(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 627 124**

(51) Int. CI.:

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

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **16.12.2011 PCT/JP2011/079176**

(87) Fecha y número de publicación internacional: **04.10.2012 WO12132136**

(96) Fecha de presentación y número de la solicitud europea: **16.12.2011 E 11862602 (7)**

(97) Fecha y número de publicación de la concesión europea: **26.04.2017 EP 2690817**

(54) Título: **Sistema de procesamiento de código, dispositivo de generación de claves, dispositivo codificador, dispositivo descodificador, procedimiento de procesamiento de código y programa de procesamiento de código**

(30) Prioridad:

25.03.2011 JP 2011067471

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
26.07.2017

(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 SLP, .

DESCRIPCIÓN

Sistema de procesamiento de código, dispositivo de generación de claves, dispositivo codificador, dispositivo descodificador, procedimiento de procesamiento de código y programa de procesamiento de código

Campo técnico

La presente invención se refiere a la encriptación funcional multi-autoridad descentralizada.

Antecedentes

La literatura distinta de la de patentes 31 describe la encriptación funcional.

Las literaturas distintas de la de patentes 12, 13, 25, 26 y 28 describen la encriptación multi-autoridad basada en atributos. La encriptación basada en atributo es una clase de encriptación funcional.

La literatura distinta de la de patentes 25 describe la encriptación basada en atributos multi-autoridad descentralizada.

Lista de citas

Literatura de patentes

Literatura distinta de la de patentes 1: Beimel, A., Secure schemes for secret sharing and key distribution. PhD Thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

Literatura distinta de la de patentes 2: Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. En: 2007 IEEE Symposium on Security and Privacy, pp. 321-34. IEEE Press (2007)

Literatura distinta de la de patentes 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, pp. 223-38. Springer Heidelberg (2004)

Literatura distinta de la de patentes 4: Boneh, D., Boyen, X.: Secure identity based encryption without random oracles. En: Franklin, M.K. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 443-59. Springer Heidelberg (2004)

Literatura distinta de la de patentes 5: Boneh, D., Boyen, X., Goh, E.: Hierarchical identity based encryption with constant size ciphertext. En: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 440-56. Springer Heidelberg (2005)

Literatura distinta de la de patentes 6: Boneh, D., Franklin, M.: Identity-based encryption from the Weil pairing. En: Kilian, J. (ed.) CRYPTO02001. LNCS, vol. 2139, pp. 213-29. Springer Heidelberg (2001)

Literatura distinta de la de patentes 7: Boneh, D., Hamburg, M.: Generalized identity based and broadcast encryption scheme. En: Pieprzyk, J.(ed.) ASIACRYPT 2008. LNCS, vol. 5350, pp. 455-70. Springer Heidelberg (2008)

Literatura distinta de la de patentes 8: Boneh, D., Katz, J., Improved efficiency for CCA-secure cryptosystems built using identity based encryption.RSA-CT 2005, LNCS, Springer Verlag (2005)

Literatura distinta de la de patentes 9: Boneh, D., Waters, B.: Conjunctive, subset, and range queries on encrypted data. En: Vadhan, S.P. (ed.) TCC 2007. LNCS, vol. 4392, pp. 535- 54. Springer Heidelberg (2007)

Literatura distinta de la de patentes 10: Boyen, X., Waters, B.: Anonymous hierarchical identity-based encryption (without random oracles). En: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 290-07. Springer Heidelberg (2006)

Literatura distinta de la de patentes 11: Canetti, R., Halevi S., Katz J.: Chosen-ciphertext security from identity-based encryption. EUROCRYPT2004, LNCS, Springer Heidelberg (2004)

Literatura distinta de la de patentes 12: Chase, M.: Multi-authority attribute based encryption. TCC, LNCS, pp. 515-34, Springer Heidelberg (2007).

Literatura distinta de la de patentes 13: Chase, M. y Chow, S.: Improving privacy and security in multi-authority attribute-based encryption, ACM Conference on Computer and Communications Security, pp. 121-30, ACM (2009).

Literatura distinta de la de patentes 14: Cocks, C.: An identity based encryption scheme based on quadratic residues. En: Honary, B. (ed.) IMAInt. Conf. LNCS, vol. 2260, pp. 360-63. Springer Heidelberg (2001)

Literatura distinta de la de patentes 15: Estibals, N.: Compact hardware for computing the Tate pairing over 128-bit-

security supersingular curves, IACR ePrint Archive: Report 2010/371 (2010).

Literatura distinta de la de patentes 16: SECURE HASH STANDARD, FIPS PUB 180-1, 180-2, NIST, USA (1995,2002)

Literatura distinta de la de patentes 17: Gentry, C.: Practical identity-based encryption without random oracles. En: Vaudenay, S. (ed.) EUROCRYPT2006. LNCS, vol. 4004, pp. 445-64. Springer Heidelberg (2006)

Literatura distinta de la de patentes 18: Gentry, C., Halevi, S.: Hierarchical identity-based encryption with polynomially many levels. En: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 437-56. Springer Heidelberg (2009)

Literatura distinta de la de patentes 19: Gentry, C., Silverberg, A.: Hierarchical ID-based cryptography. En: Zheng, Y. (ed.) ASIACRYPT 2002.LNCS, vol. 2501, pp. 548-66. Springer Heidelberg (2002)

Literatura distinta de la de patentes 20: Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control of encrypted data. En: ACM Conference on Computer and Communication Security 2006, pp. 89-8, ACM (2006)

Literatura distinta de la de patentes 21: ISO/IEC 15946-5, Information technology · Security techniques ·Cryptographic techniques based on elliptic curves ·Part 5: Elliptic curve generation (2009).

Literatura distinta de la de patentes 22: 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, pp. 146-62. Springer Heidelberg (2008)

Literatura distinta de la de patentes 23: Lewko, A., Okamoto, T., Sahai, A., Takashima, K., Waters, B.: Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption, EUROCRYPT 2010. LNCS, Springer Heidelberg (2010)

Literatura distinta de la de patentes 24: Lewko, A.B., Waters, B.: New techniques for dual system encryption and fully secure HIBE with short ciphertexts. En: Micciancio, D. (ed.) TCC 2010. LNCS, vol. 5978, pp. 455-79. Springer Heidelberg (2010)

Literatura distinta de la de patentes 25: Lewko, A.B., Waters, B.: Decentralizing Attribute-Based Encryption, IACR ePrint Archive: Report 2010/351 (2010).

Literatura distinta de la de patentes 26: H. Lin, Z. Cao, X. Liang, and J. Shao.: Secure threshold multi authority attribute based encryption without a central authority, INDOCRYPT, LNCS, vol. 5365, pp. 426-36, Springer Heidelberg (2008).

Literatura distinta de la de patentes 27: Maji, H., Prabhakaran, M., Rosulek, M.: Attribute-Based Signatures. http://www.cs.uiuc.edu/-mmp/research.html

Literatura distinta de la de patentes 28: S. Muller, S. Katzenbeisser, and C. Eckert.; On multi-authority ciphertext-policy attribute-based encryption, Bull. Korean Math Soc. 46, No.4, pp. 803-19 (2009).

Literatura distinta de la de patentes 29: Okamoto, T., Takashima, K.: Homomorphic encryption and signatures from vector decomposition. En: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5209, pp. 57-4, Springer Heidelberg (2008)

Literatura distinta de la de patentes 30: Okamoto, T., Takashima, K.: Hierarchical predicate encryption for inner-products, En: ASIACRYPT 2009, Springer Heidelberg (2009)

Literatura distinta de la de patentes 31: Okamoto, T., Takashima, K.: Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption, 5 de Noviembre 2010, recuperado de Internet: URL: http://eprint.iacr.org/eprint-bin/getfile.pl?entry=2010/563&version=20101105:113344& file=563.pdf

Literatura distinta de la de patentes 32: Ostrovsky, R., Sahai, A., Waters, B.: Attribute-based encryption with non-monotonic access structures. En: ACM Conference on Computer and Communication Security 2007, pp. 195-03, ACM (2007)

Literatura distinta de la de patentes 33: Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: Secure attribute-based systems. En: ACM Conference on Computer and Communication Security 2006, pp. 99-12, ACM, (2006)

Literatura distinta de la de patentes 34: Sahai, A., Waters, B.: Fuzzy identity-based encryption. En: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 457-73. Springer Heidelberg (2005)

Literatura distinta de la de patentes 35: Shi, E., Waters, B.: Delegating capability in predicate encryption systems. En:

Aceto, L., Damgard, I., Goldberg, L.A., Halldorsson, M.M., Ingolfsdottir, A., Walukiewicz, I. (eds.) ICALP (2) 2008. LNCS, vol.5126, pp. 560.578. Springer Heidelberg (2008)

Literatura distinta de la de patentes 36: Waters, B.: Efficient identity based encryption without random oracles. Eurocrypt 2005, LNCS, vol. 3152, pp.443-59. Springer Verlag, (2005)

Literatura distinta de la de patentes 37: Waters, B.: Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization. ePrint, IACR, http://eprint.iacr.org/2008/290

Literatura distinta de la de patentes 38: Waters, B.: Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions. En: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5677, pp. 619-36. Springer Heidelberg (2009)

Sumario de la invención

Problema técnico

La encriptación funcional tiene un problema en el sentido de que la seguridad de todo el sistema depende de una sola parte.

La presente invención tiene por objeto proporcionar una encriptación funcional multi-autoridad descentralizada en la que la seguridad de todo el sistema no dependa de una sola parte.

Solución al problema

Un sistema de procesamiento criptográfico según la presente invención es un sistema de procesamiento criptográfico que incluye d (d es un número entero de 1 o más) unidades de dispositivos de generación de claves, un dispositivo de encriptación y un dispositivo de desencriptación y que sirve para ejecutar un procedimiento criptográfico usando una base B_t y una base B^*_t para al menos un número entero t = 1, ..., d,
en el que cada dispositivo de generación de claves de las d unidades de dispositivos de generación de claves incluye

una primera parte de entrada de información que toma como entrada información de atributo $\vec{x}_t := (x_{t,i})$ (i = 1, ..., n_t) donde n_t es un número entero de 1 o más) para un número entero t entre los números enteros t = 1, ..., d que está predeterminado para cada dispositivo de generación de claves,
una parte de generación de elementos de clave que genera un elemento k^*_t de clave que incluye un vector indicado en la Fórmula 1 basado en el número entero t, la información $\vec{x}_t$ de atributos introducida por la primera parte de entrada de información, un valor δ predeterminado, y un vector base $b^*_{t,i}$ (i = 1, ..., $2n_t$) de la base B^*_t, y
una parte de transmisión de clave de desencriptación que transmite al dispositivo de desencriptación, una clave usk de desencriptación que incluye el elemento k^*_t de clave generado por la parte de generación de elementos de clave y la información $\vec{x}_t$ de atributos,

en el que el dispositivo de encriptación incluye

una segunda 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 una de entre una tupla $(t, \vec{v}_i)$ positiva y una tupla $\neg(t, \vec{v}_i)$ negativa de la información t de identificación (t es cualquier número entero de t = 1, ..., d) e información de atributos $\vec{v}_i := (v_{i,i'})$ (i'= 1, ..., n_t); 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 vectores que genera un vector columna $\vec{s}^T := (s_{1, ..., } s_L)^T := M \cdot \vec{f}^T$ basado en un vector $\vec{f}$ que tiene r piezas de elementos y la matriz M introducida por la segunda parte de entrada de información, y genera un vector columna $(\vec{s}')^T := (s_{1, ..., } s_L)^T := M\text{-} (\vec{f}')^T$ basado en la matriz M y un vector $\vec{f}'$ que tiene r piezas de elementos y que satisface $s_0 = \vec{h} \cdot (\vec{f}')^T$ donde $s_0 = \vec{h} \cdot \vec{f}^T$,

una parte de generación de elemento c_i de encriptación que, para cada número entero i = 1, ..., L y en base al vector columna $\vec{s}^T$ y el vector columna $(\vec{s}')^T$ que son generados por la parte de generación de vectores, y valores θ_i y θ_i' predeterminados para cada número entero i = 1, ..., L, genera un elemento c_i de encriptación, incluyendo un vector indicado en la Fórmula 2, cuando la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, usando un vector base $b_{t,i}'$ (i' = 1, ..., $2n_t$) de la base B_t indicado por la información t de identificación de la tupla positiva, y genera un elemento c_i de encriptación incluyendo un vector indicado en la Fórmula 3, cuando la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, usando un vector base $b_{t,i}$ (i = 1, ..., $2n_t$) indicado por la información t de identificación de la tupla negativa, y

una parte de transmisión de texto encriptado que transmite al dispositivo de desencriptación, un texto ct_s encriptado incluyendo: el elemento c_i de encriptación generado para cada número entero i = 1, ..., L por la parte de generación de elemento c_i de encriptación; la variable ρ(i); y la matriz M, y

en el que el dispositivo de desencriptación incluye

una parte de recepción de clave de desencriptación que recibe la clave usk de desencriptación transmitida por la parte de transmisión de clave de desencriptación de al menos un dispositivo de generación de clave de entre las d unidades de dispositivos de generación de claves,

una parte de recepción de datos que recibe el texto ct_s encriptado transmitido por la parte de transmisión de texto encriptado,

una parte de cálculo de coeficiente complementario que, en base a la información $\vec{x}_t$ de atributos incluida en la clave usk de desencriptación recibida por la parte de recepción de clave de desencriptación, y la variable ρ(i) incluida en el texto ct_s encriptado recibido por la parte de recepción de datos, especifica, de 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 $(t, \vec{v}_i)$, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla positiva recibida por la parte de recepción de clave de desencriptación, y con la que un producto interno de $\vec{v}_i$ de la tupla positiva y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla positiva es igual a 0 y un número entero i para el cual la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla negativa recibida por la parte de recepción de clave de desencriptación y con la que un producto interno de $\vec{v}_i$ de la tupla negativa y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla negativa no es igual a 0; y calcula, con respecto a i incluido en el conjunto I especificado, un coeficiente α_i complementario con el cual un total de $\alpha_i M_i$ en base a M_i que es un elemento en una i-ésima fila de la matriz M incluida en el texto ct_s encriptado se convierte en el vector $\vec{h}$ predeterminado, y

una parte de operación de emparejamiento que calcula una información K predeterminada realizando una operación de emparejamiento indicada en la Fórmula 4 para el elemento c_i de encriptación incluido en el texto ct_s encriptado y el elemento k^*_t de clave incluido en la clave usk de desencriptación en base al conjunto I y el coeficiente α_i complementario que son calculados por la parte de cálculo del coeficiente complementario.

[Fórmula 1]

$$(\overbrace{(\delta+1)x_{t,1},\ldots,(\delta+1)x_{t,n_t}}^{n_t},\overbrace{-\delta x_{t,1},\ldots,-\delta x_{t,n_t}}^{n_t},0,\ldots,0)_{\mathbb{B}_t^*}$$

[Fórmula 2]

$$(\overbrace{s_i+\theta_i v_{i,1},\theta_i v_{i,2},\ldots,\theta_i v_{i,n_t}}^{n_t},\overbrace{s_i'+\theta_i' v_{i,1},\theta_i' v_{i,2},\ldots,\theta_i' v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbb{B}_i}$$

[Fórmula 3]

$$(\overbrace{s_i v_{i,1},\ldots,s_i v_{i,n_t}}^{n_t},\overbrace{s_t' v_{i,1},\ldots,s' v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbb{B}_t}$$

[Fórmula 4]

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

Efectos ventajosos de la invención

En un sistema de procesamiento criptográfico según la presente invención, cada uno de entre la pluralidad de dispositivos de generación de claves genera una parte de una clave de desencriptación. Por lo tanto, incluso si las seguridades de algunos de los dispositivos de generación de claves se ven comprometidas, sólo se pierde la función de una parte de la clave de desencriptación y no se compromete la seguridad de todo el sistema.

Breve descripción de los dibujos

La Fig. 1 es un dibujo explicativo de multi-autoridad.

La Fig. 2 es un dibujo explicativo de una matriz M^.

La Fig. 3 es un dibujo explicativo de una matriz M_{δ}.

La Fig. 4 es un dibujo explicativo de s_0.

La Fig. 5 es un dibujo explicativo de $s^{\rightarrow T}$.

La Fig. 6 es un diagrama de configuración de un sistema 10 de procesamiento criptográfico que ejecuta la encriptación funcional multi-autoridad descentralizada.

La Fig. 7 es un diagrama de bloques de funciones que muestra una función de un dispositivo 100 de generación de claves.

La Fig. 8 es un diagrama de bloques de funciones que muestra una función de un dispositivo 200 de encriptación.

La Fig. 9 es un diagrama de bloques de funciones que muestra una función de un dispositivo 300 de desencriptación.

La Fig. 10 es un diagrama de flujo que muestra el procedimiento del algoritmo GSetup.

La Fig. 11 es un diagrama de flujo que muestra el procedimiento del algoritmo ASetup.

La Fig. 12 es un diagrama de flujo que muestra el procedimiento del algoritmo AttrGen.

La Fig. 13 es un diagrama de flujo que muestra el procedimiento del algoritmo Enc.

La Fig. 14 es un diagrama de flujo que muestra el procedimiento del algoritmo Dec.

La Fig. 15 es un diagrama que muestra un ejemplo de la configuración de hardware del dispositivo 100 de generación de claves, el dispositivo 200 de encriptación y el dispositivo de desencriptación.

Descripción de realizaciones

A continuación, se describirán realizaciones de la presente invención con referencia a los dibujos adjuntos.

En la siguiente descripción, un dispositivo de procesamiento es, por ejemplo, una CPU 911 (que se describirá más adelante). Un dispositivo de almacenamiento es, por ejemplo, una ROM 913, una RAM 914, o un disco 920 magnético (cada uno se describirá más adelante). Un dispositivo de comunicación es, por ejemplo, una placa 915 de comunicación (que se describirá más adelante). Un dispositivo de entrada es, por ejemplo, un teclado 902 o la placa 915 de comunicación (cada uno se describirá más adelante). Concretamente, 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 descripción siguiente.

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

[Fórmula 101]

$$y \xleftarrow{R} A$$

Cuando A es un conjunto, la Fórmula 102 denota que y se selecciona de manera uniforme de A. Concretamente, en la Fórmula 102, y es un número aleatorio uniforme.

[Fórmula 102]

$$y \xleftarrow{U} A$$

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

[Fórmula 103]

$$y := z$$

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

[Fórmula 104]

$$A(x) \to a$$

Por ejemplo,

$$A(x) \to 1$$

La Fórmula 105, es decir, F_q, denota un campo finito de orden q.

[Fórmula 105]

$$\mathbb{F}_q$$

Un símbolo vectorial denota una representación vectorial sobre el campo F_q finito. Es decir, se establece la Fórmula 106.

[Fórmula 106]

$\vec{x}$ denota

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

La Fórmula 107 denota el producto interno, indicado por la Fórmula 109, de dos vectores $\vec{x}$ y $\vec{v}$ 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 denota la transpuesta de la matriz M.

Obsérvese que para las bases B y B* indicadas en la Fórmula 110, se establece la Fórmula 111.

[Fórmula 110]

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

[Fórmula 111]

$$(x_1, \ldots, x_N)_{\mathbb{B}} := \sum_{i=1}^{N} x_i b_i,$$
$$(y_1, \ldots, y_N)_{\mathbb{B}^*} := \sum_{i=1}^{N} y_i b_i^*$$

Obsérvese que $\vec{e}_{t,j}$ indica un vector base normal mostrado en la Fórmula 112.

[Fórmula 112]

$$\vec{e}_{t,j} : (\overbrace{0 \cdots 0}^{j-1}, 1, \overbrace{0 \cdots 0}^{n_t - j}) \in \mathbb{F}_q^{n_t} \quad \text{for } j = 1, \ldots, n_t$$

En la siguiente descripción, cuando "nt" se indica como un subíndice o superíndice, nt es n_t. De manera similar, cuando "Vt" se indica como un subíndice o superíndice, Vt es V_t. De manera similar, cuando "δi,j" se indica como superíndice, δi,j es $\delta_{i,j}$.

Cuando "→", que indica un vector, está asociado a un subíndice o superíndice, "→" se adjunta como un superíndice al subíndice o superíndice.

Además, xt en una clave $usk_{Gid,(t,Xt)}$ de desencriptación representa x_t.

En la siguiente descripción, un procedimiento criptográfico incluye un procedimiento de generación de claves, un procedimiento de encriptación y un procedimiento de desencriptación.

Realización 1.

Esta realización describe un concepto básico para implementar la "encriptación funcional multi-autoridad descentralizada" y, a continuación, describe la estructura de la encriptación funcional multi-autoridad descentralizada.

En primer lugar, se describirá brevemente la encriptación funcional multi-autoridad descentralizada: se describirá la encriptación funcional y, a continuación, se describirá la multi-autoridad descentralizada.

En segundo lugar, se describirá un espacio que tiene una rica estructura matemática denominado "espacios vectoriales de doble emparejamiento (DPVS, Dual Pairing Vector Spaces)" que es un espacio para implementar la encriptación funcional.

En tercer lugar, se describirá un concepto para implementar la encriptación funcional. Aquí se describirá el "programa span", "el producto interno de los vectores de atributo y una estructura de acceso" y "la distribución secreta (intercambio secreto)".

En cuarto lugar, se describirá la "encriptación funcional multi-autoridad descentralizada" según esta realización. Inicialmente, se describirá la estructura básica de la “encriptación funcional multi-autoridad descentralizada". Posteriormente, se describirá la estructura básica de un "sistema 10 de procesamiento criptográfico" que implementa la “encriptación funcional multi-autoridad descentralizada". A continuación, se describirán en detalle la “encriptación funcional multi-autoridad descentralizada" y un "sistema 10 de procesamiento criptográfico" según esta realización.

<1. Encriptación funcional multi-autoridad descentralizada>

<1-1. Encriptación funcional>

La encriptación funcional es una noción avanzada (detallada) de encriptación de clave pública que cubre a encriptación basada en ID (encriptación basada en identidad, Identity-Based Encryption, IBE) (véanse las Literaturas distintas de las de patentes 3, 4, 6, 14 y 17), encriptación de vector oculto (véase la Literatura distinta de la de patentes 9), encriptación de predicado (véase la Literatura distinta de la de patentes 22) y encriptación basada en atributo (ABE, véase las Literaturas distintas de las de patentes, 2, 20 y 32-34) como casos especiales.

Una clave de desencriptación, sk_ψ (clave secreta), en la encriptación funcional se asocia con un parámetro, ψ, y un mensaje m es encriptado a un texto encriptado Enc(m, pk, γ) usando una clave pk pública junto con un parámetro γ que es diferente del parámetro ψ. La clave sk_ψ de desencriptación puede desencriptar un texto encriptado Enc(m, pk, γ) si y sólo si se cumple una relación R(ψ, γ).

La encriptación funcional requiere un grupo de confianza denominado una autoridad. La autoridad genera una clave pk pública y una clave msk secreta maestra. La clave pk pública es distribuida como un parámetro del sistema. La clave secreta maestra es usada para generar la clave sk_ψ de desencriptación de un usuario que está asociado con el parámetro del usuario, ψ.

En el caso de encriptación basada en ID, el parámetro ψ es el ID del usuario y la relación R es la igualdad, es decir, la relación R(ψ, γ) se cumple si y sólo si ψ = γ.

En una encriptación basada en atributos CP (Ciphertext-Policy, política de acceso por texto encriptado) para una estructura de acceso general, un parámetro ψ es una tupla (x_1, ... , x_i) de atributos de usuario, y una relación R(·, γ) es una estructura de acceso general. Más precisamente, la relación R(·, γ) se expresa mediante (M^, (v_1, …, v_i)), y las relaciones de igualdad entre componentes para componentes de atributo, es decir, $\{x_t = v_t\}_{t\in\{1,\ldots,i\}}$, se introducen en un programa (monótono) M^, y la relación R(ψ, γ) se cumple si y sólo si el vector de valor verdadero de ($T(x_1 = v_1)$, ..., $T(x_i = v_i)$ es aceptado por el programa span M^, donde T(ψ) := 1 si ψ es verdadero, y T(ψ) := 0 si ψ es falso. Por ejemplo, T (x = v) := 1 si x = v, y T(x = v) := 0 si $x \neq v$.

Aunque la encriptación funcional tiene muchas aplicaciones, un gran problema en la noción es que la seguridad de todo el sistema depende de una sola parte. En otras palabras, si la autoridad está corrompida o la clave secreta maestra está comprometida, el sistema dejará de funcionar.

<1-2. Multi-Autoridad descentralizada>

Inicialmente, se explicará la "multi-autoridad". Multi-autoridad significa la presencia de una pluralidad de autoridades que generan la clave de desencriptación del usuario.

Tal como se ha descrito anteriormente, en una encriptación funcional ordinaria, la seguridad de todo el sistema depende de una parte determinada (autoridad). Con el esquema multi-autoridad, sin embargo, incluso si la seguridad de alguna autoridad está dañada o la clave secreta (clave maestra) de alguna autoridad está comprometida, sólo parte del sistema deja de funcionar y la parte restante del sistema puede funcionar normalmente.

