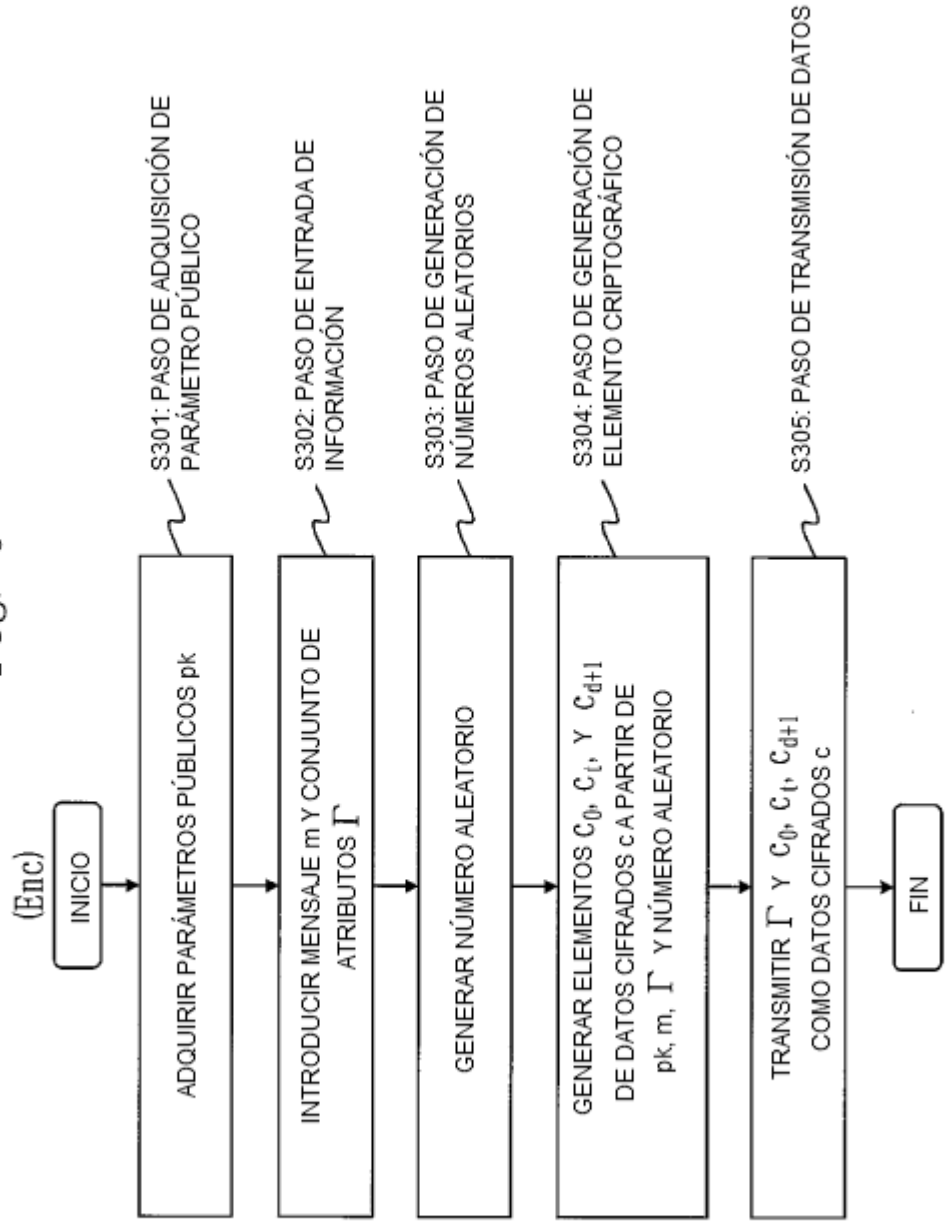

(Enc)
INICIO
ADQUIRIR PARÁMETROS PÚBLICOS pk
S301: PASO DE ADQUISICIÓN DE PARÁMETRO PÚBLICO
INTRODUCIR MENSAJE m Y CONJUNTO DE ATRIBUTOS Γ
S302: PASO DE ENTRADA DE INFORMACIÓN
GENERAR NÚMERO ALEATORIO
S303: PASO DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR ELEMENTOS C_0, C_t, Y C_{d+1} DE DATOS CIFRADOS c A PARTIR DE pk, m, Γ Y NÚMERO ALEATORIO
S304: PASO DE GENERACIÓN DE ELEMENTO CRIPTOGRÁFICO
TRANSMITIR Γ Y C_0, C_t, C_{d+1} COMO DATOS CIFRADOS c
S305: PASO DE TRANSMISIÓN DE DATOS
FIN

Fig. 10

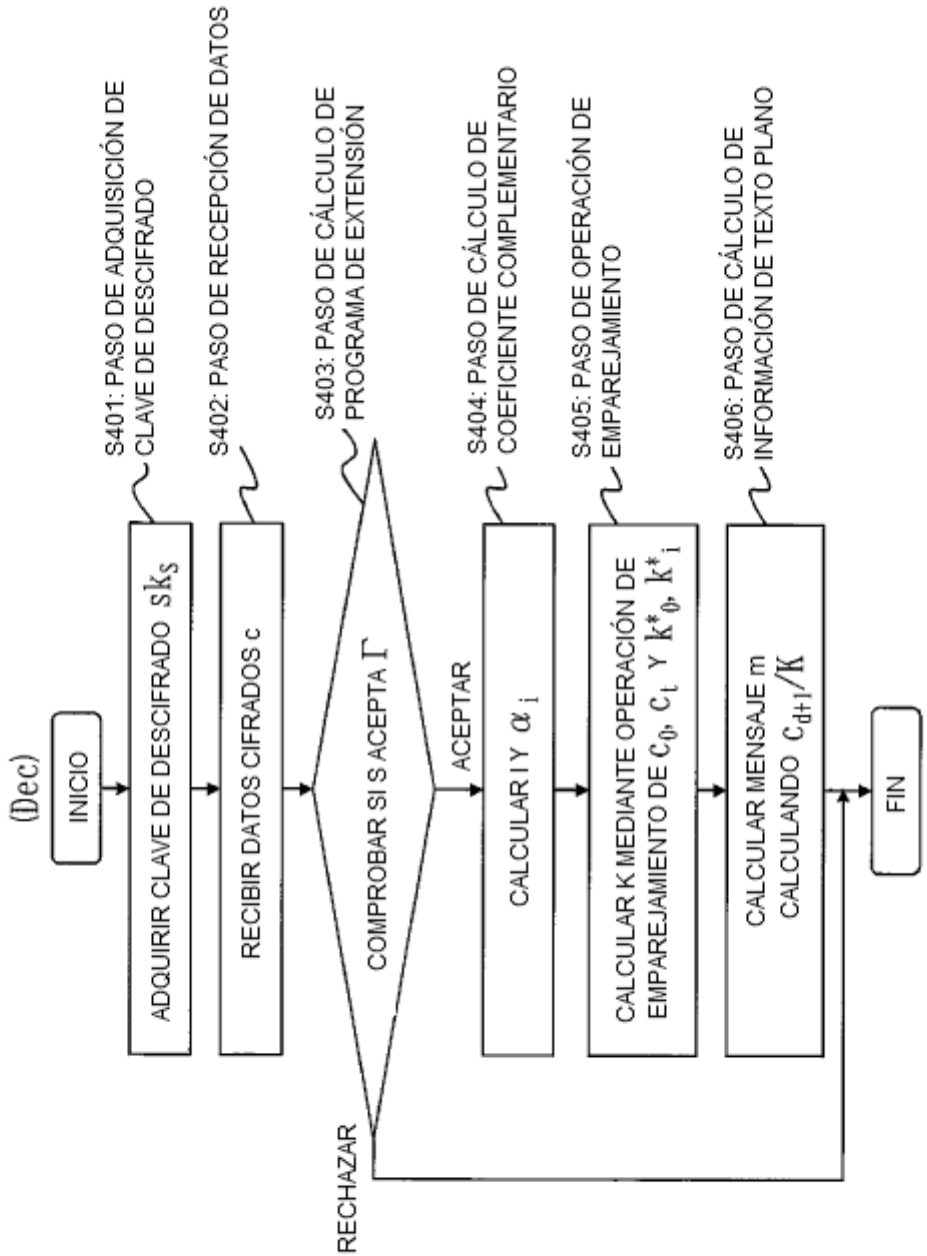

(Dec)
INICIO
ADQUIRIR CLAVE DE DESCIFRADO sk_S
S401: PASO DE ADQUISICIÓN DE CLAVE DE DESCIFRADO
RECIBIR DATOS CIFRADOS c
S402: PASO DE RECEPCIÓN DE DATOS
COMPROBAR SI S ACEPTA Γ
S403: PASO DE CÁLCULO DE PROGRAMA DE EXTENSIÓN
RECHAZAR
ACEPTAR
CALCULAR I Y α_i
S404: PASO DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
CALCULAR K MEDIANTE OPERACIÓN DE EMPAREJAMIENTO DE c_0, c_I Y k^*_ϕ, k^*_i
S405: PASO DE OPERACIÓN DE EMPAREJAMIENTO
CALCULAR MENSAJE m CALCULANDO c_{d+1}/K
S406: PASO DE CÁLCULO DE INFORMACIÓN DE TEXTO PLANO
FIN

