(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 645 072**

(51) Int. CI.:

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

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **17.07.2013 PCT/JP2013/069364**

(87) Fecha y número de publicación internacional: **24.04.2014 WO14061324**

(96) Fecha de presentación y número de la solicitud europea: **17.07.2013 E 13846805 (3)**

(97) Fecha y número de publicación de la concesión europea: **27.09.2017 EP 2911137**

(54) Título: **Sistema criptográfico**

(30) Prioridad:

19.10.2012 JP 2012231293

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
04.12.2017

(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, S.L.P

DESCRIPCIÓN

Sistema criptográfico

Campo técnico

La presente invención se refiere a un esquema generalizado de cifrado mediante predicados de productos internos, y a un esquema de cifrado funcional y un esquema de firma basado en atributos que presentan, como estructura inferior, cada uno de ellos, un esquema generalizado de cifrado mediante predicados de productos internos.

Antecedentes de la técnica

El documento EP 2 613 472 A1 se refiere a un sistema de procesado de cifrado, a un dispositivo de generación de claves, a un dispositivo de cifrado, a un dispositivo de descifrado, a un método de procesado de cifrado, y a un programa de procesado de cifrado. El objetivo de este documento es proporcionar un esquema de cifrado funcional y seguro que disponga de muchas funciones criptográficas. Una estructura de acceso se constituye aplicando el producto interno de vectores de atributos a un programa de subespacios generados. La estructura de acceso tiene un grado de libertad en el diseño del programa de subespacios generados y en el diseño de los vectores de atributos, disponiendo, así, de un grado alto de libertad en el diseño del control de acceso. Se implementa un proceso de cifrado funcional dotando de la estructura de acceso a cada uno de un texto cifrado y una clave de descifrado.

Las referencias bibliográficas 30 y 31, que no son patentes, describen esquemas de cifrado mediante predicados de productos internos.

La referencia bibliográfica 31, que no es una patente, describe un esquema de cifrado funcional.

La referencia bibliográfica 32, que no es patente, describe un esquema de firma basado en atributos.

Lista de referencias

Bibliografía que no forma parte del ámbito de las patentes

Referencia bibliográfica 1, no documento de patente: Attrapadung, N. y Libert, B., *Functional Encryption for Inner Product: Achieving Constant-Size Ciphertexts with Adaptive Security or Support for Negation,* PKC 2010, págs. 384 a 402. Springer Heidelberg (2010)

Referencia bibliográfica 2, no documento de patente: Beimel, A., *Secure schemes for secret sharing and key distribution.* PhD Thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

Referencia bibliográfica 3, no documento de patente: Bellare, M., Waters, B., Yilek, S.: *Identity-based encryption secure against selective opening attack*. En: Ishai, Y.(ed.) TCC 2011. págs. 235 a 252. Springer Heidelberg (2011)

Referencia bibliográfica 4, no documento de patente: Bethencourt, J., Sahai, A., Waters, B.: *Ciphertext-policy attribute-based encryption*. En: 2007 IEEE Symposium on Security and Privacy, págs. 321 a 334. IEEE Press (2007)

Referencia bibliográfica 5, no documento de patente: Boneh, D., Boyen, X.: *Efficient selective-ID secure identity based encryption without random oracles*. En: Cachin, C., Camenisch, J. (eds.) EUROCRYPT 2004. LNCS, vol. 3.027, págs. 223 a 238. Springer Heidelberg (2004)

Referencia bibliográfica 6, no documento de patente : Boneh, D., Boyen, X.: *Secure identity based encryption without random oracles*. En: Franklin, M.K. (ed.) CRYPT02004. LNCS, vol. 3.152, págs. 443 a 459. Springer Heidelberg (2004)

Referencia bibliográfica 7, no documento de patente: Boneh, D., Boyen, X., Goh, E.: *Hierarchical identity based encryption with constant size ciphertext*. En: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3.494, págs. 440 a 456. Springer Heidelberg (2005)

Referencia bibliográfica 8, no documento de patente: Boneh, D., Boyen, X., Shacham, H.: *Short group signatures.* En: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3.152, págs. 41 a 55. Springer, Heidelberg (2004)

Referencia bibliográfica 9, no documento de patente: Boneh, D., Franklin, M.: *Identity-based encryption from the Weil pairing*. En: Kilian, J. (ed.) CRYPTO 2001.LNCS, vol. 2.139, págs. 213 a 229. Springer Heidelberg (2001)

Referencia bibliográfica 10, no documento de patente: Boneh, D., Hamburg, M.: *Generalized identity based and broadcast encryption scheme*. En: Pieprzyk, J. (ed.) ASIACRYPT 2008. LNCS, vol. 5.350, págs. 455 a 470. Springer Heidelberg (2008)

Referencia bibliográfica 11, no documento de patente: Boneh, D., Katz, J., *Improved efficiency for CCA-secure*

cryptosystems built using identity based encryption. RSA-CT 2005, LNCS, Springer Verlag (2005)

Referencia bibliográfica 12, no documento de patente: Boneh, D., Waters, B.: *Conjunctive, subset, and range queries on encrypted data*. En: Vadhan, S.P. (ed.) TCC2007. LNCS, vol. 4.392, págs. 535 a 554. Springer Heidelberg (2007)

Referencia bibliográfica 13, no documento de patente: Boyen, X., Waters, B.: *Anonymous hierarchical identity-based encryption (without random oracles)*. En: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4.117, págs. 290 a 307. Springer Heidelberg (2006)

Referencia bibliográfica 14, no documento de patente: Canetti, R., Halevi S., Katz J.: *Chosen-ciphertext security from identity-based encryption.* EUROCRYPT 2004, LNCS, Springer Heidelberg (2004)

Referencia bibliográfica 15, no documento de patente: Chase, M.: *Multi-authority attribute based encryption*. TCC, LNCS, págs. 515 a 534, Springer Heidelberg (2007).

Referencia bibliográfica 16, no documento de patente: Chase, M. y Chow, S.: *Improving privacy and security in multi-authority attribute-based encryption*, ACM Conference on Computer and Communications Security, págs. 121 a 130, ACM (2009).

Referencia bibliográfica 17, no documento de patente: Cocks, C.: *An identity based encryption scheme based on quadratic residues*. En: Honary, B. (ed.) IMA Int. Conf. LNCS, vol. 2.260, págs. 360 a 363. Springer Heidelberg (2001)

Referencia bibliográfica 18, no documento de patente: Gentry, C.: *Practical identity-based encryption without random oracles*. En: Vaudenay, S. (ed.) EUROCRYPT2006. LNCS, vol. 4.004, págs. 445 a 464. Springer Heidelberg (2006)

Referencia bibliográfica 19, no documento de patente: Gentry, C., Halevi, S.: *Hierarchical identity-based encryption with polynomially many levels*. En: Reingold, O.(ed.) TCC 2009. LNCS, vol. 5.444, págs. 437 a 456. Springer Heidelberg (2009)

Referencia bibliográfica 20, no documento de patente: Gentry, C., Silverberg, A.: *Hierarchical ID-based cryptography*. En: Zheng, Y. (ed.) ASIACRYPT 2002. LNCS, vol. 2.501, págs. 548 a 566. Springer Heidelberg (2002)

Referencia bibliográfica 21, no documento de patente: 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, págs. 89 a 98, ACM (2006)

Referencia bibliográfica 22, no documento de patente: Katz, J., Sahai, A., Waters, B.: *Predicate encryption supporting disjunctions, polynomial equations, and inner products*. En: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4.965, págs. 146 a 162. Springer Heidelberg (2008)

Referencia bibliográfica 23, no documento de patente: 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) La versión completa está disponible en http://eprint.iacr.org/2010/110

Referencia bibliográfica 24, no documento de patente: 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. 5.978, págs. 455 a 479. Springer Heidelberg (2010)

Referencia bibliográfica 25, no documento de patente: Lewko, A.B., Waters, B.: *Decentralizing Attribute-Based Encryption*, EUROCRYPT 2011. LNCS, vol. 6.632, págs. 568 a 588. Springer Heidelberg (2011)

Referencia bibliográfica 26, no documento de patente: Lewko, A.B., Waters, B.: *Unbounded HIBE and attribute-based encryption*, EUROCRYPT 2011. LNCS, vol. 6.632, págs. 547 a 567. Springer Heidelberg (2011)

Referencia bibliográfica 27, no documento de patente: H. Lin, Z. Cao, X. Liang, y J. Shao.: *Secure threshold multi authority attribute based encryption without a central authority*, INDOCRYPT, LNCS, vol. 5.365, págs. 426 a 436, Springer Heidelberg (2008).

Referencia bibliográfica 28, no documento de patente: S. Mueller, S. Katzenbeisser, y C. Eckert.; *On multi-authority ciphertext-policy attribute-based encryption*, Bull. Korean Math Soc. 46, n.º 4, págs. 803 a 819 (2009).

Referencia bibliográfica 29, no documento de patente: Okamoto, T., Takashima, K.: *Homomorphic encryption and signatures from vector decomposition*. En: Galbraith, S.D., Paterson, K.G. (eds.) Pairing 2008. LNCS, vol. 5.209, págs. 57 a 74, Springer Heidelberg (2008)

Referencia bibliográfica 30, no documento de patente: Okamoto, T., Takashima, K.: *Hierarchical predicate encryption for inner-products*, En: ASIACRYPT 2009, Springer Heidelberg (2009)

Referencia bibliográfica 31, no documento de patente: Okamoto, T., Takashima, K.: *Fully secure functional encryption with general relations from the decisional linear assumption*. En: Rabin, T. (ed.) CRYPTO 2010. LNCS, vol. 6.223, págs. 191 a 208. Springer Heidelberg (2010). La versión completa está disponible en http://eprint.iacr.org/2010/563

Referencia bibliográfica 32, no documento de patente: Okamoto, T., Takashima, K.: *Efficient attribute-based signatures for non-monotone predicates in the standard model*, En: PKC 2011, Springer Heidelberg (2011)

Referencia bibliográfica 33, no documento de patente: Okamoto, T., Takashima, K.: *Achieving Short Ciphertexts or Short Secret-Keys for Adaptively Secure General Inner-Product Encryption*. CANS 2011, LNCS, vol. 7.092, págs. 138 a 159 Springer Heidelberg (2011).

Referencia bibliográfica 34, no documento de patente: Okamoto, T., Takashima, K.: *Adaptively Attribute-Hiding (Hierarchical) Inner Product Encryption*. EUROCRYPT 2012, LNCS, vol. 7.237, págs. 591 a 608, Springer Heidelberg (2012)

Referencia bibliográfica 35, no documento de patente: Ostrovsky, R., Sahai, A., Waters, B.: *Attribute-based encryption with non-monotonic access structures*. En: ACM Conference on Computer and Communication Security 2007, págs. 195 a 203, ACM (2007)

Referencia bibliográfica 36, no documento de patente: Pirretti, M., Traynor, P., McDaniel, P., Waters, B.: *Secure attribute-based systems*. En: ACM Conference on Computer and Communication Security 2006, págs. 99 a 112, ACM, (2006)

Referencia bibliográfica 37, no documento de patente: Sahai, A., Waters, B.: *Fuzzy identity-based encryption*. En: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3.494, págs. 457 a 473. Springer Heidelberg (2005)

Referencia bibliográfica 38, no documento de patente: Shi, E., Waters, B.: *Delegating capability in predicate encryption systems*. En: Aceto, L., Damgaard, I, Goldberg, L.A., Halldoersson, M.M., Ingoelfsdoettir, A., Walukiewicz, I. (eds.) ICALP (2) 2008. LNCS, vol. 5.126, págs. 560 a 578. Springer Heidelberg (2008)

Referencia bibliográfica 39, no documento de patente: Waters, B.: *Efficient identity based encryption without random oracles*. Eurocrypt 2005, LNCS, vol. 3.152, págs. 443 a 459. Springer Verlag, (2005)

Referencia bibliográfica 40, no documento de patente: Waters, B.: *Ciphertext-policy attribute-based encryption: an expressive, efficient, and provably secure realization*. PKC 2011, LNCS, vol. 6.571, págs. 53 a 70. Springer Heidelberg (2011). ePrint, IACR, http://eprint.iacr.org/2008/290

Referencia bibliográfica 41, no documento de patente: Waters, B.: *Dual system encryption: realizing fully secure IBE and HIBE under simple assumptions*. En: Halevi, S. (ed.) CRYPTO 2009. LNCS, vol. 5.677, págs. 619 a 636. Springer Heidelberg (2009)

Sumario de la invención

Problema técnico

En los esquemas de cifrado por predicados de productos internos descritos en las referencias bibliográficas 30 y 31, que no son documentos de patente, existe la restricción de que las dimensiones de un vector de atributos $\vec{x}$ (parámetro usado para el cifrado) y un vector de predicados $\vec{v}$ (parámetro usado para una clave secreta) deberían ser equivalentes.

Es un objetivo de la presente invención lograr un esquema de cifrado mediante predicados de productos internos con una flexibilidad mejorada.

Solución al problema

El objetivo de la presente invención se logra con las reivindicaciones independientes. Las reivindicaciones dependientes describen realizaciones ventajosas.

Un sistema criptográfico de acuerdo con la presente invención es un sistema criptográfico que incluye un dispositivo de cifrado y un dispositivo de descifrado,

en donde el dispositivo de cifrado incluye

una unidad de generación de textos cifrados que genera un texto cifrado que tiene un elemento c_0 en el cual un valor $\tilde{\omega}$ se fija como coeficiente de un vector de base $b_{0,r}$, y un elemento c_t en el cual la información de atributo x_t se fija como coeficiente de un vector de base b_p y el valor $\tilde{\omega}$ se fija como coeficiente de un vector de base b_q, para cada índice t incluido en un conjunto $I_{x\_}$, y

en donde el dispositivo de descifrado incluye

una unidad de almacenamiento de claves de descifrado que almacena una clave de descifrado que tiene un elemento k_0 y un elemento k_t que se generan usando un valor s_t y un valor s_0 el cual es una suma del valor s_t para cada índice t incluido en un conjunto $I_{v\rightarrow}$, siendo el elemento k_0 un elemento en el cual un valor $-s_0$ se fija como coeficiente de un vector de base $b^*_{0,r}$ correspondiente al vector de base $b_{0,r}$, siendo el elemento k_t un elemento en el cual la información de predicado v_t se fija como coeficiente de un vector de base b^*_p correspondiente al vector de base b_p y el valor s_t se fija como coeficiente de un vector de base b^*_q correspondiente al vector de base b_q, para cada índice t incluido en el conjunto $I_{v\rightarrow}$; y

una unidad de descifrado que descifra el texto cifrado generado por la unidad de generación de textos cifrados con la clave de descifrado almacenada por la unidad de almacenamiento de claves de descifrado, de manera que la unidad de descifrado descifra el texto cifrado calculando un producto de operaciones de emparejamiento entre pares correspondientes de los vectores de base en el elemento c_0 y el elemento k_0 y en el elemento c_t y el elemento k_t para cada índice t incluido en el conjunto $I_{v\rightarrow}$.

Efectos ventajosos de la invención

En un sistema criptográfico de acuerdo con la presente invención, se lleva a cabo una operación de emparejamiento sobre solamente un índice t incluido en un conjunto $I_{v\rightarrow}$, y no se requiere que las dimensiones de un vector de atributos $\vec{x}$ y un vector de predicados $\vec{v}$ sean equivalentes. De este modo, el sistema criptográfico de acuerdo con la presente invención puede materializar un esquema de cifrado mediante predicados de productos internos con una flexibilidad mejorada.