La Fig. 1 es un dibujo explicativo de la multi-autoridad.

En la Fig. 1, una oficina pública gestiona atributos tales como la dirección, el número de teléfono y la edad. La policía gestiona atributos tales como el tipo de licencia de conducir. Una empresa A gestiona atributos tales como la posición en la empresa A y el departamento de pertenencia en la empresa A. Una clave 1 de desencriptación asociada con los atributos gestionados por la oficina pública es emitida por la oficina pública. Una clave 2 de desencriptación asociada con los atributos gestionados por la policía es emitida por la policía. Una clave 3 de desencriptación asociada con los atributos gestionados por la empresa A es emitida por la empresa A.

El desencriptador que desencripta un texto encriptado desencripta el texto encriptado usando una clave de desencriptación formada por la unión de las claves 1, 2 y 3 de desencriptación emitidas por las autoridades respectivas tales como la oficina pública, la policía y la empresa A. Concretamente, desde el punto de vista del

desencriptador, una clave de desencriptación formada mediante la unión de las claves de desencriptación emitidas por las respectivas autoridades es la única clave de desencriptación emitida al mismo.

Por ejemplo, en un caso en el que la clave maestra de la empresa A está comprometida, aunque el sistema de procesamiento criptográfico no funcione con respecto a los atributos de la empresa A, funciona con respecto a los atributos gestionados por las otras autoridades. Concretamente, aunque con respecto a los atributos gestionados por la empresa A existe un riesgo de desencriptación por un usuario que tiene atributos distintos de los atributos especificados, con respecto a los atributos distintos de los gestionados por la empresa A, la desencriptación es posible únicamente por un usuario que tiene los atributos especificados.

Tal como se observa en el ejemplo de la Fig. 1, según la encriptación funcional, es normal que una pluralidad de autoridades estén presentes y que cada autoridad gestione una cierta categoría (subespacio) o rango de definición en los atributos y emita (una parte de) una clave de desencriptación con respecto al atributo del usuario en esta categoría.

Se explicará la noción "descentralizada". “Descentralizada” significa que cualquier parte puede servir como una autoridad y puede emitir (una parte de) la clave de desencriptación sin interactuar con las otras partes, y que cada usuario puede adquirir (una parte de) la clave de desencriptación sin interactuar con las otras partes.

Por ejemplo, si existe una autoridad central, el sistema no es descentralizado. Una autoridad central es una autoridad superior a las demás autoridades. Si la seguridad de la autoridad central está corrompida, la seguridad de cada autoridad estará corrompida.

<2. Espacios vectoriales de doble emparejamiento>

En primer lugar, se describirán grupos de emparejamientos bilineales simétricos.

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

En la siguiente descripción, supóngase que la Fórmula 113 es un algoritmo que toma como entrada 1^{λ} y emite los valores de un parámetro $param_G := (q, G, G_T, G, e)$ de grupos de emparejamiento bilineales con un parámetro λ de seguridad.

[Fórmula 113]

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

Ahora se describirán espacios vectoriales de doble emparejamiento.

Los espacios vectoriales de doble emparejamiento (q, V, G_T, A, e) pueden estar constituidos por un producto directo de grupos de emparejamientos bilineales simétricos ($param_G := (q, G, G_T, G, e)$). Los espacios vectoriales de doble emparejamiento (q, V, G_T, A, e) son una tupla de un número primo q, un espacio V vectorial N-dimensional sobre F_q indicado en la Fórmula 114, un grupo G_T cíclico de 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 tal como se indica en la Fórmula 115.

[Fórmula 114]

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

[Fórmula 115]

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

Operación (1): Emparejamiento bilineal no degenerado

El emparejamiento en el espacio V está definido por la Fórmula 116.

[Fórmula 116]

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

en la que

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

Esto es bilineal no degenerado, es decir, e(sx, ty) = e(s, y)st y si e(x, y) = 1 para todo y ∈ V, entonces x = 0. Para todo i y j, e(a_i, a_j) = e(g, g)$^{\delta i,j}$ en la que $\delta_{i,j}$ = 1 si i = j, y $\delta_{i,j}$ = 0 si i ≠ j. Además, e(g, g) ≠ 1 ∈ G_T.

Operación (2): Mapas de distorsión

Una transformación lineal $\Phi_{i,j}$ en el espacio V indicado en la Fórmula 117 puede conseguir la Fórmula 118.

[Fórmula 117]

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

[Fórmula 118]

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

Obsérvese que

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

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

En la siguiente descripción, supóngase que la Fórmula 119 es un algoritmo que toma como entrada, 1^{λ} (λ ∈ números naturales), N ∈ números naturales, y los valores del parámetro $param_G$:= (q, G, G_T, G, e) de grupos de emparejamiento bilineales, y emite los valores de un parámetro $param_v$:= (q, V, G_T, A, e) de los espacios vectoriales de doble emparejamiento que tienen un parámetro λ de seguridad y que forman un espacio V N-dimensional.

[Fórmula 119]

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

Se describirá un caso en el que los espacios vectoriales de doble emparejamiento se construyen a partir de los grupos de emparejamientos bilineales simétricos descritos anteriormente. También pueden construirse espacios

vectoriales de doble emparejamiento a partir de grupos de emparejamiento bilineales asimétricos. La siguiente descripción puede aplicarse fácilmente a un caso en el que los espacios vectoriales de doble emparejamiento se construyen a partir de grupos de emparejamiento bilineales asimétricos.

<3. Concepto para implementar la encriptación funcional>

<3-1. Programa Span>

La Fig. 2 es un dibujo explicativo de una matriz M^.

Supóngase que $\{p_1, ..., p_n\}$ es 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 cada fila de la matriz M y está relacionada con uno de los literales $\{ p_1, ..., p_n \neg p_1, ..., \neg p_n\}$. Una etiqueta ρ_i (i = 1, ..., L) de cada fila de M está relacionado con uno de los literales, concretamente, ρ: $\{1, ..., L\} \rightarrow \{p_1, ..., p_n \neg p_1, ..., \neg p_n\}$.

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 están relacionadas con el valor "1" por la secuencia δ de entrada. Concretamente, la matriz M_δ es una submatriz que consiste en las filas de la matriz M que están relacionadas con p_i con el que $\delta_i = 1$ y las filas de la matriz M que están relacionadas con $\neg p_i$ con el que $\delta_i = 0$.

La Fig. 3 es un dibujo explicativo de la matriz M_δ. Obsérvese que en la Fig. 3, n = 7, L = 6 y r = 5. Es decir, el conjunto de variables es $\{p_1, ..., p_7\}$, y la matriz M es una matriz (6 filas x 5 columnas). En la Fig. 3, supóngase 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 $\neg p_5$.

Supóngase 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, $\neg p_2$, $\neg p_4$, $\neg p_5$) rodeada de líneas discontinuas es la matriz M_δ. Es decir, la submatriz que consiste en la primera fila (M_1), la 2ª fila (M_2) y la cuarta fila (M_4) de la matriz M es la matriz M_δ.

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

Es decir, en la Fig. 3, el mapa γ(j) = 1 (j = 1, 2, 4), y el mapa γ(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, la inclusión o no de la j-ésima fila de la matriz M en la matriz M_δ se determina en función de si el valor del mapa γ(j) es "0" o "1".

El programa span M^ acepta una secuencia de entrada δ si y sólo si $1^{\rightarrow} \in$ span $<M_\delta>$, y de lo contrario rechaza la secuencia de entrada δ. Concretamente, el programa span 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 a partir de la matriz M^ por la secuencia de entrada δ da $1^{\rightarrow}$. $1^{\rightarrow}$ es un vector de fila que tiene el valor "1" en cada elemento.

Por ejemplo, en la Fig. 3, el programa span M^ acepta la secuencia de entrada δ si y sólo si la combinación lineal de las filas respectivas de la matriz M_δ que consisten en las filas 1ª, 2ª y 4ª de la matriz M da $1^{\rightarrow}$. Es decir, si existen α_1, α_2, y α_4 con los que $\alpha_1(M_1)+ \alpha_2(M_2)+ (\alpha_4(M_4) = 1^{\rightarrow}$, el programa span M^ acepta la secuencia de entrada δ.

El programa span se denomina monótono si sus etiquetas ρ están relacionadas sólo con literales positivos $\{p_1, ..., p_n\}$. El programa span se denomina no monótono si sus etiquetas ρ están relacionadas con los literales $\{p_1, ...,p_n, \neg p_1, ..., \neg p_n\}$. Supóngase que el programa span es no monótono. Se constituye una estructura de acceso (estructura de acceso no monótona) usando el programa span no monótono. En pocas palabras, una estructura de acceso controla el acceso a la encriptación, es decir, controla si un texto encriptado debe ser desencriptado o no.

Debido a que el programa span no es monótono sino no monótono, tal como se describirá más adelante en detalle, se amplía la aplicación del esquema de encriptación funcional constituido usando el programa span.

Obsérvese que en la matriz M, $M_i \neq 0^{\rightarrow}$ para cada número entero i = 1, .., L donde M_i es la i-ésima fila de la matriz M.

<3-2. Producto Interno de los vectores de atributos y la estructura de acceso>

El mapa γ(j) descrito anteriormente se calculará utilizando el producto interno de los vectores de atributo. Concretamente, la determinación de qué fila de la matriz M debe incluirse en la matriz M_δ se realizará usando el producto interno de los vectores de atributo.

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

Supóngase que $U_t := (t, v^{\rightarrow})$ es una variable p del programa span M^ := (M, p), es decir, p:= (t, $v^{\rightarrow}$). Supóngase que el programa span M^:= (M, ρ) que tiene la variable (p := (t, $v^{\rightarrow}$), (t', $v'^{\rightarrow}$), ...) es una estructura de acceso S.

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

Supóngase que r es un conjunto de atributos, es decir, $\Gamma := \{(t, x^{\rightarrow}_t) \mid x^{\rightarrow}_t \in F_q^{nt}, 1 \leq t \leq d\}$.

Cuando se proporciona Γ a la estructura de acceso S, el mapa γ: {1, ..., L} → {0, 1} para el programa span M^ := (M, ρ) se define como sigue. Para cada número entero i = 1, ..., L, se establece γ(i) = 1 si $[\rho(i) = (t, v^{\rightarrow}_i)] \wedge [(t, x^{\rightarrow}_t) \in \Gamma] \wedge [v^{\rightarrow}_i \cdot x^{\rightarrow}_t = 0]$ o $[\rho(i) = \neg(t, v^{\rightarrow}_i)] \wedge [(t, x^{\rightarrow}_t) \in \Gamma] \wedge [v^{\rightarrow}_i \cdot x^{\rightarrow}_t \neq 0]$. Establezca γ(i) = 0 en caso contrario.

Concretamente, el mapa γ se calcula en base al producto interno de los vectores de atributos $v^{\rightarrow}$ y $x^{\rightarrow}$. Tal como se ha descrito anteriormente, la decisión de qué fila de la matriz M debe incluirse en la matriz M_δ está determinada por el mapa γ. Más específicamente, la decisión de qué fila de la matriz M debe incluirse en la matriz M_δ está determinada por el producto interno de los vectores de atributos $v^{\rightarrow}$ y $x^{\rightarrow}$. La estructura de acceso S := (M, ρ) acepta Γ si y sólo si $1^{\rightarrow} \in \text{span} \langle (M_i)_{\gamma(i)=1} \rangle$.

<3-3. Esquema de distribución secreto>

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

El esquema de distribución secreto está distribuyendo información secreta para hacer que sea una información distribuida sin sentido. Por ejemplo, se permite que la información s secreta sea distribuida entre 10 partes para generar 10 piezas de información distribuida. Cada una de las 10 piezas de información distribuida no tiene información acerca de la información s secreta. Por lo tanto, incluso si se obtiene una determinada información distribuida, no puede obtenerse información acerca de la información s secreta. Por otro lado, si se obtienen todas las 10 piezas de información distribuida, puede recuperarse la información s secreta.

También está disponible otro esquema de distribución secreto según el cual incluso cuando no pueden obtenerse la totalidad de las 10 piezas de información distribuida, si pueden obtenerse una o más piezas, pero no todas (por ejemplo, 8 piezas), de información distribuida, entonces la información s secreta puede ser recuperada. Un caso como este en el que la información s secreta puede ser recuperada usando 8 piezas de entre 10 piezas de información distribuida se denominará 8-de-10. Es decir, un caso en el que la información s secreta puede ser recuperada usando t piezas de entre n piezas de información distribuida se denominará t-de-n. Este t se denominará un umbral.

Todavía está disponible otro esquema de distribución secreto según el cual cuando se generan 10 piezas de información d_1, d_{10} distribuida, la información s secreta puede ser recuperada si se proporcionan las 8 piezas de información distribuida d_1, d_8, pero no puede ser recuperada si se proporcionan las 8 piezas de información distribuida d_3, d_{10}. Concretamente, este es un esquema de distribución secreto con el que se controla si la información s secreta puede ser recuperada o no, no sólo por el número de piezas de información distribuida obtenidas, sino también en función de la combinación de la información distribuida.

La Fig. 4 es un dibujo explicativo de s_0. La Fig. 5 es un dibujo explicativo de $S^{\rightarrow T}$.

Sea una matriz M una matriz (L filas x r columnas). Sea $f^{\rightarrow}$ un vector columna indicado en la Fórmula 120.

[Fórmula 120]

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

Sea s_0 indicado en la Fórmula 121 una información s secreta a ser compartida.

[Fórmula 121]

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

Sea $s^{\rightarrow T}$ indicado en la Fórmula 122 un vector de L piezas de información distribuida de s_0.

[Fórmula 122]

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

Supóngase que la información s_i distribuida pertenecen a ρ(i).

Si la estructura de acceso S := (M, ρ) acepta Γ, es decir, $1^{\rightarrow} \in$ span $<(M_i)_{\gamma(i)=1}>$ para γ: {1, ..., L} → {0. 1}, entonces existen constantes $\{\alpha_i \in F_q | \ i \in I\}$ tal que I ⊂ {i ∈ {1, ..., L} | γ(i) = 1}.

Esto es obvio a partir de la explicación de la Fig. 3 en el sentido de que si existe α_1, α_2 y α_4 con los que $\alpha_1(M_1)+ \alpha_2(M_2)+ (\alpha_4(M_4) = 1^{\rightarrow}$, el programa span M^ acepta la secuencia de entrada δ. Es decir, si el programa span M^ acepta la secuencia de entrada δ cuando existe α_1, α_2 y α_4 con los 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 los cuales $\alpha_1(M_1)+ \alpha_2(M_2)+ (\alpha_4(M_4) = 1^{\rightarrow}$.

Obsérvese la Fórmula 123.

[Fórmula 123]

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

Obsérvese que las constantes $\{\alpha_i\}$ puede ser calculadas en tiempo polinomial en el tamaño de la matriz M.

Con el esquema de encriptación funcional según esta y las siguientes realizaciones, se construye una estructura de acceso aplicando el predicado de producto interno y el esquema de distribución secreto al programa span, tal como se ha descrito anteriormente. Por lo tanto, el control de acceso puede ser diseñado de manera flexible diseñando la matriz M en el programa span y la información x de atributos y la información v de atributos (información de predicado) en el predicado de producto interno. Concretamente, el control de acceso puede ser diseñado de manera muy flexible. El diseño de la matriz M corresponde a condiciones de diseño tales como el umbral del esquema de distribución secreto.

Por ejemplo, el esquema de encriptación basado en atributos corresponde a un caso, en la estructura de acceso en el esquema de encriptación funcional según esta realización, en el que el diseño del predicado del producto interno está limitado a cierta condición. Es decir, cuando se compara con la estructura de acceso en el esquema de encriptación funcional según esta realización, la estructura de acceso en el esquema de encriptación basado en atributos tiene una menor flexibilidad en diseño de control de acceso ya que carece de flexibilidad en el diseño de la información x de atributos y la información v de atributo (información de predicado) en el predicado de producto interno. Más específicamente, el esquema de encriptación basado en atributos corresponde a un caso en el que la información de atributo $\{x^{\rightarrow}_t\}_{t \in \{1,\ldots,d\}}$ y $\{v^{\rightarrow}_t\}_{t \in \{1,\ldots,d\}}$ están limitadas a vectores bidimensionales para la relación de igualdad, por ejemplo, $x^{\rightarrow}_t := (1, x_t)$ y $v^{\rightarrow}_t := V_t, -1)$.

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

Más específicamente, debido a que el programa span no monótono incluye un literal negativo (¬p), puede establecerse una condición negativa. Por ejemplo, supóngase que la primera empresa incluye cuatro departamentos de A, B, C y D. Supóngase que debe realizarse un control de acceso de manera que sólo los

usuarios pertenecientes a departamentos distintos del departamento B de la primera empresa puedan acceder (tienen capacidad de desencriptación). En este caso, si no puede establecerse una condición negativa, debe establecerse una condición de que "el usuario pertenece a uno cualquiera de los departamentos A, C y D de la primera empresa". Por otro lado, si puede establecerse una condición negativa, puede establecerse una condición de que "el usuario sea un empleado de la primera empresa y pertenezca a un departamento que no sea el departamento B". Concretamente, debido a que puede establecerse una condición negativa, es posible establecer una condición natural. Aunque el número de departamentos es pequeño en este caso, este esquema es muy efectivo en un caso donde el número de departamentos es grande.

<4. Estructura básica de la encriptación funcional multi-autoridad descentralizada>

<4-1. Estructura Básica de la encriptación funcional multi-autoridad descentralizada>

Se describirá brevemente la estructura de una encriptación funcional multi-autoridad descentralizada.

El esquema de encriptación funcional multi-autoridad descentralizada consiste en cinco algoritmos: GSetup, ASetup, AttrGen, Enc y Dec.

(GSetup)

Un algoritmo GSetup es un algoritmo aleatorio que toma como entrada un parámetro λ de seguridad y genera un parámetro gparam público global. El algoritmo GSetup es ejecutado por una parte determinada. El parámetro gparam público global se publica.

(ASetup)

Un algoritmo ASetup es un algoritmo aleatorio que toma como entrada el parámetro gparam público global, la información t de identificación de autoridad y un formato $\vec{n}$ de atributo, y genera una clave apk_t pública de autoridad y una clave ask_t secreta de autoridad. El algoritmo ASetup es ejecutado por una autoridad t que tiene como información de identificación al menos un t que satisface $1 \leq t \leq d$. La clave apk_t pública de autoridad se divulga, y la clave ask_t secreta de autoridad la posee la autoridad t.

(AttrGen)

Un algoritmo AttrGen es un algoritmo aleatorio que toma como entrada el parámetro gparam público global, la información t de identificación de autoridad, la clave ask_t secreta de autoridad, información gid de identificación del usuario y un atributo $\vec{x}_t := (x_{t,i})$ $(i = 1, ..., n_t) \in F_q$, y emite una clave $usk_{gid(t,xt)}$ de desencriptación. El algoritmo AttrGen es ejecutado por la autoridad t cuando la autoridad t genera una clave de desencriptación relacionada con el atributo $\vec{x}_t$, al usuario indicado por la información gid de identificación. La autoridad t proporciona la clave $usk_{gid(t,xt)}$ de desencriptación al usuario indicado por la información gid de identificación.

(Enc)

Un algoritmo Enc es un algoritmo aleatorio que toma como entrada el parámetro gparam público global, la clave apk_t pública de autoridad, un mensaje $m \in G_T$ y una estructura de acceso S, y emite un texto ct_s encriptado. El algoritmo Enc es ejecutado por un usuario que genera el texto ct_s encriptado.

(Dec)

Un algoritmo Dec es un algoritmo que toma como entrada el parámetro gparam público global, el parámetro apk_t público de autoridad, la clave $usk_{gid(t,xt)}$ de desencriptación y el texto ct_s encriptado, y emite el mensaje m o símbolo $\perp$ distinguido. El algoritmo Dec es ejecutado por un usuario que desencripta el texto ct_s encriptado.

<4-2 Sistema de procesamiento criptográfico 10>

Se describirá el sistema 10 de procesamiento criptográfico que ejecuta los algoritmos de la encriptación funcional multi-autoridad descentralizada descrita anteriormente.

La Fig. 6 es un diagrama de configuración del sistema 10 de procesamiento criptográfico que ejecuta la encriptación funcional multi-autoridad descentralizada.

Un (único) dispositivo de generación de claves 100 ejecuta el algoritmo GSetup tomando como entrada el parámetro λ de seguridad, y genera el parámetro gparam público global. Este dispositivo 100 de generación de claves publica el parámetro gparam público global generado.

Cada dispositivo 100 de generación de claves ejecuta el algoritmo ASetup tomando como entrada el parámetro gparam público global, la información t de identificación asignada a este dispositivo 100 de generación de claves y el formato $\vec{n}$ de atributo, y genera la clave apk_t pública de autoridad y la clave ask_t secreta de autoridad. Cada dispositivo 100 de generación de claves ejecuta el algoritmo AttrGen tomando como entrada global el parámetro público gparam, la información t de identificación asignada a este dispositivo 100 de generación de claves, la clave ask_t secreta de autoridad, la información gid de identificación del usuario y el atributo $\vec{x}_t := (x_{t,i})$ $(i = 1, ..., n_t) \in F_q$, y genera la clave $usk_{gid(t,xt)}$ de desencriptación y la distribuye al dispositivo 300 de desencriptación en secreto.

El dispositivo 200 de encriptación ejecuta el algoritmo Enc tomando como entrada el parámetro gparam público global, la clave apk_t pública de autoridad, el mensaje $m \in G_T$, y la estructura de acceso S, y genera el texto ct_s encriptado. El dispositivo 200 de encriptación transmite el texto ct_s encriptado generado al dispositivo 300 de desencriptación.

El dispositivo 300 de desencriptación ejecuta el algoritmo Dec tomando como entrada el parámetro gparam público global, la clave apk_t pública de autoridad, la clave $usk_{gid(t,xt)}$ de desencriptación y el texto ct_s encriptado, y emite el mensaje m o símbolo ⊥ distinguido.

<4-3. Encriptación funcional multi-autoridad descentralizada y sistema 10 de procesamiento criptográfico en detalle>

La encriptación funcional multi-autoridad descentralizada según la Realización 1, y la función y operación del sistema 10 de procesamiento criptográfico que ejecuta la encriptación funcional multi-autoridad descentralizada se describirán con referencia a las Figs. 7 a 14.

La Fig. 7 es un diagrama de bloques de funciones que muestra la función del dispositivo 100 de generación de claves. La Fig. 8 es un diagrama de bloques de funciones que muestra la función del dispositivo 200 de encriptación. La Fig. 9 es un diagrama de bloques de funciones que muestra la función del dispositivo 300 de desencriptación.

Las Figs. 10 a 12 son diagramas de flujo que muestran la operación del dispositivo 100 de generación de claves. Obsérvese que la Fig. 10 es un diagrama de flujo que muestra el procedimiento del algoritmo GSetup, que la Fig. 11 es un diagrama de flujo que muestra el procedimiento del algoritmo ASetup, y que la Fig. 12 es un diagrama de flujo que muestra el procedimiento del algoritmo AttrGen. La Fig. 13 es un diagrama de flujo que muestra la operación del dispositivo 200 de encriptación y el procedimiento del algoritmo Enc. La Fig. 14 es un diagrama de flujo que muestra la operación del dispositivo 300 de desencriptación y el procedimiento del algoritmo Dec.

Se describirá la función y la operación del dispositivo 100 de generación de claves.

Tal como se muestra en la Fig. 7, el dispositivo 100 de generación de claves está provisto de una parte 110 de generación de clave maestra, una parte 120 de almacenamiento de clave maestra, una parte 130 de entrada de información (primera parte de entrada de información), una parte 140 de generación de clave de desencriptación y una parte 150 de distribución de claves (parte de transmisión de clave de desencriptación).

La parte 110 de generación de clave maestra está provista de una parte 111 de generación de parámetros globales y de una parte 112 de generación de clave secreta de autoridad. La parte 140 de generación de clave de desencriptación está provista de una parte 141 de generación de números aleatorios y de una parte 145 de generación de elemento de clave.

El procedimiento del algoritmo GSetup ejecutado por el dispositivo 100 de generación de claves se describirá en primer lugar con referencia a la Fig. 10. Tal como se ha descrito anteriormente, el algoritmo GSetup puede ser ejecutado por un dispositivo 100 de generación de claves de entre la pluralidad de dispositivos 100 de generación de claves.

(S101: Etapa de entrada de parámetros de seguridad)

Con el dispositivo de entrada, la parte 111 de generación de parámetros globales toma como entrada un parámetro λ de seguridad (1^{λ}).

(S102: Etapa de generación de grupo de emparejamiento bilineal)

Con el dispositivo de procesamiento, la parte 111 de generación de parámetros globales ejecuta el algoritmo G_{bpg} tomando como entrada el parámetro λ de seguridad (1^{λ}) introducido en S101, y genera aleatoriamente los valores de un parámetro $param_G := (q, G, G_T, g, e)$ del grupo de emparejamiento bilineal.