Fig. 11

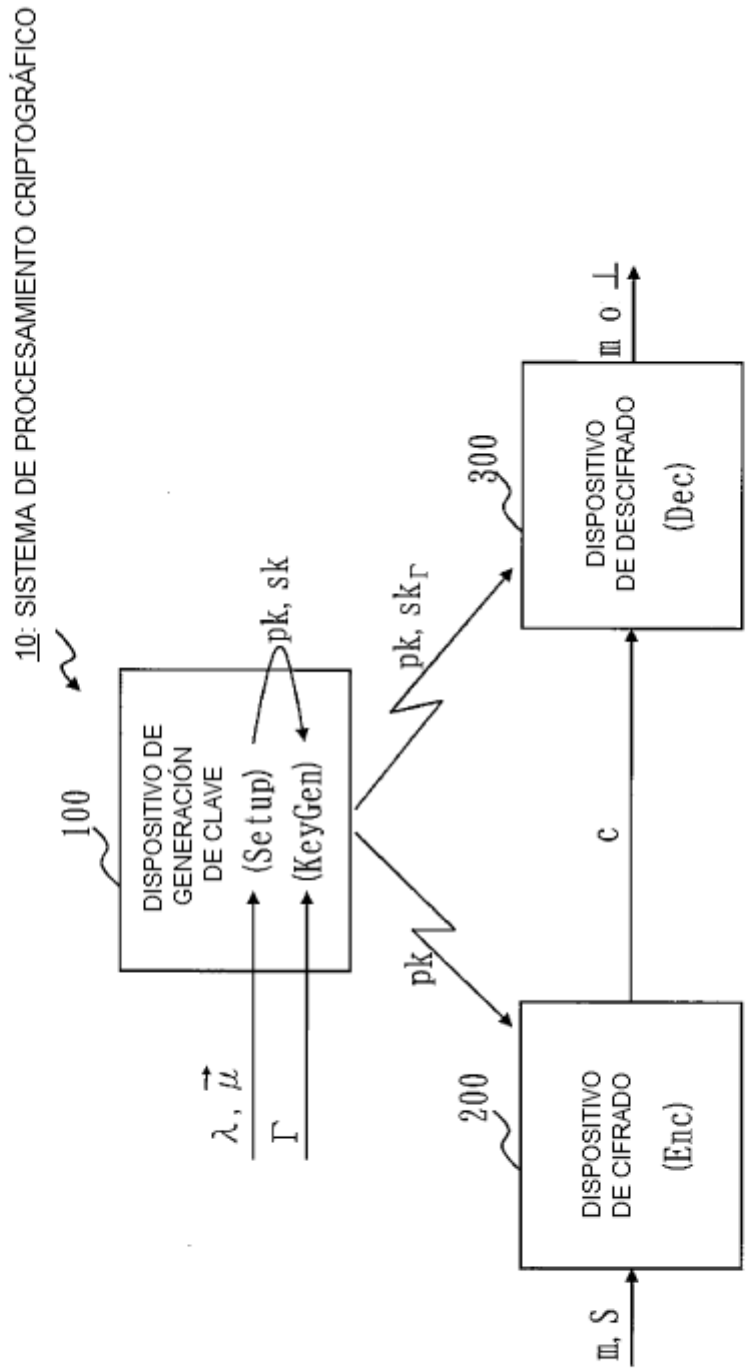

10: SISTEMA DE PROCESAMIENTO CRIPTOGRÁFICO
100
DISPOSITIVO DE GENERACIÓN DE CLAVE
(Setup)
(KeyGen)
pk, sk
λ, μ
Γ
pk, skΓ
pk
300
DISPOSITIVO DE DESCIFRADO
(Dec)
m o ⊥
c
200
DISPOSITIVO DE CIFRADO
(Enc)
m, S