Breve descripción de los dibujos

La Fig. 1 es un diagrama de configuración de un sistema 10 de procesado criptográfico según la Realización 1;

la Fig. 2 es un diagrama de bloques funcional que ilustra la función de un dispositivo 100 de generación de claves según la Realización 1;

la Fig. 3 es un diagrama de bloques funcional que ilustra la función de un dispositivo 200 de cifrado según la Realización 1;

la Fig. 4 es un diagrama de bloques funcional que ilustra la función de un dispositivo 300 de descifrado según la Realización 1;

la Fig. 5 es un diagrama de flujo que ilustra el proceso de un algoritmo *Setup* según la Realización 1;

la Fig. 6 es un diagrama de flujo que ilustra el proceso de un algoritmo *KeyGen* según la Realización 1;

la Fig. 7 es un diagrama de flujo que ilustra el proceso de un algoritmo *Enc* según la Realización 1;

la Fig. 8 es un diagrama de flujo que ilustra el proceso de un algoritmo *Dec* según la Realización 1;

la Fig. 9 es un diagrama de flujo que ilustra el proceso de un algoritmo *KeyGen* según la Realización 2;

la Fig. 10 es un diagrama de flujo que ilustra el proceso de un algoritmo *Enc* según la Realización 2;

la Fig. 11 es un diagrama de flujo que ilustra el proceso de un algoritmo *Dec* según la Realización 2;

la Fig. 12 es un diagrama de flujo que ilustra el proceso de un algoritmo *Setup* según la Realización 3;

la Fig. 13 es un diagrama de flujo que ilustra el proceso de un algoritmo *KeyGen* según la Realización 3;

la Fig. 14 es un diagrama de flujo que ilustra el proceso de un algoritmo *Enc* según la Realización 3;

la Fig. 15 es un diagrama de flujo que ilustra el proceso de un algoritmo *Dec* según la Realización 3; y

la Fig. 16 es un diagrama que ilustra un ejemplo de una configuración de hardware del dispositivo 100 de generación de claves, el dispositivo 200 de cifrado y el dispositivo 300 de descifrado.

Descripción de realizaciones

En lo sucesivo en la presente se describirán realizaciones de esta invención en referencia a los dibujos adjuntos.

En la siguiente descripción, un dispositivo de procesado es una CPU 911 ó similar, que se describirá posteriormente. Un dispositivo de almacenamiento es una ROM 913, una RAM 914, un disco magnético 920 ó similares que se describirán posteriormente. Un dispositivo de comunicaciones es una placa 915 de comunicaciones o similar que se describirá posteriormente. Un dispositivo de entrada es un teclado 902, la placa 915 de comunicaciones o similares

que se describirán posteriormente. Un dispositivo de salida es la RAM 914, el disco magnético 920, la placa 915 de comunicaciones, un LCD 901 ó similares que se describirán posteriormente. Es decir, el dispositivo de procesado, el dispositivo de almacenamiento, el dispositivo de comunicaciones, el dispositivo de entrada y el dispositivo de salida son hardware.

Se explicarán las notaciones que se van a utilizar en la siguiente descripción.

Cuando A es una distribución o variable aleatoria, la Fórmula 101 designa que y se selecciona aleatoriamente de A de acuerdo con la distribución de A. Es decir, y es un número aleatorio en la Fórmula 101.

[Fórmula 101]

$$y \xleftarrow{R} A$$

Cuando A es un conjunto, la Fórmula 102 designa que y se selecciona uniformemente de A. Es decir, y es un número aleatorio uniforme en la Fórmula 102.

[Fórmula 102]

$$y \xleftarrow{U} A$$

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

[Fórmula 103]

$$y := z$$

Cuando a es un valor fijo, la Fórmula 104 designa que una máquina (algoritmo) A da salida a a con una entrada x.

[Fórmula 104]

$$A(x) \to a$$

Por ejemplo,

$$A(x) \to 1$$

La Fórmula 105, a saber F_q, designa un cuerpo finito de orden q.

[Fórmula 105]

$$\mathbb{F}_q$$

Un símbolo de vector designa una representación vectorial sobre el cuerpo finito F_q, según se indica en la Fórmula 106.

[Fórmula 106]

$\vec{x}$ designa

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

La Fórmula 107 designa el producto interno, indicado en 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 designa la transposición de una matriz X.

Para una base B y una base B^* indicadas en la Fórmula 110, se establece la Fórmula 111.

[Fórmula 110]

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

[Fórmula 111]

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

En la siguiente descripción, cuando "δi,j" se representa como un superíndice, este δi,j designa $\delta_{i,j}$.

Cuando "→" que representa un vector se adjunta a un subíndice o superíndice, se pretende significar que este "→" se adjunta como superíndice al subíndice o superíndice.

En la siguiente descripción, los procesos de primitivas criptográficas incluyen no solamente un proceso criptográfico estrechamente definido para mantener la seguridad de información con respecto a un tercero, sino que incluyen también un proceso de firma. Los procesos de las primitivas criptográficas incluyen un proceso de generación de claves, un proceso de cifrado, un proceso de descifrado, un proceso de firma y un proceso de verificación.

Realización 1

En la Realización 1, se describirán conceptos básicos en los cuales se basa un esquema de cifrado por predicados de productos internos, y, a continuación, se describirá una realización del esquema de cifrado por predicados de productos internos.

En primer lugar, se describirán las dimensiones de un vector de atributos $\vec{x}$ y un vector de predicados $\vec{v}$.

En segundo lugar, se describirá la adición de una categoría de atributo.

En tercer lugar, se describirá un espacio que presenta una estructura matemática enriquecida denominada "espacios vectoriales con emparejamiento dual (DPVS)", que es un espacio para implementar el esquema de cifrado mediante predicados de productos internos.

En cuarto lugar, se describirá un esquema de cifrado mediante predicados de productos internos (Tipo 1) según la Realización 1.

<1. Dimensiones del vector de atributos $\vec{x}$ y de $\vec{v}$>

En los esquemas de cifrado mediante predicados de productos internos, descritos en las referencias bibliográficas 30 y 31, que no son documentos de patente, existe la restricción de que las dimensiones del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ deberían ser equivalentes. Se considera que esta restricción es inevitable para la relación de producto interno en $\vec{x} \cdot \vec{v}$. No obstante, es necesario que esta restricción se relaje para mejorar la eficiencia en varias aplicaciones.

Como ejemplo, se considerarán datos genéticos de un individuo. Los datos genéticos de un individuo deberían tratarse con cuidado, y se deberían cifrar con vistas al procesado y las recuperaciones de datos. Aunque los datos genéticos pueden incluir una cantidad enorme de información, en muchas aplicaciones se usa característicamente solo una parte de los datos genéticos.

Por ejemplo, para comprobar si los datos genéticos de Alice presentan una cierta característica, se determina para unas pocas (por ejemplo, tres) propiedades genéticas diana de entre muchas (por ejemplo, 100) propiedades genéticas, si los datos genéticos de Alice cumplen la condición. No es necesario determinar para el resto (97 elementos) de las propiedades genéticas si se cumple la condición.

Por ejemplo, sean $X_1, ..., X_{100}$ 100 propiedades genéticas y sean $x_1, ..., x_{100}$ los valores de Alice de las 100 propiedades genéticas. Para evaluar si $f(x_1, ..., x_{100}) = 0$ para un polinomio de comprobación (multivariable) f con grado 3, o para evaluar el valor de verdad de un predicado correspondiente $\varphi_f(x_1, ..., x_{100})$, el vector de atributos de

Alice $\vec{x}$ se convierte en un vector monómico de valores de Alice con grado 3, $\vec{x}$:= (1, x_1, ..., x_{100}, x_1^2, x_1x_2, ..., x_{100}^2, x_1^3, $x_1^2x_2$, ..., x_{100}^3). La dimensión de este vector de atributos $\vec{x}$ es aproximadamente 10^6.

Sea una expresión de comprobación (predicado) usada para la comprobación bc $((X_5 = a) \vee (X_{16}=b)) \wedge (X_{57} = c)$, que se centra en solamente tres propiedades genéticas X_5, X_{16} y X_{57}. Esto se representa con un polinomio $r_1(X_5 – a)(X_{16} – b) + r_2(X_{57} – c) = 0$ (donde r_1 y r_2 son números aleatorios uniformes). Este polinomio se puede convertir en $(r_1ab-r_2c) – r_1bX_5 – r_1aX_{16} + r_2X_{57} + r_1X_5X_{16} = 0$. Para que $r_1(X_5-a)(X_{16}-b) + r_2(X_{57} – c) = 0$ si y solamente si $\vec{x}\cdot\vec{v} = 0$, el vector de predicados $\vec{v}$ se convierte en $((r_1ab – r_2c), 0, ..., 0, -r_1b, 0, ..., 0, -r_1a, 0, ..., 0, r_2, 0, ..., 0, r_1, 0, ..., 0)$. La dimensión de este vector de predicados $\vec{v}$ es equivalente a la del vector de atributos $\vec{x}$, es decir, aproximadamente 10^6, aunque la dimensión efectiva (dimensión con elementos diferentes de 0) es solamente 5.

De esta manera, se requiere que la dimensión del vector de predicados $\vec{v}$ sea aproximadamente 10^6, aunque la dimensión efectiva real es 5. Esto es debido a la restricción de que las dimensiones del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ deberían ser equivalentes. La eliminación de esta restricción permite que el vector de predicados $\vec{v}$ se construya con solamente la dimensión efectiva (5 en este caso).

Se ha descrito en la presente que el vector de predicados $\vec{v}$ se construye con solamente la dimensión efectiva. De manera similar, el vector de atributos $\vec{x}$ se puede construir con solamente la dimensión efectiva.

<2. Adición de categoría de atributo>

Una categoría de atributo es una clasificación de un atributo de cada usuario, tal como organización de pertenencia, departamento de pertenencia, posición en la empresa, edad y género.

Los esquemas de cifrado mediante predicados de productos internos que se describirán en las siguientes realizaciones materializan un control de acceso basándose en el atributo del usuario. Por ejemplo, con un proceso criptográfico estrechamente definido para proteger información contra un tercero, el hecho de si el usuario puede descifrar o no un texto cifrado se controla basándose en el atributo del usuario.

En general, las categorías de atributo usadas para el control de acceso se determinan de antemano en la fase de diseño de un sistema. No obstante, puede darse un caso en el que las reglas operativas del sistema se cambien en una fase posterior, requiriendo la adición de una categoría de atributo usada para el control de acceso.

Por ejemplo, supóngase que se construye un sistema criptográfico considerando que el sistema se va a utilizar solamente dentro de la Empresa A. En este caso, se supone que las categorías de atributo a utilizar serán, por ejemplo, departamento de pertenencia, posición en la empresa, e ID individual. No obstante, supóngase que las reglas operativas se cambian en una fase posterior, de manera que el sistema criptográfico se usa, no solamente en la Empresa A, sino también en empresas asociadas de la Empresa A. En este caso, es necesario fijar nuevamente la empresa de pertenencia como categoría de atributo a utilizar.

Si las categorías de atributo usadas para el control de acceso se especifican mediante un parámetro público, la adición de una categoría de atributo en una fase posterior requiere que el parámetro público se vuelva a emitir y que se vuelva a distribuir para cada usuario. Por este motivo, una categoría de atributo no se puede añadir fácilmente en una fase posterior, y no se puede adoptar de manera flexible un modo operativo que no se haya tenido en cuenta en la fase de diseño del sistema.

Por lo tanto, es importante permitir la adición de una categoría de atributo sin volver a emitir el parámetro público.

Para permitir la adición de una categoría de atributo sin volver a emitir el parámetro público, se aplica una técnica de indexación a un cifrado de sistema dual en espacios vectoriales con emparejamiento dual.

En el cifrado de sistema dual en espacios vectoriales con emparejamiento dual, se generan aleatoriamente una base B y una base B^* las cuales son bases duales. A continuación, una parte (base B^) de la base B usa como parámetro público.

En el esquema de cifrado por predicados de productos internos que se describe en la referencia bibliográfica 31, la cual no es documento de patente, se generan una base $B^\wedge_1$..., y una base $B^\wedge_d$ como parámetro público. A continuación, se asigna una categoría de atributo a una base $B^\wedge_t$ para cada entero t =1,..., d. Es decir, pueden gestionarse d elementos de categorías de atributo.

Obsérvese en este caso que la base $B^\wedge_1$,..., y la base $B^\wedge_d$ se usan como parámetro público. Tal como resulta evidente a partir de esto, se requiere que el parámetro público se vuelva a emitir para añadir una base B^, es decir, para incrementar el valor de d en una fase posterior. Es decir, el valor de d queda acotado por el parámetro público.

En el esquema de cifrado mediante predicados de productos internos que se describirá posteriormente en la presente, como parámetro público se genera una base B^. A continuación, vectores de índices bidimensionales, $\sigma_t(1, t)$ y $\mu_i(t, -1)$, se fijan en un texto cifrado c y una clave secreta k^*, respectivamente, para cada entero t = 1, ..., d, y se asigna una categoría de atributo a cada entero t. Es decir, se pueden gestionar d elementos de categorías de

atributo.

Obsérvese en este caso que el parámetro público incluye la base B^, pero no incluye los vectores de índices. Así, no se requiere que el parámetro público se vuelva a emitir para añadir un vector de índice con el fin de incrementar el valor de d en una fase posterior. Es decir, el valor de d no queda acotado por el parámetro público.

<3. Espacios vectoriales con emparejamiento dual>

El esquema de cifrado mediante predicados de productos internos que se describirá posteriormente en la presente se materializa en espacios vectoriales con emparejamiento dual.

En primer lugar, se describirán grupos con emparejamiento bilineal simétrico.

Los grupos con emparejamiento bilineal simétrico (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 polinómico e: G x G → G_T. Emparejamiento bilineal no degenerado significa e(sg, tg) = e(g, g)st, y e(g, g) ≠ 1.

En la siguiente descripción, sea G_{bpg} un algoritmo que toma como entrada 1^λ y da salida a valores de un parámetro $param_G := (q, G, G_T, g, e)$ de grupos de emparejamiento bilineal con un parámetro de seguridad λ.

A continuación se describirán espacios vectoriales con emparejamiento dual.

Los espacios vectoriales con emparejamiento dual (q, V, G_T, A, e) se pueden construir mediante un producto directo de los grupos de emparejamiento bilineal simétrico ($param_G := (q, G, G_T, g, e)$). Los espacios vectoriales con emparejamiento dual (q, V, G_T, A, e) son una tupla de un número primo q, un espacio vectorial N-dimensional V sobre F_q indicado en la Fórmula 112, un grupo cíclico G_T de orden q, y una base canónica A := (a_1, ..., a_N) del espacio V, y presentan las siguientes operaciones (1) y (2), donde a_i es tal como se indica en la Fórmula 113.

[Fórmula 112]

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

[Fórmula 113]

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

Operación (1): emparejamiento bilineal no degenerado

Un emparejamiento en el espacio V se define mediante la Fórmula 114.

[Fórmula 114]

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

donde

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

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

Operación (2): mapas de distorsión

Transformaciones lineales $\varphi_{i,j}$ en el espacio V indicado en la Fórmula 115 pueden materializar la Fórmula 116.

[Fórmula 115]

Si $\phi_{i,j}(a_j) = a_i$ y

$k \neq j$ entonces $\phi_{i,j}(a_k) = 0$.

[Fórmula 116]

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

donde

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

A las transformaciones lineales $\varphi_{i,j}$ se les denominarán mapas de distorsión.