(S103: Etapa de generación de parámetros)

Una función hash H se determina como una función hash indicada en la Fórmula 124.

[Fórmula 124]

$$H : \{0,1\}^* \rightarrow \mathbb{G}$$

Con el dispositivo de procesamiento, la parte 111 de generación de parámetros globales genera los elementos G_0 y G_1 del parámetro gparam global indicado en la Fórmula 125.

[Fórmula 125]

$$G_0 := H(0^{\lambda}) \in \mathbb{G},$$

$$G_1 := H(1^{\lambda}) \in \mathbb{G}$$

La parte 111 de generación de parámetros globales establece también $g_T := e(G_0, G_1)$.

S104: Etapa de almacenamiento de parámetros)

La parte 120 de almacenamiento de la clave maestra almacena el $param_G$ generado en (S102), y la función hash H, los elementos G_0, G_1, y el valor g_T que se establecen en (S103), como el parámetro gparam global en el dispositivo de almacenamiento.

En resumen, de (S101) a (S103), el dispositivo 100 de generación de claves genera el parámetro gparam global ejecutando el algoritmo GSetup indicado en la Fórmula 126. A continuación, en (S104), el dispositivo 100 de generación de claves almacena el parámetro gparam global público generado, en el dispositivo de almacenamiento.

Obsérvese que el parámetro gparam global es publicado a través, por ejemplo, de una red, de manera que otros dispositivos 100 de generación de claves, el dispositivo 200 de encriptación y el dispositivo 300 de desencriptación puedan adquirirlo.

[Fórmula 126]

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

$$H : \{0,1\}^* \rightarrow \mathbb{G};\ G_0 := H(0^{\lambda}) \in \mathbb{G},\ G_1 := H(1^{\lambda}) \in \mathbb{G},\ g_T := e(G_0, G_1),$$

$$\text{devuelve}\ \mathrm{gparam} := (\mathrm{param}_{\mathbb{G}}, H, G_0, G_1, g_T).$$

El procedimiento del algoritmo ASetup ejecutado por el dispositivo 100 de generación de claves se describirá con referencia a la Fig. 11. Tal como se ha descrito anteriormente, el algoritmo ASetup puede ser ejecutado por todos los dispositivos 100 de generación de claves, o por sólo algunos de entre la pluralidad de dispositivos 100 de generación de claves.

(S201: Etapa de entrada de información)

Con el dispositivo de entrada, la parte 130 de entrada de información toma como entrada la información t de identificación asignada a sí misma (su dispositivo 100 de generación de claves). Obsérvese que se asignan diferentes informaciones t de identificación a los dispositivos 100 de generación de claves respectivos.

Por ejemplo, con el dispositivo de comunicación, la parte 130 de entrada de información adquiere el parámetro gparam global a través de la red. Si esta parte 130 de entrada de información pertenece al dispositivo 100 de generación de claves que ha generado el parámetro gparam global, la parte 130 de entrada de información puede leer el parámetro gparam global a partir de la parte 120 de generación de clave maestra.

(S202: Etapa de generación de espacio)

Con el dispositivo de procesamiento, la parte 112 de generación de clave secreta de autoridad ejecuta el algoritmo G_{dpvs} tomando como entrada el parámetro λ de seguridad (1^{λ}), $N_t = 2n_t+u_t+w_t+z_t$, y los valores de $param_G := (q, G, G_T, g, e)$, para generar los valores de un parámetro $param_{Vt} := (q, V_t, G_T, A_t, e)$ de los espacios vectoriales de doble emparejamiento.

Obsérvese que n_t, u_t, w_t y z_t son cada uno un número entero de 1 o más.

(S203) Etapa de generación de base U)

Con el dispositivo de procesamiento, la parte 112 de generación de clave secreta de autoridad genera una base U_j para cada número entero j = 0, 1, tal como se indica en la Fórmula 127.

[Fórmula 127]

$$\mathbb{U}_j := (u_{j,1},\ldots,u_{j,N_t}),$$

$$\text{donde } u_{j,i} := (\overbrace{0,\ldots,0}^{i-1},G_j,\overbrace{0,\ldots,0}^{N_t-i})$$

$$\text{para } j=0,1;\ i=1,\ldots,N_t$$

(S204: Etapa de generación de transformación lineal)

Con el dispositivo de procesamiento, la parte 112 de generación de clave secreta de autoridad toma como entrada N_t y F_q, y genera una transformación lineal $X_t := (\chi_{t,i,j})_{i,j}$ aleatoriamente, tal como se indica en la Fórmula 128.

[Fórmula 128]

$$X_t \xleftarrow{\cup} GL(N_t,\mathbb{F}_q)$$

Obsérvese que GL significa General Lineal. Concretamente, GL es un grupo lineal general, un conjunto de matrices cuadradas en las que el determinante no es 0, y un grupo con respecto a la multiplicación. Obsérvese que $(\chi_{t,i,j})_{i,j}$ significa una matriz relativa a los sufijos i y j de la matriz $\chi_{t,i,j}$ donde i, j = 1, ..., N_t.

(S605: Etapa de generación de base B)

Con el dispositivo de procesamiento, la parte 112 de generación de clave secreta de autoridad genera una base B_t y una base B^*_t, tal como se indica en la Fórmula 129.

[Fórmula 129]

$$(\mathbb{B}_t,\mathbb{B}_t^*) := (X_t(\mathbb{U}_0),(X_t^{\mathrm{T}})^{-1}(\mathbb{U}_1))$$

(S206: Etapa de generación de Base B^)

Con el dispositivo de procesamiento, la parte 112 de generación de clave secreta de autoridad genera una sub-base $B^\wedge_t$ de la base B_t y una sub-base $B^{\wedge *}_t$ de la base B^*_t tal como se indica en la Fórmula 130.

[Formula 130]

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

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

(S207: Etapa de almacenamiento de la clave maestra)

La parte 120 de almacenamiento de la clave maestra almacena el parámetro $param_{Vt}$ generado en (S202), y la sub-base $B^{\wedge}_t$ generada en (S206), en el dispositivo de almacenamiento como una clave apk_t pública de autoridad. La parte 120 de almacenamiento de la clave maestra almacena también la transformación X_t lineal generada en (S204), en el dispositivo de almacenamiento como la clave ask_t secreta de autoridad.

Brevemente, de (S201) a (S206), el dispositivo 100 de generación de claves genera el parámetro apk_t público de autoridad y la clave ask_t secreta de autoridad ejecutando el algoritmo ASetup indicado en la Fórmula 131. A continuación, en (S207), el dispositivo 100 de generación de claves almacena el parámetro apk_t público de autoridad generado y la clave ask_t secreta de autoridad, en el dispositivo de almacenamiento.

Obsérvese que parámetro apk_t público de autoridad es divulgado, por ejemplo, a través de una red, de manera que el dispositivo 200 de encriptación y el dispositivo 300 de desencriptación puedan adquirirlo.

[Fórmula 131]

$$\begin{aligned}
&\mathsf{ASetup}(\mathsf{gparam}, t, n_t, u_t, w_t, z_t):\\
&N_t := 2n_t + u_t + w_t + z_t,\\
&\mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_t, \mathsf{param}_{\mathbb{G}}),\\
&\mathbb{U}_j := (u_{j,1}, \ldots, u_{j,N_t}), \text{ donde } u_{j,i} := (\overbrace{0,\ldots,0}^{i-1}, G_j, \overbrace{0,\ldots,0}^{N_t-i})\\
&\text{para } j = 0,1;\ i = 1,\ldots,N_t,\\
&X_t \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q),\ (\mathbb{B}_t, \mathbb{B}_t^*) := (X_t(\mathbb{U}_0), (X_t^{\mathrm{T}})^{-1}(\mathbb{U}_1)),\\
&\hat{\mathbb{B}}_t := (b_{t,1}, \ldots, b_{t,2n_t}, b_{t,2n_t+u_t+w_t+1}, \ldots, b_{t,2n_t+u_t+w_t+z_t}),\\
&\hat{\mathbb{B}}_t^* := (b_{t,1}^*, \ldots, b_{t,2n_t}^*, b_{t,2n_t+u_t+1}^*, \ldots, b_{t,2n_t+u_t+w_t}^*),\\
&\mathsf{ask}_t := X_t,\ \mathsf{apk}_t := (\mathsf{param}_{\mathbb{V}_t}, \hat{\mathbb{B}}_t),\\
&\text{devuelve } (\mathsf{ask}_t, \mathsf{apk}_t).
\end{aligned}$$

El procedimiento del algoritmo AttrGen ejecutado por el dispositivo 100 de generación de claves se describirá con referencia a la Fig. 12. Obsérvese que, tal como se ha descrito anteriormente, el algoritmo AttrGen es ejecutado por el dispositivo 100 de generación de claves, de entre la pluralidad de dispositivos 100 de generación de claves, que ha ejecutado el algoritmo ASetup.

(S301: Etapa de entrada de información)

Con el dispositivo de entrada, la parte 130 de entrada de información toma como entrada la información t de identificación asignada a sí misma (su dispositivo 100 de generación de claves), la información gid de identificación del usuario al que se va a emitir la clave de desencriptación, y la información de atributo $\vec{x}_t := (x_{t,i})$ $(i = 1, \ldots, n_t)$ indicada en la Fórmula 132.

Por ejemplo, con el dispositivo de comunicación, la parte 130 de entrada de información adquiere también el parámetro gparam global a través de la red. Si esta parte 130 de entrada de información pertenece al dispositivo 100 de generación de claves que ha generado el parámetro gparam global, la parte 130 de entrada de información puede leer el parámetro gparam global a partir de la parte 120 de almacenamiento de clave maestra.

La parte 130 de entrada de información lee también la clave ask_t secreta de autoridad a partir de la parte 120 de almacenamiento de la clave maestra.

[Fórmula 132]

$$\vec{x}_t := (x_{t,1},\ldots,x_{t,n_t}) \in \mathbb{F}_q^{n_t} - \{\ \vec{0}\ \} \text{ de manera que } x_{t,1} := 1$$

(S302: Etapa de generación de números aleatorios)

Con el dispositivo de procesamiento, la parte 141 de generación de números aleatorios genera un número aleatorio $\Phi^{\rightarrow}_t$ para la información t de identificación, tal como se indica en la Fórmula 133.

[Fórmula 133]

$$\vec{\varphi}_t := (\varphi_{t,1},\ldots,\varphi_{t,w_t}) \xleftarrow{\cup} \mathbb{F}_q^{w_t}$$

(S303: Etapa de generación de elementos de clave)

Supóngase que se establece la Fórmula 134.

[Fórmula 134]

$$G_{\mathsf{gid}}(=\delta G_1) := H(\mathsf{gid}) \in \mathbb{G}$$

Con el dispositivo de procesamiento, la parte 142 de generación de elementos de clave genera un elemento k^*_t de clave que es un elemento de la clave $usk_{gid(t,xt)}$ de desencriptación, para la información t de identificación, tal como se indica en la Fórmula 135.

[Fórmula 135]

$$k_t^* := (X_t^{\mathrm{T}})^{-1}((x_{t,1}(G_{\mathsf{gid}}+G_1),\ldots,x_{t,n_t}(G_{\mathsf{gid}}+G_1),-x_{t,1}G_{\mathsf{gid}},\ldots,-x_{t,n_t}G_{\mathsf{gid}},$$
$$0^{u_t},\varphi_{t,1}G_1,\ldots,\varphi_{t,w_t}G_1,0^{z_t})),$$

$$\text{es decir, } k_t^* = (\overbrace{(\delta+1)\vec{x}_t}^{n_t},\ \overbrace{-\delta\vec{x}_t}^{n_t},\ \overbrace{0^{u_t}}^{u_t},\ \overbrace{\vec{\varphi}_t}^{w_t},\ \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*}$$

Tal como se ha descrito anteriormente, para las bases B y B* indicadas en la Fórmula 110, se establece la Fórmula 111. Por lo tanto, la Fórmula 135 significa que el coeficiente para el vector base de una base B^*_t se establece tal como se describe a continuación. Para el propósito de representación simple, un vector base $b^*_{t,i}$ se especifica sólo por su parte i. Por ejemplo, un vector base 1 significa un vector base $b^*_{t,1}$. Los vectores base 1, ..., 3 significan vectores base $b^*_{t,1}$, …, $b^*_{t,3}$, respectivamente.

Cada uno de entre $(\delta+1)x_{t,1}$, ..., $(\delta+1)x_{t,nt}$ (donde nt representa n_t) se establece como el coeficiente para los vectores base 1, ..., n_t. Cada uno de entre $-\delta x_{t,1}$, ..., $-\delta x_{t,nt}$ (donde nt representa n_t) se establece como el coeficiente para los vectores base n_t+1, ..., $2n_t$. 0 se establece como el coeficiente para los vectores base $2n_t+1$, ..., $2n_t+u_t$. Cada uno de los números aleatorios $\Phi_{t,1}$, …, $\Phi_{t,wt}$ (donde wt representa w_t) se establece como el coeficiente para los vectores base $2n_t+u_t+1$, ..., $2n_t+u_t+w_t$. 0 se establece como el coeficiente para los vectores base $2n_t+u_t+w_t+1$, ..., $2n_t+u_t+w_t+z_t$.

(S304: Etapa de distribución de claves)

Por ejemplo, con el dispositivo de comunicación, la parte 150 de distribución de claves distribuye la clave $usk_{gid(t,xt)}$ de desencriptación, constituida como elementos por la información gid de identificación de usuario, la información t de identificación y la información $x^{\rightarrow}_t$ de atributo y el elemento k^*_t de clave al dispositivo 300 de desencriptación en secreto a través de la red. Por supuesto, la clave $usk_{gid(t,xt)}$ de desencriptación puede ser distribuida al dispositivo 300 de desencriptación mediante otro procedimiento.

Brevemente, de (S301) a (S303), el dispositivo 100 de generación de claves genera la clave $usk_{gid(t,xt)}$ de desencriptación ejecutando el algoritmo AttrGen indicado en la Fórmula 136. En (S304), el dispositivo 100 de generación de claves distribuye la clave $usk_{gid(t,xt)}$ de desencriptación generada al dispositivo 300 de desencriptación.

[Fórmula 136]

$$\mathsf{AttrGen}(\mathsf{gparam}, t, \mathsf{ask}_t, \mathsf{gid}, \vec{x}_t := (x_{t,1},\ldots,x_{t,n_t}) \in \mathbb{F}_q^{n_t} - \{\vec{0}\} \text{ tal que } x_{t,1} := 1):$$

$$G_{\mathsf{gid}}(=\delta G_1) := H(\mathsf{gid}) \in \mathbb{G},\ \vec{\varphi}_t := (\varphi_{t,1},\ldots,\varphi_{t,w_t}) \xleftarrow{\cup} \mathbb{F}_q^{w_t},$$

$$k_t^* := (X_t^{\mathrm{T}})^{-1}((x_{t,1}(G_{\mathsf{gid}}+G_1),\ldots,x_{t,n_t}(G_{\mathsf{gid}}+G_1),-x_{t,1}G_{\mathsf{gid}},\ldots,-x_{t,n_t}G_{\mathsf{gid}}, 0^{u_t},\varphi_{t,1}G_1,\ldots,\varphi_{t,w_t}G_1,0^{z_t})),$$

$$\text{es decir, } k_t^* = (\overbrace{(\delta+1)\vec{x}_t}^{n_t}, \overbrace{-\delta\vec{x}_t}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\vec{\varphi}_t}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*},$$

$$\text{devuelve } \mathsf{usk}_{\mathsf{gid},(t,x_t)} := (\mathsf{gid},(t,\vec{x}_t),k_t^*).$$

Se describirán la función y la operación del dispositivo 200 de encriptación.

Tal como se muestra en la Fig. 8, el dispositivo 200 de encriptación está provisto de una parte 210 de adquisición de clave pública, una parte 220 de entrada de información (segunda parte de entrada de información), una parte 230 de generación de texto encriptado y una parte 240 de transmisión de texto encriptado.

La parte 220 de entrada de información está provista de una parte 221 de entrada de información de atributo y una parte 222 de entrada de mensaje. La parte 230 de generación de texto encriptado está provista de una parte 231 de generación de número aleatorio, una parte 232 de generación de vector f, una parte 233 de generación de vector s, una parte 234 de generación de elemento c_i de encriptación y una parte 235 de generación de elemento c_{d+1} de encriptación.

El procedimiento del algoritmo Enc ejecutado por el dispositivo 200 de encriptación se describirá con referencia a la Fig. 13.

(S401: Etapa de adquisición de clave pública)

Por ejemplo, con el dispositivo de comunicación, la parte 210 de adquisición de clave pública adquiere la clave apk_t pública de autoridad generada por cada dispositivo 100 de generación de claves, a través de la red. La parte 210 de adquisición de clave pública adquiere también el parámetro gparam global generado por el dispositivo 100 de generación de claves.

(S402: Etapa de entrada de información)

Con el dispositivo de entrada, la parte 221 de entrada de información de atributo toma como entrada la estructura de acceso S := (M, ρ). La matriz M es una matriz de L filas x r columnas. L y r son cada uno un número entero de 1 o más.

Con el dispositivo de entrada, la parte 220 de entrada de mensaje toma como entrada el mensaje m a ser encriptado.

La estructura de acceso S se establecerá en función del estado del sistema a ser implementado.

(S403: Etapa de generación de vector f)

Con el dispositivo de procesamiento, la parte 232 de generación de vector f genera un vector $\vec{f}$ que tiene r piezas de elementos, aleatoriamente tal como se indica en la Fórmula 137.

[Fórmula 137]

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

(S404: Etapa de generación de vector s)

Con el dispositivo de procesamiento, la parte 233 de generación de vector s genera un vector $s^{\rightarrow T}$, en base a la matriz M de la estructura de acceso S introducida en (S402) y el vector $f^{\rightarrow}$ generado (L filas x r columnas) en (S403) y que tiene r piezas de elementos, tal como se indica en la Fórmula 138.

[Fórmula 138]

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

Con el dispositivo de procesamiento, la parte 233 de generación de vector s genera un valor s_0, en base al vector $f^{\rightarrow}$ generado en (S403), tal como se indica en la Fórmula 139. Obsérvese que $1^{\rightarrow}$ es un vector que tiene un valor 1 en todos sus elementos.

[Fórmula 139]

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

(S405: Etapa de generación de vector f)

Con el dispositivo de procesamiento, la parte 232 de generación de vector f genera un vector $f^{\rightarrow}$' que tiene r piezas de elementos, aleatoriamente tal como se indica en la Fórmula 140 bajo la condición de $s_0 = 1^{\rightarrow} \cdot f^{\rightarrow}$'.

[Fórmula 140]

$$\vec{f}' \xleftarrow{\mathsf{R}} \mathbb{F}_q^r \text{ s.t. } s_0 = \vec{1} \cdot \vec{f}'^{\mathrm{T}}$$

(S406: Etapa de generación de vector s')

Con el dispositivo de procesamiento, la parte 233 de generación de vector s genera un vector $(s^{\rightarrow}')^T$, en base a la matriz M de la estructura de acceso S introducida en (S402) y el vector $f^{\rightarrow}$' (L filas x r columnas) que tiene r piezas de elementos, tal como se indica en la Fórmula 141.

[Fórmula 141]

$$\vec{s}'^{\mathrm{T}} := (s'_1,\ldots,s'_L)^{\mathrm{T}} := M \cdot \vec{f}'^{\mathrm{T}}$$

(S407: Etapa de generación de números aleatorios)

Con el dispositivo de procesamiento, la parte 231 de generación de números aleatorios genera números aleatorios $\eta^{\rightarrow}_i$, θ_i y θ'_i, para cada número entero i = 1, ..., L, tal como se indica en la Fórmula 142.

[Fórmula 142]

$$\vec{\eta}_i := (\eta_{i,1},\ldots,\eta_{i,z_t}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{z_t} \ (i = 1,\ldots,L),$$

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

(S408: Etapa de generación de elemento c_i de encriptación)

Con el dispositivo de procesamiento, la parte 234 de generación de elemento de encriptación genera el elemento c_i de encriptación que es un elemento del texto ct_s encriptado, Para cada número entero i = 1, ..., L, tal como se indica en la Fórmula 143.

[Fórmula 143]

$$\text{para } i = 1, \ldots, L,$$

$$\text{si } \rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q^{n_t} - \{\vec{0}\} \text{ tal que } v_{i,n_t} \neq 0),$$

$$c_i := (\overbrace{s_i \vec{e}_{t,1} + \theta_i \vec{v}_i}^{n_t}, \overbrace{s_i' \vec{e}_{t,1} + \theta_i' \vec{v}_i}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\eta}_i}^{z_t})_{\mathbb{B}_t},$$

$$\text{si } \rho(i) = \neg(t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q^{n_t} - \{\vec{0}\} \text{ tal que } v_{i,n_t} \neq 0),$$

$$c_i := (\overbrace{s_i \vec{v}_i}^{n_t}, \overbrace{s_t' \vec{v}_i}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\eta}_i}^{z_t})_{\mathbb{B}_t}$$

Tal como se ha descrito anteriormente, la Fórmula 111 se establece para las bases B y B* indicadas en la Fórmula 110. Por lo tanto, la Fórmula 143 significa que el coeficiente para el vector base de la base B_t se establece tal como se describe a continuación. Para el propósito de representación simple, un vector base $b_{t,i}$ se especifica sólo por su parte i. Por ejemplo, un vector base 1 significa un vector base $b_{t,1}$. Los vectores base 1, ..., 3 significan vectores base $b_{t,1}$, …, $b_{t,3}$, respectivamente.

Cuando ρ(i) es una tupla positiva (t, $\vec{v}_i$), $s_i+\theta_i v_{i,1}$ se establece como el coeficiente para el vector base 1. Cada uno de entre $\theta_i v_{i,2}$, ..., $\theta_i v_{i,nt}$ (donde nt representa n_t) se establece como el coeficiente para los vectores base 2, ..., n_t. $s_i'+\theta_i' v_{i,1}$ se establece como el coeficiente para un vector base n_t+1. Cada uno de entre $\theta_i' v_{i,2}$, …, $\theta_i' v_{i,nt}$ (donde nt representa n_t) se establece como el coeficiente para los vectores base n_t+2, ..., $2n_t$. 0 se establece como el coeficiente para los vectores base $2n_t+1$, ..., $2n_t+u_t+w_t$. Cada uno de entre $\eta_{i,1}$, ..., $\eta_{i,zt}$ (donde zt representa z_t) se establece como el coeficiente para los vectores base $2n_t+u_t+w_t+1$, ..., $2n_t+u_t+w_t+z_t$.

Cuando ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$), cada uno de entre $s_i v_{i,1}$, …., $s_i v_{i,nt}$ (donde nt representa n_t) se establece como el coeficiente para los vectores base 1, ..., n_t. Cada uno de entre $s_i' v_{i,1}$, …, $s_i' v_{i,nt}$ (donde nt representa n_t) se establece como el coeficiente para los vectores base n_t+1, ..., $2n_t$. 0 se establece como el coeficiente para los vectores base $2n_t+1$, ..., $2n_t+u_t+w_t$. Cada uno de entre $\eta_{i,1}$, ..., $\eta_{i,zt}$ (donde zt representa z_t) se establece como el coeficiente para los vectores base $2n_t+u_t+w_t+1$, ..., $2n_t+u_t+w_t+z_t$.

(S409: Etapa de generación de elemento c_{d+1} de encriptación)

Con el dispositivo de procesamiento, la parte 235 de generación de elemento c_{d+1} de encriptación genera un elemento c_{d+1} de encriptación que es un elemento del texto ct_s encriptado, tal como se indica en la Fórmula 144.

[Fórmula 144]

$$c_{d+1} := g_T^{s_0} m$$

(S410: Etapa de transmisión de datos)

Por ejemplo, con el dispositivo de comunicación, la parte 240 de transmisión de texto encriptado transmite el texto ct_s encriptado, incluyendo la estructura de acceso S := (M, ρ), el elemento c_i de encriptación (i = 1, ..., L), y el elemento c_{d+1} de encriptación, al dispositivo 300 de desencriptación a través de la red. Por supuesto, el texto ct_s encriptado puede ser transmitido al dispositivo 300 de desencriptación mediante otro procedimiento.

Brevemente, de (S401) a (S409), el dispositivo 200 de encriptación genera el texto ct_s encriptado ejecutando el algoritmo Enc indicado en la Fórmula 145. En (S410), el dispositivo 200 de encriptación distribuye el texto ct_s encriptado generado al dispositivo 300 de desencriptación.