Fig. 12

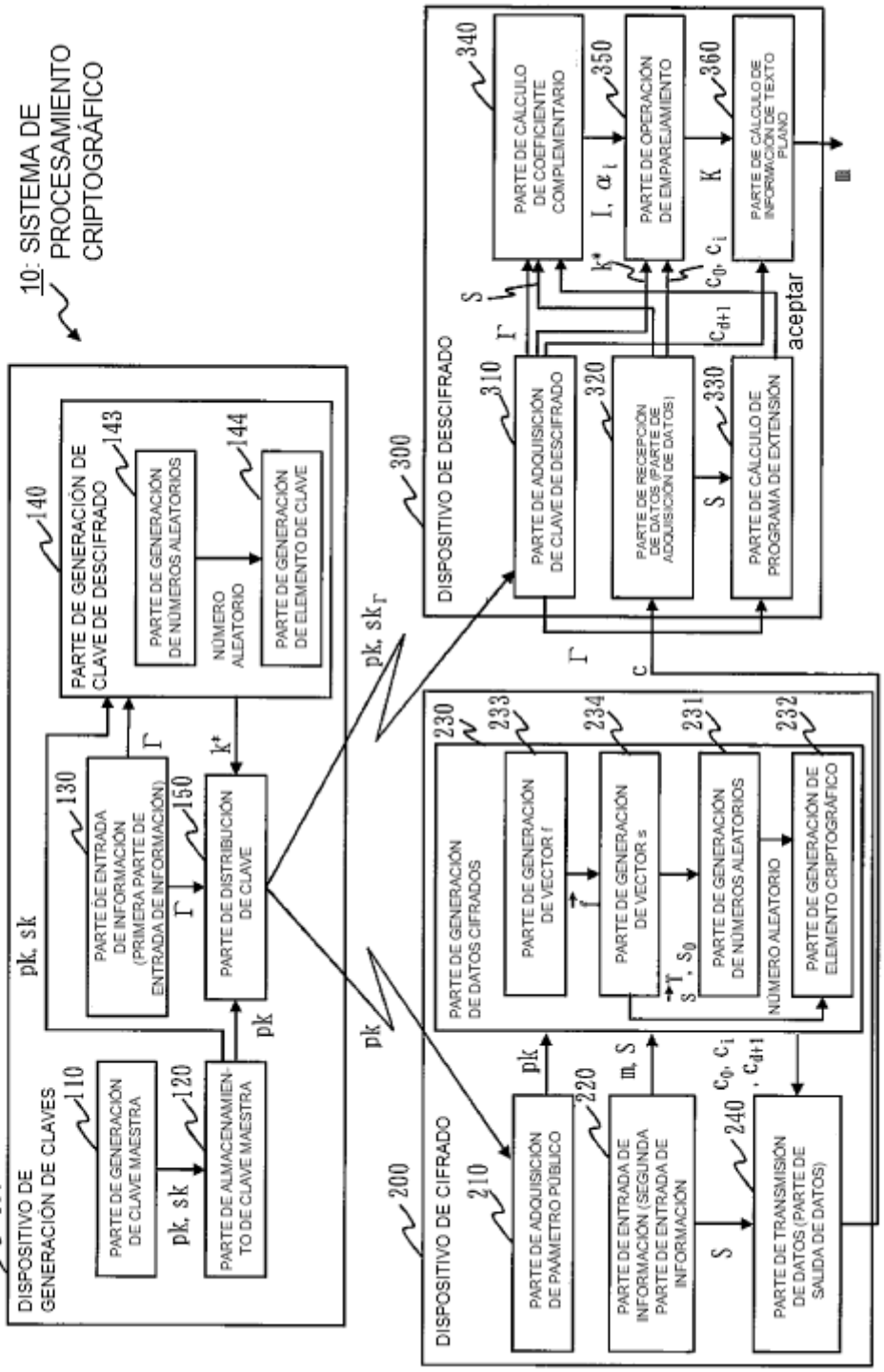

10: SISTEMA DE PROCESAMIENTO CRIPTOGRÁFICO
100
DISPOSITIVO DE GENERACIÓN DE CLAVES
110
PARTE DE GENERACIÓN DE CLAVE MAESTRA
120
PARTE DE ALMACENAMIENTO DE CLAVE MAESTRA
pk, sk
130
PARTE DE ENTRADA DE INFORMACIÓN (PRIMERA PARTE DE ENTRADA DE INFORMACIÓN)
140
PARTE DE GENERACIÓN DE CLAVE DE DESCIFRADO
143
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
144
PARTE DE GENERACIÓN DE ELEMENTO DE CLAVE
NÚMERO ALEATORIO
150
PARTE DE DISTRIBUCIÓN DE CLAVE
pk
pk, sk_Γ
200
DISPOSITIVO DE CIFRADO
210
PARTE DE ADQUISICIÓN DE PARÁMETRO PÚBLICO
220
PARTE DE ENTRADA DE INFORMACIÓN (SEGUNDA PARTE DE ENTRADA DE INFORMACIÓN)
230
PARTE DE GENERACIÓN DE DATOS CIFRADOS
233
PARTE DE GENERACIÓN DE VECTOR r
234
PARTE DE GENERACIÓN DE VECTOR s
231
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
NÚMERO ALEATORIO
232
PARTE DE GENERACIÓN DE ELEMENTO CRIPTOGRÁFICO
240
PARTE DE TRANSMISIÓN DE DATOS (PARTE DE SALIDA DE DATOS)
300
DISPOSITIVO DE DESCIFRADO
310
PARTE DE ADQUISICIÓN DE CLAVE DE DESCIFRADO
320
PARTE DE RECEPCIÓN DE DATOS (PARTE DE ADQUISICIÓN DE DATOS)
330
PARTE DE CÁLCULO DE PROGRAMA DE EXTENSIÓN
340
PARTE DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
350
PARTE DE OPERACIÓN DE EMPAREJAMIENTO
360
PARTE DE CÁLCULO DE INFORMACIÓN DE TEXTO PLANO
aceptar

Fig. 13

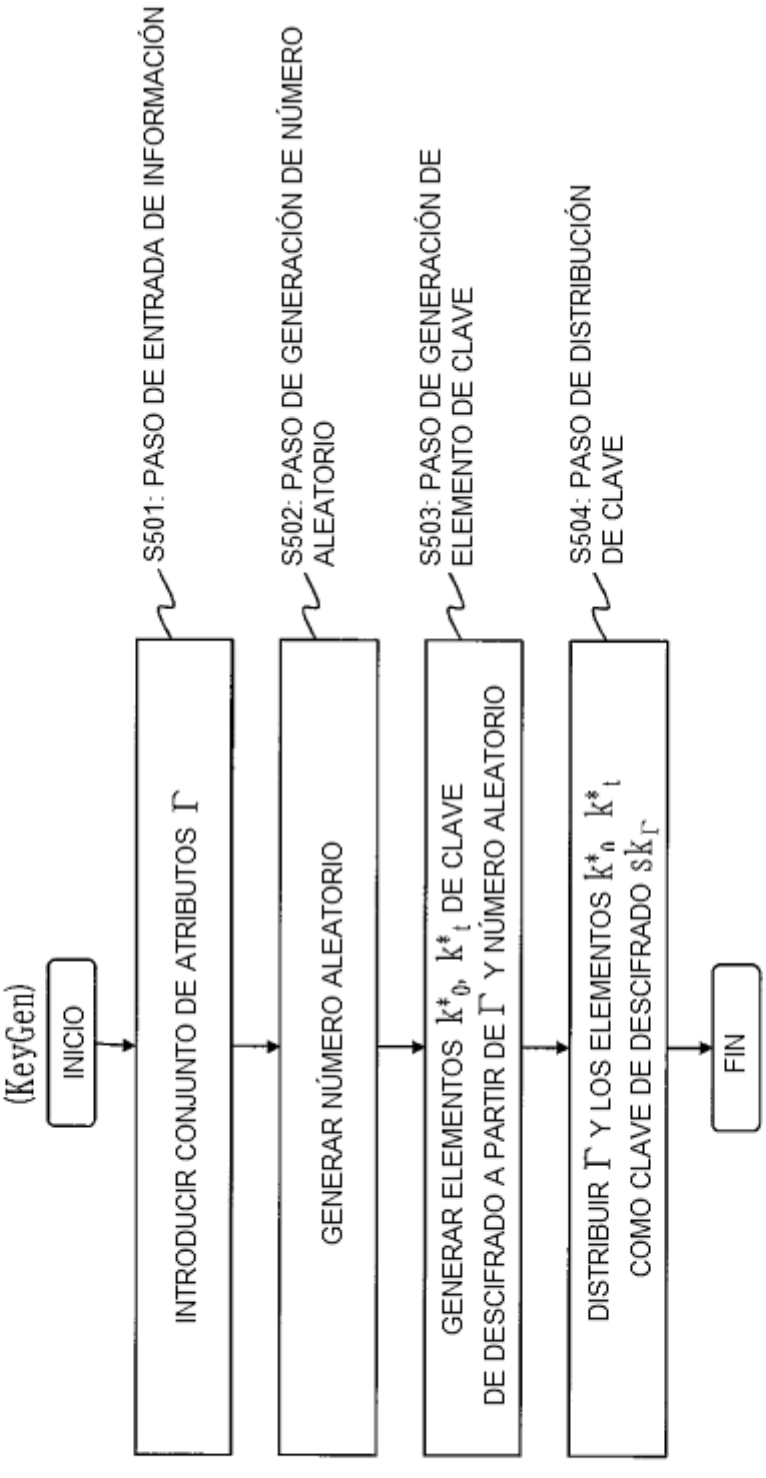

(KeyGen)
INICIO
S501: PASO DE ENTRADA DE INFORMACIÓN
INTRODUCIR CONJUNTO DE ATRIBUTOS Γ
S502: PASO DE GENERACIÓN DE NÚMERO ALEATORIO
GENERAR NÚMERO ALEATORIO
S503: PASO DE GENERACIÓN DE ELEMENTO DE CLAVE
GENERAR ELEMENTOS k^*_0, k^*_t DE CLAVE DE DESCIFRADO A PARTIR DE Γ Y NÚMERO ALEATORIO
S504: PASO DE DISTRIBUCIÓN DE CLAVE
DISTRIBUIR Γ Y LOS ELEMENTOS k^*_n k^*_t COMO CLAVE DE DESCIFRADO sk_Γ
FIN