En la siguiente descripción, sea G_{dpvs} un algoritmo que toma como entrada 1^{λ} ($\lambda \in$ número natural), $N \in$ número natural, y valores de un parámetro $param_G := (q, G, G_T, g, e)$ de grupos de emparejamiento bilineal, y da salida a valores de un parámetro $param_v := (q, V, G_T, A, e)$ de espacios vectoriales con emparejamiento dual con un parámetro de seguridad λ y un espacio N-dimensional V.

La descripción se centrará aquí en un caso en el que los espacios vectoriales con emparejamiento dual se construyen usando los grupos de emparejamiento bilineal simétrico antes descritos. Los espacios vectoriales con emparejamiento dual también se pueden construir usando grupos de emparejamiento bilineal asimétrico. La siguiente descripción se puede adaptar fácilmente a un caso en el que los espacios vectoriales con emparejamiento dual se construyen usando grupos de emparejamiento bilineal asimétrico.

<4. Esquema de cifrado mediante predicados de productos internos>

El esquema de cifrado mediante predicados de productos internos tiene cuatro algoritmos probabilísticos de tiempo polinómico: *Setup, KeyGen, Enc y Dec.*

(Setup)

Un algoritmo *Setup* toma como entrada un parámetro de seguridad 1^{λ}, y da salida a una clave pública maestra pk y a una clave secreta maestra sk.

(KeyGen)

Un algoritmo *KeyGen* toma como entrada la clave pública maestra pk, la clave secreta maestra sk, y un vector de predicado $\vec{v}$, y da salida a una clave secreta sk_v.

(Enc)

Un algoritmo *Enc* toma como entrada la clave pública maestra pk, un vector de atributos $\vec{x}$, y un mensaje m, y da salida a un texto cifrado ct_x.

(Dec)

Un algoritmo *Dec* toma como entrada la clave pública maestra pk, la clave secreta sk_v, y el texto cifrado ct_x, y da salida al mensaje m o a un símbolo distinguido $\perp$. El símbolo distinguido $\perp$ es información que indica un fallo de descifrado.

Se describirá un sistema 10 de procesado criptográfico que ejecuta los algoritmos del esquema de cifrado mediante predicados de productos internos.

La Fig. 1 es un diagrama de configuración del sistema 10 de procesado criptográfico según la Realización 1.

El sistema 10 de procesado criptográfico tiene un dispositivo 100 de generación de claves, un dispositivo 200 de cifrado (transmisor), y un dispositivo 300 de descifrado (receptor).

El dispositivo 100 de generación de claves ejecuta el algoritmo *Setup* tomando como entrada un parámetro de seguridad λ, y genera así una clave pública maestra pk y una clave secreta maestra sk. A continuación, el dispositivo 100 de generación de claves publica la clave pública maestra generada pk. El dispositivo 100 de generación de claves ejecuta también el algoritmo *KeyGen* tomando como entrada la clave pública maestra pk, la clave secreta maestra sk, y un vector de predicado $\vec{v}$, y genera así una clave secreta sk_v, y distribuye la clave secreta sk_v al dispositivo 300 de descifrado en secreto.

El dispositivo 200 de cifrado ejecuta el algoritmo *Enc* tomando como entrada la clave pública maestra pk, un vector de atributo $\vec{x}$, y un mensaje m, y genera así un texto cifrado ct_x. El dispositivo 200 de cifrado transmite el texto cifrado generado ct_x al dispositivo 300 de descifrado.

El dispositivo 300 de descifrado ejecuta el algoritmo *Dec* tomando como entrada la clave pública maestra pk, la clave

secreta sk_v, y el texto cifrado ct_x, y da salida al mensaje m o al símbolo distinguido $\perp$.

La Fig. 2 es un diagrama de bloques funcional que ilustra la función del dispositivo 100 de generación de claves según la Realización 1. La Fig. 3 es un diagrama de bloques funcional que ilustra la función del dispositivo 200 de cifrado según la Realización 1. La Fig. 4 es un diagrama de bloques funcional que ilustra la función del dispositivo 300 de descifrado según la Realización 1.

Las Figs. 5 y 6 son diagramas de flujo que ilustran el funcionamiento del dispositivo 100 de generación de claves según la Realización 1. La Fig. 5 es un diagrama de flujo que ilustra el proceso del algoritmo *Setup* según la Realización 1, y la Fig. 6 es un diagrama de flujo que ilustra el proceso del algoritmo *KeyGen* según la Realización 1. La Fig. 7 es un diagrama de flujo que ilustra el funcionamiento del dispositivo 200 de cifrado según la Realización 1 y que ilustra el proceso del algoritmo *Enc* según la Realización 1. La Fig. 8 es un diagrama de flujo que ilustra el funcionamiento del dispositivo 300 de descifrado según la Realización 1 y que ilustra el proceso del algoritmo *Dec* según la Realización 1.

Se describirán la función y el funcionamiento del dispositivo 100 de generación de claves.

Tal como se ilustra en la Fig. 2, el dispositivo 100 de generación de claves tiene una unidad 110 de generación de claves maestras, una unidad 120 de almacenamiento de claves maestras, una unidad 130 de introducción de información, una unidad 140 de generación de claves de descifrado, y una unidad 150 de distribución de claves.

En primer lugar, en referencia a la Fig. 5, se describirá el proceso del algoritmo *Setup*.

(S101: etapa de generación de bases ortonormales)

Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras calcula la Fórmula 117, y genera así un parámetro param, una base B_0 y una base B^*_0, y una base B_1 (base B) y una base B^*_1 (base B^*).

[Fórmula 117]

(1) entrada 1^λ

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

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

$N_0 := 1 + u_0 + 1 + w_0 + z_0, \quad N_1 := 4 + u + w + z$

El proceso (4) a (8) se ejecuta para cada t =0, 1.

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

(5) $X_t := (\chi_{t,i,j})_{i,j=1,\ldots,N_t} \xleftarrow{U} GL(N_t, \mathbb{F}_q)$

(6) $X_t^* := (\vartheta_{t,i,j})_{i,j=1,\ldots,N_t} := \psi \cdot (X_t^T)^{-1}$

(7) $\boldsymbol{b}_{t,i} := (\vec{x}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j}$ para i = 1, ..., N_t,

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

(8) $\boldsymbol{b}^*_{t,i} := (\vec{\vartheta}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j}$ para i = 1, ..., N_t,

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

(9) $g_T := e(g,g)^\psi$,
$\text{param} := (\{\text{param}_{\mathbb{V}_t}\}_{t=0,1}, g_T)$

Es decir, la unidad 110 de generación de claves maestras ejecuta el siguiente proceso.

(1) Usando el dispositivo de entrada, la unidad 110 de generación de claves maestras toma como entrada un parámetro de seguridad λ (1^{λ}).

(2) Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras ejecuta el algoritmo G_{bpg} tomando como entrada el parámetro de seguridad $\lambda(1^{\lambda})$ introducido en (1), y genera así valores de un parámetro $param_G := (q, G, G_T, g, e)$ de grupos de emparejamiento bilineal.

(3) Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras genera un número aleatorio ψ, fija $1 + u_0 + 1 + w_0 + z_0$ en N_0, y fija $4 + u + w + z +$ en N_1, donde u_0, w_0, z_0, u, w, y z son, cada uno de ellos, un entero de 0 ó más.

A continuación, la unidad 110 de generación de claves maestras ejecuta el siguiente proceso (4) a (8) para cada t = 0, 1.

(4) Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras ejecuta el algoritmo G_{dpvs} tomando como entrada el parámetro de seguridad $\lambda(1^{\lambda})$ introducido en (1), la N_t fijada en (3), y los valores de $param_G := (q, G, G_T, g, e)$ generados en (2), y genera así valores de un parámetro $param_{Vt} := (q, V_t, G_T, A_t, e)$ de espacios vectoriales con emparejamiento dual.

(5) Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras toma como entrada la N_t fijada en (3) y F_q, y genera aleatoriamente una transformación lineal $X_t := (\chi_{t,i,j})_{i,j}$. Obsérvese que GL significa lineal general. En otras palabras, GL es un grupo lineal general, un conjunto de matrices cuadradas con determinantes diferentes de 0, y un grupo bajo multiplicación. Obsérvese que $(\chi_{t,i,j})_{i,j}$ indica una matriz con respecto a los sufijos i y j de la matriz $\chi_{t,i,j}$, donde i, j, = 1, .., N_t.

(6) Usando el dispositivo de procesado y basándose en el número aleatorio ψ y la transformación lineal X_t, la unidad 110 de generación de claves maestras genera $(v_{t,i,j})_{i,j} := \psi \cdot (X_t^T)^{-1}$. Igual que $(\chi_{t,i,j})_{i,j}$, $(v_{t,i,j})_{i,j}$ indica una matriz con respecto a los sufijos i y j de la matriz $v_{t,i,j}$, donde i, j = 1, ..., N_t.

(7) Usando el dispositivo de procesado y basándose en la transformación lineal X_t generada en (5), la unidad 110 de generación de claves maestras genera una base B_t a partir de la base ortonormal A_t generada en (4). Obsérvese que $\vec{x}_{t,i}$ indica la fila i-ésima de la transformación lineal X_t.

(8) Usando el dispositivo de procesado y basándose en la $(v_{t,i,j})_{i,j}$ generada en (6), la unidad 110 de generación de claves maestras genera una base B^*_t a partir de la base ortonormal A_t generada en (4). Obsérvese que $\vec{v}_{t,i}$ indica la fila i-ésima de la transformación lineal X^*_t.

(9) Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras fija $e(g,g)^{\psi}$ en g_T. La unidad 110 de generación de claves maestras fija también $\{param_{vt}\}_{t=0,1}$ generada en (4) y g_T en param.

Resumiendo, en S101, la unidad 110 de generación de claves maestras genera param, la base B_0 y la base B^*_0, y la base B_1 (base B) y la base B^*_1 (base B^*) ejecutando el algoritmo G_{ob} indicado en la Fórmula 118.

[Fórmula 118]

$$\mathcal{G}_{\mathrm{ob}}(1^{\lambda}):$$

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

$$N_0 := 1 + u_0 + 1 + w_0 + z_0,\ N_1 := 4 + u + w + z,$$

para $t = 0,1$,

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

$$X_t := (\chi_{t,i,j})_{i,j=1,\ldots,N_t} \xleftarrow{U} GL(N_t, \mathbb{F}_q),$$

$$X_t^* := (\vartheta_{t,i,j})_{i,j=1,\ldots,N_t} := \psi \cdot (X_t^T)^{-1},\ \text{en lo sucesivo,}\ \vec{x}_{t,i}$$

y $\vec{\vartheta}_{t,i}$ indica que las filas *i*-ésimas de X_t y X^*_t para i= 1, ..., N_t, respectivamente,

$\boldsymbol{b}_{t,i} := (\vec{\chi}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j}$ para i = 1, ..., N_t, $\mathbb{B}_t := (\boldsymbol{b}_{t,1},...,\boldsymbol{b}_{t,N_t})$,

$\boldsymbol{b}^*_{t,i} := (\vec{\vartheta}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j}$ para i = 1, ..., N_t, $\mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1},...,\boldsymbol{b}^*_{t,N_t})$,

$g_T := e(g,g)^{\psi}$, $\mathsf{param} := (\{\mathsf{param}_{\mathbb{V}_t}\}_{t=0,1}, g_T)$,

devolver, $(\mathsf{param}, \mathbb{B}_t, \mathbb{B}^*_t)$.

Por motivos de simplicidad, la base B_1 y la base B^*_1 se describirán como la base B y la base B^*.

(S102: etapa de generación de parámetros públicos)

Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras genera una sub-base $B\hat{}_0$ de la base B_0 y una sub-base B^ de la base B, según se indica en la Fórmula 119, habiéndose generado las bases B_0 y B en S101.

[Fórmula 119]

$$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1},\ \boldsymbol{b}_{0,1+u_0+1},\ \boldsymbol{b}_{0,1+u_0+1+w_0+1},...,\boldsymbol{b}_{0,1+u_0+1+w_0+z_0}),$$

$$\hat{\mathbb{B}} := (\boldsymbol{b}_1,..,\boldsymbol{b}_4,\ \boldsymbol{b}_{4+u+w+1},..,\ \boldsymbol{b}_{4+u+w+z})$$

La unidad 110 de generación de claves maestras genera un parámetro público pk juntando la sub-base $B\hat{}_0$ y la sub-base B^ generadas, el parámetro de seguridad $\lambda(1^{\lambda})$ introducido en S101, y el param generado en S101.

(S103: etapa de generación de claves maestras)

Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras genera una sub-base $B\hat{}^*_0$ de la base B^*_0 y una sub-base $B\hat{}^*$ de la base B^*, según se indica en la Fórmula 120, habiéndose generado las bases B^*_0 y B^* en S101.

[Fórmula 120]

$$\hat{\mathbb{B}}^*_0 := (\boldsymbol{b}^*_{0,1},\ \boldsymbol{b}^*_{0,1+u_0+1},\ \boldsymbol{b}^*_{0,1+u_0+1+1,...,}\boldsymbol{b}^*_{0,1+u_0+1+w_0}),$$

$$\hat{\mathbb{B}}^* := (\boldsymbol{b}^*_1,..,\boldsymbol{b}^*_4,\ \boldsymbol{b}^*_{4+u+1},..,\ \boldsymbol{b}^*_{4+u+w})$$

La unidad 110 de generación de claves maestras genera una clave maestra sk que está constituida por la sub-base $B\hat{}^*_0$ y la sub-base $B\hat{}^*$ generadas.

(S104: etapa de almacenamiento de claves maestras)

La unidad 120 de almacenamiento de claves maestras almacena el parámetro público pk generado en S102 en el dispositivo de almacenamiento. La unidad 120 de almacenamiento de claves maestras almacena también la clave maestra sk generada en S103 en el dispositivo de almacenamiento.

Resumiendo, en S101 a S103, el dispositivo 100 de generación de claves genera el parámetro público pk y la clave maestra sk ejecutando el algoritmo *Setup* indicado en la Fórmula 121. En S104, el dispositivo 100 de generación de claves almacena el parámetro público generado pk y la clave maestra sk en el dispositivo de almacenamiento.

El parámetro público se publica por medio de la red, por ejemplo, y se hace que esté disponible para el dispositivo 200 de cifrado y el dispositivo 300 de descifrado.

[Fórmula 121]

$$\text{Setup}(1^\lambda):$$

$$(\text{param}, (\mathbb{B}_0, \mathbb{B}_0^*), (\mathbb{B}, \mathbb{B}^*)) \xleftarrow{R} \mathcal{G}_{\text{ob}}(1^\lambda),$$

$$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,1+u_0+1}, \boldsymbol{b}_{0,1+u_0+1+w_0+1}, \ldots, \boldsymbol{b}_{0,1+u_0+1+w_0+z_0}),$$

$$\hat{\mathbb{B}} := (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_4, \boldsymbol{b}_{4+u+w+1}, \ldots, \boldsymbol{b}_{4+u+w+z}),$$

$$\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,1+u_0+1}^*, \boldsymbol{b}_{0,1+u_0+1+1}^*, \ldots, \boldsymbol{b}_{0,1+u_0+1+w_0}^*),$$

$$\hat{\mathbb{B}}^* := (\boldsymbol{b}_1^*, \ldots, \boldsymbol{b}_4^*, \boldsymbol{b}_{4+u+1}^*, \ldots, \boldsymbol{b}_{4+u+w}^*),$$

$$\text{devolver pk} := (1^\lambda, \text{param}, \hat{\mathbb{B}}_0, \hat{\mathbb{B}}), \ \text{sk} := (\hat{\mathbb{B}}_0^*, \hat{\mathbb{B}}^*).$$

En referencia a la Fig. 6, se describirá el proceso del algoritmo *KeyGen*.