[Fórmula 145]

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

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

$$\vec{f}' \xleftarrow{\mathrm{R}} \mathbb{F}_q^r \text{ s.t. } s_0 = \vec{1} \cdot \vec{f}'^{\mathrm{T}},\ \vec{s}'^{\mathrm{T}} := (s_1', \ldots, s_L')^{\mathrm{T}} := M \cdot \vec{f}'^{\mathrm{T}},$$

$$\vec{\eta}_i \xleftarrow{\mathrm{U}} \mathbb{F}_q^{z_t} (i = 1, \ldots, L),$$

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

para $i = 1, \ldots, L$,

si $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q^{n_t} - \{\vec{0}\}$ tal que $v_{i,n_t} \neq 0)$,

$$c_i := (\overbrace{s_i \vec{e}_{t,1} + \theta_i \vec{v}_i}^{n_t}, \overbrace{s_i' \vec{e}_{t,1} + \theta_i' \vec{v}_i}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\eta}_i}^{z_t})_{\mathbb{B}_t},$$

si $\rho(i) = \neg(t, \vec{v}_i)$,

$$c_i := (\overbrace{s_i \vec{v}_i}^{n_t}, \overbrace{s_i' \vec{v}_i}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\eta}_i}^{z_t})_{\mathbb{B}_t},$$

$$c_{d+1} := g_T^{s_0} m, \quad \mathrm{ct}_{\mathbb{S}} := (\mathbb{S}, c_1, \ldots, c_L, c_{d+1}).$$

devuelve $\mathrm{ct}_{\mathbb{S}}$.

Se describirá la función y la operación del dispositivo 300 de desencriptación.

Tal como se muestra en la Fig. 9, el dispositivo 300 de desencriptación está provisto de una parte 310 de recepción de clave de desencriptación (parte de adquisición de clave de desencriptación), una parte 320 de recepción de datos (parte de adquisición de datos), una parte 330 de cálculo de programa span, una parte 340 de cálculo de coeficiente complementario, una parte 350 de operación de emparejamiento y una parte 360 de cálculo de mensaje.

El procedimiento del algoritmo Dec se describirá con referencia a la Fig. 14.

(S501: Etapa de adquisición de clave de desencriptación)

Por ejemplo, con el dispositivo de comunicación, la parte 310 de adquisición de clave de desencriptación recibe la clave $usk_{gid(t,xt)}$ de desencriptación distribuida por el dispositivo 100 de generación de claves, a través de la red. La parte 310 de adquisición de clave de desencriptación adquiere también la clave apk_t pública de autoridad generada por el dispositivo 100 de generación de claves.

(S502: Etapa de recepción de datos)

Por ejemplo, con el dispositivo de comunicación, la parte 320 de recepción de datos recibe el texto ct_s encriptado transmitido por el dispositivo 200 de encriptación, a través de la red.

(S503: Etapa de cálculo del programa span)

Con el dispositivo de procesamiento, la parte 330 de cálculo del programa span comprueba si la estructura de acceso S incluida en el texto ct_s encriptado recibido en (S502) acepta o no el conjunto Γ de la información $\vec{x}_t$ de atributos incluida en la clave $usk_{gid(t,xt)}$ de desencriptación adquirida en (S501). El procedimiento para comprobar si la estructura de acceso S acepta o no Γ es el mismo que el descrito en "3. Concepto para implementar la encriptación funcional".

Si la estructura de acceso S acepta Γ (acepta en S503), la parte 330 de cálculo del programa span avanza el procedimiento a (S504). Si la estructura de acceso S rechaza Γ (rechazo en S503), la parte 330 de cálculo de programa span determina que el texto ct_s encriptado no puede ser desencriptado con la clave $sk_{gid,(t,xt)}$ de desencriptación, y termina el procedimiento.

(S504: Etapa de cálculo del coeficiente complementario)

Con el dispositivo de procesamiento, la parte 340 de cálculo de coeficiente complementario calcula I y una constante (coeficiente complementario) $\{\alpha_i\}_{i\in I}$, cuyos I y $\{\alpha_i\}_{i\in I}$ satisfacen la Fórmula 146.

[Fórmula 146]

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

donde M_i es la i-ésima fila de M, y

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

(S505: Etapa de operación de emparejamiento)

Con el dispositivo de procesamiento, la parte 350 de operación de emparejamiento calcula la Fórmula 147, generando de esta manera una clave de sesión $K = g_T^{s0}$ (donde s0 representa s_0).

[Fórmula 147]

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

Tal como se indica en la Fórmula 148, la clave $K = g_T^{s0}$ (donde s0 representa s_0) se obtiene calculando la Fórmula 147.

[Fórmula 148]

$$\prod_{i\in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_i,k_t^*)^{\alpha_i} \cdot \prod_{i\in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_i,k_t^*)^{\alpha_i/(\vec{v}_i\cdot\vec{x}_t)}$$
$$= \prod_{i\in I \wedge \rho(i)=(t,\vec{v}_i)} g_T^{(\delta+1)\alpha_i s_i - \delta\alpha_i s_i'}$$
$$\prod_{i\in I \wedge \rho(i)=\neg(t,\vec{v}_i)} g_T^{((\delta+1)\alpha_i s_i - \delta\alpha_i s_i')(\vec{v}_i\cdot\vec{x}_t)/(\vec{v}_i\cdot\vec{x}_t)}$$
$$= g_T^{(\delta+1)s_0 - \delta s_0} = g_T^{s_0}.$$

(S506: Etapa de cálculo del mensaje)

La parte 360 de cálculo de mensaje genera un mensaje m' (= m) calculando m' = c_{d+1}/K con el dispositivo de procesamiento. Obsérvese que c_{d+1} es g_T^{s0}m (donde s0 representa s_0), tal como se indica en la Fórmula 144. Debido a que K es g_T^{s0} (donde s0 representa s_0), el mensaje m puede obtenerse calculando m' = c_{d+1}/K.

Brevemente, de (S501) a (S506), el dispositivo 300 de desencriptación genera el mensaje m' (= m) ejecutando el algoritmo Dec indicado en la Fórmula 149.

[Fórmula 149]

$$\mathsf{Dec}(\mathsf{gparam}, \{\mathsf{apk}_t, \mathsf{usk}_{\mathsf{gid},(t,x_t)} := (\mathsf{gid},(t,\vec{x}_t),k_t^*)\},$$
$$\mathsf{ct}_{\mathbb{S}} := (\mathbb{S},c_1,\ldots,c_L,c_{d+1})):$$
$$\text{si } \mathbb{S} := (M,\rho) \text{ acepta } \Gamma := \{(t,\vec{x}_t) \in \mathsf{usk}_{\mathsf{gid},(t,x_t)}\},$$
$$\text{entonces, calcula } I \text{ y } \{\alpha_i\}_{i\in I} \text{ tal que}$$
$$\vec{1} = \sum_{i\in I}\alpha_i M_i, \text{ donde } M_i \text{ es la i-ésima fila de M, y}$$
$$I \subseteq \{i \in \{1,\ldots,L\} \mid [\rho(i)=(t,\vec{v}_i) \wedge (t,\vec{x}_t)\in\Gamma \wedge \vec{v}_i\cdot\vec{x}_t = 0]$$
$$\vee [\rho(i)=\neg(t,\vec{v}_i) \wedge (t,\vec{x}_t)\in\Gamma \wedge \vec{v}_i\cdot\vec{x}_t \neq 0]\},$$
$$K := \prod_{i\in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_i,k_t^*)^{\alpha_i} \cdot \prod_{i\in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_i,k_t^*)^{\alpha_i/(\vec{v}_i\cdot\vec{x}_t)},$$
$$\text{devuelve } m' := c_{d+1}/K.$$

Tal como se ha descrito anteriormente, el sistema 10 de procesamiento criptográfico según la Realización 1 implementa el esquema de encriptación funcional multi-autoridad en el que la pluralidad de dispositivos 100 de generación de claves genera claves de desencriptación. En particular, el esquema de encriptación implementado por el sistema 10 de procesamiento criptográfico es un esquema de encriptación funcional multi-autoridad descentralizada sin autoridad central.

Obsérvese que el sistema 10 de procesamiento criptográfico según la Realización 1 implementa un esquema de encriptación funcional con un predicado no monótono.

En la descripción anterior, las dimensiones u_t, w_t y z_t (t = 1, ..., d) se proporcionan para mejorar la seguridad. Por lo tanto, cada uno de entre u_t, w_t y z_t (t = 1, ..., d) puede establecerse a 0, es decir, no es necesario proporcionar las dimensiones u_t, w_t y z_t (t = 1, ..., d), aunque la seguridad puede ser degradada.

En la descripción anterior, el número de dimensiones de cada una de las bases B_t y la base B^*_t se establece a $N_t = 2n_t+u_t+w_t+z_t$. De manera alternativa, $2n_t+u_t+w_t+z_t$ puede ser reemplazado por $2n_t+3n_t+2n_t+ 1 (= 7n_t +1)$, de manera que el número de dimensiones de cada una de las bases B_t y la base B^*_t puede establecerse a $7n_t+1$.

En este caso, el algoritmo ASetup indicado en la Fórmula 131 se reescribe como la Fórmula 150.

[Fórmula 150]

$$\begin{aligned}
&\mathsf{ASetup}(\mathsf{gparam}, t, n_t):\\
&\quad \mathsf{param}_{\mathbb{V}_t} := (q, \mathbb{V}_t, \mathbb{G}_T, \mathbb{A}_t, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, 7n_t+1, \mathsf{param}_{\mathbb{G}}),\\
&\quad \mathbb{U}_j := (u_{j,1},\ldots,u_{j,7n_t+1}), \text{ donde } u_{j,i} := (\overbrace{0,\ldots,0}^{i-1}, G_j, \overbrace{0,\ldots,0}^{7n_t+1-i})\\
&\quad \text{para } j=0,1; i=1,\ldots,7n_t+1,\\
&\quad X_t \xleftarrow{\cup} GL(7n_t+1, \mathbb{F}_q),\ (\mathbb{B}_t, \mathbb{B}^*_t) := (X_t(\mathbb{U}_0), (X_t^{\mathrm{T}})^{-1}(\mathbb{U}_1)),\\
&\quad \hat{\mathbb{B}}_t := (b_{t,1},\ldots,b_{t,2n_t}, b_{t,7n_t+1}),\ \hat{\mathbb{B}}^*_t := (b^*_{t,1},\ldots,b^*_{t,2n_t}, b^*_{t,5n_t+1},\ldots,b^*_{t,7n_t}),\\
&\quad \mathsf{ask}_t := X_t,\ \mathsf{apk}_t := (\mathsf{param}_{\mathbb{V}_t}, \hat{\mathbb{B}}_t),\\
&\mathsf{devuelve}\ (\mathsf{ask}_t, \mathsf{apk}_t).
\end{aligned}$$

El algoritmo AttrGen indicado en la Fórmula 136 se reescribe como la Fórmula 151.

[Fórmula 151]

$$\begin{aligned}
&\mathsf{AttrGen}(\mathsf{gparam}, t, \mathsf{ask}_t, \mathsf{gid}, \vec{x}_t := (x_{t,1},\ldots,x_{t,n_t})\\
&\quad \in \mathbb{F}_q^{n_t} - \{\vec{0}\} \text{ tal que } x_{t,1} := 1):\\
&\quad G_{\mathsf{gid}}(=\delta G_1) := H(\mathsf{gid}) \in \mathbb{G},\ \vec{\varphi}_t := (\varphi_{t,1},\ldots,\varphi_{t,2n_t}) \xleftarrow{\cup} \mathbb{F}_q^{2n_t},\\
&\quad k^*_t := (X_t^{\mathrm{T}})^{-1}((x_{t,1}(G_{\mathsf{gid}}+G_1),\ldots,x_{t,n_t}(G_{\mathsf{gid}}+G_1), -x_{t,1}G_{\mathsf{gid}},\ldots,-x_{t,n_t}G_{\mathsf{gid}},\\
&\qquad\qquad 0^{3n_t}, \varphi_{t,1}G_1,\ldots,\varphi_{t,2n_t}G_1, 0)),\\
&\quad \text{i.e., } k^*_t = (\overbrace{(\delta+1)\vec{x}_t}^{n_t}, \overbrace{-\delta\vec{x}_t}^{n_t}, \overbrace{0^{3n_t}}^{3n_t}, \overbrace{\vec{\varphi}_t}^{2n_t}, \overbrace{0}^{1})_{\mathbb{B}^*_t},\\
&\quad \mathsf{return}\ \mathsf{usk}_{\mathsf{gid},(t,x_t)} := (\mathsf{gid}, (t, \vec{x}_t), k^*_t).
\end{aligned}$$

El algoritmo Enc indicado en la Fórmula 145 se reescribe como Fórmula 152.

[Fórmula 152]

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

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

$$\vec{f}' \xleftarrow{\mathrm{R}} \mathbb{F}_q^r \ \text{s.t.}\ s_0 = \vec{1} \cdot \vec{f}'^{\mathrm{T}},\ \vec{s}'^{\mathrm{T}} := (s'_1, \ldots, s'_L)^{\mathrm{T}} := M \cdot \vec{f}'^{\mathrm{T}},$$

$$\eta_i, \theta_i, \theta'_i \xleftarrow{\mathrm{U}} \mathbb{F}_q \ (i = 1, \ldots, L),$$

para $i = 1, \ldots, L$,

si $\rho(i) = (t, \vec{v}_i := (v_{i,1}, \ldots, v_{i,n_t}) \in \mathbb{F}_q^{n_t} - \{\vec{0}\}$ tal que $v_{i,n_t} \neq 0)$,

$$c_i := (\overbrace{s_i \vec{e}_{t,1} + \theta_i \vec{v}_i}^{n_t}, \overbrace{s'_i \vec{e}_{t,1} + \theta'_i \vec{v}_i}^{n_t}, \overbrace{0^{3n_t}}^{3n_t}, \overbrace{0^{2n_t}}^{2n_t}, \overbrace{\eta_i}^{1})_{\mathbb{B}_t},$$

si $\rho(i) = \neg(t, \vec{v}_i)$,

$$c_i := (\overbrace{s_i \vec{v}_i}^{n_t}, \overbrace{s'_i \vec{v}_i}^{n_t}, \overbrace{0^{3n_t}}^{3n_t}, \overbrace{0^{2n_t}}^{2n_t}, \overbrace{\eta_i}^{1})_{\mathbb{B}_t},$$

$$c_{d+1} := g_T^{s_0} m, \quad \mathrm{ct}_{\mathbb{S}} := (\mathbb{S}, c_1, \ldots, c_L, c_{d+1}).$$

devuelve $\mathrm{ct}_{\mathbb{S}}$.

El algoritmo GSetup y el algoritmo Dec se mantienen inalterados.

El algoritmo GSetup puede ser ejecutado sólo una vez por un dispositivo 100 de generación en la configuración del sistema 10 de procesamiento criptográfico, y no es necesario que sea ejecutado cada vez que deba generarse una clave de desencriptación. De manera similar, el algoritmo ASetup puede ser ejecutado sólo una vez por cada dispositivo 100 de generación de claves en la configuración del sistema 10 de procesamiento criptográfico y no es necesario que sea ejecutado cada vez que deba generarse una clave de desencriptación.

En la explicación anterior, el algoritmo GSetup, el algoritmo ASetup y el algoritmo KeyGen son ejecutados por el dispositivo 100 de generación de claves. De manera alternativa, el algoritmo GSetup, el algoritmo ASetup y el algoritmo KeyGen pueden ser ejecutados respectivamente por dispositivos diferentes.

Realización 2.

En la Realización 1, se ha descrito el procedimiento de implementación del procedimiento criptográfico en los espacios vectoriales dobles. En la Realización 2, se describirá un procedimiento para implementar un procedimiento criptográfico en grupos aditivos dobles.

Más específicamente, en la Realización 1, el procedimiento criptográfico se implementa en el grupo cíclico del orden primo q. Cuando un anillo R se expresa tal como se indica en la Fórmula 153 usando un número compuesto M, el procedimiento criptográfico descrito en la Realización 1 puede aplicarse también a un grupo aditivo que tiene el anillo R como un coeficiente.

[Fórmula 153]

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

donde

$\mathbb{Z}$: es un número entero, y

M : es un número compuesto

Cuando el esquema de encriptación funcional multi-autoridad descentralizada descrito en la realización 1 se implementa en el grupo aditivo que tiene el anillo R como un coeficiente, se obtienen las Fórmulas 154 a 158.

[Fórmula 154]

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

$$H:\{0,1\}^* \to \mathbb{G};\ \ G_0 := H(0^\lambda) \in \mathbb{G},\ \ G_1 := H(1^\lambda) \in \mathbb{G},\ \ g_T := e(G_0, G_1),$$

$$\text{return}\ \ \mathsf{gparam} := (\mathsf{param}_{\mathbb{G}}, H, G_0, G_1, g_T).$$

[Fórmula 155]

$$\mathsf{ASetup}(\mathsf{gparam}, t, n_t, u_t, w_t, z_t):$$

$$N_t := 2n_t + u_t + w_t + z_t,$$

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

$$\mathbb{U}_j := (u_{j,1},\ldots,u_{j,N_t}),\ \ \text{donde}\ \ u_{j,i} := (\overbrace{0,\ldots,0}^{i-1}, G_j, \overbrace{0,\ldots,0}^{N_t-i})$$

$$\text{para } j = 0,1;\ i = 1,\ldots,N_t,$$

$$X_t \xleftarrow{U} GL(N_t,\mathbb{R}),\ \ (\mathbb{B}_t, \mathbb{B}_t^*) := (X_t(\mathbb{U}_0), (X_t^{\mathrm{T}})^{-1}(\mathbb{U}_1)),$$

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

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

$$\mathsf{ask}_t := X_t,\ \ \mathsf{apk}_t := (\mathsf{param}_{\mathbb{V}_t}, \hat{\mathbb{B}}_t),$$

$$\text{devuelve}\ \ (\mathsf{ask}_t, \mathsf{apk}_t).$$

[Fórmula 156]

AttrGen(gparam, t, ask_t, gid, $\vec{x}_t := (x_{t,1},\ldots,x_{t,n_t})$

$\in \mathbb{R}^{n_t} - \{\vec{0}\}$ tal que $x_{t,1} := 1$):

$G_{\mathsf{gid}}(=\delta G_1) := H(\mathsf{gid}) \in \mathbb{G}$, $\vec{\varphi}_t := (\varphi_{t,1},\ldots,\varphi_{t,w_t}) \xleftarrow{\mathsf{U}} \mathbb{R}^{w_t}$,

$$k_t^* := (X_t^{\mathrm{T}})^{-1}((x_{t,1}(G_{\mathsf{gid}}+G_1),\ldots,x_{t,n_t}(G_{\mathsf{gid}}+G_1),-x_{t,1}G_{\mathsf{gid}},\ldots,-x_{t,n_t}G_{\mathsf{gid}},$$
$$0^{u_t},\varphi_{t,1}G_1,\ldots,\varphi_{t,w_t}G_1,0^{z_t})),$$

es decir, $k_t^* = (\overbrace{(\delta+1)\vec{x}_t}^{n_t}, \overbrace{-\delta\vec{x}_t}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\vec{\varphi}_t}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*}$,

devuelve $\mathsf{usk}_{\mathsf{gid},(t,x_t)} := (\mathsf{gid},(t,\vec{x}_t),k_t^*)$.

[Fórmula 157]

Enc(gparam, $\{\mathsf{apk}_t\}$, m, $\mathbb{S} := (M,\rho)$):

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

$\vec{f}' \xleftarrow{\mathsf{R}} \mathbb{R}^r$ s.t. $s_0 = \vec{1}\cdot\vec{f}'^{\mathrm{T}}$, $\vec{s}'^{\mathrm{T}} := (s_1',\ldots,s_L')^{\mathrm{T}} := M\cdot\vec{f}'^{\mathrm{T}}$,

$\vec{\eta}_i \xleftarrow{\mathsf{U}} \mathbb{R}^{z_t}$ $(i=1,\ldots,L)$,

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

para $i=1,\ldots,L$,

si $\rho(i) = (t,\vec{v}_i := (v_{i,1},\ldots,v_{i,n_t}) \in \mathbb{R}^{n_t} - \{\vec{0}\}$ tal que $v_{i,n_t} \neq 0)$,

$$c_i := (\overbrace{s_i\vec{e}_{t,1}+\theta_i\vec{v}_i}^{n_t}, \overbrace{s_i'\vec{e}_{t,1}+\theta_i'\vec{v}_i}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\eta}_i}^{z_t})_{\mathbb{B}_t},$$

si $\rho(i) = \neg(t,\vec{v}_i)$,

$$c_i := (\overbrace{s_i\vec{v}_i}^{n_t}, \overbrace{s_i'\vec{v}_i}^{n_t}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\eta}_i}^{z_t})_{\mathbb{B}_t},$$

$c_{d+1} := g_T^{s_0} m$, $\mathsf{ct}_{\mathbb{S}} := (\mathbb{S},c_1,\ldots,c_L,c_{d+1})$.

devuelve $\mathsf{ct}_{\mathbb{S}}$.

[Fórmula 158]

$$\mathrm{Dec}(\mathrm{gparam}, \{\mathrm{apk}_t, \mathrm{usk}_{\mathrm{gid},(t,x_t)} := (\mathrm{gid}, (t,\vec{x}_t), k_t^*)\},$$
$$\mathrm{ct}_{\mathbb{S}} := (\mathbb{S}, c_1, \ldots, c_L, c_{d+1})):$$
$$\text{si } \mathbb{S} := (M,\rho) \text{ acepta } \Gamma := \{(t,\vec{x}_t) \in \mathrm{usk}_{\mathrm{gid},(t,x_t)}\},$$
$$\text{a continuación, calcula } I \text{ y } \{\alpha_i\}_{i\in I} \text{ tal que}$$
$$\vec{1} = \sum_{i\in I} \alpha_i M_i, \text{ donde } M_i \text{ es la i-ésima fila de M, y}$$
$$I \subseteq \{i \in \{1,\ldots,L\} \mid [\rho(i) = (t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t = 0]$$
$$\vee [\rho(i) = \neg(t,\vec{v}_i) \wedge (t,\vec{x}_t) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_t \neq 0]\},$$
$$K := \prod_{i\in I \wedge \rho(i)=(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i} \cdot \prod_{i\in I \wedge \rho(i)=\neg(t,\vec{v}_i)} e(c_i, k_t^*)^{\alpha_i/(\vec{v}_i\cdot\vec{x}_t)},$$
$$\text{devuelve } m' := c_{d+1} / K.$$

Desde el punto de vista de la prueba de seguridad, en las realizaciones anteriores, ρ(i) para cada número entero i = 1, ..., L puede limitarse a una tupla positiva (t, $\vec{v}_i$) o una tupla negativa ¬(t, $\vec{v}_i$) para información t de identificación diferente.

En otras palabras, cuando ρ(i) = (t, $\vec{v}_i$) o ρ(i) = ¬(t, $\vec{v}_i$), supóngase que una función ρ~ es un mapa de {1, ..., L} → {1, ..., d} con el que se establece ρ~(I) = t. En este caso, ρ~ puede limitarse a inyección. Obsérvese que ρ(i) es ρ(i) en la estructura de acceso S := (M, ρ(i)) descrita anteriormente.

En la descripción anterior, el programa span 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 $\vec{1}$. De manera alternativa, el programa span M^ puede aceptar la secuencia de entrada δ si y sólo si se obtiene otro vector $\vec{h}$ en lugar de $\vec{1}$.

En este caso, en el algoritmo Enc, puede establecerse $s_0 := \vec{h}\cdot\vec{f}^T$ en lugar de $s_0 := \vec{1}\cdot\vec{f}^T$, y puede establecerse $s_0 := \vec{h}\cdot\vec{f}'^T$ en lugar de $s_0 := \vec{1}\cdot\vec{f}'^T$.

Se describirá la configuración de hardware del sistema 10 de procesamiento criptográfico (el dispositivo 100 de generación de claves, el dispositivo 200 de encriptación y el dispositivo 300 de desencriptación) en las realizaciones anteriores.

La Fig. 15 es un diagrama que muestra un ejemplo de la configuración de hardware de cada dispositivo 100 de generación de claves, dispositivo 200 de encriptación y dispositivo 300 de desencriptación.

Tal como se muestra en la Fig. 15, cada uno de entre el dispositivo 100 de generación de claves, el dispositivo 200 de encriptación y el dispositivo 300 de desencriptación incluye la CPU 911 (denominada también 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, una pantalla LCD 901 (pantalla de cristal líquido), un teclado 902 (K/B), la placa 915 de comunicación y el dispositivo 920 de disco magnético a través de un bus 912 y controla estos dispositivos de hardware. En lugar del dispositivo 920 de disco magnético (dispositivo de disco fijo), puede emplearse un dispositivo de almacenamiento tal como un dispositivo de disco óptico o un dispositivo de lectura/escritura de tarjeta de memoria. El dispositivo 920 de disco magnético está conectado a través de una interfaz de disco fijo predeterminada.

La ROM 913 y el dispositivo 920 de disco magnético 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 920 de disco magnético son ejemplos del dispositivo de almacenamiento (memoria). El teclado 902 y la placa 915 de comunicación son ejemplos de un

dispositivo de entrada. La placa 915 de comunicación es un ejemplo de un dispositivo de comunicación. Además, la pantalla LCD 901 es un ejemplo de un dispositivo de visualización.