Fig. 14

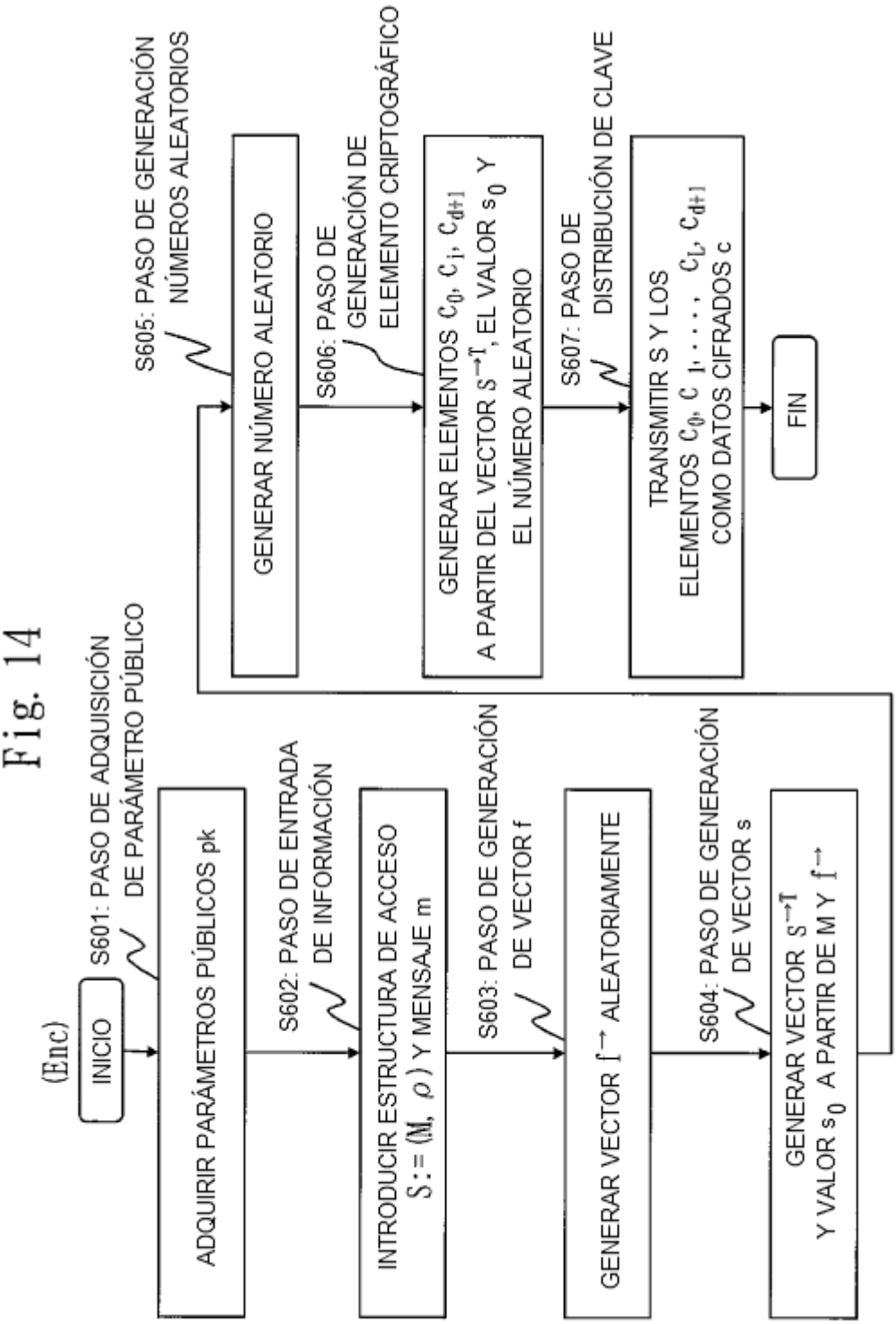

(Enc)
INICIO
S601: PASO DE ADQUISICIÓN DE PARÁMETRO PÚBLICO
ADQUIRIR PARÁMETROS PÚBLICOS pk
S602: PASO DE ENTRADA DE INFORMACIÓN
INTRODUCIR ESTRUCTURA DE ACCESO S := (M, ρ) Y MENSAJE m
S603: PASO DE GENERACIÓN DE VECTOR f
GENERAR VECTOR f→ ALEATORIAMENTE
S604: PASO DE GENERACIÓN DE VECTOR s
GENERAR VECTOR s→T Y VALOR s0 A PARTIR DE M Y f→
S605: PASO DE GENERACIÓN NÚMEROS ALEATORIOS
GENERAR NÚMERO ALEATORIO
S606: PASO DE GENERACIÓN DE ELEMENTO CRIPTOGRÁFICO
GENERAR ELEMENTOS c0, ci, cd+1 A PARTIR DEL VECTOR s→T, EL VALOR s0 Y EL NÚMERO ALEATORIO
S607: PASO DE DISTRIBUCIÓN DE CLAVE
TRANSMITIR S Y LOS ELEMENTOS c0, c1, ..., cL, cd+1 COMO DATOS CIFRADOS c
FIN

Fig. 15

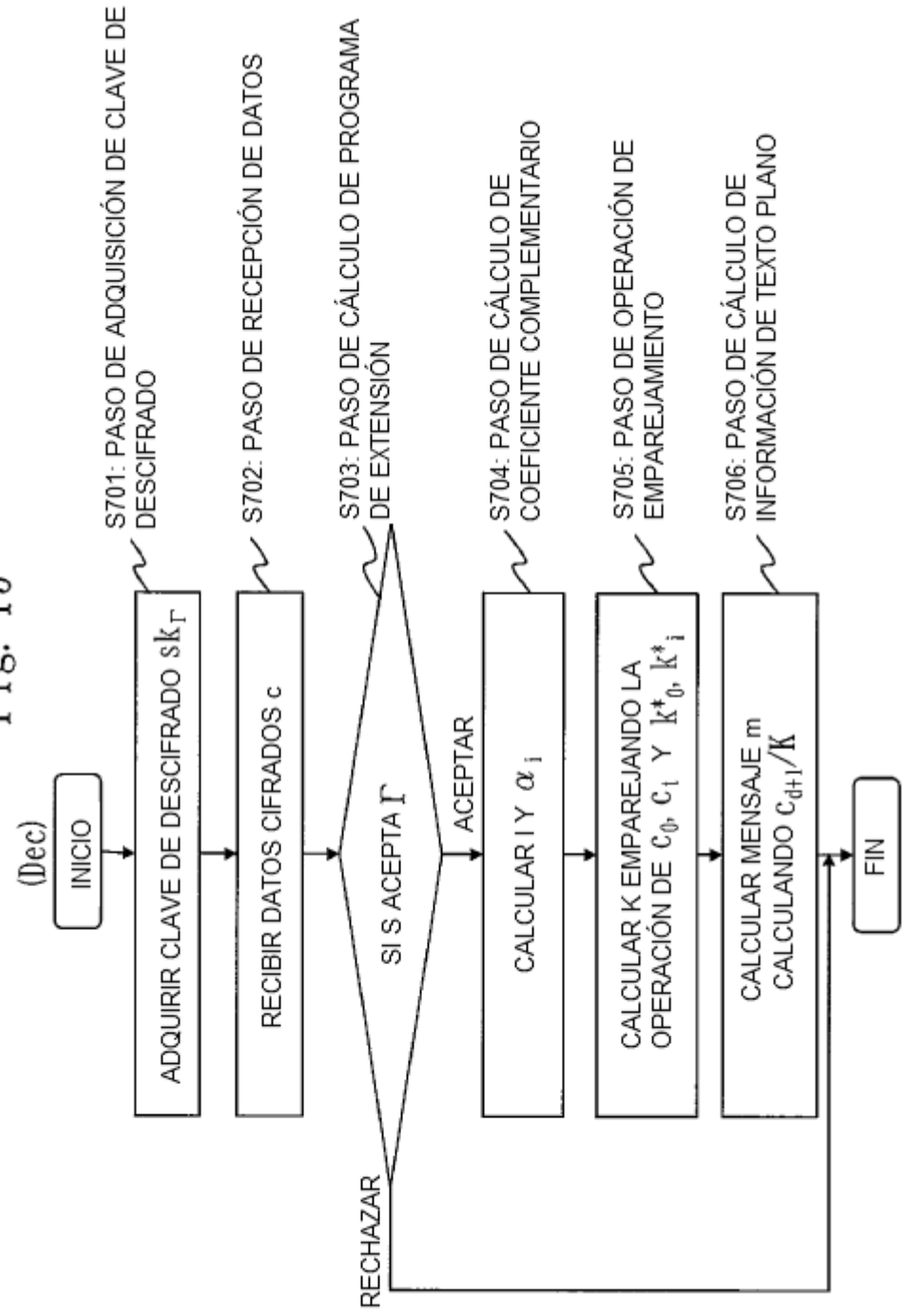

(Dec)
INICIO
S701: PASO DE ADQUISICIÓN DE CLAVE DE DESCIFRADO
ADQUIRIR CLAVE DE DESCIFRADO sk_Γ
S702: PASO DE RECEPCIÓN DE DATOS
RECIBIR DATOS CIFRADOS c
S703: PASO DE CÁLCULO DE PROGRAMA DE EXTENSIÓN
SI S ACEPTA Γ
RECHAZAR
ACEPTAR
S704: PASO DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
CALCULAR I Y α_i
S705: PASO DE OPERACIÓN DE EMPAREJAMIENTO
CALCULAR K EMPAREJANDO LA OPERACIÓN DE c_0, c_i Y k^*_0, k^*_i
S706: PASO DE CÁLCULO DE INFORMACIÓN DE TEXTO PLANO
CALCULAR MENSAJE m CALCULANDO c_{d+1}/K
FIN