(S201: etapa de introducción de información)

Usando el dispositivo de entrada, la unidad 130 de introducción de información toma como entrada un vector de predicados $v^{\rightarrow} := \{(t, v_t) \mid t \in I_{v^{\rightarrow}}\}$. Es decir, el vector de predicados $v^{\rightarrow}$ es un vector que tiene, como elementos, un índice t e información de predicado v_t para el índice t incluido en un conjunto $I_{v^{\rightarrow}}$. Como información de predicado v_t, se fija, por ejemplo, información de atributos de un usuario de una clave de descifrado sk_v.

(S202: etapa de generación de información secreta)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera información secreta s_t y s_0, tal como se indica en la Fórmula 122.

[Fórmula 122]

$$s_t \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad (t, v_t) \in \vec{v},$$

$$s_0 := \sum_{(t,v_t)\in\vec{v}} s_t$$

(S203: etapa de generación de números aleatorios)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera números aleatorios, tal como se indica en la Fórmula 123.

[Fórmula 123]

$$\vec{\eta}_0 := (\eta_{0,1}, \ldots, \eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$$

$$\delta \xleftarrow{U} \mathbb{F}_q,$$

$$\mu_t \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad (t, v_t) \in \vec{v},$$

$$\vec{\eta}_t := (\eta_{t,1}, \ldots, \eta_{t,w}) \xleftarrow{U} \mathbb{F}_q^{w} \quad \text{para} \quad (t, v_t) \in \vec{v},$$

(S204: etapa de generación de elementos de clave)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera un elemento k^*_0 de la clave de descifrado sk_v, tal como se indica en la Fórmula 124.

[Fórmula 124]

$$\boldsymbol{k}_0^* := (-s_0, \overbrace{0^{u_0}}^{u_0}, 1, \overbrace{\vec{\eta}_0}^{w_0}, \overbrace{0^{z_0}}^{z_0},)_{\mathbb{B}_0^*}$$

Tal como se ha descrito anteriormente, para la base B y la base B^* indicadas en la Fórmula 110, se establece la Fórmula 111. Así, la Fórmula 124 significa que $-s_0$ se fija como coeficiente de un vector de base $b^*_{0,1}$ de la base B^*_0, que 0 se fija como coeficiente de los vectores de base $b^*_{0,1+1}$,..., $b^*_{0,1+u0}$, que 1 se fija como coeficiente de un vector de base $b^*_{0,1+u0+1}$, que $\eta_{0,1}$, ..., $\eta_{0,w0}$ se fijan respectivamente como coeficiente de los vectores de base $b^*_{0,1+u0+1+1}$, ..., $b^*_{0,1+u0+1+w0}$, y que 0 se fija como coeficiente de los vectores de base $b^*_{0,1+u0+1+w0+1}$, ..., $b^*_{0,1+u0+1+w0+z0}$, donde u0, w0 y z0 indican respectivamente u_0, w_0 y z_0.

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera también un elemento k^*_t de la clave de descifrado sk_v para el índice t incluido en el conjunto $I_{v\rightarrow}$, según se indica en la Fórmula 125.

[Fórmula 125]

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t, -1,), \delta v_t, s_t}^{4} \overbrace{0^u,}^{u} \overbrace{\vec{\eta}_t,}^{w} \overbrace{0^z}^{z})_{\mathbb{B}^*}$$

Es decir, igual que la Fórmula 124, la Fórmula 125 significa que μ_tt se fija como coeficiente de un vector de base b^*_1 de la base B^*, que $-\mu_t$ se fija como coeficiente de un vector de base b^*_2, que δv_t se fija como coeficiente de un vector de base b^*_3, que s_t se fija como coeficiente de un vector de base b^*_4, que 0 se fija como coeficiente de los vectores de base b^*_{4+1},..., b^*_{4+u}, que $\eta_{t,1}$,..., $\eta_{t,w}$ se fijan respectivamente como coeficiente de los vectores de base b^*_{4+u+1},..., b^*_{4+u+w}, y que 0 se fija como coeficiente de los vectores de base $b^*_{4+u+w+1}$, ..., $b^*_{4+u+w+z}$.

(S205: etapa de distribución de claves)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 150 de distribución de claves distribuye la clave de descifrado sk_v que tiene, como elementos, k^*_0 y k^*_t generados en S204, para el dispositivo 300 de descifrado en secreto. Naturalmente, la clave de descifrado sk_v se puede distribuir al dispositivo 300 de descifrado mediante otro método.

Resumiendo, en S201 a S204, el dispositivo 100 de generación de claves genera la clave de descifrado sk_v ejecutando el algoritmo *KeyGen* indicado en la Fórmula 126. En S205, el dispositivo 100 de generación de claves distribuye la clave de descifrado generada sk_v al dispositivo 300 de descifrado.

[Fórmula 126]

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \vec{v} := \{(t, v_t) \mid t \in I_{\vec{v}}\}):$

$\delta, s_t \xleftarrow{U} \mathbb{F}_q$ para $(t, v_t) \in \vec{v},$

$\vec{\eta}_0 := (\eta_{0,1}, \ldots, \eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$

$s_0 := \sum_{(t,v_t)\in\vec{v}} s_t,$

$\boldsymbol{k}_0^* := (-s_0, \overbrace{0^{u_0}}^{u_0}, 1, \overbrace{\vec{\eta}_0}^{w_0}, \overbrace{0^{z_0}}^{z_0},)_{\mathbb{B}_0^*},$

para $(t, v_t) \in \vec{v},$

$\mu_t \xleftarrow{U} \mathbb{F}_q,$

$\vec{\eta}_t := (\eta_{t,1}, \ldots, \eta_{t,w}) \xleftarrow{U} \mathbb{F}_q^w,$

$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t, -1,), \delta v_t, s_t}^{4} \overbrace{0^u,}^{u} \overbrace{\vec{\eta}_t,}^{w} \overbrace{0^z}^{z})_{\mathbb{B}^*},$

devolver $\mathsf{sk}_v := (\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}_{(t,v_t)\in\vec{v}}).$

Se describirán la función y el funcionamiento del dispositivo 200 de cifrado.

El dispositivo 200 de cifrado tiene una unidad 210 de adquisición de parámetros públicos, una unidad 220 de introducción de información, una unidad 230 de generación de textos cifrados, y una unidad 240 de transmisión de datos.

En referencia a la Fig. 7, se describirá el proceso del algoritmo *Enc*.

(S301: etapa de adquisición de parámetros públicos)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 210 de adquisición de parámetros públicos obtiene el parámetro público pk generado por el dispositivo 100 de generación de claves.

(S302: etapa de introducción de información)

Usando el dispositivo de entrada, la unidad 220 de introducción de información toma como entrada un mensaje m que se va a transmitir hacia el dispositivo 300 de descifrado. Usando el dispositivo de entrada, la unidad 220 de introducción de información toma también como entrada un vector de atributos $\vec{x} := \{(t, x_t) \mid t \in I_{\vec{x}}\}$. Es decir, el vector de atributos $\vec{x}$ es un vector que tiene, como elementos, un índice t e información de atributos x_t para el índice t incluido en un conjunto $I_{\vec{x}}$. En el vector de atributos $\vec{x}$, se fija, por ejemplo, información de atributos de un usuario, compatible con el descifrado.

(S303: etapa de generación de números aleatorios)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera números aleatorios, tal como se indica en la Fórmula 127.

[Fórmula 127]

$$\omega, \tilde{\omega}, \zeta \xleftarrow{U} \mathbb{F}_q,$$

$$\vec{\varphi}_0 := (\varphi_{0,1}, \ldots, \varphi_{0,z_0}) \xleftarrow{U} \mathbb{F}_q^{z_0},$$

$$\sigma_t \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad (t, x_t) \in \vec{x},$$

$$\vec{\varphi}_t := (\varphi_{t,1}, \ldots, \varphi_{t,z}) \xleftarrow{U} \mathbb{F}_q^{z} \quad \text{para} \quad (t, x_t) \in \vec{x}$$

(S304: Etapa de generación de elementos cifrados)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera un elemento c_0 de un texto cifrado ct_x, tal como se indica en la Fórmula 128.

[Fórmula 128]

$$c_0 := (\tilde{\omega}, \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0}$$

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera también un elemento c_t del texto cifrado ct_x para el índice t incluido en el conjunto $I_{\vec{x}}$, tal como se indica en la Fórmula 129.

[Fórmula 129]

$$c_t = (\overbrace{\sigma_t(1, t), \omega x_t, \tilde{\omega}}^{4}, \overbrace{0^u}^{u}, \overbrace{0^w}^{w}, \overbrace{\vec{\varphi}_t}^{z})_{\mathbb{B}}$$

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera también un elemento c_T del texto cifrado ct_x, tal como indica en la Fórmula 130.

[Fórmula 130]

$$c_T := g_T^{\zeta} m$$

(S305: etapa de transmisión de datos)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 240 de transmisión de datos transmite el texto cifrado ct_x que tiene, como elementos, los c_0, c_t y c_T generados en S304, al dispositivo 300 de descifrado. Naturalmente, el texto cifrado ct_x se puede transmitir al dispositivo 300 de descifrado mediante otro método.

Resumiendo, en S301 a S304, el dispositivo 200 de cifrado genera el texto cifrado ct_x ejecutando el algoritmo *Enc* indicado en la Fórmula 131. En S305, el dispositivo 200 de cifrado transmite el texto cifrado generado ct_x al dispositivo 300 de descifrado.

[Fórmula 131]

$$\text{Enc}(\text{ pk}, m, \vec{x} := \{(t, x_t) \mid t \in I_{\vec{x}}\}):$$

$$\omega, \tilde{\omega}, \zeta \xleftarrow{\cup} \mathbb{F}_q,\ \vec{\varphi}_0 := (\varphi_{0,1}, \ldots, \varphi_{0,z_0}) \xleftarrow{\cup} \mathbb{F}_q^{z_0},$$

$$c_0 := (\tilde{\omega}, \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0},$$

para $(t, x_t) \in \vec{x},$

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

$$c_t = (\overbrace{\sigma_t(1, t), \omega x_t, \tilde{\omega}}^{4}\ \overbrace{0^{u}}^{u},\ \overbrace{0^{w}}^{w},\ \overbrace{\vec{\varphi}_t}^{z})_{\mathbb{B}},$$

$$c_T := g_T^{\zeta} m$$

devolver $\text{ct}_x := (c_0, \{c_t\}_{(t, x_t) \in \vec{x}}, c_T).$

Se describirán la función y el funcionamiento del dispositivo 300 de descifrado.

El dispositivo 300 de descifrado tiene una unidad 310 de adquisición de claves de descifrado, una unidad 320 de almacenamiento de claves de descifrado, una unidad 330 de adquisición de textos cifrados, una unidad 340 de operaciones de emparejamiento, y una unidad 350 de cálculo de mensajes. A la unidad 340 de operaciones de emparejamiento y a la unidad 350 de cálculo de mensajes se les hará referencia en conjunto como unidad de descifrado.

En referencia a la Fig. 8, se describirá el proceso del algoritmo *Dec*.

(S401: etapa de adquisición de claves de descifrado)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 310 de adquisición de claves de descifrado obtiene la clave de descifrado sk_v distribuida por el dispositivo 100 de generación de claves. La unidad 310 de adquisición de claves de descifrado obtiene también el parámetro público pk generado por el dispositivo 100 de generación de claves.

La unidad 310 de adquisición de claves de descifrado almacena la clave de descifrado obtenida sk_v y el parámetro público pk en la unidad 320 de almacenamiento de claves de descifrado.

(S402: etapa de adquisición de textos cifrados)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 330 de adquisición de textos cifrados recibe el texto cifrado ct_x transmitido por el dispositivo 200 de cifrado.

(S403: etapa de operaciones de emparejamiento)

Usando el dispositivo de procesado, la unidad 340 de operaciones de emparejamiento calcula la Fórmula 132, y genera así una clave de sesión $K = g_T^{\zeta}$.

[Fórmula 132]

$$K := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \prod_{t \in I_v^{\rightarrow}} e(\boldsymbol{c}_t, \boldsymbol{k}_t^*)$$

Si se cumple la Fórmula 133, la clave $K = g_T^{\zeta}$ puede obtenerse calculando la Fórmula 132, tal como se indica en la Fórmula 134.

[Fórmula 133]

$$I_v^{\rightarrow} \subset I_x^{\rightarrow} \quad \text{y}$$

$$\sum_{t \in I_v^{\rightarrow}} v_t \cdot x_t = 0$$

[Fórmula 134]

$$\begin{aligned} K &:= e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \prod_{t \in I_v^{\rightarrow}} e(\boldsymbol{c}_t, \boldsymbol{k}_t^*) \\ &= g_T^{-\tilde{\omega} s_0 + \varsigma} \cdot \prod_{t \in I_v^{\rightarrow}} g_T^{\delta \omega v_t x_t + \tilde{\omega} s_t} \\ &= g_T^{-\tilde{\omega} s_0 + \varsigma} \cdot g_T^{\delta \omega (\sum_{(t, v_t) \in I_v^{\rightarrow}} v_t x_t) + \tilde{\omega} (\sum_{(t, v_t) \in I_v^{\rightarrow}} s_t)} \\ &= g_T^{-\tilde{\omega} s_0 + \varsigma + \tilde{\omega} s_0} \\ &= g_T^{\varsigma} \end{aligned}$$

Es decir, si el conjunto $I_{v\rightarrow}$ es un subconjunto del conjunto $I_{x\rightarrow}$, y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{v\rightarrow}$, la clave $K = g_T^{\zeta}$ se puede obtener calculando la Fórmula 132.

(S404: etapa de cálculo de mensajes)

Usando el dispositivo de procesado, la unidad 350 de cálculo de mensajes calcula $m' = c_T/K$, y genera así un mensaje m' (=m). Obsérvese que c_T es $g_T^{\zeta} m$ tal como se indica en la Fórmula 130, y K es g_T^{ζ}. Así, el mensaje m se puede obtener calculando $m' = c_T/K$.

Resumiendo, en S401 a S404, el dispositivo 300 de descifrado genera el mensaje m' (=m) ejecutando el algoritmo *Dec* indicado en la Fórmula 135.

[Fórmula 135]

$\mathrm{Dec}(\mathrm{pk}, \mathrm{sk}_v := (\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}_{(t, v_t) \in \vec{v}}),\ \mathrm{ct}_x := (\boldsymbol{c}_0, \{\boldsymbol{c}_t\}_{(t, x_t) \in \vec{x}}, c_T))$:

si $I_v^{\rightarrow} \subset I_x^{\rightarrow}$ y $\sum_{t \in I_v^{\rightarrow}} v_t x_t = 0,$

$$K := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \prod_{t \in I_v^{\rightarrow}} e(\boldsymbol{c}_t, \boldsymbol{k}_t^*),$$

devolver $m' := c_T / K.$

Tal como se ha descrito anteriormente, en el esquema de cifrado mediante predicados de productos internos según la Realización 1, incluso si las dimensiones del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ no son equivalentes, el texto cifrado ct_x se puede descifrar con la clave de descifrado sk_v si el conjunto $I_{v\rightarrow}$ es un subconjunto del conjunto $I_{x\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{v\rightarrow}$.