El dispositivo 920 de disco magnético, la ROM 913 o similar almacena un sistema operativo 921 (OS), un sistema 922 de ventanas, programas 923 y archivos 924. La CPU 911, el sistema operativo 921 y el sistema 922 de ventanas ejecutan cada programa de entre los programas 923.

Los programas 923 almacenan software y programas que ejecutan las funciones descritas como "parte 110 de generación de clave maestra", "parte 120 de almacenamiento de clave maestra", "parte 130 de entrada de información", "parte 140 de generación de clave de desencriptación", “parte 150 de distribución de claves”, "parte 210 de adquisición de clave pública”, “parte 220 de entrada de información", "parte 230 de generación de texto encriptado", "parte 240 de transmisión de texto encriptado", "parte 310 de recepción de clave de desencriptación", "parte 320 de recepción de datos" “parte 330 de cálculo de programa span", "parte 340 de cálculo de coeficiente complementario", "parte 350 de operación de emparejamiento", "parte 360 de cálculo de mensaje" y similares en la descripción anterior. Los programas 923 almacenan también otros programas. Los programas son leídos y ejecutados por la CPU 911.

Los archivos 924 almacenan información, datos, valores de señal, valores de variables y parámetros tales como el "parámetro gparam global", " clave apk pública de autoridad", " clave ask secreta de autoridad", " clave $usk_{gid(t,xt)}$ de desencriptación” “texto ct_s encriptado", "estructura S de acceso", "información de atributos", "mensaje m" 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. Los datos de información, datos, valores de señal, valores de variables y parámetros almacenados en el medio de grabación tal como el disco o la memoria son leídos a la memoria principal o la memoria caché por la CPU 911 a través de un circuito de lectura/escritura y son usados para las operaciones de la CPU 911 tales como extracción, búsqueda, comparación, cálculo, computación, procesamiento, salida, impresión y visualización. La información, datos, valores de señal, valores de variables y parámetros se almacenan temporalmente en la memoria principal, memoria caché o memoria intermedia durante las operaciones de la CPU 1911 incluyendo extracción, búsqueda, comparación, cálculo, computación, procesamiento, 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 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 una "etapa", "procedimiento" o "proceso". El "dispositivo" puede ser un "circuito", "equipo", "medio" o "función"; o una "etapa", "procedimiento" o "proceso". El "procedimiento" puede ser una "etapa". Es decir, la "parte" puede ser llevada a la práctica como firmware almacenado en la ROM 913. De manera alternativa, la "parte" puede ser llevada a la práctica como sólo software; como sólo hardware tal como un elemento, un dispositivo, un sustrato o una línea de cableado; como una combinación de software y hardware; o además como una combinación de software, hardware y firmware. El firmware y el software se almacenan, como programas, en el medio de grabación tal como la ROM 913. El programa es leído por la CPU 911 y es ejecutado por la CPU 911. Concretamente, el programa causa que el ordenador funcione como una "parte" descrita anteriormente. De manera alternativa, el programa causa que el ordenador o similar ejecute el procedimiento y el método de la "parte" descrita anteriormente.

Lista de Signos de Referencia

100: dispositivo de generación de claves; 110: parte de generación de clave maestra; 111: parte de generación de parámetros globales; 112: parte de generación de clave secreta de autoridad; 120: pieza de almacenamiento de clave maestra; 130: parte de entrada de información; 140: parte de generación de clave de desencriptación; 141: parte de generación de números aleatorios; 142: parte de generación de elementos de clave; 150: parte de distribución de claves; 200: dispositivo de encriptación; 210: parte de adquisición de clave pública; 220: parte de entrada de información; 221: parte de entrada de información de atributos; 222: parte de entrada de mensaje; 230: parte de generación de texto encriptado; 231: parte de generación de números aleatorios; 232: parte de generación de vector f; 233: parte de generación de vector s; 234: parte de generación de elemento c_i de encriptación; 235: parte de generación de elemento c_{d+1} de encriptación; 240: parte de transmisión de texto encriptado; 300: dispositivo de desencriptación; 310: parte de recepción de clave de desencriptación; 320: parte de recepción de datos; 330: parte de cálculo del programa span; 340: parte del cálculo del coeficiente complementario; 350: parte de operación de emparejamiento; 360: parte de cálculo de mensaje

REIVINDICACIONES

1. Un sistema (10) de procesamiento criptográfico que comprende d, en el que d es un número entero de 1 o más, unidades de dispositivos (100) de generación de claves, un dispositivo (200) de encriptación y un dispositivo (300) de desencriptación, y que sirve para ejecutar un procedimiento criptográfico usando una base B_t y una base B^*_t para al menos un número entero t = 1, ..., d,

en el que cada dispositivo de generación de claves de entre las d unidades de dispositivos de generación de claves incluye

una primera parte (130) de entrada de información que toma como entrada información de atributo $\vec{x}_t := (x_{t,i})$, en la que i = 1, ..., n_t donde n_t es un número entero de 1 o más, para un número entero t de entre los números enteros t = 1, ..., d que está predeterminado para cada dispositivo de generación de claves,

una parte (142) de generación de elementos de clave que genera un elemento k^*_t de clave que incluye un vector indicado en la Fórmula 1 basado en el número entero t, la información $\vec{x}_t$ de atributos introducida por la primera parte de entrada de información, un valor δ predeterminado, y un vector base $b^*_{t,i}$ con i = 1, ..., $2n_t$, de la base B^*_t, y

una parte (150) de transmisión de clave de desencriptación que transmite al dispositivo de desencriptación, una clave usk de desencriptación que incluye el elemento k^*_t de clave generado por la parte de generación de elementos de clave y la información $\vec{x}_t$ de atributos,

en el que el dispositivo de encriptación incluye

una segunda parte (220) de entrada de información que toma como entrada una variable ρ(i) para cada número entero i = 1, ..., L, en el que L es un número entero de 1 o más, cuya variable ρ(i) es una de entre una tupla $(t, \vec{v}_i)$ positiva y una tupla $\neg(t, \vec{v}_i)$ negativa de la información t de identificación, en el que t es cualquier número entero de t = 1, ..., d, e información de atributos $\vec{v}_i := (v_{i,i'})$ con i'= 1, ..., n_t; y una matriz M predeterminada que tiene L filas y r columnas, en el que r es un número entero de 1 o más,

una parte (232, 233) de generación de vectores que genera vectores columna $\vec{s}^T := (s_{1, ..., } s_L)^T := M \cdot \vec{f}^T$ basado en un vector $\vec{f}$ que tiene r piezas de elementos y la matriz M introducida por la segunda parte de entrada de información, y genera un vector columna $(\vec{s}')^T := (s_{1, ..., } s_L)^T := M \cdot (\vec{f}')^T$ basado en la matriz M y un vector $\vec{f}'$ que tiene r piezas de elementos y que satisface $s_0 = \vec{h} \cdot (\vec{f}')^T$ donde $s_0 = \vec{h} \cdot \vec{f}^T$,

una parte (234) de generación de elemento c_i de encriptación que, para cada número entero i = 1, ..., L y en base a los vectores columna $\vec{s}^T$ y el vector columna $(\vec{s}')^T$ que son generados por la parte de generación de vectores, y valores θ_i y θ_i' predeterminados para cada número entero i = 1, ..., L, genera un elemento c_i de encriptación, incluyendo un vector indicado en la Fórmula 2, cuando la variable ρ(i) es una tupla positiva $(t, \vec{v}_i)$, usando un vector base $b_{t,i}'$, con i' = 1, ..., $2n_t$, de la base B_t indicado por la información t de identificación de la tupla positiva, y genera un elemento c_i de encriptación incluyendo un vector indicado en la Fórmula 3, cuando la variable ρ(i) es una tupla negativa $\neg(t, \vec{v}_i)$, usando un vector base $b_{t,i}$, con i = 1, ..., $2n_t$, indicado por la información t de identificación de la tupla negativa, y

una parte (240) de transmisión de texto encriptado que transmite al dispositivo de desencriptación, un texto ct_s encriptado incluyendo: el elemento c_i de encriptación generado para cada número entero i = 1, ..., L por la parte de generación de elemento c_i de encriptación; la variable ρ(i); y la matriz M, y

en el que el dispositivo de desencriptación incluye

una parte (310) de recepción de clave de desencriptación que recibe la clave usk de desencriptación transmitida por la parte de transmisión de clave de desencriptación de al menos un dispositivo de generación de clave de entre las d unidades de dispositivos de generación de claves,

una parte (320) de recepción de datos que recibe el texto ct_s encriptado transmitido por la parte de transmisión de texto encriptado,

una parte (340) de cálculo de coeficiente complementario que, en base a la información $\vec{x}_t$ de atributos incluida en la clave usk de desencriptación recibida por la parte de recepción de clave de desencriptación, y la variable ρ(i) incluida en el texto ct_s encriptado recibido por la parte de recepción de datos, especifica, de 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 $(t, \vec{v}_i)$, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla positiva recibida por la parte de recepción de clave de desencriptación, y con la que un producto interno de $\vec{v}_i$ de la tupla positiva y la información $\vec{x}_t$ de atributos indicada por la información t de

identificación de la tupla positiva es igual a 0 y un número entero i para el cual la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$), en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla negativa recibida por la parte de recepción de clave de desencriptación y con la que un producto interno de $\vec{v}_i$ de la tupla negativa y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla negativa no es igual a 0; y calcula, con respecto a i incluido en el conjunto I especificado, un coeficiente α_i complementario con el cual un total de $\alpha_i M_i$ en base a M_i que es un elemento en una i-ésima fila de la matriz M incluida en el texto ct_s encriptado se convierte en el vector $\vec{h}$ predeterminado, y

una parte (350) de operación de emparejamiento que calcula una información K predeterminada realizando una operación de emparejamiento indicada en la Fórmula 4 para el elemento c_i de encriptación incluido en el texto ct_s encriptado y el elemento k^*_t de clave incluido en la clave usk de desencriptación en base al conjunto I y el coeficiente α_i complementario que son calculados por la parte de cálculo del coeficiente complementario.

[Fórmula 1]

$$(\overbrace{(\delta+1)x_{t,1},\ldots,(\delta+1)x_{t,n_t}}^{n_t},\overbrace{-\delta x_{t,1},\ldots,-\delta x_{t,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}^*_t}$$

[Fórmula 2]

$$(\overbrace{s_i+\theta_i v_{i,1},\theta_i v_{i,2},\ldots,\theta_i v_{i,n_t}}^{n_t},\overbrace{s'_i+\theta'_i v_{i,1},\theta'_i v_{i,2},\ldots,\theta'_i v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}_t}$$

[Fórmula 3]

$$(\overbrace{s_i v_{i,1},\ldots,s_i v_{i,n_t}}^{n_t},\overbrace{s'_t v_{i,1},\ldots,s' v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}_t}$$

[Fórmula 4]

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

2. Sistema de procesamiento criptográfico según la reivindicación 1, que ejecuta el procedimiento criptográfico usando, para al menos un número entero t = 1, ..., d, la base B_t que tiene al menos el vector base $b_{t,i}$, en el que i = 1, ..., $2n_t$, …, $2n_t+u_t$, …, $2n_t+u_t+w_t$, …, $2n_t+u_t+w_t+z_t$ y donde u_t, w_t y z_t son cada uno un número entero de 1 o más, y la base B^*_t tiene al menos el vector base $b^*_{t,i}$, en el que i = 1, ..., $2n_t$, …, $2n_t+u_t$, …, $2n_t+u_t+w_t$, …, $2n_t+u_t+w_t+z_t$,

en el que, en el dispositivo de generación de claves,

la parte de generación del elemento de clave genera el elemento k^*_t de clave indicado en la Fórmula 5 para el entero t en base a la información $\vec{x}_t$ de atributos, el valor δ predeterminado y un número $\Phi_{t,i}$ aleatorio para cada número entero i = 1, ..., w_t, y

en el que, en el dispositivo de encriptación,

la parte de generación de elemento c_i de encriptación, para cada número entero i = 1, ..., L y en base al vector columna $\vec{s}^T$ y el vector columna $(\vec{s}')^T$, los valores θ_i y θ_i' predeterminados para cada número entero i = 1, ..., L, y un número $\eta_{i,i'}$ aleatorio para cada número entero i = 1, ..., L y cada número entero i' = 1, ..., z_t, genera el elemento c_i de encriptación indicado en la Fórmula 6 cuando la variable ρ(i) es una tupla (t, $\vec{v}_i$) positiva, y genera el elemento c_i de encriptación indicado en la Fórmula 7 cuando la variable p (i) es una tupla ¬(t, $\vec{v}_i$) negativa.

[Fórmula 5]

$$k_t^* = (\overbrace{(\delta+1)x_{t,1},\ldots,(\delta+1)x_{t,n_t}}^{n_t}, \overbrace{-\delta x_{t,1},\ldots,-\delta x_{t,n_t}}^{n_t},$$

$$\overbrace{0^{u_t}}^{u_t}, \overbrace{\varphi_{t,1},\ldots,\varphi_{t,w_t}}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbf{B}_t^*}$$

[Fórmula 6]

$$c_i := (\overbrace{s_i+\theta_i v_{i,1}, \theta_i v_{i,2},\ldots,\theta_i v_{i,n_t}}^{n_t}, \overbrace{s_i'+\theta_i' v_{i,1}, \theta_i' v_{i,2},\ldots,\theta_i' v_{i,n_t}}^{n_t},$$

$$\overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\eta_{i,1},\ldots,\eta_{i,z_t}}^{z_t})_{\mathbf{B}_t}$$

[Fórmula 7]

$$c_i := (\overbrace{s_i v_{i,1},\ldots,s_i v_{i,n_t}}^{n_t}, \overbrace{s_t' v_{i,1},\ldots,s' v_{i,n_t}}^{n_t},$$

$$\overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\eta_{i,1},\ldots,\eta_{i,z_t}}^{z_t})_{\mathbf{B}_t}$$

3. Sistema de procesamiento criptográfico según la reivindicación 1,

en el que el dispositivo de encriptación comprende además

una parte (235) de elemento c_{d+1} de encriptación que genera un elemento c_{d+1} de encriptación indicado en la Fórmula 8 y obtenido encriptando un mensaje m, en base al s_0 prescrito y un valor g_t que se calcula realizando una operación de emparejamiento del vector base $b_{t,i}$ de la base B_t y el vector base $b^*_{t,i}$ de la base B^*_t para un número entero i predeterminado,

en el que la parte de transmisión de texto encriptado transmite al dispositivo de desencriptación, un texto ct encriptado que incluye el elemento c_i de encriptación y el elemento c_{d+1} de encriptación que es generado por la parte de generación de elemento c_{d+1} de encriptación, y

en el que el dispositivo de desencriptación incluye además

una parte (360) de cálculo de mensaje que calcula el mensaje m dividiendo el elemento c_{d+1} de encriptación por la información K predeterminada calculada por la parte de operación de emparejamiento.

[Fórmula 8]

$$c_{d+1} := g_T^{s_0} m$$

4. Un dispositivo (100) de generación de claves que genera una clave usk de desencriptación en un sistema (10) de procesamiento criptográfico que ejecuta un procedimiento criptográfico usando una base B_t y una base B^*_t para al menos un número entero t = 1, ..., d, en el que d es un número entero de 1 o más, en el que el dispositivo de generación de claves comprende:

una primera parte (130) de entrada de información que toma como entrada información de atributo $\vec{x}_t := (x_{t,i})$, en el que i = 1, ..., n_t donde n_t es un número entero de 1 o más, para un número entero t predeterminado de entre t = 1, ..., d;

una parte (142) de generación de elementos de clave que genera un elemento k^*_t de clave que incluye un vector indicado en la Fórmula 9 en base al entero t, la información $\vec{x}_t$ de atributos introducida por la primera parte de entrada de información, un valor δ predeterminado y un vector base $b^*_{t,i}$, en el que i = 1, ..., $2n_t$, de la base B^*_t; y

una parte (150) de transmisión de clave de desencriptación que transmite a un dispositivo de desencriptación, una clave usk de desencriptación que incluye el elemento k^*_t de clave generado por la parte de generación de elementos de clave y la información $\vec{x}_t$ de atributos.

[Fórmula 9]

$$(\overbrace{(\delta+1)x_{t,1},\ldots,(\delta+1)x_{t,n_t}}^{n_t},\overbrace{-\delta x_{t,1},\ldots,-\delta x_{t,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}^*_t}$$

5. Un dispositivo (200) de encriptación que genera un texto ct_s encriptado en un sistema (10) de procesamiento criptográfico que ejecuta un procedimiento criptográfico usando una base B_t y una base B^*_t para al menos un número entero t = 1, ..., d, en el que d es un número entero de 1 o más, en el que el dispositivo de encriptación comprende:

una segunda parte (220) de entrada de información que toma como entrada una variable ρ(i) para cada número entero i = 1, ..., L, en el que L es un número entero de 1 o más, cuya variable ρ(i) es una de entre una tupla (t, $\vec{v}_i$) positiva y una tupla ¬(t, $\vec{v}_i$) negativa de la información t de identificación, en el que t es cualquier número entero de entre t = 1, ..., d, e información de atributos $\vec{v}_i := (v_{i,i'})$ con 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, donde r es un número entero de 1 o más;

una parte (232, 233) de generación de vector que genera un vector columna $\vec{s}^T := (s_{1,\ ...,\ }s_L)^T := M\cdot\vec{f}^T$ en base a un vector $\vec{f}$ que tiene r piezas de elementos y la matriz M introducida por la segunda parte de entrada de información, y genera un vector columna $(\vec{s}')^T := (s_1',\ ...,\ s_L')^T := M\cdot(\vec{f}')^T$ en base a la matriz M y un vector $\vec{f}'$ que tiene r piezas de elementos y que satisface $s_0 = \vec{h}\cdot(\vec{f}')^T$ donde $s_0 = \vec{h}\cdot\vec{f}^T$;

un elemento c_i de encriptación (234) que, para cada número entero i = 1, ..., L y en base al vector columna $\vec{s}^T$ y el vector columna $(\vec{s}')^T$ que son generados por la parte de generación de vectores, y valores θ_i y θ_i' predeterminados para cada número entero i = 1, ..., L, genera un elemento c_i de encriptación que incluye un vector indicado en la Fórmula 10, cuando la variable ρ(i) es una tupla (t, $\vec{v}_i$) positiva, usando un vector base $b_{t,i'}$, con i' = 1, ..., $2n_t$, de la base B_t indicada por la información t de identificación de la tupla positiva, y genera un elemento c_i de encriptación que incluye un vector indicado en la Fórmula 11, cuando la variable ρ(i) es una tupla ¬(t, $\vec{v}_i$) negativa, usando un vector base $b_{t,i}$, con i = 1, ..., $2n_t$, indicado por la información t de identificación de la tupla negativa; y

una parte (240) de transmisión de texto encriptado que transmite a un dispositivo de desencriptación, el texto ct_s encriptado incluyendo: el elemento c_i de encriptación generado para cada número entero i = 1, ..., L por la parte de generación de elemento c_i de encriptación; la variable ρ(i); y la matriz M.

[Fórmula 10]

$$(\overbrace{s_i+\theta_i v_{i,1},\theta_i v_{i,2},\ldots,\theta_i v_{i,n_t}}^{n_t},\overbrace{s_i'+\theta_i' v_{i,1},\theta_i' v_{i,2},\ldots,\theta_i' v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}_t}$$

[Fórmula 11]

$$(\overbrace{s_i v_{i,1},\ldots,s_i v_{i,n_t}}^{n_t},\overbrace{s_t' v_{i,1},\ldots,s' v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}_t}$$

6. Un dispositivo (300) de desencriptación que desencripta un texto ct_s encriptado por una clave usk de desencriptación en un sistema (10) de procesamiento criptográfico que ejecuta un procedimiento criptográfico

usando una base B_t y una base B^*_t para al menos un número entero t = 1, ..., d, en el que d es un número entero de 1 o más, en el que el dispositivo de desencriptación comprende:

una parte (310) de recepción de clave de desencriptación que recibe la clave usk de desencriptación que incluye, para al menos uno de los números enteros t de entre t = 1, ..., d, información de atributos $\vec{x}_t := (x_{t,i})$, en el que i = 1, ..., n_t donde n_t es un número entero de 1 o más, y un elemento k^*_t de clave que se genera para incluir un vector indicado en la Fórmula 12;

una parte (320) de recepción de datos que recibe el texto ct_s encriptado que incluye: una variable ρ(i) para cada número entero i = 1, ..., L, en el que L es un número entero de 1 o más, cuya variable ρ(i) es una de entre una tupla $(t, \vec{v}_i)$ positiva y una tupla $\neg(t, \vec{v}_i)$ negativa de información t de identificación, en el que t es un número entero cualquiera de entre t = 1, ..., d, e información de atributos $\vec{v}_i := (v_{i,i'})$, en el que i' = 1, ..., n_t; una matriz M predeterminada que tiene L filas y r columnas, en el que r es un número entero de 1 o más; y un elemento c_i de encriptación generado para incluir un vector indicado en la Fórmula 13 para cada número entero i = 1, .., L;

una parte (340) de cálculo de coeficiente complementario que, en base a la información $\vec{x}_t$ de atributos incluida en la clave usk de desencriptación recibida por la parte de recepción de clave de desencriptación, y la variable ρ(i) incluida en el texto ct_s encriptado recibido por la parte de recepción de datos, especifica, de 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 $(t, \vec{v}_i)$ positiva, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla positiva recibida por la parte de recepción de clave de desencriptación, y con la que un producto interno de $\vec{v}_i$ de la tupla positiva y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla positiva es igual a 0 y un número entero i para el cual la variable ρ(i) es una tupla $\neg(t, \vec{v}_i)$ negativa, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla negativa recibida por la parte de recepción de clave de desencriptación y con la que un producto interno de $\vec{v}_i$ de la tupla negativa y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla negativa no es igual a 0; y calcula, con respecto a i incluido en el conjunto I especificado, un coeficiente α_i complementario con lo que un total de $\alpha_i M_i$ basado en M_i que es un elemento en una i-ésima fila de la matriz M incluida en el texto ct_s encriptado se convierte en un vector $\vec{h}$ predeterminado; y

una parte (350) de operación de emparejamiento que calcula información K predeterminada llevando a cabo una operación de emparejamiento indicada en la Fórmula 14 para el elemento c_i de encriptación incluido en el texto ct_s encriptado y el elemento k^*_t de clave incluido en la clave usk de desencriptación basado en el conjunto I y el coeficiente α_i complementario que son calculados por la parte de cálculo del coeficiente complementario.

[Fórmula 12]

$$(\overbrace{(\delta+1)x_{t,1},\ldots,(\delta+1)x_{t,n_t}}^{n_t},\overbrace{-\delta x_{t,1},\ldots,-\delta x_{t,n_t},}^{n_t},0,\ldots,0)_{\mathbf{B}^*_t}$$

donde δ es un valor predeterminado.

[Fórmula 13]

si $\rho(i) = (t, \vec{v}_i)$,

$$(\overbrace{s_i + \theta_i v_{i,1}, \theta_i v_{i,2}, \ldots, \theta_i v_{i,n_t}}^{n_t}, \overbrace{s'_i + \theta'_i v_{i,1}, \theta'_i v_{i,2}, \ldots, \theta'_i v_{i,n_t}}^{n_t}, 0, \ldots, 0)_{\mathbf{B}_t},$$

si $\rho(i) = \neg(t, \vec{v}_i)$,

$$(\overbrace{s_i v_{i,1}, \ldots, s_i v_{i,n_t}}^{n_t}, \overbrace{s'_t v_{i,1}, \ldots, s' v_{i,n_t}}^{n_t}, 0, \ldots, 0)_{\mathbf{B}_t}$$

donde

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

$\vec{f}$ es un vector predeterminado que tiene r piezas de elementos

$s_0 = \vec{h} \cdot \vec{f}^{\mathrm{T}}$,

$\vec{h}$ es un vector predeterminado que tiene r piezas de elementos

$\vec{f}'$ es un vector predeterminado que satisface $s_0 = \vec{h} \cdot \vec{f}'^{\mathrm{T}}$,

$\vec{s}'^{\mathrm{T}} := (s'_1, \ldots, s'_L)^{\mathrm{T}} := M \cdot \vec{f}'^{\mathrm{T}}$,

$\vec{s}'^{\mathrm{T}} := (s'_1, \ldots, s'_L)^{\mathrm{T}} := M \cdot \vec{f}'^{\mathrm{T}}$, y

θ_i, θ'_i son valores predeterminados.