Fig. 16

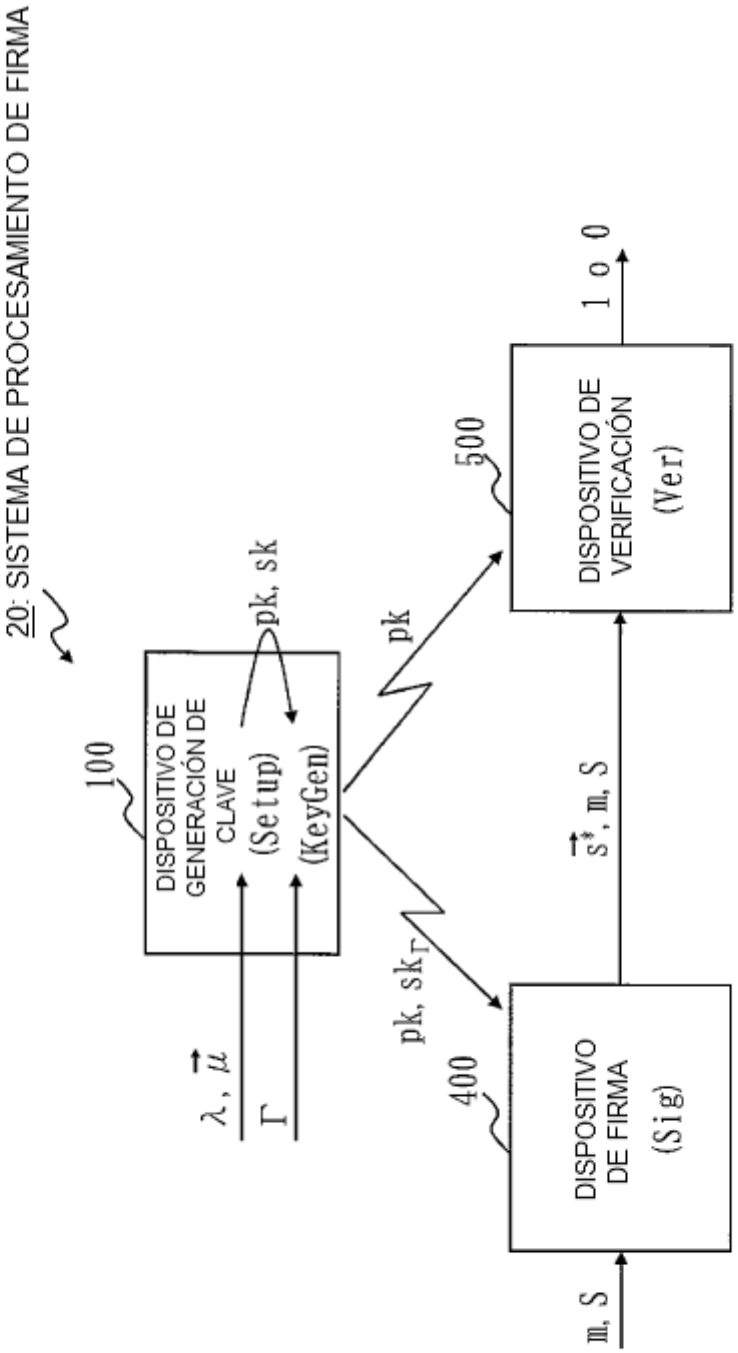

20: SISTEMA DE PROCESAMIENTO DE FIRMA
100
DISPOSITIVO DE GENERACIÓN DE CLAVE
(Setup)
(KeyGen)
pk, sk
λ, μ
Γ
pk
pk, skΓ
400
DISPOSITIVO DE FIRMA
(Sig)
m, S
500
DISPOSITIVO DE VERIFICACIÓN
(Ver)
s*, m, S
1 o 0