En el esquema de cifrado por predicados de productos internos según la Realización 1, $\mu_t t$ y $-\mu_t$ se fijan respectivamente como el coeficiente de los vectores de base b^*_1 y b^*_2 (vectores de base $b^*_{índice}$) para el elemento k^*_t de la clave de descifrado sk_v. En el sistema criptográfico 10, σ_t y $\sigma_t t$ se fijan respectivamente como el coeficiente de los vectores de base b_1 y b_2 (vectores de base $b_{índice}$) para el elemento c_t del texto cifrado ct_x.

Debido a estas disposiciones, cuando se ejecuta una operación de emparejamiento sobre el elemento k^*_t y el elemento c_t para el índice correspondiente t, el producto interno se convierte en 0 para aquellas partes constituidas por los vectores de base b^*_1 y b^*_2 y los vectores de base b_1 y b_2, que de este modo se anulan. Es decir, cuando se lleva a cabo una operación de emparejamiento sobre el elemento k^*_t y el elemento c_t para el índice correspondiente t, las partes de índice que se fijan como coeficientes de los vectores de base (partes constituidas por los vectores de base b^*_1 y b^*_2 y los vectores de base b_1 y b_2) se anulan, y se obtiene un resultado de la operación de emparejamiento para las partes restantes.

En el esquema de cifrado mediante predicados de productos internos según la Realización 1, las partes de índice se proporcionan de manera que las bases que se usan para cada categoría de atributo se pueden construir como bases comunes (base B y base B^*). Como consecuencia, solamente es necesario incluir en un parámetro público la base B y la base B^*, eliminando la necesidad de volver a emitir el parámetro público cuando vaya a añadirse una categoría de atributo en una fase posterior.

Se requiere para las partes de índice que se obtenga 0 como consecuencia de una operación de productos internos. Por lo tanto, aunque en la descripción anterior se utilizan las partes de índice bidimensionales, concretamente los vectores de base b^*_1 y b^*_2 y los vectores de base b_1 y b_2, las partes de índice no se limitan a ser bidimensionales y pueden ser tridimensionales o ser de una dimensión mayor. Los valores asignados a las partes de índice no se limitan a aquellos que se han descrito anteriormente, y puede utilizarse una disposición de asignación diferente.

En la anterior descripción, las partes de índice se proporcionan de manera que las bases que se usan para cada categoría de atributo se construyen como bases comunes (base B y base B^*). No obstante, en un caso en el que se permita volver a emitir el parámetro público para añadir una categoría de atributo en una fase posterior, no hay necesidad de proporcionar las partes de índice si se dispone que las bases que se usan para cada categoría de atributo sean respectivamente diferentes.

En este caso, el algoritmo G_{ob}, el algoritmo *Setup*, el algoritmo *KeyGen* y el algoritmo *Enc* del esquema de cifrado mediante predicados de productos internos descrito anteriormente son tal como se indica en la Fórmula 136 a la Fórmula 139. El algoritmo *Dec* es tal como se indica en la Fórmula 135, sin ningún cambio.

[Fórmula 136]

$$\mathcal{G}_{\mathrm{ob}}(1^\lambda):$$

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

$$N_0 := 1 + u_0 + 1 + w_0 + z_0,\ N_t := 2 + u_t + w_t + z_t \text{ para } t = 1,...,d,$$

para $t = 0, ..., d$,

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

$$X_t := (\chi_{t,i,j})_{i,j=1,..,N_t} \xleftarrow{U} GL(N_t, \mathbb{F}_q),$$

$$X_t^* := (\vartheta_{t,i,j})_{i,j=1,..,N_t} := \psi \cdot (X_t^T)^{-1},\ \text{en lo sucesivo, } \vec{\chi}_{t,i}$$

y $\vec{\vartheta}_{t,i}$ indican las filas i-ésimas de X_t y X^*_t para $i = 1, ..., N_t$, respectivamente,

$$\boldsymbol{b}_{t,i} := (\vec{\chi}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j} \text{ para } i = 1,..., N_t, \quad \mathbb{B}_t := (\boldsymbol{b}_{t,1},...,\boldsymbol{b}_{t,N_t}),$$

$$\boldsymbol{b}^*_{t,i} := (\vec{\vartheta}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j} \text{ para } i = 1,..., N_t, \quad \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1},...,\boldsymbol{b}^*_{t,N_t}),$$

$$g_T := e(g,g)^\psi,\ \mathrm{param} := (\{\mathrm{param}_{\mathbb{V}_t}\}_{t=0,...,d}, g_T),$$

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

[Fórmula 137]

$$\mathrm{Setup}(1^{\lambda}):$$

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

$$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,1+u_0+1}, \boldsymbol{b}_{0,1+u_0+1+w_0+1}, \ldots, \boldsymbol{b}_{0,1+u_0+1+w_0+z_0}),$$

$$\hat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,2}, \boldsymbol{b}_{t,2+u_t+w_t+1}, \ldots, \boldsymbol{b}_{t,2+u_t+w_t+z_t}) \quad \text{para } t = 0,\ldots,d,$$

$$\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,1+u_0+1}^*, \boldsymbol{b}_{0,1+u_0+1+1}^*, \ldots, \boldsymbol{b}_{0,1+u_0+1+w_0}^*),$$

$$\hat{\mathbb{B}}_t^* := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{t,2}^*, \boldsymbol{b}_{t,2+u_t+1}^*, \ldots, \boldsymbol{b}_{t,2+u_t+w_t}^*) \quad \text{para } t = 0,\ldots,d,$$

devolver $\mathrm{pk} := (1^{\lambda}, \mathrm{param}, \hat{\mathbb{B}}_0, \{\hat{\mathbb{B}}_t\}_{t=1,\ldots,d})$, $\mathrm{sk} := (\hat{\mathbb{B}}_0^*, \{\hat{\mathbb{B}}_t^*\}_{t=1,\ldots,d})$.

[Fórmula 138]

$$\mathrm{KeyGen}(\mathrm{pk}, \mathrm{sk}, \vec{v} := \{(t, v_t) \mid t \in I_v^-\}):$$

$$\delta, s_t \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad (t, v_t) \in \vec{v},$$

$$\vec{\eta}_0 := (\eta_{0,1}, \ldots, \eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$$

$$s_0 := \sum_{(t,v_t)\in\vec{v}} s_t,$$

$$\boldsymbol{k}_0^* := (-s_0, \overbrace{0^{u_0}}^{u_0}, 1, \overbrace{\vec{\eta}_0}^{w_0}, \overbrace{0^{z_0}}^{z_0})_{\mathbb{B}_0^*},$$

para $(t, v_t) \in \vec{v}$,

$$\mu_t \xleftarrow{U} \mathbb{F}_q,$$

$$\vec{\eta}_t := (\eta_{t,1}, \ldots, \eta_{t,w_t}) \xleftarrow{U} \mathbb{F}_q^{w_t},$$

$$\boldsymbol{k}_t^* := (\overbrace{\delta v_t, s_t}^{2}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\vec{\eta}_t}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*},$$

devolver $\mathrm{sk}_v := (\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}_{(t,v_t)\in\vec{v}})$.

[Fórmula 139]

$$\mathrm{Enc}(\mathrm{pk}, m, \vec{x} := \{(t, x_t) \mid t \in I_{\vec{x}}\}):$$

$$\omega, \tilde{\omega}, \zeta \xleftarrow{\cup} \mathbb{F}_q, \ \vec{\varphi}_0 := (\varphi_{0,1}, \ldots, \varphi_{0,z_0}) \xleftarrow{\cup} \mathbb{F}_q^{z_0},$$

$$c_0 := (\tilde{\omega}, \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0},$$

para $(t, x_t) \in \vec{x}$,

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

$$c_t = (\overbrace{\omega x_t, \tilde{\omega}}^{2}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\varphi}_t}^{z_t})_{\mathbb{B}_t},$$

$$c_T := g_T^{\zeta} m$$

devolver $\mathrm{ct}_{\vec{x}} := (c_0, \{c_t\}_{(t,x_t) \in \vec{x}}, c_T)$.

Es decir, se usan una base B_t diferente y una base B^*_t diferente para cada índice t.

En la anterior descripción, se especifica que u_0, w_0, z_0, u, w y z son, cada uno de ellos, un entero de 0 o más. Por ejemplo, se puede especificar que $u_0 = 1$, $w_0 = 1$, $z_0 = 1$, u= 9, w = 2, y z= 2.

Realización 2

En la Realización 1, se ha descrito el esquema de cifrado mediante predicados de productos internos (Tipo 1) en el que el texto cifrado ct_x se puede descifrar con la clave de descifrado sk_v si el conjunto $I_{v\rightarrow}$ es un subconjunto del conjunto $I_{x\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{v\rightarrow}$.

En la Realización 2, se describirá un esquema de cifrado mediante predicados de productos internos (Tipo 2) en el cual el texto cifrado ct_x se puede descifrar con la clave de descifrado sk_v si el conjunto $I_{x\rightarrow}$ es un subconjunto del conjunto $I_{v\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{x\rightarrow}$.

En la Realización 2, se describirán principalmente diferencias con respecto a la Realización 1.

La configuración del sistema 10 de procesado criptográfico según la Realización 2 es la misma que la configuración del sistema 10 de procesado criptográfico según la Realización 1 ilustrado en la Fig. 1. Las configuraciones del dispositivo 100 de generación de claves, del dispositivo 200 de cifrado, y del dispositivo 300 de descifrado según la Realización 2, son respectivamente iguales que las configuraciones del dispositivo 100 de generación de claves, del dispositivo 200 de cifrado, y del dispositivo 300 de descifrado según la Realización 1 ilustrada en la Fig. 2 a la Fig. 4.

La Fig. 9 es un diagrama de flujo que ilustra el proceso de un algoritmo *KeyGen* según la Realización 2. La Fig. 10 es un diagrama de flujo que ilustra el proceso de un algoritmo *Enc* según la Realización 2. La Fig. 11 es un diagrama de flujo que ilustra el proceso de un algoritmo *Dec* según la Realización 2.

Un algoritmo *Setup* de acuerdo con la Realización 2 es igual al algoritmo *Setup* según la Realización 1.

En referencia a la Fig. 9, se describirá el proceso del algoritmo *KeyGen*.

El proceso de S501 es igual al proceso de S201 ilustrado en la Fig. 6.

(S502: etapa de generación de números aleatorios)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera números aleatorios, tal como se indica en la Fórmula 140.

[Fórmula 140]

$$\vec{\eta}_0 := (\eta_{0,1},...,\eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$$

$$\delta, \tilde{\delta} \xleftarrow{U} \mathbb{F}_q,$$

$$\mu_t \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad (t, v_t) \in \vec{v},$$

$$\vec{\eta}_t := (\eta_{t,1},...,\eta_{t,w}) \xleftarrow{U} \mathbb{F}_q^{w} \quad \text{para} \quad (t, v_t) \in \vec{v}$$

(S503: etapa de generación de elementos de clave)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera un elemento k^*_0 de una clave de descifrado sk_v, tal como se indica en la Fórmula 141.

[Fórmula 141]

$$\boldsymbol{k}_0^* := (\tilde{\delta}, \overbrace{0^{u_0}}^{u_0}, 1, \overbrace{\vec{\eta}_0}^{w_0}, \overbrace{0^{z_0}}^{z_0},)_{\mathbb{B}_0^*}$$

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera también un elemento k^*_t de la clave de descifrado sk_v para el índice t incluido en el conjunto $I_{v\rightarrow}$, tal como se indica en la Fórmula 142.

[Fórmula 142]

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t, -1,), \delta v_t, \tilde{\delta}}^{4}, \overbrace{0^u}^{u}, \overbrace{\vec{\eta}_t}^{w}, \overbrace{0^z}^{z})_{\mathbb{B}^*}$$

(S504: etapa de distribución de claves)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 150 de distribución de claves distribuye en secreto la clave de descifrado sk_v que tiene, como elementos, k^*_0 y k^*_t generadas en S503, al dispositivo 300 de descifrado. Naturalmente, la clave de descifrado sk_v se puede distribuir al dispositivo 300 de descifrado mediante otro método.

Resumiendo, en S501 a S503, el dispositivo 100 de generación de claves genera la clave de descifrado sk_v ejecutando el algoritmo *KeyGen* indicado en la Fórmula 143. En S504, el dispositivo 100 de generación de claves distribuye la clave de descifrado generada sk_v al dispositivo 300 de descifrado.

[Fórmula 143]

$$\text{KeyGen(pk, sk, } \vec{v} := \{(t, v_t) \mid t \in I_{\vec{v}}\}):$$

$$\delta, \tilde{\delta} \xleftarrow{U} \mathbb{F}_q, \ \vec{\eta}_0 := (\eta_{0,1},...,\eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$$

$$\boldsymbol{k}_0^* := (\tilde{\delta}, \overbrace{0^{u_0}}^{u_0}, 1, \overbrace{\vec{\eta}_0}^{w_0}, \overbrace{0^{z_0}}^{z_0},)_{\mathbb{B}_0^*},$$

para $(t, v_t) \in \vec{v}$,

$$\mu_t \xleftarrow{U} \mathbb{F}_q,$$

$$\vec{\eta}_t := (\eta_{t,1},...,\eta_{t,w}) \xleftarrow{U} \mathbb{F}_q^{w},$$

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t, -1,), \delta v_t, \tilde{\delta}}^{4}, \overbrace{0^u}^{u}, \overbrace{\vec{\eta}_t}^{w}, \overbrace{0^z}^{z})_{\mathbb{B}^*},$$

devolver $\mathsf{sk}_v := (k_0^*, \{k_t^*\}_{(t,v_t)\in\vec{v}})$.

En referencia a la Fig. 10, se describirá el proceso del algoritmo *Enc*.

El proceso de S601 y S602 es igual al proceso de S301 y S302 ilustrado en la Fig. 7.

(S603: etapa de generación de información secreta)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera información secreta f_t y f_0, tal como se indica en la Fórmula 144.

[Fórmula 144]

$$f_t \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad (t, x_t) \in \vec{x},$$

$$f_0 := \sum_{(t,v_t)\in\vec{x}} f_t$$

(S604: etapa de generación de números aleatorios)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera números aleatorios, tal como se indica en la Fórmula 145.

[Fórmula 145]

$$\omega, \zeta \xleftarrow{\mathsf{U}} \mathbb{F}_q,$$

$$\vec{\varphi}_0 := (\varphi_{0,1}, ..., \varphi_{0,z_0}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{z_0},$$

$$\sigma_t \xleftarrow{\mathsf{U}} \mathbb{F}_q \quad \text{para} \quad (t, x_t) \in \vec{x},$$

$$\vec{\varphi}_t := (\varphi_{t,1}, ..., \varphi_{t,z}) \xleftarrow{\mathsf{U}} \mathbb{F}_q^{z} \quad \text{para} \quad (t, x_t) \in \vec{x}$$

(S605: etapa de generación de elementos cifrados)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera un elemento c_0 de un texto cifrado ct_x, tal como se indica en la Fórmula 146.

[Fórmula 146]

$$c_0 := (-f_0, \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0}$$

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera también un elemento c_t del texto cifrado ct_x para el índice t incluido en el conjunto $I_{x\rightarrow}$, tal como se indica en la Fórmula 147.

[Fórmula 147]

$$c_t = (\overbrace{\sigma_t(1, t), \omega x_t, f_t}^{4}, \overbrace{0^u}^{u}, \overbrace{0^w}^{w}, \overbrace{\vec{\varphi}_t}^{z})_{\mathbb{B}}$$

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera también un elemento c_T del texto cifrado ct_x, tal como se indica en la Fórmula 148.