[Fórmula 14]

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

7. Un procedimiento de procesamiento criptográfico para ejecutar un procedimiento criptográfico usando una base B_t y una base B^*_t para al menos un número entero t = 1, ..., d, en el que d es un número entero de 1 o más, en el que el procedimiento de procesamiento criptográfico comprende:

una primera etapa de entrada de información para, con al menos un dispositivo de generación de claves de entre una pluralidad de dispositivos (100) de generación de claves, tomar como entrada información de atributos $\vec{x}_t := (x_{t,i})$, en el que i = 1, ..., n_t donde n_t es un número entero de 1 o más, para un número entero t de entre los números enteros t = 1, ..., d que está predeterminado para cada uno de los dispositivos de generación de claves;

una etapa de generación de elementos de clave para, con al menos un dispositivo de generación de claves, generar un elemento k^*_t de clave que incluye un vector indicado en la Fórmula 15 basado en el número entero t, la información $\vec{x}_t$ de atributos introducida en la primera etapa de entrada de información, un valor δ predeterminado, y un vector base $b^*_{t,i}$, en el que i = 1, ..., $2n_t$, de la base B^*_t;

una etapa de transmisión de clave de desencriptación para, con al menos un dispositivo de generación de claves, transmitir a un dispositivo (300) de desencriptación una clave de desencriptación que incluye el elemento k^*_t de clave generado en la etapa de generación de elementos de clave y la información $\vec{x}_t$ de atributos;

una segunda etapa de entrada de información para, con un dispositivo (200) de encriptación, tomar como entrada una variable ρ(i) para cada número entero i = 1, ..., L, en el que L es un número entero de 1 o más, cuya variable ρ (i) es una de entre una tupla (t, $\vec{v}_i$) positiva y una tupla ¬(t, $\vec{v}_i$) negativa de la información t de identificación, en el que t es cualquier número entero de entre t = 1, ..., d, e información de atributos $\vec{v}_i := (v_{i,i'})$, en el que i' = 1, ..., n_t; y una matriz M predeterminada que tiene L filas y r columnas, en el que r es un número entero de 1 o más;

una etapa de generación de vectores para, con el dispositivo de encriptación, generar un vector columna $\vec{s}^T := (s_{1, ..., } s_L)^T := M \cdot \vec{f}^T$ basado en un vector $\vec{f}$ que tiene r piezas de elementos y la matriz M introducida en la segunda etapa de entrada de información, y generar un vector columna $(\vec{s}')^T := (s_1', ..., s_L')^T := M \cdot (\vec{f}')^T$ basado en la matriz M y un vector $\vec{f}$' que tiene r piezas de elementos y satisface $s_0 = \vec{h} \cdot (\vec{f}')^T$ en el que $s_0 = \vec{h} \cdot \vec{f}^T$;

una etapa de generación de elemento c_i de encriptación para, con el dispositivo de encriptación, para cada número entero i = 1, ..., L y basado en el vector columna $\vec{s}^T$ y el vector columna $(\vec{s}')^T$ que son generados en la etapa de generación de vectores, y valores θ_i y θ_i' predeterminados para cada número entero i = 1, ..., L, generar un elemento c_i de encriptación que incluye un vector indicado en la Fórmula 16, cuando la variable ρ(i) es una tupla (t, $\vec{v}_i$) positiva, usando un vector base $b_{t,i'}$, en el que i' = 1, ..., $2n_t$, de la base B_t indicado por la información t de identificación de la tupla positiva, y generar un elemento c_i de encriptación que incluye un vector indicado en la Fórmula 17, cuando la variable ρ(i) es una tupla negativa ¬(t, $\vec{v}_i$), usando un vector base $b_{t,i}$, en el que i = 1, ..., $2n_t$, indicado por la información t de identificación de la tupla negativa;

una etapa de transmisión de texto encriptado para, con el dispositivo de encriptación, transmitir al dispositivo de desencriptación, un texto ct_s encriptado que incluye: el elemento c_i de encriptación generado para cada número entero i = 1, ..., L en la etapa de generación de elemento c_i de encriptación; la variable ρ(i); y la matriz M;

una etapa de recepción de clave de desencriptación para, con el dispositivo de desencriptación, recibir la clave de desencriptación transmitida en la etapa de transmisión de clave de desencriptación de al menos un dispositivo de generación de claves de entre la pluralidad de dispositivos de generación de claves;

una etapa de recepción de datos para recibir el texto ct_s encriptado transmitido en la etapa de transmisión de texto encriptado;

una etapa de cálculo de coeficiente complementario para, con el dispositivo de desencriptación y en base a la información de atributos $\vec{x}_t$ incluida en la clave usk de desencriptación recibida en la etapa de recepción de clave de desencriptación, y la variable ρ(i) incluida en el texto ct_s encriptado recibido en la etapa de recepción de datos, especificar, de 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 (t, $\vec{v}_i$) positiva, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla positiva recibida en la etapa de recepción de clave de desencriptación, y con el que un producto interno de $\vec{v}_i$ de la tupla positiva y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla positiva es igual a 0, y un número entero i para el cual la variable ρ(i) es una tupla ¬(t, $\vec{v}_i$) negativa, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla negativa recibida en la etapa de recepción de clave de desencriptación, y con el que un producto interno de $\vec{v}_i$ de la tupla negativa y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla negativa no es igual a 0; y calcular, con respecto a i incluido en el conjunto I especificado, un coeficiente α_i complementario con el cual un total de $\alpha_i M_i$ basado en M_i que es un elemento en una i-ésima fila de la matriz M incluida en el texto ct_s encriptado se convierte en un vector $\vec{h}$ predeterminado; y

una etapa de operación de emparejamiento para, con el dispositivo de desencriptación, calcular información K predeterminada realizando una operación de emparejamiento indicada en la Fórmula 18 para el elemento c_i de encriptación incluido en el texto ct_s encriptado y el elemento k^*_t de clave incluido en la clave usk de desencriptación basado en el conjunto I y el coeficiente α_i complementario que se calculan en la etapa de cálculo del coeficiente complementario.

[Fórmula 15]

$$(\overbrace{(\delta+1)x_{t,1},\ldots,(\delta+1)x_{t,n_t}}^{n_t},\overbrace{-\delta x_{t,1},\ldots,-\delta x_{t,n_t},}^{n_t},0,\ldots,0)_{\mathbf{B}_t^*}$$

[Fórmula 16]

$$(\overbrace{s_i+\theta_i v_{i,1},\theta_i v_{i,2},\ldots,\theta_i v_{i,n_t}}^{n_t},\overbrace{s_i'+\theta_i' v_{i,1},\theta_i' v_{i,2},\ldots,\theta_i' v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}_t}$$

[Fórmula 17]

$$(\overbrace{s_i v_{i,1},\ldots,s_i v_{i,n_t},}^{n_t}\overbrace{s_t' v_{i,1},\ldots,s' v_{i,n_t},}^{n_t}0,\ldots,0)_{\mathbf{B}_t}$$

[Fórmula 18]

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

8. Un programa de procesamiento criptográfico que comprende un programa de generación de claves a ser ejecutado sobre d, en el que d es un número entero de 1 o más, unidades de dispositivos (100) de generación de claves, un programa de encriptación a ser ejecutado en un dispositivo (200) de encriptación y un programa de desencriptación a ser ejecutado en un dispositivo (300) de desencriptación, y que sirve para ejecutar un procedimiento criptográfico usando una base B_t y una base B^*_t para al menos un número entero t = 1, ..., d,

en el que el programa de generación de claves causa que un ordenador ejecute

un primer procedimiento de entrada de información que toma como entrada información de atributos $\vec{x}_t := (x_{t,i})$, en el que i = 1, ..., n_t donde n_t es un número entero de 1 o más, para un número entero t de entre los números enteros t = 1, ..., d que está predeterminado para cada uno de los dispositivos de generación de claves,

un procedimiento de generación de elementos de clave para generar un elemento k^*_t de clave que incluye un vector indicado en la Fórmula 19 basado en el entero t, la información $\vec{x}_t$ de atributos introducida en el primer procedimiento de entrada de información, un valor δ predeterminado, y un vector base $b^*_{t,i}$, en el que i = 1, ..., $2n_t$, de la base B^*_t, y

un procedimiento de transmisión de clave de desencriptación para transmitir al dispositivo de desencriptación, una clave usk de desencriptación que incluye el elemento k^*_t de clave generado en el procedimiento de generación de elementos de clave y la información $\vec{x}_t$ de atributos,

en el que el programa de encriptación causa que el ordenador ejecute

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

un procedimiento de generación de vectores para generar un vector columna $\vec{s}^T := (s_1, \ldots, s_L)^T := M\cdot\vec{f}^T$ basado en un vector $\vec{f}$ que tiene r piezas de elementos y la matriz M introducida en el segundo procedimiento de entrada de información, y generar un vector columna $(\vec{s}')^T := (s_1', \ldots, s_L')^T := M\cdot(\vec{f}')^T$ basado en la matriz M y un vector $\vec{f}'$ que tiene r piezas de elementos y satisface $s_0 = \vec{h}\cdot(\vec{f}')^T$ donde $s_0 = \vec{h}\cdot\vec{f}^T$,

un elemento c_i de encriptación para, para cada número entero i = 1, ..., L y en base al vector columna $\vec{s}^T$ y el vector columna $(\vec{s}')^T$ que son generados en el procedimiento de generación de vectores, y valores θ_i y θ_i' predeterminados para cada número entero i = 1, ..., L, generar un elemento c_i de encriptación que incluye un vector indicado en la Fórmula 20, cuando la variable ρ(i) es una tupla $(t, \vec{v}_i)$ positiva, usando un vector base

$b_{t,i'}$, en el que i' = 1, ..., $2n_t$, de la base B_t indicada por la información t de identificación de la tupla positiva, y generar un elemento c_i de encriptación que incluye un vector indicado en la Fórmula 21, cuando la variable ρ(i) es una tupla ¬(t, $\vec{v}_i$) negativa, usando un vector base $b_{t,i}$, en el que i = 1, ..., $2n_t$, indicado por la información t de identificación de la tupla negativa, y

un procedimiento de transmisión de texto encriptado para transmitir al dispositivo de desencriptación, un texto ct_s encriptado que incluye: el elemento c_i de encriptación generado para cada número entero i = 1, ..., L en el procedimiento de generación de elemento c_i de encriptación; la variable ρ(i); y la matriz M, y

en el que el programa de desencriptación causa que el ordenador ejecute

un procedimiento de recepción de clave de desencriptación para recibir la clave usk de desencriptación transmitida en el procedimiento de transmisión de clave de desencriptación,

un procedimiento de recepción de datos para recibir el texto ct_s encriptado transmitido en el procedimiento de transmisión de texto encriptado,

un procedimiento de cálculo de coeficiente complementario para, en base a la información de atributos $\vec{x}_t$ incluida en la clave usk de desencriptación recibida en el procedimiento de recepción de clave de desencriptación, y la variable ρ(i) incluida en el texto ct_s encriptado recibido en el procedimiento de recepción de datos, especificar, de 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 (t, $\vec{v}_i$) positiva, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla positiva recibida en el procedimiento de recepción de clave de desencriptación, y con la que un producto interno de $\vec{v}_i$ de la tupla positiva y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla positiva es igual a 0 y un número entero i para el cual la variable ρ(i) es una tupla ¬(t, $\vec{v}_i$) negativa, en el que la clave usk de desencriptación incluye $\vec{x}_t$ indicado por la información t de identificación de la tupla negativa recibida en el procedimiento de recepción de clave de desencriptación, y con la que un producto interno de $\vec{v}_i$ de la tupla negativa y la información $\vec{x}_t$ de atributos indicada por la información t de identificación de la tupla negativa no es igual a 0; y calcular, con respecto a i incluido en el conjunto I especificado, un coeficiente α_i complementario con el que un total de $\alpha_i M_i$, basado en M_i que es un elemento en una i-ésima fila de la matriz M incluida en el texto ct_s de encriptación se convierte en un vector $\vec{h}$ predeterminado, y

un procedimiento de operación de emparejamiento para calcular información K predeterminada mediante la realización de una operación de emparejamiento indicada en la Fórmula 22 para el elemento c_i de encriptación incluido en el texto ct_s encriptado y el elemento k^*_t de clave incluido en la clave usk de desencriptación en base al conjunto I y el coeficiente α_i complementario que se calculan en el procedimiento de cálculo del coeficiente complementario.

[Fórmula 19]

$$(\overbrace{(\delta+1)x_{t,1},\ldots,(\delta+1)x_{t,n_t}}^{n_t},\overbrace{-\delta x_{t,1},\ldots,-\delta x_{t,n_t},}^{n_t},0,\ldots,0)_{\mathbf{B}_t^*}$$

[Fórmula 20]

$$(\overbrace{s_i+\theta_i v_{i,1},\theta_i v_{i,2},\ldots,\theta_i v_{i,n_t}}^{n_t},\overbrace{s_i'+\theta_i' v_{i,1},\theta_i' v_{i,2},\ldots,\theta_i' v_{i,n_t}}^{n_t},0,\ldots,0)_{\mathbf{B}_t}$$

[Fórmula 21]

$$(\overbrace{s_i v_{i,1},\ldots,s_i v_{i,n_t},}^{n_t}\overbrace{s_t' v_{i,1},\ldots,s' v_{i,n_t},}^{n_t}0,\ldots,0)_{\mathbf{B}_t}$$

[Fórmula 22]

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

Fig. 1

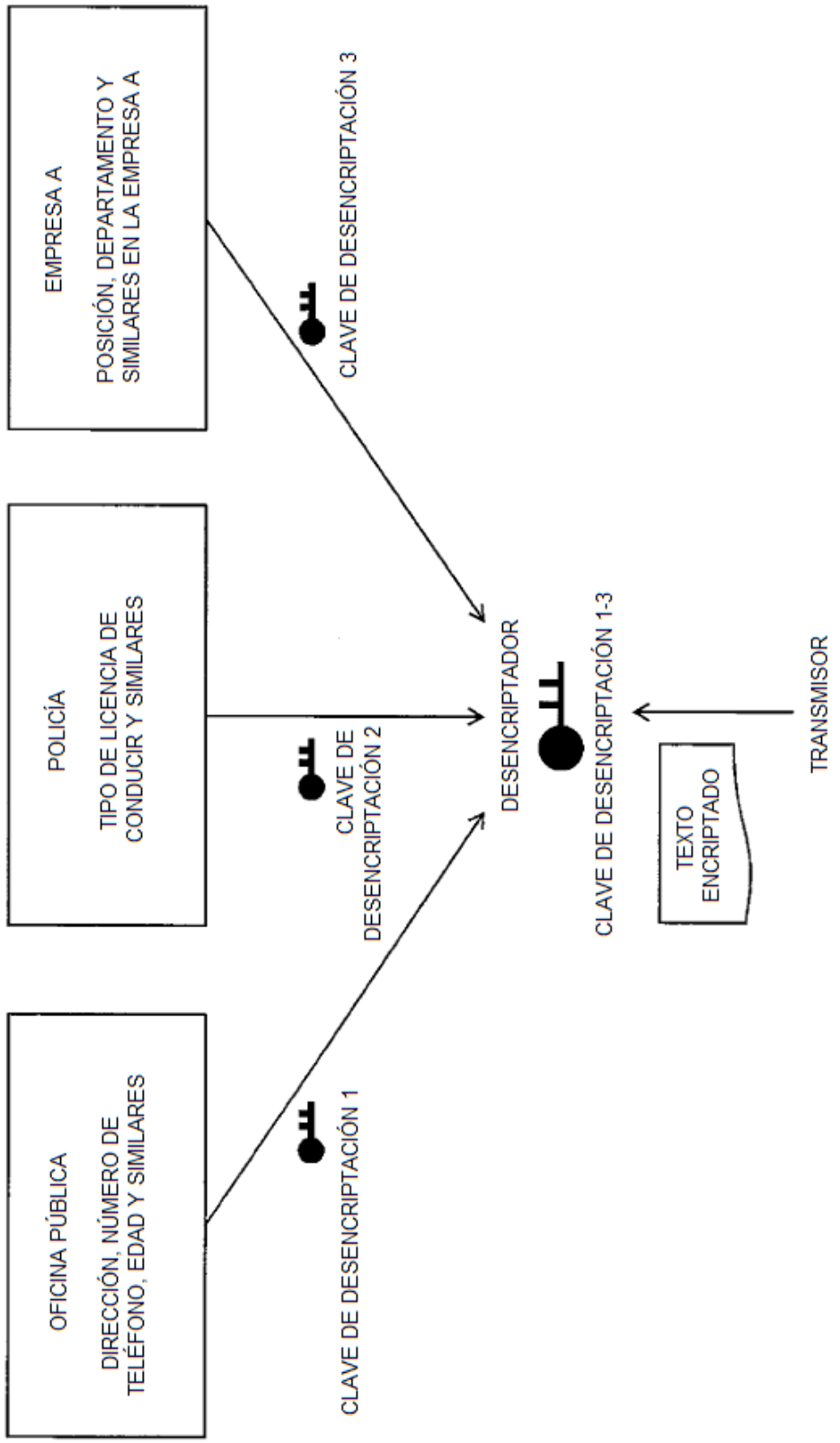

EMPRESA A
POSICIÓN, DEPARTAMENTO Y SIMILARES EN LA EMPRESA A
POLICÍA
TIPO DE LICENCIA DE CONDUCIR Y SIMILARES
OFICINA PÚBLICA
DIRECCIÓN, NÚMERO DE TELÉFONO, EDAD Y SIMILARES
CLAVE DE DESENCRIPTACIÓN 3
CLAVE DE DESENCRIPTACIÓN 2
CLAVE DE DESENCRIPTACIÓN 1
DESENCRIPTADOR
CLAVE DE DESENCRIPTACIÓN 1-3
TEXTO ENCRIPTADO
TRANSMISOR

Fig. 2

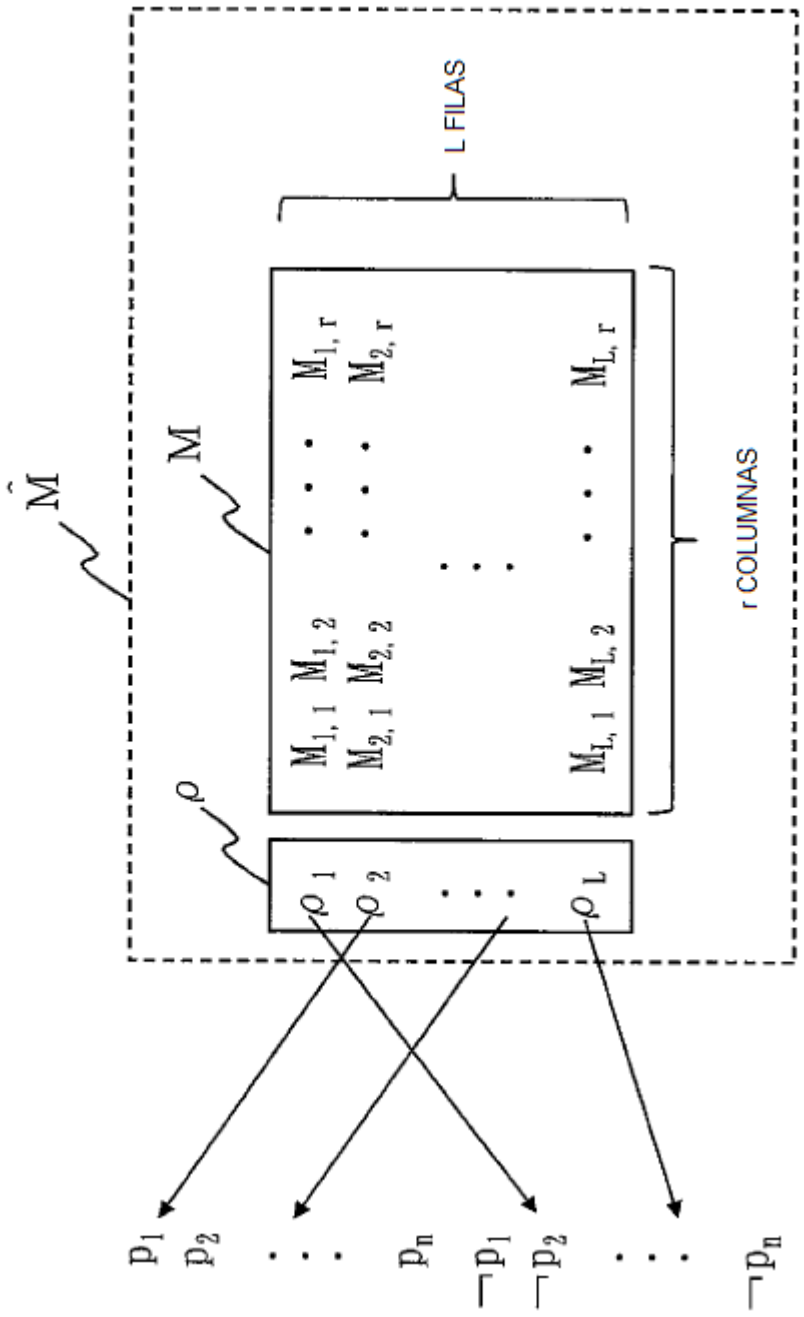

L FILAS
r COLUMNAS
$\hat{M}$
M
ρ
ρ_1 ρ_2 ρ_L
$M_{1,1}$ $M_{1,2}$ $M_{1,r}$
$M_{2,1}$ $M_{2,2}$ $M_{2,r}$
$M_{L,1}$ $M_{L,2}$ $M_{L,r}$
p_1 p_2 p_n $\neg p_1$ $\neg p_2$ $\neg p_n$

Fig. 3

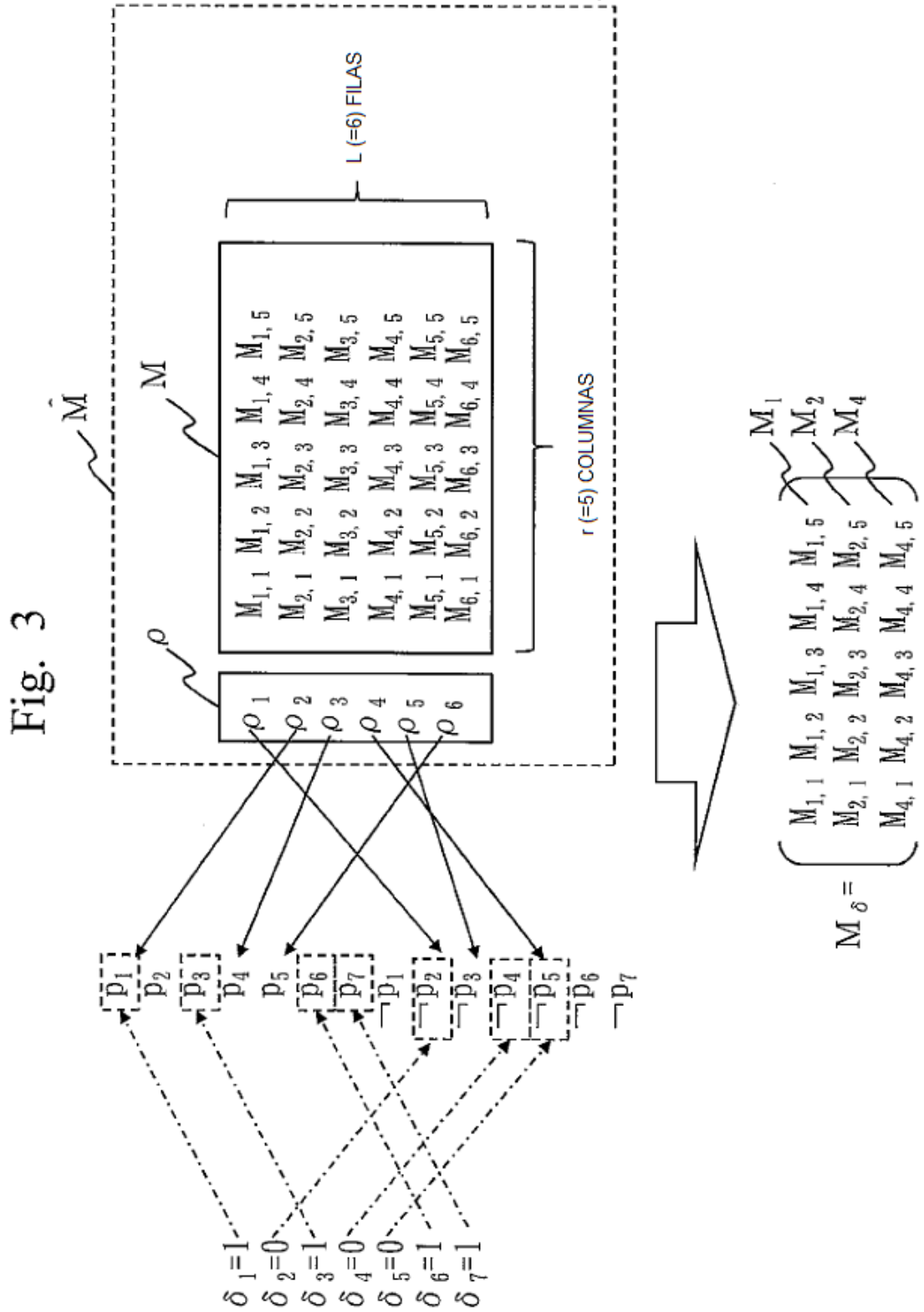


Fig. 4

$$s_0 = \begin{bmatrix} 1, \ldots, 1 \\ \text{r COLUMNAS} \end{bmatrix} \begin{pmatrix} f_1 \\ , \\ \cdot \\ \cdot \\ \cdot \\ , \\ f_r \end{pmatrix} = \sum_{k=1}^{r} f_k$$