Fig. 17

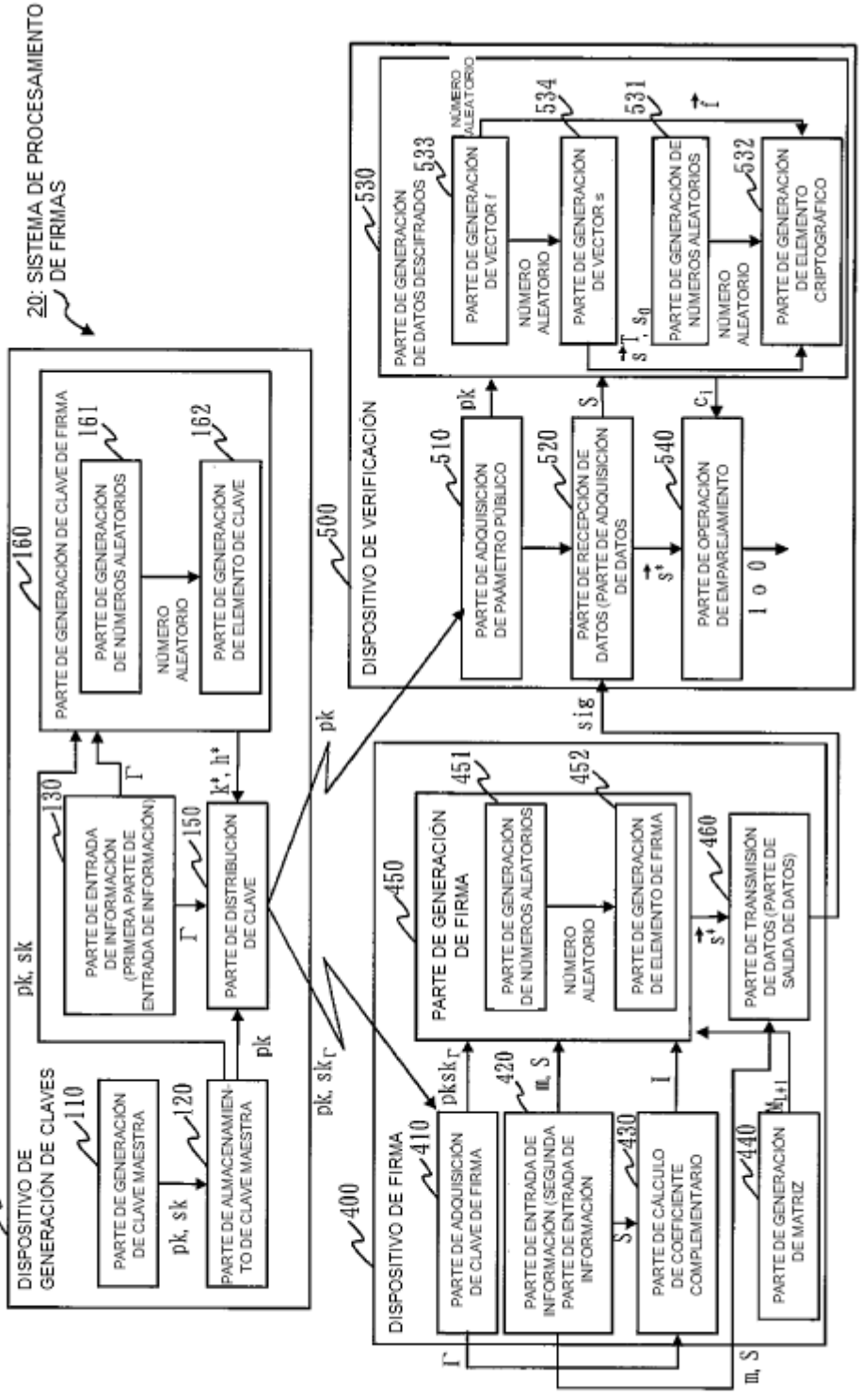

20: SISTEMA DE PROCESAMIENTO DE FIRMAS
100
DISPOSITIVO DE GENERACIÓN DE CLAVES
110
PARTE DE GENERACIÓN DE CLAVE MAESTRA
120
PARTE DE ALMACENAMIENTO DE CLAVE MAESTRA
pk, sk
pk
130
PARTE DE ENTRADA DE INFORMACIÓN (PRIMERA PARTE DE ENTRADA DE INFORMACIÓN)
Γ
150
PARTE DE DISTRIBUCIÓN DE CLAVE
k*, h*
160
PARTE DE GENERACIÓN DE CLAVE DE FIRMA
161
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
NÚMERO ALEATORIO
162
PARTE DE GENERACIÓN DE ELEMENTO DE CLAVE
pk, skΓ
400
DISPOSITIVO DE FIRMA
410
PARTE DE ADQUISICIÓN DE CLAVE DE FIRMA
pkskΓ
420
PARTE DE ENTRADA DE INFORMACIÓN (SEGUNDA PARTE DE ENTRADA DE INFORMACIÓN
m, S
430
PARTE DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
S
l
440
PARTE DE GENERACIÓN DE MATRIZ
k_{l+1}
450
PARTE DE GENERACIÓN DE FIRMA
451
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
NÚMERO ALEATORIO
452
PARTE DE GENERACIÓN DE ELEMENTO DE FIRMA
$\vec{s}^t$
460
PARTE DE TRANSMISIÓN DE DATOS (PARTE DE SALIDA DE DATOS)
sig
500
DISPOSITIVO DE VERIFICACIÓN
510
PARTE DE ADQUISICIÓN DE PARÁMETRO PÚBLICO
pk
520
PARTE DE RECEPCIÓN DE DATOS (PARTE DE ADQUISICIÓN DE DATOS
S
$\vec{s}^t$
540
PARTE DE OPERACIÓN DE EMPAREJAMIENTO
c_i
1 o 0
530
PARTE DE GENERACIÓN DE DATOS DESCIFRADOS
533
PARTE DE GENERACIÓN DE VECTOR f
NÚMERO ALEATORIO
534
PARTE DE GENERACIÓN DE VECTOR s
$\vec{T}$, s_0
531
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
NÚMERO ALEATORIO
532
PARTE DE GENERACIÓN DE ELEMENTO CRIPTOGRÁFICO
$\vec{f}$
Γ
m, S

Fig. 18

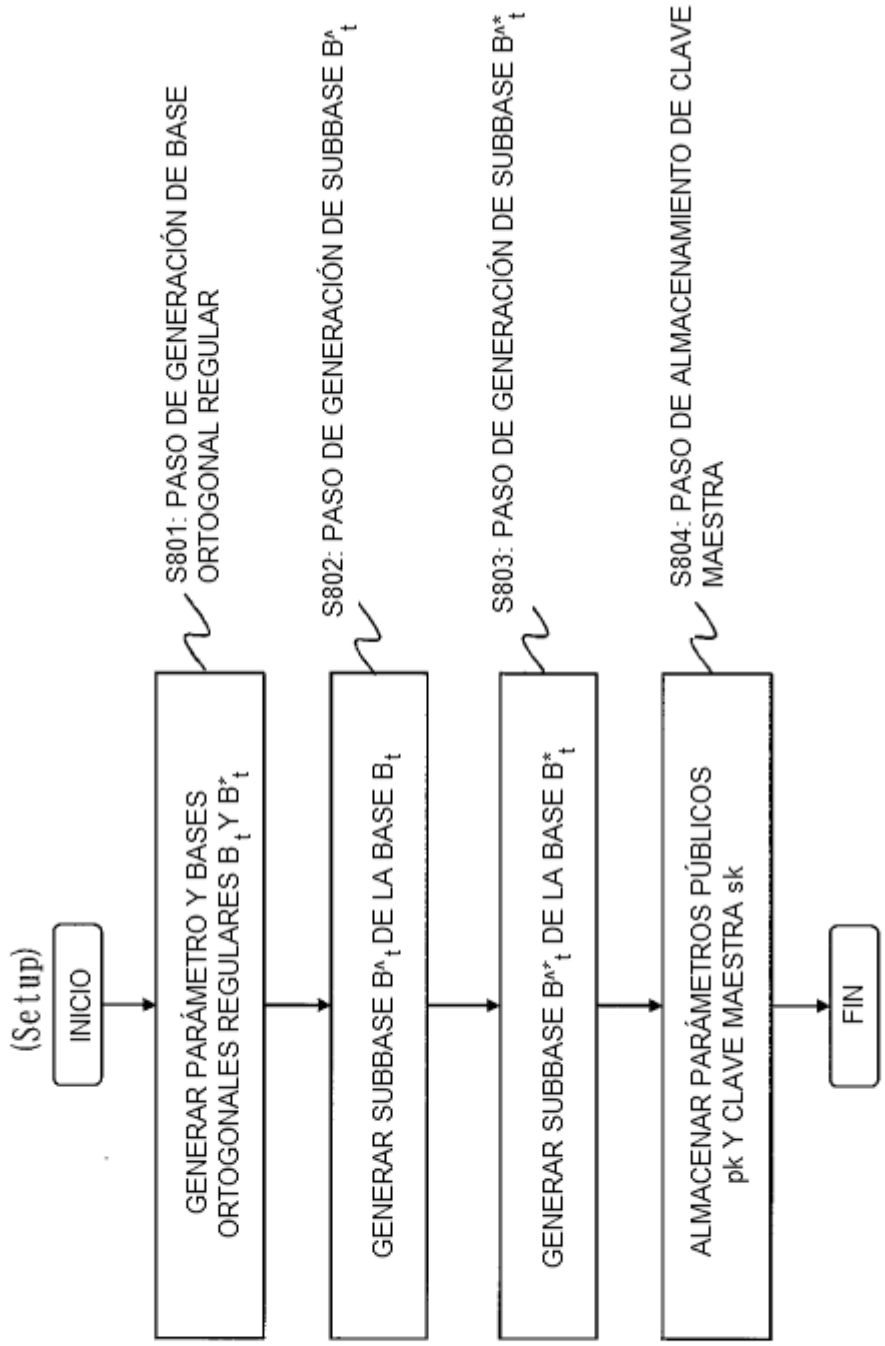

(Setup)
INICIO
GENERAR PARÁMETRO Y BASES ORTOGONALES REGULARES B_t Y B^*_t
S801: PASO DE GENERACIÓN DE BASE ORTOGONAL REGULAR
GENERAR SUBBASE $B^\wedge_t$ DE LA BASE B_t
S802: PASO DE GENERACIÓN DE SUBBASE $B^\wedge_t$
GENERAR SUBBASE $B^{\wedge *}_t$ DE LA BASE B^*_t
S803: PASO DE GENERACIÓN DE SUBBASE $B^{\wedge *}_t$
ALMACENAR PARÁMETROS PÚBLICOS pk Y CLAVE MAESTRA sk
S804: PASO DE ALMACENAMIENTO DE CLAVE MAESTRA
FIN