[Fórmula 148]

$$c_T := g_T^{\zeta} m$$

(S606: etapa de transmisión de datos)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 240 de transmisión de datos transmite el texto cifrado ct_x que tiene, como elementos, c_0, c_t y c_T generados en S605, al dispositivo 300 de descifrado. Naturalmente, el texto cifrado ct_x se puede transmitir al dispositivo 300 de descifrado mediante otro método.

Resumiendo, en S601 a S605, el dispositivo 200 de descifrado genera el texto cifrado ct_x ejecutando el algoritmo *Enc* indicado en la Fórmula 149. En S606, el dispositivo 200 de cifrado transmite el texto cifrado generado ct_x al dispositivo 300 de descifrado.

[Fórmula 149]

$$\text{Enc}(\text{ pk}, m, \vec{x} := \{(t, x_t) \mid t \in I_x^-\}):$$

$$f_t, \omega, \zeta \xleftarrow{\cup} \mathbb{F}_q \text{ para } (t, x_t) \in \vec{x},$$

$$\vec{\varphi}_0 := (\varphi_{0,1}, \ldots, \varphi_{0,z_0}) \xleftarrow{\cup} \mathbb{F}_q^{z_0}, f_0 := \sum_{(t,v_t)\in \vec{x}} f_t,$$

$$c_0 := (-f_0, \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0},$$

para $(t, x_t) \in \vec{x}$,

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

$$c_t = (\overbrace{\sigma_t(1, t), \omega x_t, f_t}^{4}, \overbrace{0^u}^{u}, \overbrace{0^w}^{w}, \overbrace{\vec{\varphi}_t}^{z})_{\mathbb{B}},$$

$$c_T := g_T^{\zeta} m$$

devolver $\text{ct}_x := (c_0, \{c_t\}_{(t,x_t)\in\vec{x}}, c_T)$.

En referencia a la Fig. 11, se describirá el proceso del algoritmo *Dec*.

El proceso de S701 y S702 es igual al proceso de S401 y S402 ilustrado en la Fig. 8. El proceso de S704 es el mismo que el proceso de S404 ilustrado en la Fig. 8.

(S703: etapa de operaciones de emparejamiento)

Usando el dispositivo de procesado, la unidad 340 de operaciones de emparejamiento calcula la Fórmula 150, y genera así una clave de sesión $K = g_T^{\zeta}$.

[Fórmula 150]

$$K := e(c_0, k_0^*) \prod_{t\in I_x^-} e(c_t, k_t^*)$$

Si se cumple la Fórmula 151, la clave $K = g_T^{\zeta}$ puede obtenerse calculando la Fórmula 150, según se indica en la Fórmula 152.

[Fórmula 151]

$$I_x^- \subset I_v^- \quad \text{y}$$

$$\sum_{t\in I_x^-} v_t \cdot x_t = 0$$

[Fórmula 152]

$$K := e(c_0, k_0^*) \prod_{t \in I_{\vec{x}}} e(c_t, k_t^*)$$
$$= g_T^{-\tilde{\delta} f_0 + \varsigma} \cdot \prod_{t \in I_{\vec{x}}} g_T^{\omega \delta v_t x_t + \tilde{\delta} f_t}$$
$$= g_T^{-\tilde{\delta} f_0 + \varsigma} \cdot g_T^{\omega \delta (\sum_{(t,x_t) \in I_{\vec{x}}} v_t x_t) + \tilde{\delta} (\sum_{(t,x_t) \in I_{\vec{x}}} f_t)}$$
$$= g_T^{-\tilde{\delta} f_0 + \varsigma + \tilde{\delta} f_0}$$
$$= g_T^{\varsigma}$$

Es decir, si el conjunto $I_{\vec{x}}$ es un subconjunto del conjunto $I_{\vec{v}}$, y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{\vec{x}}$, la clave $K=g_T^{\zeta}$ puede obtenerse calculando la Fórmula 150.

Resumiendo, en S701 a S704, el dispositivo 300 de descifrado genera el mensaje m' (=m) ejecutando el algoritmo *Dec* en la Fórmula 153.

[Fórmula 153]

$\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_{\vec{v}} := (k_0^*, \{k_t^*\}_{(t,v_t) \in \vec{v}}),\ \mathsf{ct}_{\vec{x}} := (c_0, \{c_t\}_{(t,x_t) \in \vec{x}}, c_T))$:

si $I_{\vec{x}} \subset I_{\vec{v}}$ y $\sum_{t \in I_{\vec{x}}} v_t x_t = 0,$

$$K := e(c_0, k_0^*) \prod_{t \in I_{\vec{x}}} e(c_t, k_t^*),$$

devolver $m' := c_T / K.$

Tal como se ha descrito anteriormente, en el esquema de cifrado por predicados de productos internos según la Realización 2, incluso si las dimensiones del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ no son equivalentes, el texto cifrado $\mathsf{ct}_{\vec{x}}$ se puede descifrar con la clave de descifrado $\mathsf{sk}_{\vec{v}}$ si el conjunto $I_{\vec{x}}$ es un subconjunto del conjunto $I_{\vec{v}}$, y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{\vec{x}}$.

En la anterior descripción, las partes de los índices se proporcionan de manera que las bases que se usan para cada categoría de atributo se construyen como bases comunes (base B y base B^*). No obstante, en un caso en el que se permita volver a emitir el parámetro público para añadir una categoría de atributo en una fase posterior, no hay necesidad de proporcionar las partes de los índices si se dispone que las bases que se utilizan para cada categoría de atributo sean respectivamente diferentes.

En este caso, el algoritmo *KeyGen* y el algoritmo *Enc* del esquema de cifrado por predicados de productos internos antes descrito son tal como se indica en la Fórmula 154 y la Fórmula 155. El algoritmo G_{ob} es el mismo que el que se indica en la Fórmula 136, el algoritmo *Setup* es el mismo que el que se indica en la Fórmula 137, y el algoritmo *Dec* es tal como se indica en la Fórmula 153, sin ningún cambio.

[Fórmula 154]

$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \vec{v} := \{(t, v_t) \mid t \in I_{\vec{v}}\})$:

$$\delta, \tilde{\delta} \xleftarrow{U} \mathbb{F}_q,\ \vec{\eta}_0 := (\eta_{0,1}, \ldots, \eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$$
$$k_0^* := (\tilde{\delta}, \overbrace{0^{u_0}}^{u_0}, 1, \overbrace{\vec{\eta}_0}^{w_0}, \overbrace{0^{z_0}}^{z_0},)_{\mathbb{B}_0^*},$$

para $(t, v_t) \in \vec{v}$,

$$\mu_t \xleftarrow{U} \mathbb{F}_q,$$

$$\vec{\eta}_t := (\eta_{t,1}, ..., \eta_{t,w_t}) \xleftarrow{U} \mathbb{F}_q^{w_t},$$

$$\boldsymbol{k}_t^* := (\overbrace{\delta v_t}^{2}, \overbrace{\tilde{\delta}}^{u_t}, \overbrace{0^{u_t}}^{w_t}, \overbrace{\vec{\eta}_t}^{z_t}, \overbrace{0^{z_t}}^{}) \mathbb{B}_t^*,$$

devolver $\mathsf{sk}_v := (\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}_{(t,v_t)\in\vec{v}})$.

[Fórmula 155]

Enc(pk, m, $\vec{x} := \{(t, x_t) \mid t \in I_{\vec{x}}\}$) :

$$f_t, \omega, \zeta \xleftarrow{U} \mathbb{F}_q \text{ para } (t, x_t) \in \vec{x},$$

$$\vec{\varphi}_0 := (\varphi_{0,1}, ..., \varphi_{0,z_0}) \xleftarrow{U} \mathbb{F}_q^{z_0}, f_0 := \sum_{(t,v_t)\in\vec{x}} f_t$$

$$c_0 := (-f_0, \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0}) \mathbb{B}_0,$$

para $(t, x_t) \in \vec{x}$,

$$\sigma_t \xleftarrow{U} \mathbb{F}_q, \vec{\varphi}_t := (\varphi_{t,1}, ..., \varphi_{t,z_t}) \xleftarrow{U} \mathbb{F}_q^{z_t},$$

$$\boldsymbol{c}_t = (\overbrace{\omega x_t, f_t}^{2}, \overbrace{0^{u_t}}^{u_t}, \overbrace{0^{w_t}}^{w_t}, \overbrace{\vec{\varphi}_t}^{z_t}) \mathbb{B}_t,$$

$$\boldsymbol{c}_T := g_T^{\zeta} m$$

devolver $\mathsf{ct}_x := (\boldsymbol{c}_0, \{\boldsymbol{c}_t\}_{(t,x_t)\in\vec{x}}, \boldsymbol{c}_T)$.

Es decir, para cada índice t se usan una base diferente B_t y una base diferente B^*_t.

Realización 3

En la Realización 1, se ha descrito el esquema de cifrado por predicados de productos internos (Tipo 1) en el cual el texto cifrado ct_x se puede descifrar con la clave de descifrado sk_v si el conjunto $I_{v\rightarrow}$ es un subconjunto del conjunto $I_{x\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{v\rightarrow}$.

En la Realización 2, se ha descrito el esquema de cifrado por predicados de productos internos (Tipo 2) en el cual el texto cifrado ct_x se puede descifrar con la clave de descifrado sk_v si el conjunto $I_{x\rightarrow}$ es un subconjunto del conjunto $I_{v\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{x\rightarrow}$.

En la Realización 3, se describirá un esquema de cifrado por predicados de productos internos (Tipo 0) en el cual el texto cifrado ct_x se puede descifrar con la clave de descifrado sk_v si el conjunto $I_{v\rightarrow}$ es igual al conjunto $I_{x\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{v\rightarrow}$ (o el conjunto $I_{x\rightarrow}$).

Obsérvese que "el conjunto $I_{v\rightarrow}$ es igual al conjunto $I_{x\rightarrow}$" significa que el conjunto $I_{v\rightarrow}$ es un subconjunto del conjunto $I_{x\rightarrow}$ y que el conjunto $I_{x\rightarrow}$ es un subconjunto del conjunto $I_{v\rightarrow}$. Así, el esquema de cifrado por predicados de productos internos (Tipo 0) puede considerarse como una combinación del esquema de cifrado por predicados de productos internos (Tipo 1) descrito en la Realización 1 y el esquema de cifrado por predicados de productos internos (Tipo 2) descrito en la Realización 2.

En la Realización 3, se describirán principalmente diferencias con respecto a las Realizaciones 1 y 2.

La configuración del sistema 10 de procesado criptográfico según la Realización 3 es la misma que la configuración del sistema 10 de procesado criptográfico según la Realización 1 ilustrada en la Fig. 1. Las configuraciones del

dispositivo 100 de generación de claves, el dispositivo 200 de cifrado, y el dispositivo 300 de descifrado según la Realización 3 son respectivamente las mismas que las configuraciones del dispositivo 100 de generación de claves, el dispositivo 200 de cifrado, y el dispositivo 300 de descifrado según la Realización 1 ilustrada en la Fig. 2 a la Fig. 4.

La Fig. 12 es un diagrama de flujo que ilustra el proceso de un algoritmo *Setup* según la Realización 3. La Fig. 13 es un diagrama de flujo que ilustra el proceso de un algoritmo *KeyGen* según la Realización 3. La Fig. 14 es un diagrama de flujo que ilustra el proceso de un algoritmo *Enc* según la Realización 3. La Fig. 15 es un diagrama de flujo que ilustra el proceso de un algoritmo *Dec* según la Realización 3.

Se describirá el algoritmo *Setup* en referencia a la Fig. 12.

El proceso de S801 es el mismo que el proceso de S101 ilustrado en la Fig. 5, excepto que $N_0 = 2 + u_0 + 1 + w_0 + z_0$ y $N_1 = 5 + u + w + z$.

Resumiendo, en S801, la unidad 110 de generación de claves maestras genera param, la base B_0 y la base B^*_0, y la base B_1 (base B) y la base B^*_1 (base B^*) ejecutando el algoritmo G_{ob} indicado en la Fórmula 156.

[Fórmula 156]

$$\mathcal{G}_{\mathrm{ob}}(1^\lambda):$$

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

$$N_0 := 2+u_0+1+w_0+z_0,\ N_1 := 5+u+w+z,$$

para t = 0,1,

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

$$X_t := (\chi_{t,i,j})_{i,j=1,..,N_t} \xleftarrow{U} GL(N_t,\mathbb{F}_q),$$

$$X_t^* := (\vartheta_{t,i,j})_{i,j=1,..,N_t} := \psi\cdot(X_t^T)^{-1},\ \text{en lo sucesivo, } \vec{\chi}_{t,i}$$

y $\vec{\vartheta}_{t,i}$ indican las filas i-ésimas de X_t y X^*_t para i = 1, ..., N_t, respectivamente,

$$\boldsymbol{b}_{t,i} := (\vec{\chi}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j}\boldsymbol{a}_{t,j} \quad \text{para} \quad i=1,...,N_t,\ \mathbb{B}_t := (\boldsymbol{b}_{t,1},...,\boldsymbol{b}_{t,N_t}),$$

$$\boldsymbol{b}^*_{t,i} := (\vec{\vartheta}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \vartheta_{t,i,j}\boldsymbol{a}_{t,j} \quad \text{para} \quad i=1,...,N_t,\ \mathbb{B}^*_t := (\boldsymbol{b}^*_{t,1},...,\boldsymbol{b}^*_{t,N_t}),$$

$$g_T := e(g,g)^\psi,\ \mathrm{param} := (\{\mathrm{param}_{\mathbb{V}_t}\}_{t=0,1}, g_T),$$

devolver $(\mathrm{param},\mathbb{B}_t,\mathbb{B}^*_t)$.

(S802: etapa de generación de parámetros públicos)

Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras genera una sub-base $B^\wedge_0$ de la base B_0 y una sub-base $B^\wedge$ de la base B, tal como se indica en la Fórmula 157, habiéndose generado las bases B_0 y B en S801.

[Fórmula 157]

$$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1},\ \boldsymbol{b}_{0,2},\ \boldsymbol{b}_{0,2+u_0+1},\ \boldsymbol{b}_{0,2+u_0+1+w_0+1},...,\boldsymbol{b}_{0,2+u_0+1+w_0+z_0}),$$

$$\hat{\mathbb{B}} := (\boldsymbol{b}_1,..,\boldsymbol{b}_5,\ \boldsymbol{b}_{5+u+w+1},..,\ \boldsymbol{b}_{5+u+w+z})$$

La unidad 110 de generación de claves maestras genera un parámetro público pk agrupando la sub-base $B^\wedge_0$ y la sub-base $B^\wedge$ generadas, el parámetro de seguridad λ (1^λ) introducido en S801, y el param generado en S801.

(S803: etapa de generación de claves maestras)

Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras genera una sub-base $B^{\wedge *}_0$ de la base B^*_0 y una sub-base $B^{\wedge *}$ de la base B^*, tal como se indica en la Fórmula 158, habiéndose generado las bases B^*_0 y B^* en S801.

[Fórmula 158]

$\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,2}^*, \boldsymbol{b}_{0,2+u_0+1}^*, \boldsymbol{b}_{0,2+u_0+1+1,\ldots,}^*\boldsymbol{b}_{0,2+u_0+1+w_0}^*)$,

$\hat{\mathbb{B}}^* := (\boldsymbol{b}_1^*,..,\boldsymbol{b}_5^*, \boldsymbol{b}_{5+u+1}^*,.., \boldsymbol{b}_{5+u+w}^*)$

La unidad 110 de generación de claves maestras genera una clave maestra sk que está constituida por la sub-base $B^{\wedge *}_0$ y la sub-base $B^{\wedge *}$ generadas.