Fig. 5

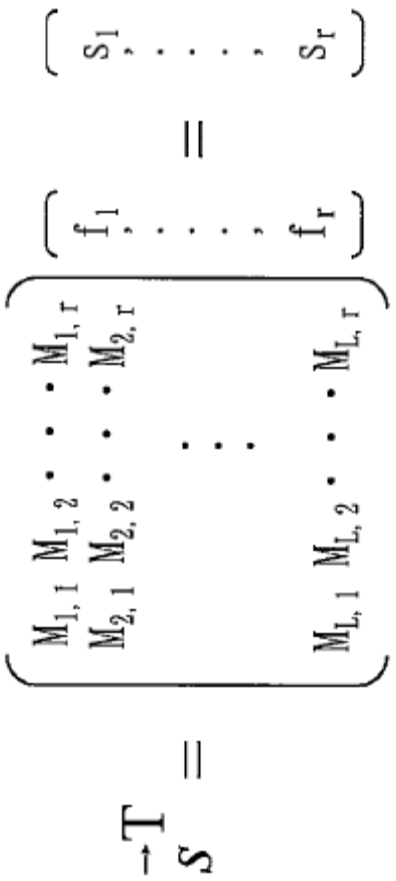

$\vec{s} = \vec{T}$
$\begin{pmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,r} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,r} \\ & & \ddots & \\ M_{L,1} & M_{L,2} & \cdots & M_{L,r} \end{pmatrix}$
$\begin{pmatrix} f_1 \\ \cdots \\ f_r \end{pmatrix}$
$=$
$\begin{pmatrix} s_1 \\ \cdots \\ s_r \end{pmatrix}$

Fig. 6

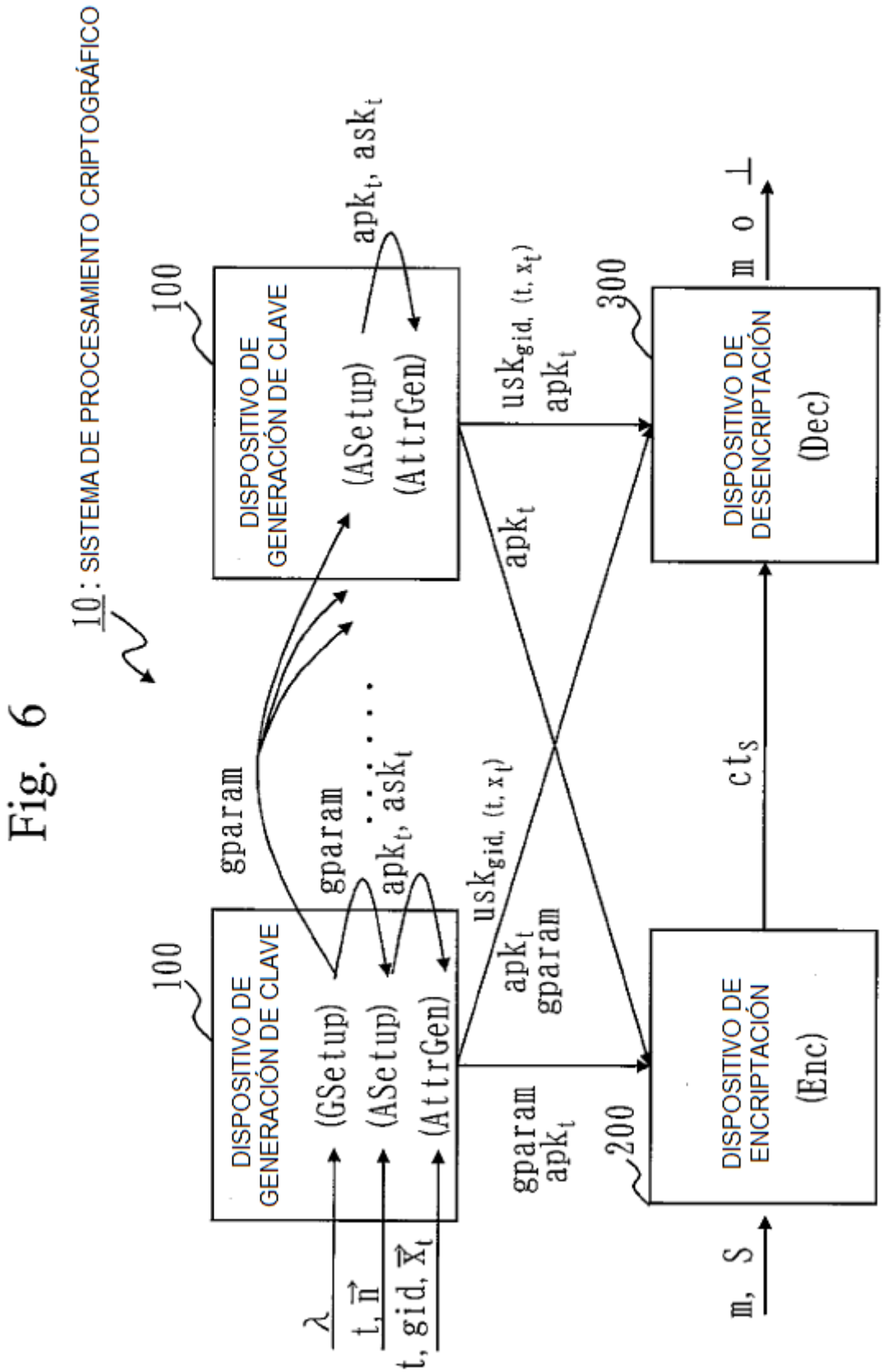

10: SISTEMA DE PROCESAMIENTO CRIPTOGRÁFICO
100
DISPOSITIVO DE GENERACIÓN DE CLAVE
(GSetup)
(ASetup)
(AttrGen)
100
DISPOSITIVO DE GENERACIÓN DE CLAVE
(ASetup)
(AttrGen)
200
DISPOSITIVO DE ENCRIPTACIÓN
(Enc)
300
DISPOSITIVO DE DESENCRIPTACIÓN
(Dec)
gparam
gparam
apk_t, ask_t
apk_t, ask_t
usk_gid, (t, x_t)
apk_t
apk_t
gparam
gparam
apk_t
usk_gid, (t, x_t)
apk_t
gparam
ct_S
λ
t, n
t, gid, x_t
m, S
m o ⊥

Fig. 7

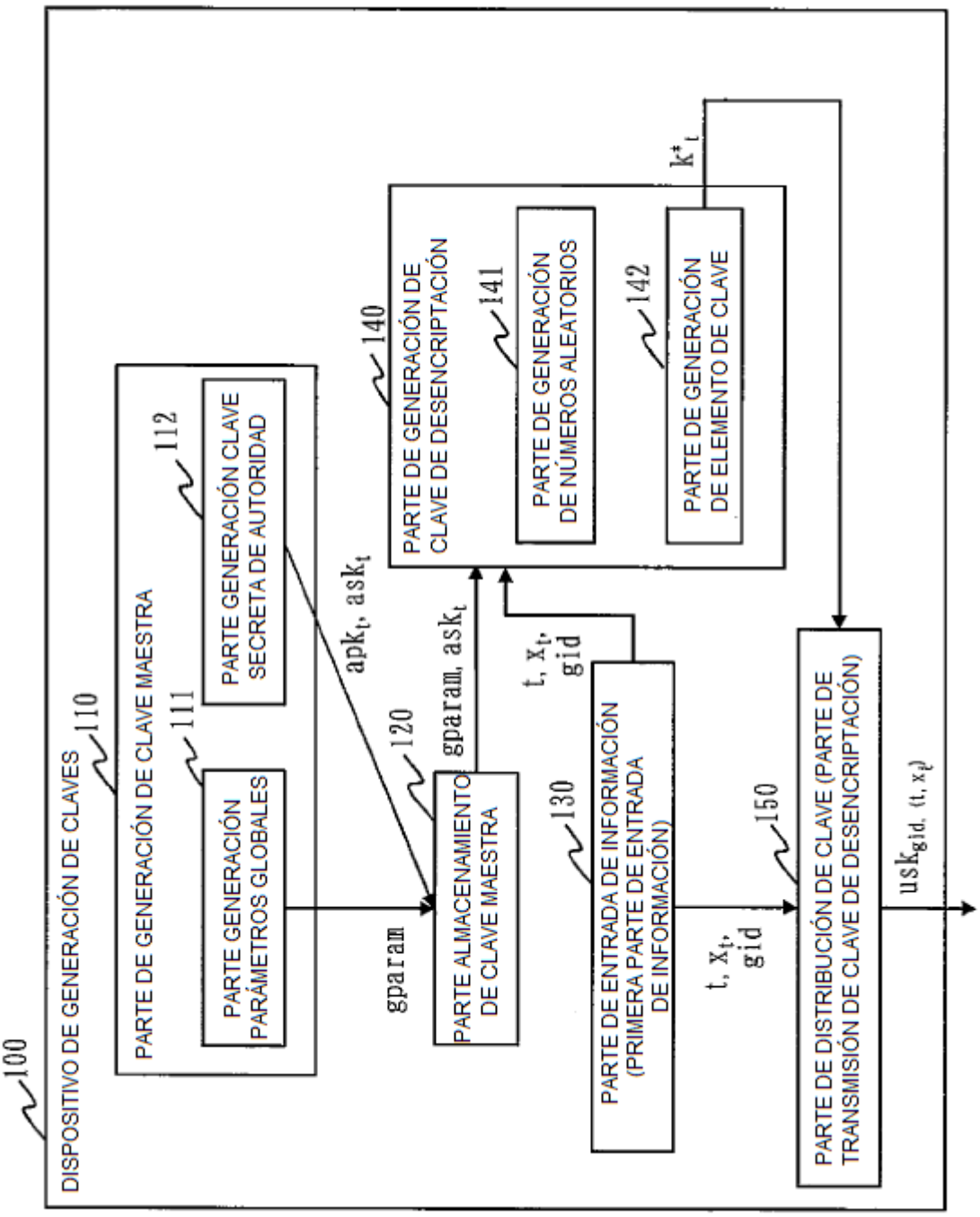

100
DISPOSITIVO DE GENERACIÓN DE CLAVES
110
PARTE DE GENERACIÓN DE CLAVE MAESTRA
111
PARTE GENERACIÓN PARÁMETROS GLOBALES
112
PARTE GENERACIÓN CLAVE SECRETA DE AUTORIDAD
140
PARTE DE GENERACIÓN DE CLAVE DE DESENCRIPTACIÓN
141
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
142
PARTE DE GENERACIÓN DE ELEMENTO DE CLAVE
120
PARTE ALMACENAMIENTO DE CLAVE MAESTRA
130
PARTE DE ENTRADA DE INFORMACIÓN (PRIMERA PARTE DE ENTRADA DE INFORMACIÓN)
150
PARTE DE DISTRIBUCIÓN DE CLAVE (PARTE DE TRANSMISIÓN DE CLAVE DE DESENCRIPTACIÓN)
gparam
apk_t, ask_t
gparam, ask_t
t, x_t, gid
t, x_t, gid
k*_t
usk_gid, (t, x_t)

Fig. 8

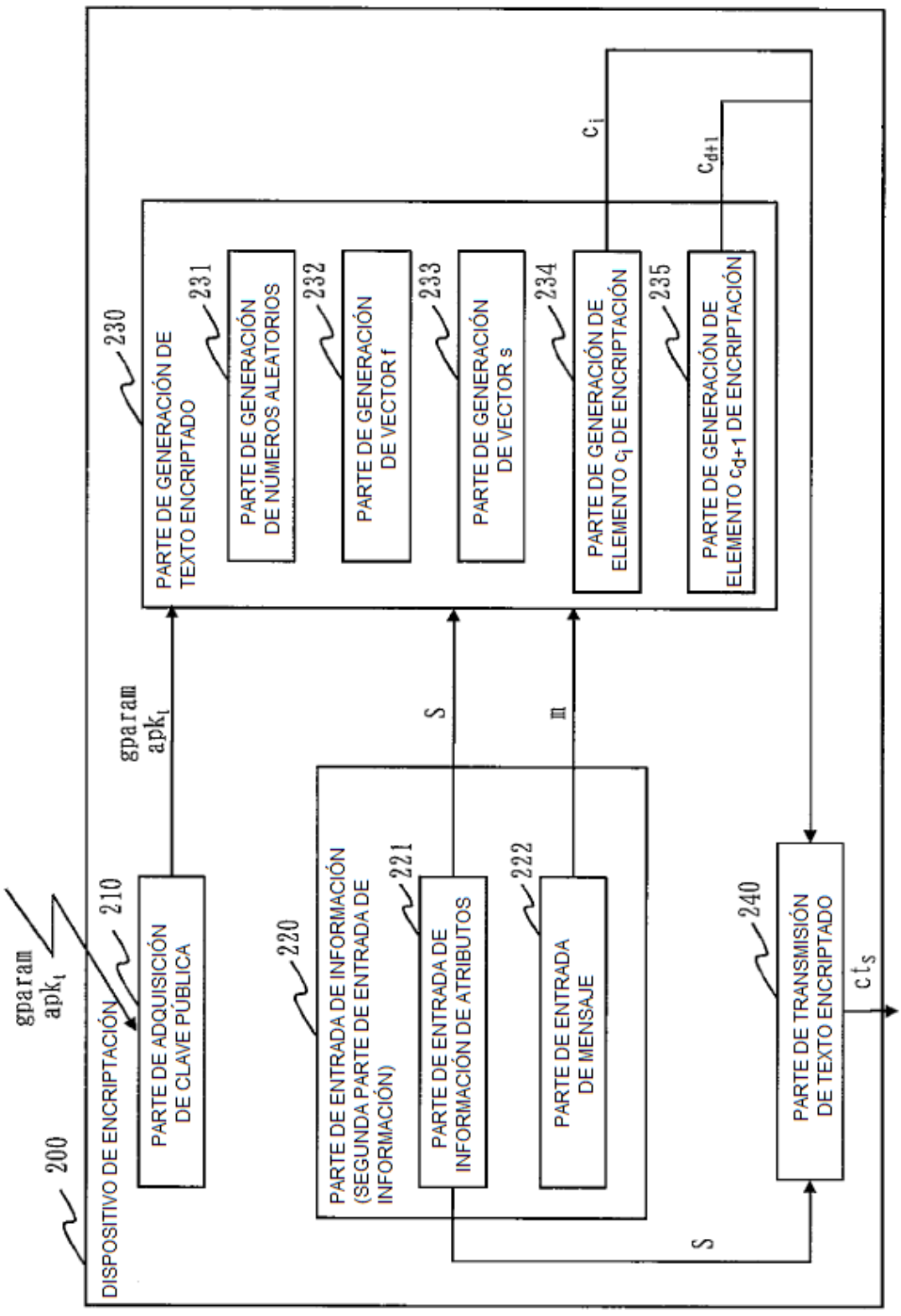

200
DISPOSITIVO DE ENCRIPTACIÓN
gparam apk_t
210
PARTE DE ADQUISICIÓN DE CLAVE PÚBLICA
gparam apk_t
220
PARTE DE ENTRADA DE INFORMACIÓN (SEGUNDA PARTE DE ENTRADA DE INFORMACIÓN)
221
PARTE DE ENTRADA DE INFORMACIÓN DE ATRIBUTOS
S
222
PARTE DE ENTRADA DE MENSAJE
m
230
PARTE DE GENERACIÓN DE TEXTO ENCRIPTADO
231
PARTE DE GENERACIÓN DE NÚMEROS ALEATORIOS
232
PARTE DE GENERACIÓN DE VECTOR f
233
PARTE DE GENERACIÓN DE VECTOR s
234
PARTE DE GENERACIÓN DE ELEMENTO c_i DE ENCRIPTACIÓN
235
PARTE DE GENERACIÓN DE ELEMENTO c_{d+1} DE ENCRIPTACIÓN
c_i
c_{d+1}
S
240
PARTE DE TRANSMISIÓN DE TEXTO ENCRIPTADO
ct_S

Fig. 9

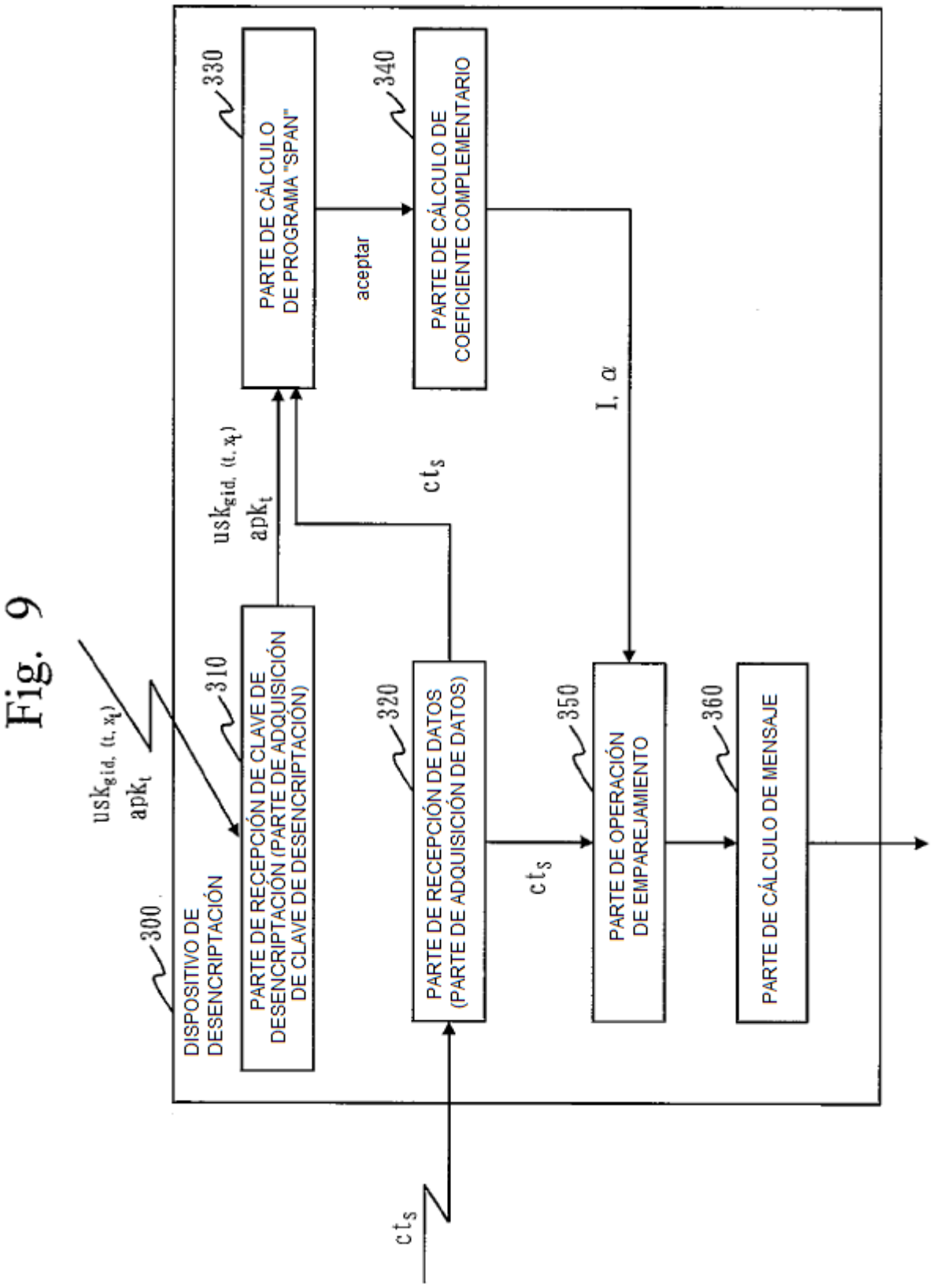

300
DISPOSITIVO DE DESENCRIPTACIÓN
310
PARTE DE RECEPCIÓN DE CLAVE DE DESENCRIPTACIÓN (PARTE DE ADQUISICIÓN DE CLAVE DE DESENCRIPTACIÓN)
320
PARTE DE RECEPCIÓN DE DATOS (PARTE DE ADQUISICIÓN DE DATOS)
330
PARTE DE CÁLCULO DE PROGRAMA "SPAN"
340
PARTE DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
350
PARTE DE OPERACIÓN DE EMPAREJAMIENTO
360
PARTE DE CÁLCULO DE MENSAJE
aceptar
$usk_{gid,(t,x_t)}$
apk_t
ct_s
I, α

Fig. 10

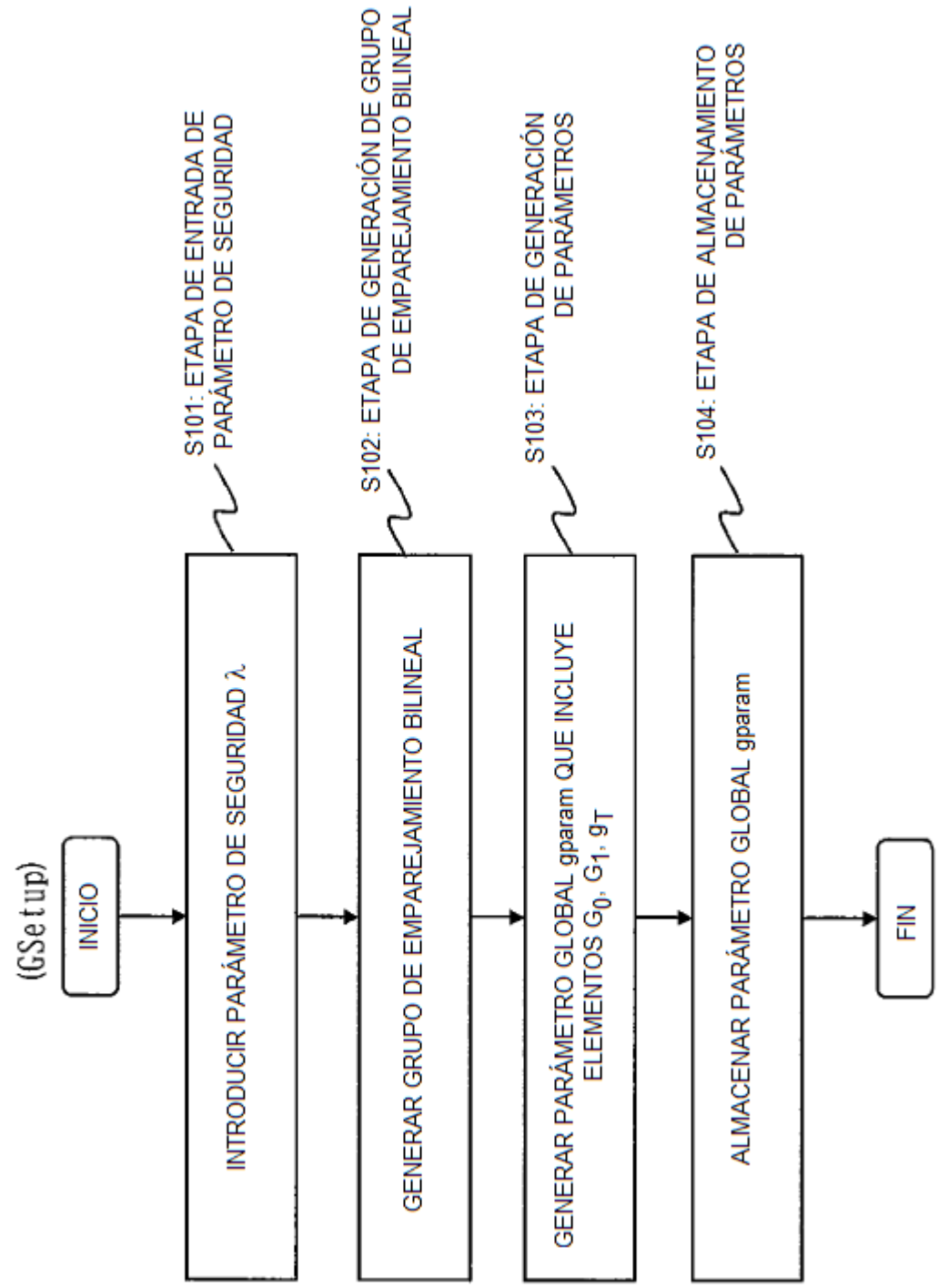

(GSetup)
INICIO
S101: ETAPA DE ENTRADA DE PARÁMETRO DE SEGURIDAD
INTRODUCIR PARÁMETRO DE SEGURIDAD λ
S102: ETAPA DE GENERACIÓN DE GRUPO DE EMPAREJAMIENTO BILINEAL
GENERAR GRUPO DE EMPAREJAMIENTO BILINEAL
S103: ETAPA DE GENERACIÓN DE PARÁMETROS
GENERAR PARÁMETRO GLOBAL gparam QUE INCLUYE ELEMENTOS G_0, G_1, g_T
S104: ETAPA DE ALMACENAMIENTO DE PARÁMETROS
ALMACENAR PARÁMETRO GLOBAL gparam
FIN