Fig. 19

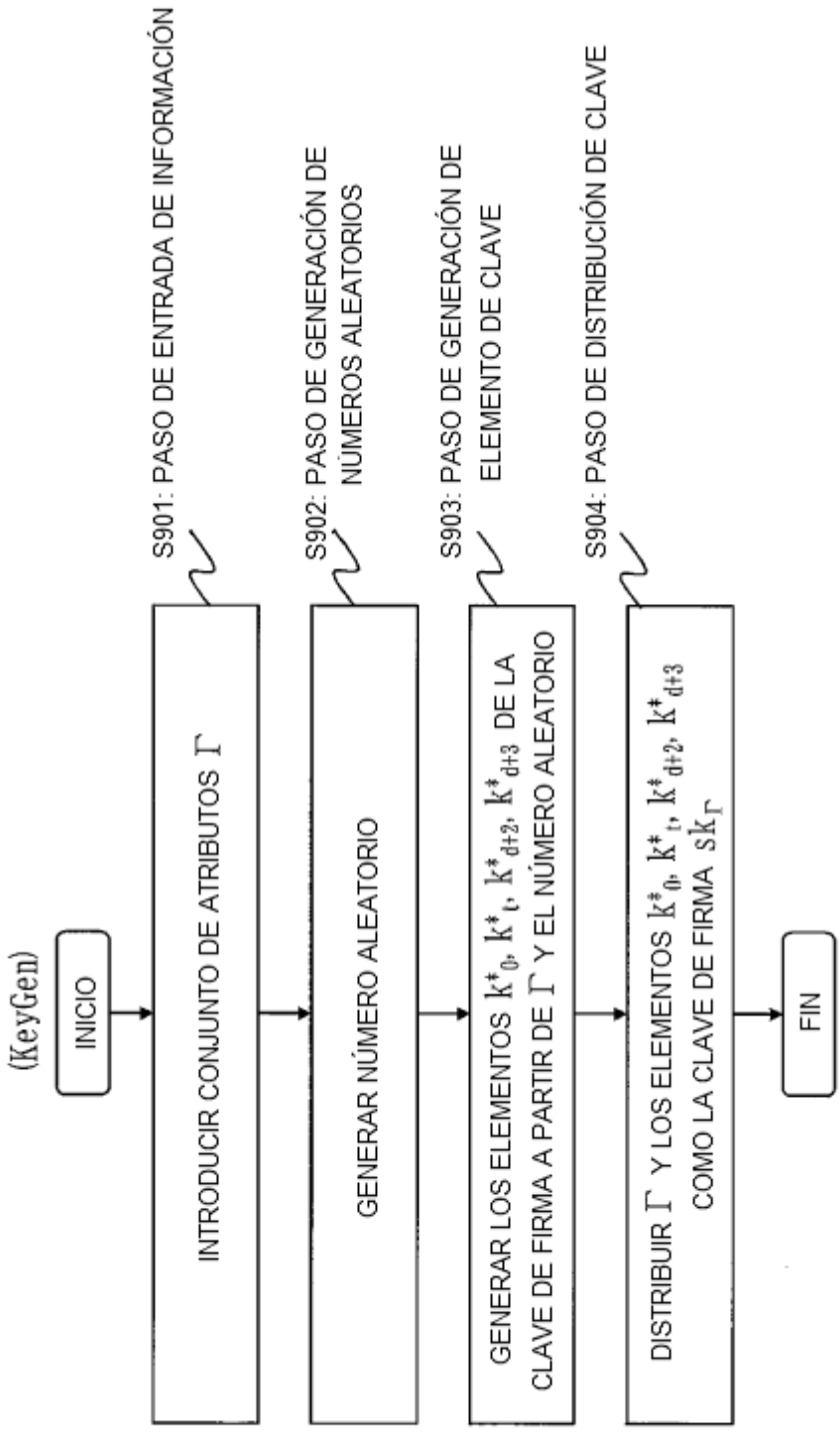

(KeyGen)
INICIO
S901: PASO DE ENTRADA DE INFORMACIÓN
INTRODUCIR CONJUNTO DE ATRIBUTOS Γ
S902: PASO DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMERO ALEATORIO
S903: PASO DE GENERACIÓN DE ELEMENTO DE CLAVE
GENERAR LOS ELEMENTOS k^*_0, k^*_t, k^*_{d+2}, k^*_{d+3} DE LA CLAVE DE FIRMA A PARTIR DE Γ Y EL NÚMERO ALEATORIO
S904: PASO DE DISTRIBUCIÓN DE CLAVE
DISTRIBUIR Γ Y LOS ELEMENTOS k^*_0, k^*_t, k^*_{d+2}, k^*_{d+3} COMO LA CLAVE DE FIRMA sk_Γ
FIN

Fig. 20

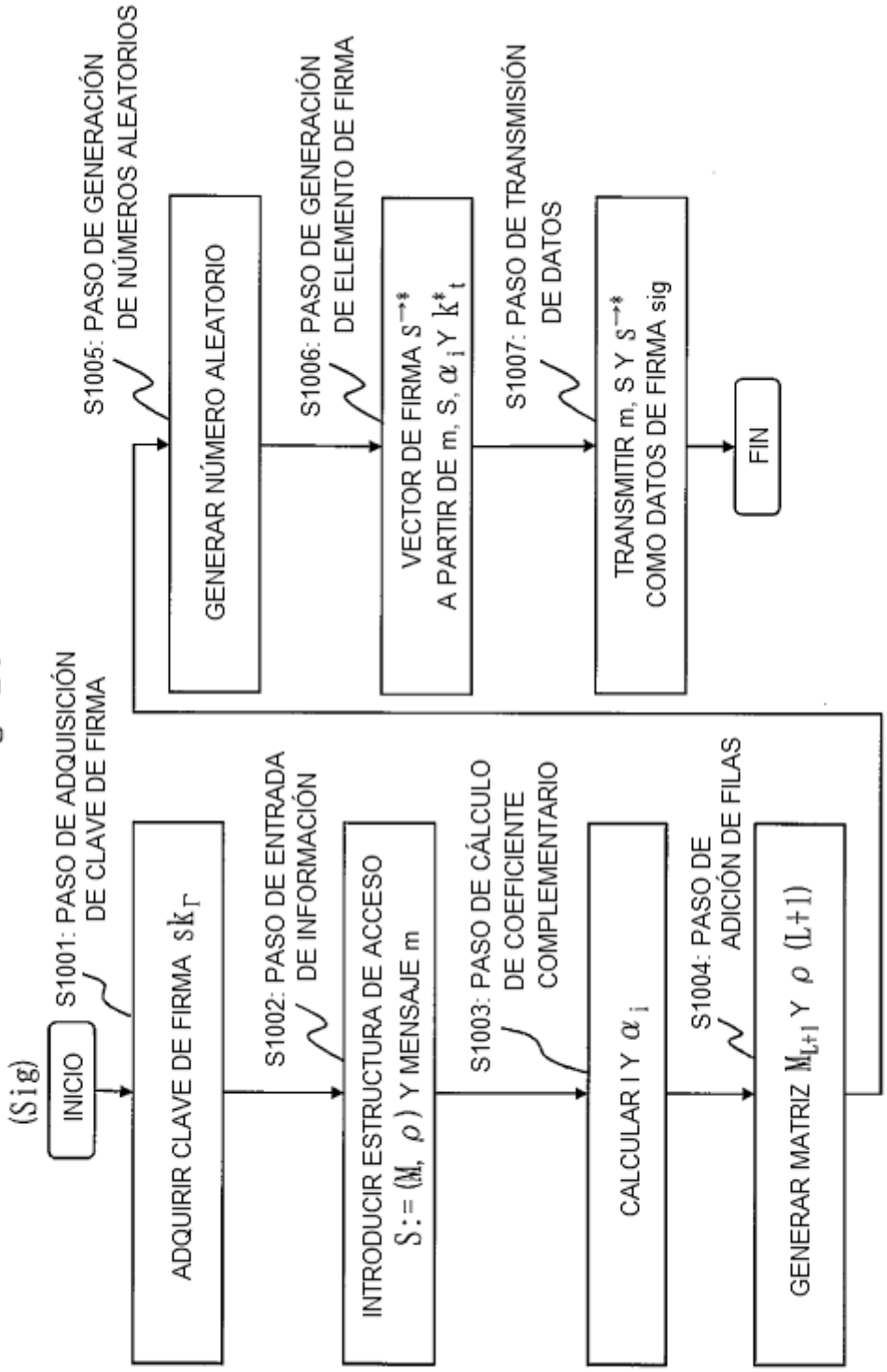

(Sig)
INICIO
S1001: PASO DE ADQUISICIÓN DE CLAVE DE FIRMA
ADQUIRIR CLAVE DE FIRMA sk_Γ
S1002: PASO DE ENTRADA DE INFORMACIÓN
INTRODUCIR ESTRUCTURA DE ACCESO $S := (\mathbb{M}, \rho)$ Y MENSAJE m
S1003: PASO DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
CALCULAR I Y α_i
S1004: PASO DE ADICIÓN DE FILAS
GENERAR MATRIZ $\mathbb{M}_{L+1}$ Y ρ (L+1)
S1005: PASO DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMERO ALEATORIO
S1006: PASO DE GENERACIÓN DE ELEMENTO DE FIRMA
VECTOR DE FIRMA $S^{\rightarrow *}$ A PARTIR DE m, S, α_i Y k^*_t
S1007: PASO DE TRANSMISIÓN DE DATOS
TRANSMITIR m, S Y $S^{\rightarrow *}$ COMO DATOS DE FIRMA sig
FIN