El proceso de S804 es el mismo que el proceso de S104 ilustrado en la Fig. 5.

Resumiendo, en S801 a S803, el dispositivo 100 de generación de claves genera el parámetro público pk y la clave maestra sk ejecutando el algoritmo *Setup* indicado en la Fórmula 159. En S804, el dispositivo 100 de generación de claves almacena el parámetro público generado pk y la clave maestra sk en el dispositivo de almacenamiento.

El parámetro público se publica por medio de la red, por ejemplo, y se hace que esté disponible para el dispositivo 200 de cifrado y el dispositivo 300 de descifrado.

[Fórmula 159]

$\mathsf{Setup}(1^\lambda)$:

$(\mathsf{param}, (\mathbb{B}_0, \mathbb{B}_0^*), (\mathbb{B}, \mathbb{B}^*)) \xleftarrow{R} \mathcal{G}_{\mathsf{ob}}(1^\lambda)$,

$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,2}, \boldsymbol{b}_{0,2+u_0+1}, \boldsymbol{b}_{0,2+u_0+1+w_0+1},\ldots,\boldsymbol{b}_{0,2+u_0+1+w_0+z_0})$,

$\hat{\mathbb{B}} := (\boldsymbol{b}_1,..,\boldsymbol{b}_5, \boldsymbol{b}_{5+u+w+1},.., \boldsymbol{b}_{5+u+w+z})$,

$\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,2}^*, \boldsymbol{b}_{0,2+u_0+1}^*, \boldsymbol{b}_{0,2+u_0+1+1,\ldots,}^*\boldsymbol{b}_{0,2+u_0+1+w_0}^*)$,

$\hat{\mathbb{B}}^* := (\boldsymbol{b}_1^*,..,\boldsymbol{b}_5^*, \boldsymbol{b}_{5+u+1}^*,.., \boldsymbol{b}_{5+u+w}^*)$,

devolver $\mathsf{pk} := (1^\lambda, \mathsf{param}, \hat{\mathbb{B}}_0, \hat{\mathbb{B}})$, $\mathsf{sk} := (\hat{\mathbb{B}}_0^*, \hat{\mathbb{B}}^*)$.

En referencia a la Fig. 13 se describirá el algoritmo *KeyGen*.

(S901: etapa de entrada de información)

Usando el dispositivo de entrada, la unidad 130 de introducción de información toma como entrada un vector de predicados $\vec{v} := (v_1, \ldots, v_n)$. Como información de predicado v_t, se fija, por ejemplo, información de atributo de un usuario de una clave de descifrado sk_v.

(S902: etapa de generación de información secreta)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera información secreta s_t y s_0, tal como se indica en la Fórmula 160.

[Fórmula 160]

$s_t \xleftarrow{U} \mathbb{F}_q$ para $t = 1, \ldots, n$,

$s_0 := \sum_{t=1}^{n} s_t$

(S903: etapa de generación de números aleatorios)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera números aleatorios, tal como se indica en la Fórmula 161.

[Fórmula 161]

$\vec{\eta}_0 := (\eta_{0,1},\ldots,\eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0}$,

$\delta, \tilde{\delta} \xleftarrow{U} \mathbb{F}_q$,

$\mu_t \xleftarrow{U} \mathbb{F}_q$ para $t = 1,\ldots, n$

$\vec{\eta}_t := (\eta_{t,1},\ldots,\eta_{t,w}) \xleftarrow{U} \mathbb{F}_q^{w}$ para $t = 1,\ldots, n$

(S904: etapa de generación de elementos de clave)

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera un elemento k^*_0 de la clave de descifrado sk_v, tal como se indica en la Fórmula 162.

[Fórmula 162]

$$\boldsymbol{k}_0^* := (-s_0,\ \tilde{\delta},\ \overbrace{0^{u_0}}^{u_0},\ 1,\ \overbrace{\vec{\eta}_0}^{w_0},\ \overbrace{0^{z_0}}^{z_0},\ )_{\mathbb{B}_0^*}$$

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado también genera un elemento k^*_t de la clave de descifrado sk_v para cada índice t =1,..., n, tal como se indica en la Fórmula 163.

[Fórmula 163]

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t,\ -1,),\ \delta v_t,\ s_t,\ \tilde{\delta}}^{5}\ \overbrace{0^{u}}^{u},\ \overbrace{\vec{\eta}_t}^{w},\ \overbrace{0^{z}}^{z},\ )_{\mathbb{B}^*}$$

(S905: etapa de distribución de claves)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 150 de distribución de claves distribuye la clave de descifrado sk_v que tiene, como elementos, k^*_0 y k^*_t generadas en S904, al dispositivo 300 de descifrado en secreto. Naturalmente, la clave de descifrado sk_v se puede distribuir al dispositivo 300 de descifrado mediante otro método.

Resumiendo, en las etapas S901 a S904, el dispositivo 100 de generación de claves genera la clave de descifrado sk_v ejecutando el algoritmo *KeyGen* indicado en la Fórmula 164. En S905, el dispositivo 100 de generación de claves distribuye la clave de descifrado generada sk_v al dispositivo 300 de descifrado.

[Fórmula 164]

KeyGen(pk, sk, $\vec{v} := (v_1,...,v_n)$):

$$\delta, \tilde{\delta}, s_t \xleftarrow{U} \mathbb{F}_q \quad \text{para } t = 1,..., n,$$

$$\vec{\eta}_0 := (\eta_{0,1},...,\eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$$

$$s_0 := \sum_{t=1}^{n} s_t,$$

$$\boldsymbol{k}_0^* := (-s_0,\ \tilde{\delta},\ \overbrace{0^{u_0}}^{u_0},\ 1,\ \overbrace{\vec{\eta}_0}^{w_0},\ \overbrace{0^{z_0}}^{z_0},\ )_{\mathbb{B}_0^*},$$

para $t = 1,..., n$,

$$\mu_t \xleftarrow{U} \mathbb{F}_q,$$

$$\vec{\eta}_t := (\eta_{t,1},...,\eta_{t,w}) \xleftarrow{U} \mathbb{F}_q^{w},$$

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t,\ -1,),\ \delta v_t,\ s_t,\ \tilde{\delta}}^{5}\ \overbrace{0^{u}}^{u},\ \overbrace{\vec{\eta}_t}^{w},\ \overbrace{0^{z}}^{z},\ )_{\mathbb{B}^*},$$

devolver $\mathrm{sk}_v := (\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}_{t=1,...,n})$.

En referencia a la Fig. 14, se describirá el proceso del algoritmo *Enc*.

El proceso de S1001 es el mismo que el proceso de S301 ilustrado en la Fig. 7.

(S1002: etapa de introducción de información)

Usando el dispositivo de entrada, la unidad 220 de introducción de información toma como entrada un mensaje m a transmitir hacia el dispositivo 300 de descifrado. Usando el dispositivo de entrada, la unidad 220 de introducción de información también toma como entrada un vector de atributos $\vec{x} := (x_1, ..., x_{n'})$. En el vector de atributos $\vec{x}$, se fija, por ejemplo, información de atributo de un usuario con capacidad de descifrado.

(S1003: etapa de generación de información secreta)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera información secreta f_t y f_0, tal como se indica en la Fórmula 165.

[Fórmula 165]

$$f_t \xleftarrow{U} \mathbb{F}_q \quad \text{para } t = 1, ..., n',$$

$$f_0 := \sum_{t=1}^{n'} f_t$$

(S1004: etapa de generación de números aleatorios)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera números aleatorios, tal como se indica en la Fórmula 166.

[Fórmula 166]

$$\omega, \tilde{\omega}, \zeta \xleftarrow{U} \mathbb{F}_q,$$

$$\vec{\varphi}_0 := (\varphi_{0,1}, ..., \varphi_{0,z_0}) \xleftarrow{U} \mathbb{F}_q^{z_0},$$

$$\sigma_t \xleftarrow{U} \mathbb{F}_q \quad \text{para } t = 1, ..., n',$$

$$\vec{\varphi}_t := (\varphi_{t,1}, ..., \varphi_{t,z}) \xleftarrow{U} \mathbb{F}_q^{z} \quad \text{para } t = 1, ..., n'$$

(S1005: etapa de generación de elementos cifrados)

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera un elemento c_0 de un texto cifrado ct_x, tal como se indica en la Fórmula 167.

[Fórmula 167]

$$c_0 := (\tilde{\omega}, \ -f_0 \overbrace{0^{u_0}}^{u_0}, \ \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0}$$

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera también un elemento c_t del texto cifrado ct_x para cada índice t = 1,..., n', tal como se indica en la Fórmula 168.

[Fórmula 168]

$$c_t = (\overbrace{\sigma_t(1, t), \omega x_t, \tilde{\omega}, f_t}^{5} \overbrace{0^u}^{u}, \overbrace{0^w}^{w}, \overbrace{\vec{\varphi}_t}^{z})_{\mathbb{B}}$$

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera un elemento c_T del texto cifrado ct_x, tal como se indica en la Fórmula 169.

[Fórmula 169]

$$c_T := g_T^{\zeta} m$$

(S1006: etapa de transmisión de datos)

Usando el dispositivo de comunicaciones y por medio de la red, por ejemplo, la unidad 240 de transmisión de datos transmite el texto cifrado ct_x que tiene, como elementos, c_0, c_t y c_T generados en S1005 al dispositivo 300 de descifrado. Naturalmente, el texto cifrado ct_x se puede transmitir al dispositivo 300 de descifrado a través de otro método.

Resumiendo, en las etapas S1001 a S1005, el dispositivo 200 de cifrado genera el texto cifrado ct_x ejecutando el algoritmo *Enc* indicado en la Fórmula 170. En S1006, el dispositivo 200 de cifrado transmite el texto cifrado ct_x al

dispositivo 300 de descifrado.

[Fórmula 170]

Enc(pk, m, $\vec{x} := (x_1,...,x_{n'})$) :

$f_t, \omega, \tilde{\omega}, \zeta \xleftarrow{U} \mathbb{F}_q$ para t = 1, ..., n',

$\vec{\varphi}_0 := (\varphi_{0,1},...,\varphi_{0,z_0}) \xleftarrow{U} \mathbb{F}_q^{z_0}, f_0 := \sum_{t=1}^{n'} f_t,$

$c_0 := (\tilde{\omega}, -f_0 \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0},$

para t = 1, ..., n',

$\sigma_t \xleftarrow{U} \mathbb{F}_q, \vec{\varphi}_t := (\varphi_{t,1},...,\varphi_{t,z}) \xleftarrow{U} \mathbb{F}_q^z,$

$c_t = (\overbrace{\sigma_t(1, t), \omega x_t, \tilde{\omega}, f_t}^{5} \overbrace{0^u}^{u}, \overbrace{0^w}^{w}, \overbrace{\vec{\varphi}_t}^{z})_{\mathbb{B}},$

$c_T := g_T^{\zeta} m$

devolver $\mathrm{ct}_x := (c_0, \{c_t\}_{t=1,...,n'}, c_T).$

En referencia a la Fig. 15, se describirá el proceso del algoritmo *Dec*.

El proceso de S1101 y S1102 es el mismo que el proceso de S401 y S402 ilustrado en la Fig. 8.

(S1103: etapa de operaciones de emparejamiento)

Usando el dispositivo de procesado, la unidad 340 de operaciones de emparejamiento calcula la Fórmula 171, y genera así una clave de sesión $K = g_T^{\zeta}$.

[Fórmula 171]

$$K := e(c_0, k_0^*) \prod_{t=1}^{n} e(c_t, k_t^*)$$

Si se cumple la Fórmula 172, la clave $K = g_T^{\zeta}$ puede obtenerse calculando la Fórmula 171, tal como se indica en la Fórmula 173.

[Fórmula 172]

$n = n'$ y

$\vec{v} \cdot \vec{x} = 0$

[Fórmula 173]

$$\begin{aligned} K &:= e(c_0, k_0^*) \prod_{t=1}^{n} e(c_t, k_t^*) \\ &= g_T^{-\tilde{\omega} s_0 - \tilde{\delta} f_0 + \zeta} \cdot \prod_{t=1}^{n} g_T^{\delta \omega v_t x_t + \tilde{\omega} s_t + \tilde{\delta} f_t} \\ &= g_T^{-\tilde{\omega} s_0 - \tilde{\delta} f_0 + \zeta} \cdot g_T^{\delta \omega (\sum_{t=1}^{n} v_t x_t) + \tilde{\omega} (\sum_{t=1}^{n} s_t) + \tilde{\delta} (\sum_{t=1}^{n} f_t)} \\ &= g_T^{-\tilde{\omega} s_0 - \tilde{\delta} f_0 + \zeta + \tilde{\omega} s_0 + \tilde{\delta} f_0} \\ &= g_T^{\zeta} \end{aligned}$$

Es decir, si el conjunto $I_{x\rightarrow}$ es igual al conjunto $I_{v\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{v\rightarrow}$ (o el conjunto $I_{x\rightarrow}$), la clave $K=g_T^{\zeta}$ puede obtenerse calculando la Fórmula 171.

El proceso de S1104 es el mismo que el proceso de S404 ilustrado en la Fig. 8.

Resumiendo, en las etapas S1101 a S1104, el dispositivo 300 de descifrado ejecuta el algoritmo *Dec* en la Fórmula 174, y genera así el mensaje m' (= m).

[Fórmula 174]

$$\mathsf{Dec}(\mathsf{pk}, \mathsf{sk}_v := (\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}_{t=1,\ldots,n}),\ \mathsf{ct}_x := (\boldsymbol{c}_0, \{\boldsymbol{c}_t\}_{t=1,\ldots,n'}, c_T)):$$

si $n = n'$ y $\vec{v}_t \vec{x}_t = 0$,

$$K := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \prod_{t=1}^{n} e(\boldsymbol{c}_t, \boldsymbol{k}_t^*),$$

devolver $m' := c_T / K$.

Tal como se ha descrito anteriormente, en el esquema de cifrado por predicados de productos internos según la Realización 3, el texto cifrado ct_x se puede descifrar con la clave de descifrado sk_v si el conjunto $I_{x\rightarrow}$ es igual al conjunto $I_{v\rightarrow}$ y si la suma de $v_t x_t$ es 0 para el índice t incluido en el conjunto $I_{v\rightarrow}$ (el conjunto $I_{x\rightarrow}$).

En la anterior descripción, las partes de índice se proporcionan de manera que las bases que se usan para cada categoría de atributo se construyen como bases comunes (base B y base B^*). No obstante, en un caso en el que se permita volver a emitir el parámetro público para añadir una categoría de atributo en una fase posterior, no hay necesidad de proporcionar las partes de índice si se dispone que las bases que se usan para cada categoría de atributo sean respectivamente diferentes.

En este caso, el algoritmo G_{ob}, el algoritmo *Setup*, el algoritmo *KeyGen* y el algoritmo *Enc* del esquema de cifrado mediante predicados de productos internos descrito anteriormente son tal como se indica en la Fórmula 175 a la Fórmula 178. El algoritmo *Dec* es tal como se indica en la Fórmula 174, sin ningún cambio.