Fig. 11

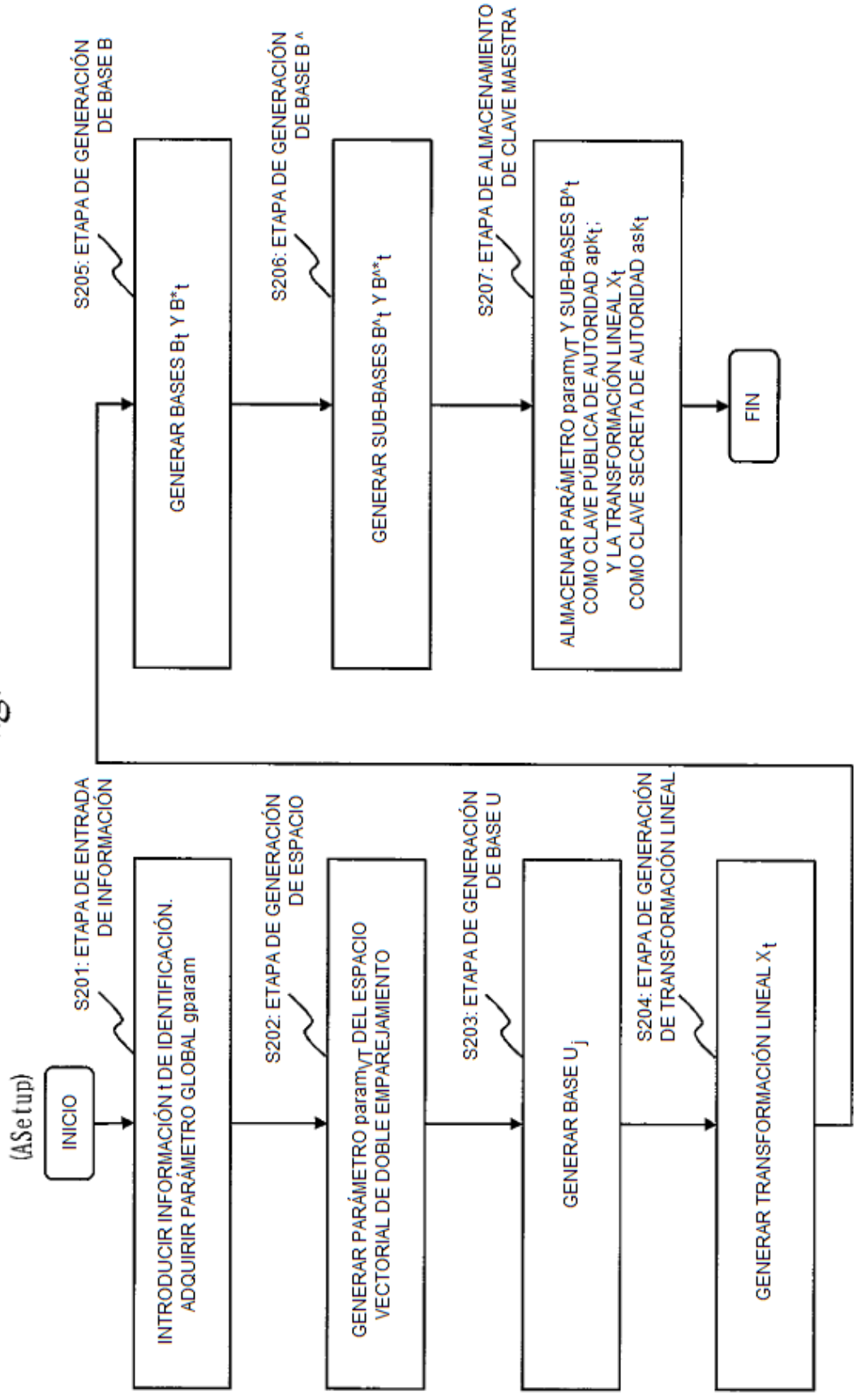

(ASetup)
INICIO
S201: ETAPA DE ENTRADA DE INFORMACIÓN
INTRODUCIR INFORMACIÓN t DE IDENTIFICACIÓN. ADQUIRIR PARÁMETRO GLOBAL gparam
S202: ETAPA DE GENERACIÓN DE ESPACIO
GENERAR PARÁMETRO paramVT DEL ESPACIO VECTORIAL DE DOBLE EMPAREJAMIENTO
S203: ETAPA DE GENERACIÓN DE BASE U
GENERAR BASE Uj
S204: ETAPA DE GENERACIÓN DE TRANSFORMACIÓN LINEAL
GENERAR TRANSFORMACIÓN LINEAL Xt
S205: ETAPA DE GENERACIÓN DE BASE B
GENERAR BASES Bt Y B*t
S206: ETAPA DE GENERACIÓN DE BASE B^
GENERAR SUB-BASES B^t Y B^*t
S207: ETAPA DE ALMACENAMIENTO DE CLAVE MAESTRA
ALMACENAR PARÁMETRO paramVT Y SUB-BASES B^t COMO CLAVE PÚBLICA DE AUTORIDAD apkt; Y LA TRANSFORMACIÓN LINEAL Xt COMO CLAVE SECRETA DE AUTORIDAD askt
FIN

Fig. 12

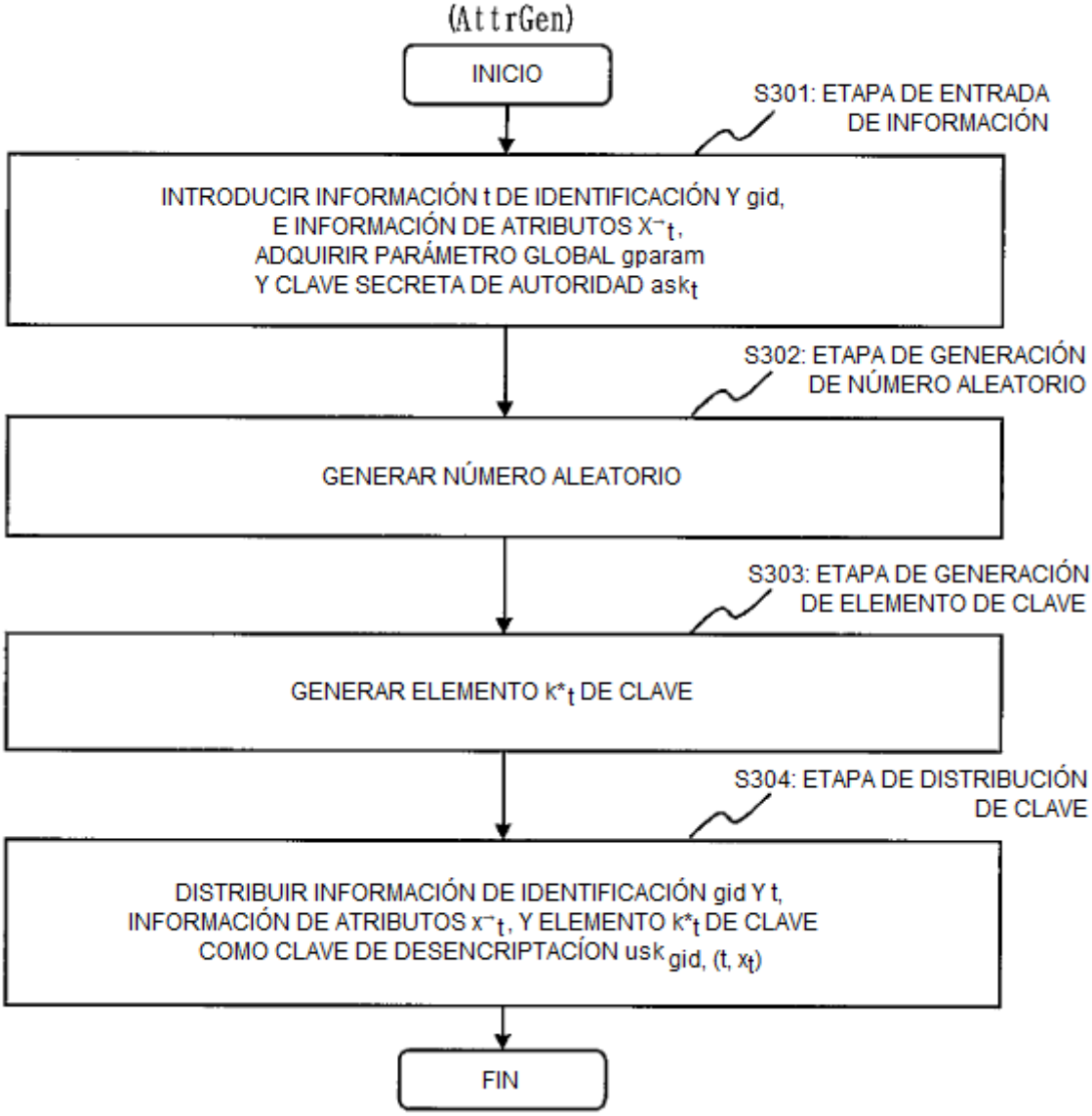

(AttrGen)
INICIO
S301: ETAPA DE ENTRADA DE INFORMACIÓN
INTRODUCIR INFORMACIÓN t DE IDENTIFICACIÓN Y gid,
E INFORMACIÓN DE ATRIBUTOS X¯t,
ADQUIRIR PARÁMETRO GLOBAL gparam
Y CLAVE SECRETA DE AUTORIDAD askt
S302: ETAPA DE GENERACIÓN DE NÚMERO ALEATORIO
GENERAR NÚMERO ALEATORIO
S303: ETAPA DE GENERACIÓN DE ELEMENTO DE CLAVE
GENERAR ELEMENTO k*t DE CLAVE
S304: ETAPA DE DISTRIBUCIÓN DE CLAVE
DISTRIBUIR INFORMACIÓN DE IDENTIFICACIÓN gid Y t,
INFORMACIÓN DE ATRIBUTOS x¯t, Y ELEMENTO k*t DE CLAVE
COMO CLAVE DE DESENCRIPTACÍON usk gid, (t, xt)
FIN

Fig. 13

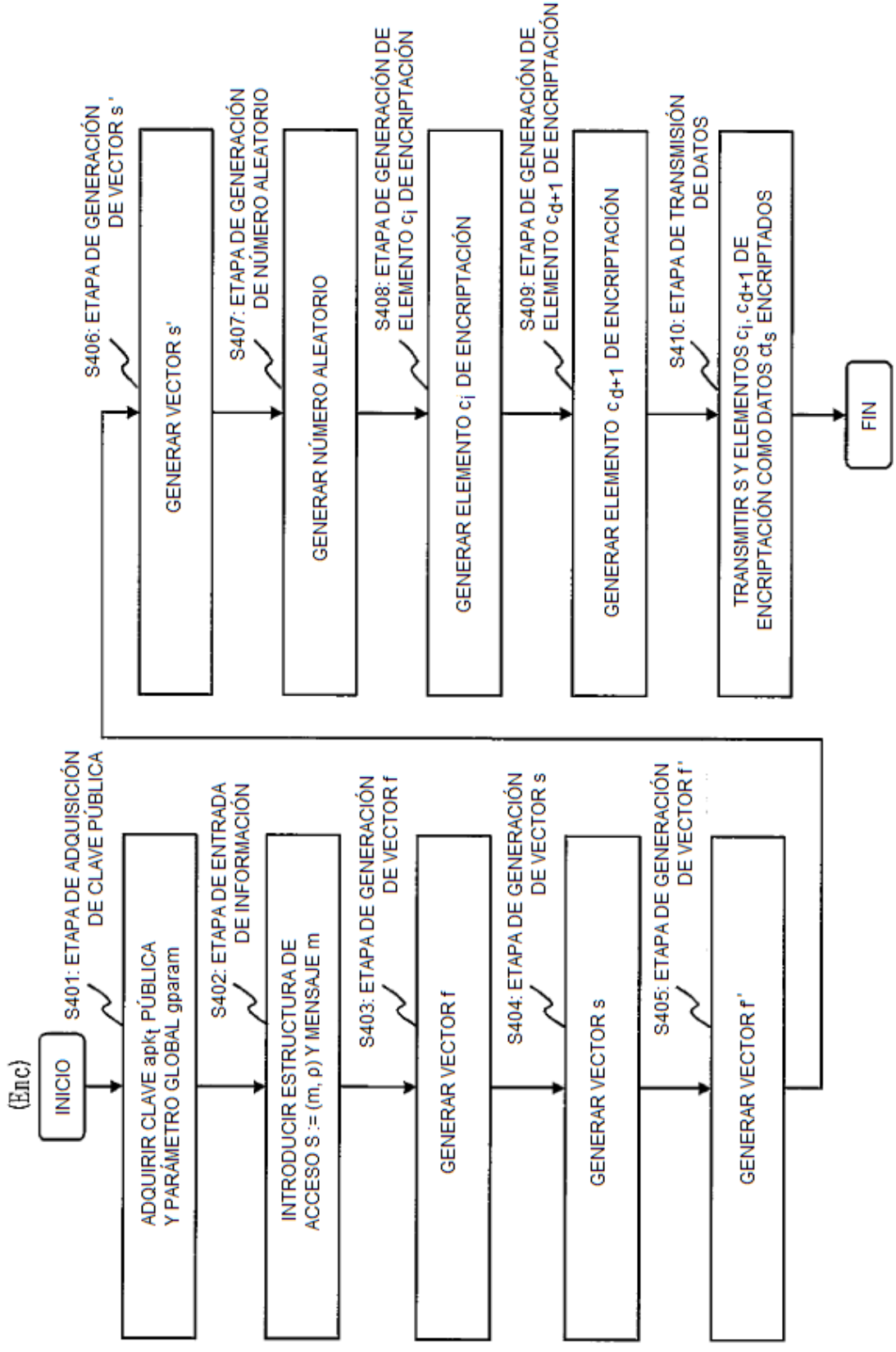

(Enc)
INICIO
S401: ETAPA DE ADQUISICIÓN DE CLAVE PÚBLICA
ADQUIRIR CLAVE apk$_t$ PÚBLICA Y PARÁMETRO GLOBAL gparam
S402: ETAPA DE ENTRADA DE INFORMACIÓN
INTRODUCIR ESTRUCTURA DE ACCESO S := (m, ρ) Y MENSAJE m
S403: ETAPA DE GENERACIÓN DE VECTOR f
GENERAR VECTOR f
S404: ETAPA DE GENERACIÓN DE VECTOR s
GENERAR VECTOR s
S405: ETAPA DE GENERACIÓN DE VECTOR f'
GENERAR VECTOR f'
S406: ETAPA DE GENERACIÓN DE VECTOR s'
GENERAR VECTOR s'
S407: ETAPA DE GENERACIÓN DE NÚMERO ALEATORIO
GENERAR NÚMERO ALEATORIO
S408: ETAPA DE GENERACIÓN DE ELEMENTO c_j DE ENCRIPTACIÓN
GENERAR ELEMENTO c_j DE ENCRIPTACIÓN
S409: ETAPA DE GENERACIÓN DE ELEMENTO c_{d+1} DE ENCRIPTACIÓN
GENERAR ELEMENTO c_{d+1} DE ENCRIPTACIÓN
S410: ETAPA DE TRANSMISIÓN DE DATOS
TRANSMITIR S Y ELEMENTOS c_j, c_{d+1} DE ENCRIPTACIÓN COMO DATOS ct_S ENCRIPTADOS
FIN

Fig. 14

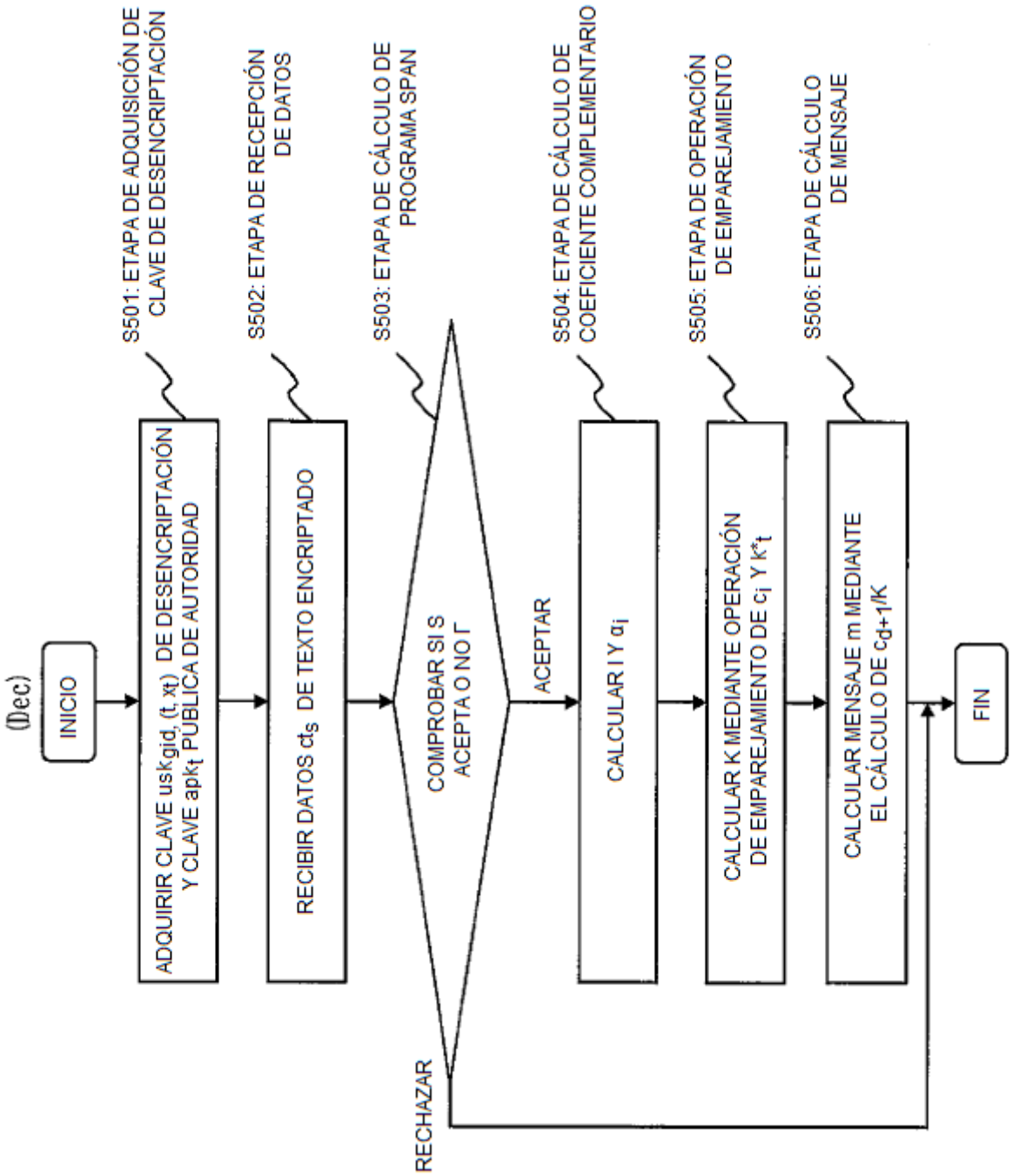
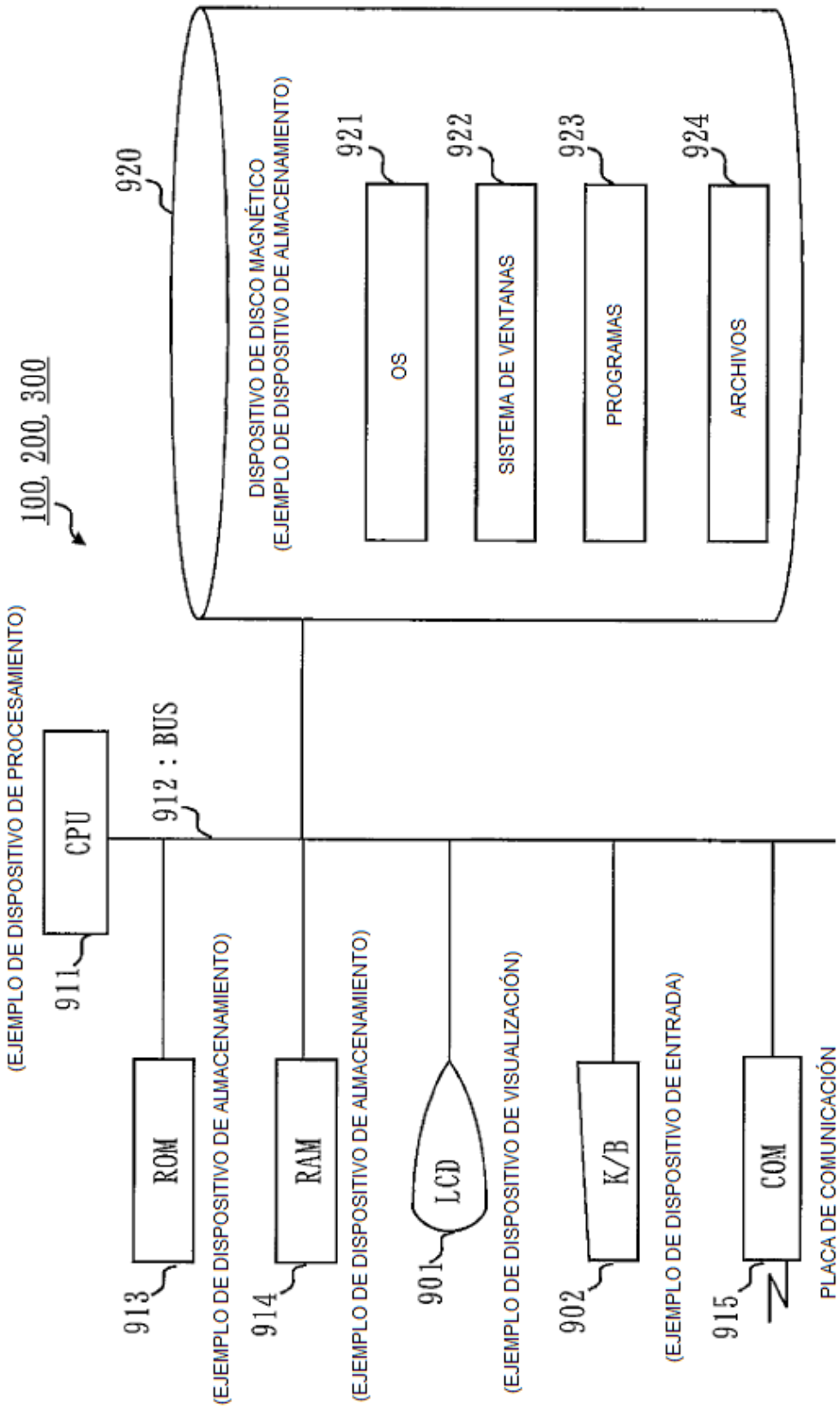

(Dec)
INICIO
S501: ETAPA DE ADQUISICIÓN DE CLAVE DE DESENCRIPTACIÓN
ADQUIRIR CLAVE uskgid, (t, xt) DE DESENCRIPTACIÓN Y CLAVE apkt PUBLICA DE AUTORIDAD
S502: ETAPA DE RECEPCIÓN DE DATOS
RECIBIR DATOS cts DE TEXTO ENCRIPTADO
S503: ETAPA DE CÁLCULO DE PROGRAMA SPAN
COMPROBAR SI S ACEPTA O NO Γ
ACEPTAR
RECHAZAR
S504: ETAPA DE CÁLCULO DE COEFICIENTE COMPLEMENTARIO
CALCULAR I Y αi
S505: ETAPA DE OPERACIÓN DE EMPAREJAMIENTO
CALCULAR K MEDIANTE OPERACIÓN DE EMPAREJAMIENTO DE ci Y k*t
S506: ETAPA DE CÁLCULO DE MENSAJE
CALCULAR MENSAJE m MEDIANTE EL CÁLCULO DE cd+1/K
FIN

Fig. 15


100, 200, 300
920
DISPOSITIVO DE DISCO MAGNÉTICO
(EJEMPLO DE DISPOSITIVO DE ALMACENAMIENTO)
921
OS
922
SISTEMA DE VENTANAS
923
PROGRAMAS
924
ARCHIVOS
912 : BUS
(EJEMPLO DE DISPOSITIVO DE PROCESAMIENTO)
911
CPU
913
ROM
(EJEMPLO DE DISPOSITIVO DE ALMACENAMIENTO)
914
RAM
(EJEMPLO DE DISPOSITIVO DE ALMACENAMIENTO)
901
LCD
(EJEMPLO DE DISPOSITIVO DE VISUALIZACIÓN)
902
K/B
(EJEMPLO DE DISPOSITIVO DE ENTRADA)
915
COM
PLACA DE COMUNICACIÓN