Fig. 21

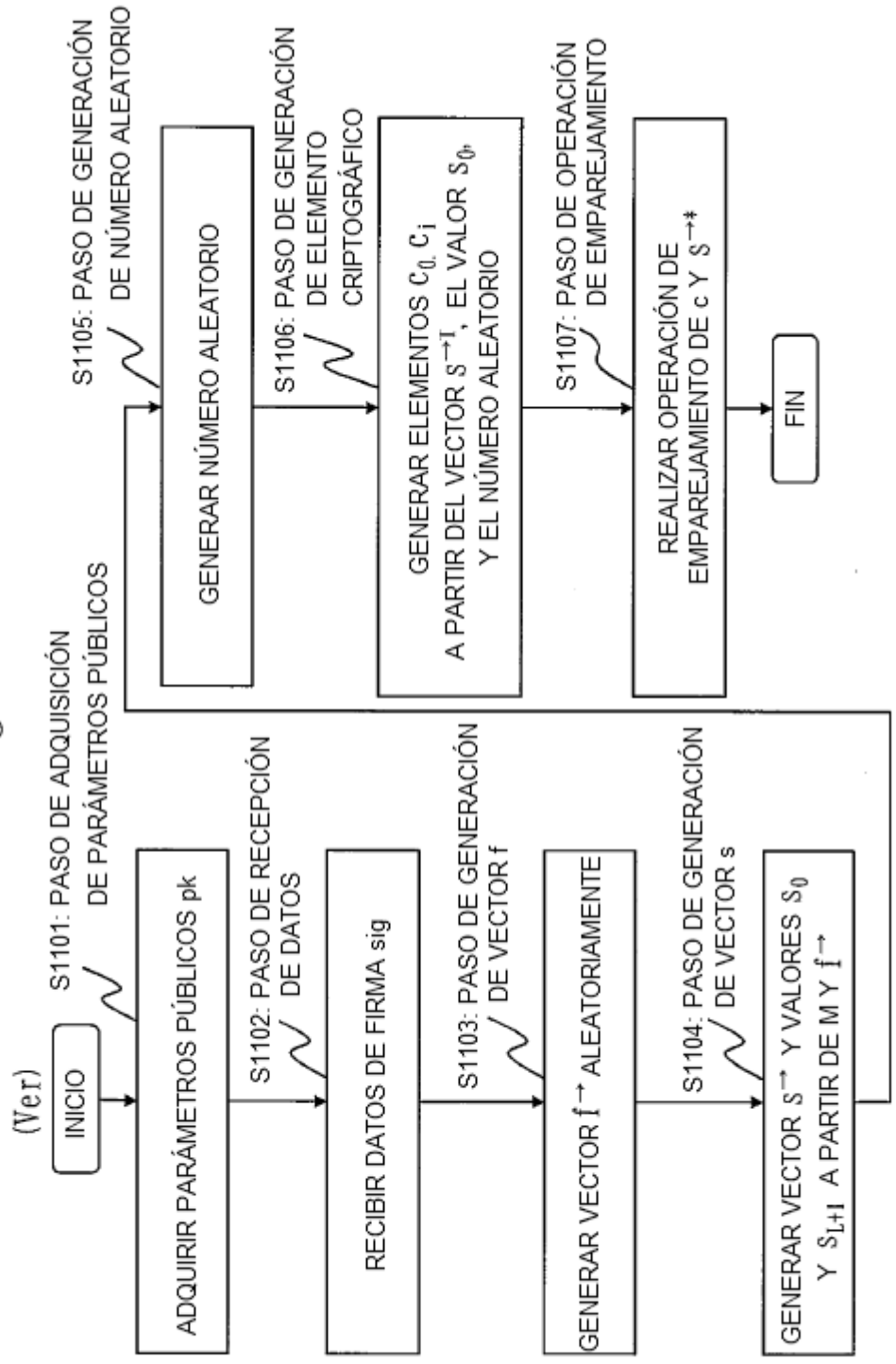
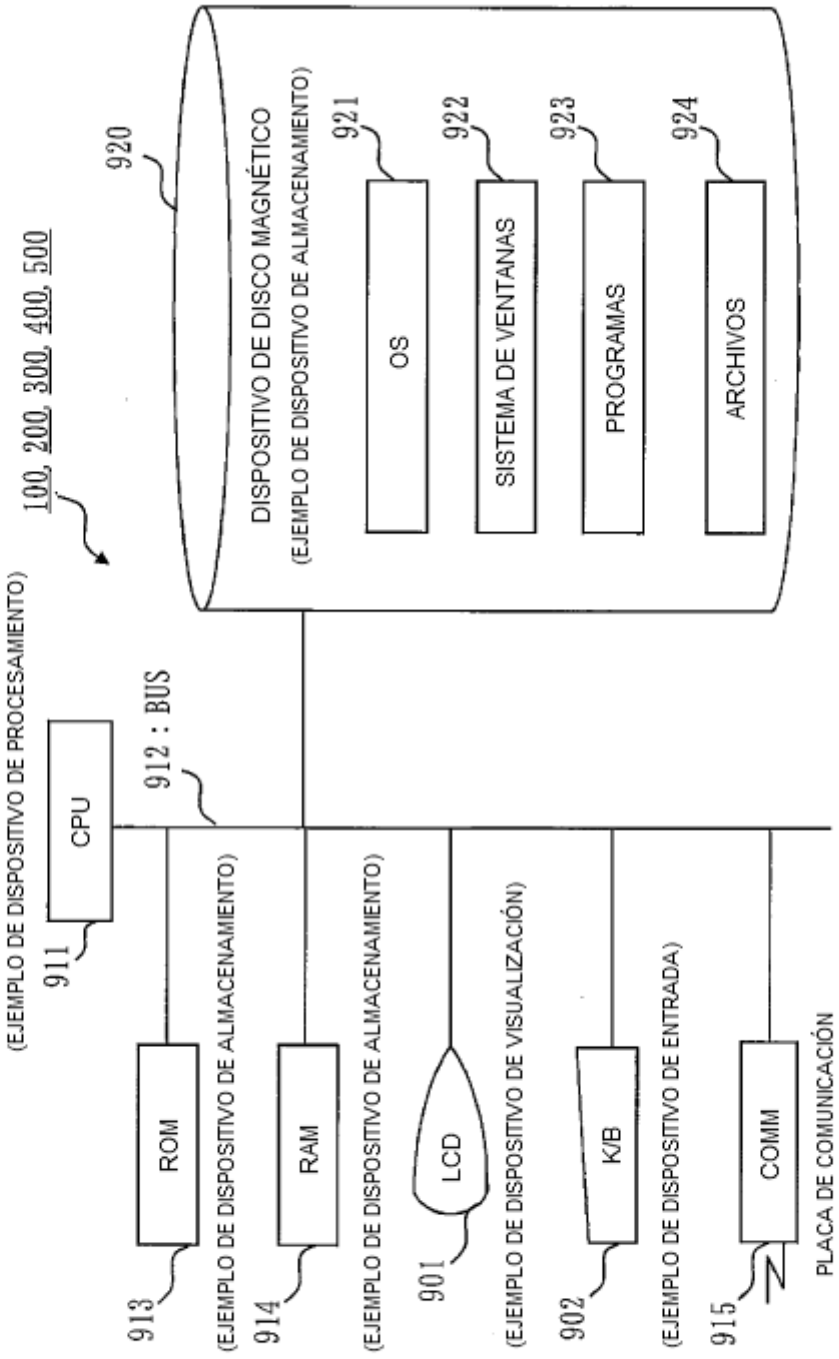

(Ver)
INICIO
S1101: PASO DE ADQUISICIÓN DE PARÁMETROS PÚBLICOS
ADQUIRIR PARÁMETROS PÚBLICOS pk
S1102: PASO DE RECEPCIÓN DE DATOS
RECIBIR DATOS DE FIRMA sig
S1103: PASO DE GENERACIÓN DE VECTOR f
GENERAR VECTOR f→ ALEATORIAMENTE
S1104: PASO DE GENERACIÓN DE VECTOR s
GENERAR VECTOR s→ Y VALORES S0 Y SL+1 A PARTIR DE M Y f→
S1105: PASO DE GENERACIÓN DE NÚMERO ALEATORIO
GENERAR NÚMERO ALEATORIO
S1106: PASO DE GENERACIÓN DE ELEMENTO CRIPTOGRÁFICO
GENERAR ELEMENTOS c0, ci A PARTIR DEL VECTOR s→I, EL VALOR S0, Y EL NÚMERO ALEATORIO
S1107: PASO DE OPERACIÓN DE EMPAREJAMIENTO
REALIZAR OPERACIÓN DE EMPAREJAMIENTO DE c Y s→*
FIN

Fig. 22


100, 200, 300, 400, 500
920
DISPOSITIVO DE DISCO MAGNÉTICO
(EJEMPLO DE DISPOSITIVO DE ALMACENAMIENTO)
921
OS
922
SISTEMA DE VENTANAS
923
PROGRAMAS
924
ARCHIVOS
(EJEMPLO DE DISPOSITIVO DE PROCESAMIENTO)
CPU
911
912 : BUS
ROM
913
(EJEMPLO DE DISPOSITIVO DE ALMACENAMIENTO)
RAM
914
(EJEMPLO DE DISPOSITIVO DE ALMACENAMIENTO)
LCD
901
(EJEMPLO DE DISPOSITIVO DE VISUALIZACIÓN)
K/B
902
(EJEMPLO DE DISPOSITIVO DE ENTRADA)
COMM
915
PLACA DE COMUNICACIÓN