[Fórmula 175]

$\mathcal{G}_{\mathrm{ob}}(1^\lambda)$:

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

$$N_0 := 2 + u_0 + 1 + w_0 + z_0,\ N_t := 3 + u_t + w_t + z_t \quad \text{para } t = 1, \ldots, d,$$

para $t = 0, \ldots, d$,

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

$$X_t := (\chi_{t,i,j})_{i,j=1,\ldots,N_t} \xleftarrow{\mathsf{U}} GL(N_t, \mathbb{F}_q),$$

$$X_t^* := (\vartheta_{t,i,j})_{i,j=1,\ldots,N_t} := \psi \cdot (X_t^T)^{-1},\quad \text{en lo sucesivo, } \vec{\chi}_{t,i}$$

y $\vec{\vartheta}_{t,i}$ indican las filas i-ésimas de X_t y X^*_t para $i = 1, \ldots, N_t$, respectivamente,

$$\boldsymbol{b}_{t,i} := (\vec{\chi}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \chi_{t,i,j} \boldsymbol{a}_{t,j} \quad \text{para } i = 1, \ldots, N_t,\quad \mathbb{B}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,N_t}),$$

$$\boldsymbol{b}_{t,i}^* := (\vec{\vartheta}_{t,i})_{\mathbb{A}_t} = \sum_{j=1}^{N_t} \vartheta_{t,i,j} \boldsymbol{a}_{t,j} \quad \text{para } i = 1, \ldots, N_t,\quad \mathbb{B}_t^* := (\boldsymbol{b}_{t,1}^*, \ldots, \boldsymbol{b}_{t,N_t}^*),$$

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

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

[Fórmula 176]

$$\mathsf{Setup}(1^{\lambda}):$$

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

$$\hat{\mathbb{B}}_0 := (\boldsymbol{b}_{0,1}, \boldsymbol{b}_{0,2}, \boldsymbol{b}_{0,2+u_0+1}, \boldsymbol{b}_{0,2+u_0+1+w_0+1}, \ldots, \boldsymbol{b}_{0,2+u_0+1+w_0+z_0}),$$

$$\hat{\mathbb{B}}_t := (\boldsymbol{b}_{t,1}, \ldots, \boldsymbol{b}_{t,3}, \boldsymbol{b}_{t,3+u_t+w_t+1}, \ldots, \boldsymbol{b}_{t,3+u_t+w_t+z_t}),$$

$$\hat{\mathbb{B}}_0^* := (\boldsymbol{b}_{0,1}^*, \boldsymbol{b}_{0,2}^*, \boldsymbol{b}_{0,2+u_0+1}^*, \boldsymbol{b}_{0,2+u_0+1+1}^*, \ldots, \boldsymbol{b}_{0,2+u_0+1+w_0}^*),$$

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

devolver $\mathsf{pk} := (1^{\lambda}, \mathsf{param}, \hat{\mathbb{B}}_0, \{\hat{\mathbb{B}}_t\}_{t=1,\ldots,d})$, $\mathsf{sk} := (\hat{\mathbb{B}}_0^*, \{\hat{\mathbb{B}}_t^*\}_{t=1,\ldots,d})$.

[Fórmula 177]

$$\mathsf{KeyGen}(\mathsf{pk}, \mathsf{sk}, \vec{v} := (v_1, \ldots, v_n)):$$

$$\delta, \tilde{\delta}, s_t \xleftarrow{U} \mathbb{F}_q \qquad \text{para } t = 1, \ldots, n,$$

$$\vec{\eta}_0 := (\eta_{0,1}, \ldots, \eta_{0,w_0}) \xleftarrow{U} \mathbb{F}_q^{w_0},$$

$$s_0 := \sum_{t=1}^{n} s_t,$$

$$\boldsymbol{k}_0^* := (-s_0, \tilde{\delta}, \overbrace{0^{u_0}}^{u_0}, 1, \overbrace{\vec{\eta}_0}^{w_0}, \overbrace{0^{z_0}}^{z_0})_{\mathbb{B}_0^*},$$

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

$$\mu_t \xleftarrow{U} \mathbb{F}_q,$$

$$\vec{\eta}_t := (\eta_{t,1}, \ldots, \eta_{t,w_t}) \xleftarrow{U} \mathbb{F}_q^{w_t},$$

$$\boldsymbol{k}_t^* := (\overbrace{\delta v_t, s_t, \tilde{\delta}}^{3}, \overbrace{0^{u_t}}^{u_t}, \overbrace{\vec{\eta}_t}^{w_t}, \overbrace{0^{z_t}}^{z_t})_{\mathbb{B}_t^*},$$

devolver $\mathsf{sk}_v := (\boldsymbol{k}_0^*, \{\boldsymbol{k}_t^*\}_{t=1,\ldots,n})$.

[Fórmula 178]

$$\mathsf{Enc}(\mathsf{pk}, m, \vec{x} := (x_1, \ldots, x_{n'})):$$

$$f_t, \omega, \tilde{\omega}, \zeta \xleftarrow{U} \mathbb{F}_q \qquad \text{para } t = 1, \ldots, n',$$

$$\vec{\varphi}_0 := (\varphi_{0,1}, \ldots, \varphi_{0,z_0}) \xleftarrow{U} \mathbb{F}_q^{z_0}, f_0 := \sum_{t=1}^{n'} f_t$$

$$\boldsymbol{c}_0 := (\tilde{\omega}, -f_0 \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbb{B}_0},$$

para $t = 1, ..., n'$,

$$\sigma_t \xleftarrow{U} \mathbb{F}_q,\ \vec{\varphi}_t := (\varphi_{t,1}, ..., \varphi_{t,z_t}) \xleftarrow{U} \mathbb{F}_q^{z_t},$$

$$\boldsymbol{c}_t = (\overbrace{\omega x_t,\ \tilde{\omega},\ f_t}^{3}\ \overbrace{0^{u_t},}^{u_t}\ \overbrace{0^{w_t},}^{w_t}\ \overbrace{\vec{\varphi}_t}^{z_t})_{\mathbb{B}_t},$$

$$\boldsymbol{c}_T := g_T^{\zeta} m$$

devolver $\mathrm{ct}_x := (\boldsymbol{c}_0, \{\boldsymbol{c}_t\}_{t=1,...,n'}, \boldsymbol{c}_T).$

Es decir, se usan una base B_t diferente y una base B^*_t diferente para cada índice t.

En la anterior descripción, se especifica que u_0, w_0, z_0, u, w y z son, cada uno de ellos, un entero de 0 ó mayor. Por ejemplo, se puede especificar que $u_0 = 2$, $w_0 = 2$, $z_0 = 2$, u = 11, w = 3, y z = 3.

Realización 4

Se describirán un esquema de cifrado funcional y un esquema de firma basado en atributos que tienen, cada uno de ellos, el esquema de cifrado por predicados de productos internos descrito en una de las realizaciones anteriores como estructura inferior.

La referencia bibliográfica 31, que no es documento de patente, describe un esquema de cifrado funcional.

En el esquema de cifrado funcional descrito en la referencia bibliográfica 31, no documento de patente, se calcula el producto interno del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ para cada índice t. A continuación, si el producto interno del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ es 0 para todo índice t dado, puede descifrarse un texto cifrado con una clave de descifrado.

El esquema de cifrado por predicados de productos internos descrito en una de las realizaciones anteriores se aplica al cálculo del producto interno del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ para cada índice t. Con esta disposición, se construye el esquema de cifrado funcional que presenta el esquema de cifrado por predicados de productos internos descrito en una de las realizaciones anteriores como estructura inferior.

En el esquema de cifrado funcional descrito en la referencia bibliográfica 31, no documento de patente, se requiere que las dimensiones del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ sean equivalentes para cada índice t. No obstante, en el esquema de cifrado funcional que presenta el esquema de cifrado por predicados de productos internos descrito en una de las realizaciones anteriores como estructura inferior, no se requiere que las dimensiones del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ sean equivalentes para cada índice t.

En la siguiente descripción, se expondrá un caso en el que el esquema de cifrado por predicados de productos internos descrito en la Realización 1 se aplica al esquema de cifrado funcional con política de claves (esquema KP-FE) descrito en la referencia bibliográfica 31, no documento de patente, como ejemplo.

En la presente se describirán principalmente cambios en el esquema KP-FE que se producen como consecuencia de aplicar el esquema de cifrado por predicados de productos internos descrito en la Realización 1.

En el esquema de cifrado funcional descrito en la referencia bibliográfica 31, no documento de patente, se utiliza una variable ρ(i) para especificar para cada índice t si se permite el descifrado si el producto interno del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ es 0 ó se permite el descifrado si el producto interno del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ es diferente de 0. No obstante, para simplificar la descripción, en la presente se describirá un caso en el que se permite el descifrado si el producto interno del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ es 0 para todo índice t dado.

En la Realización 1, las partes de los índices se proporcionan de manera que las bases que se usan para cada categoría de atributos se construyen como bases comunes (base B y base B^*). No obstante, para simplificar la descripción, en la presente se describirá un caso en el que las partes de los índices no se proporcionan y las bases que se usan para cada categoría de atributo son respectivamente diferentes.

Obsérvese también que, en la referencia bibliográfica 31, no documento de patente, y en la Realización 1, se produce un uso solapado de los alfabetos t y s del índice t y la información secreta s. Así, el índice t de la referencia bibliográfica 31, no documento de patente, se describirá en la presente como índice τ, y la información secreta s de la referencia bibliográfica 31, no documento de patente, se describirá como información secreta σ.

El esquema KP-FE tiene cuatro algoritmos: un algoritmo *Setup*, un algoritmo *KeyGen*, un algoritmo *Enc* y un algoritmo *Dec*.

Se describirá el algoritmo *Setup*.

Usando el dispositivo de procesado, la unidad 110 de generación de claves maestras genera una base $B_{\tau,t}$ y una base $B^*_{\tau,t}$ para cada índice τ,t. Es decir, la unidad 110 de generación de claves maestras genera la base $B_{\tau,t}$ y la base $B^*_{\tau,t}$ para cada índice τ,t, en lugar de generar la base B_t y la base B^*_t para cada índice t.

Resumiendo, el algoritmo *Setup* es tal como se indica en la Fórmula 179. El algoritmo G_{ob} usado en el algoritmo *Setup* es tal como se indica en la Fórmula 180.

[Fórmula 179]

$$\mathsf{Setup}(1^\lambda, \vec{n} := (\Delta; d; n_{1,1}, \ldots, n_{\Delta,d}))$$

$$(\mathsf{param}_{\vec{n}}, \mathbb{B}_0, \mathbb{B}^*_0, \{\mathbb{B}_{\tau,t}, \mathbb{B}^*_{\tau,t}\}_{\tau=1,\ldots,\Delta;t=1,\ldots,d}) \xleftarrow{R} \mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n})$$

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

$$\hat{\mathbb{B}}_{\tau,t} := (b_{\tau,t,1}, \ldots, b_{\tau,t,n_t}, b_{\tau,t,3n_t+1}) \qquad \text{para } = \tau, 1, \ldots, \Delta; t = 1, \ldots, d,$$

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

$$\hat{\mathbb{B}}^*_{\tau,t} := (b^*_{\tau,t,1}, \ldots, b^*_{\tau,t,n_t}, b^*_{\tau,t,2n_t+1}, \ldots, b^*_{\tau,t,3n_t}) \qquad \text{para } = \tau, 1, \ldots, \Delta; t = 1, \ldots, d,$$

$$\mathsf{pk} := (1^\lambda, \mathsf{param}_{\vec{n}}, \hat{\mathbb{B}}_0, \{\hat{\mathbb{B}}_{\tau,t}\}_{\tau=1,\ldots,\Delta;t=1,\ldots,d}),$$

$$\mathsf{sk} := (\hat{\mathbb{B}}^*_0, \{\hat{\mathbb{B}}^*_{\tau,t}\}_{\tau=1,\ldots,\Delta;t=1,\ldots,d}).$$

devolver pk, sk.

[Fórmula 180]

$$\mathcal{G}_{\mathsf{ob}}(1^\lambda, \vec{n}):$$

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

$$N_0 := 5, \quad N_{\tau,t} := 2 + 2n_{\tau,t} + 1 \quad \text{para } = \tau = 1, \ldots, \Delta; t = 1, \ldots, d,$$

$$\mathsf{param}_{\mathbb{V}_0} := (q, \mathbb{V}_0, \mathbb{G}_T, \mathbb{A}_0, e) := \mathcal{G}_{\mathsf{dpvs}}(1^\lambda, N_0, \mathsf{param}_{\mathbb{G}}),$$

$$X_0 := (\chi_{0,i,j})_{i,j} \xleftarrow{U} GL(N_0, \mathbb{F}_q), \quad (\nu_{0,i,j})_{i,j} := \psi \cdot (X_0^{\mathrm{T}})^{-1},$$

$$b_{0,i} := (\chi_{0,i,1}, \ldots, \chi_{0,i,N_{\tau,t}})_{\mathbb{A}_0} = \sum_{j=1}^{N_0} \chi_{0,i,j} a_{0,j}, \quad \mathbb{B}_0 := (b_{0,1}, \ldots, b_{0,N_0}),$$

$$b^*_{0,i} := (\nu_{0,i,1}, \ldots, \nu_{0,i,N_0})_{\mathbb{A}_0} = \sum_{j=1}^{N_0} \nu_{0,i,j} a_{0,j}, \quad \mathbb{B}^*_0 := (b^*_{0,1}, \ldots, b^*_{0,N_0}),$$

Para τ =1, ..., Δ; *t* = 1, ..., *d*,

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

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

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

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

$$g_T := e(g,g)^\psi, \quad \mathsf{param}_{\vec{n}} := (\{\mathsf{param}_{\mathbb{V}_{\tau,t}}\}_{\tau=1,\ldots,\Delta;t=0,\ldots,d}, g_T)$$

devolver $(\mathsf{param}_{\vec{n}}, \{\mathbb{B}_{\tau,t}, \mathbb{B}^*_{\tau,t}\}_{\tau=1,\ldots,\Delta;t=0,\ldots,d})$.

Se describirá el algoritmo *KeyGen*.

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera información secreta σ e información secreta s, tal como se indica en la Fórmula 181.

[Fórmula 181]

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

$$\vec{\sigma}^T := (\sigma_1,\ldots,\sigma_L)^T := M \cdot \vec{f}^T,$$

$$\sigma_0 := \vec{1} \cdot \vec{f}^T,$$

$$s_{i,t} \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad i=1,\ldots,L;\ (t,v_{i,t}) \in \vec{v}_i,$$

$$\sigma_i := \sum_{(t,v_{i,t}) \in \vec{v}_i} s_{i,t} \quad \text{para} \quad i=1,\ldots,L$$

Usando el dispositivo de procesado, la unidad 140 de generación de claves de descifrado genera un elemento k^*_0 de una clave de descifrado sk_s tal como se indica en la Fórmula 182, y genera un elemento $k^*_{i,t}$ de la clave de descifrado sk_s para cada i = 1,..., L (L es un entero de 1 ó mayor) y cada índice t incluido en un conjunto $I_{\vec{vi}}$, tal como se indica en la Fórmula 183.

[Fórmula 182]

$$k_0^* := (-\sigma_0, 0, 1, \eta_0, 0)_{\mathbb{B}_0^*}$$

[Fórmula 183]

$$k_{i,t}^* := (\overbrace{\delta_i v_{i,t},\ s_{i,t}}^{2},\ \overbrace{0^{n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{\eta_{i,1},\ldots,\eta_{i,n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{0,}^{1})_{\mathbb{B}_{\tau,t}^*}$$

Resumiendo, el algoritmo *KeyGen* es tal como se indica en la Fórmula 184.

[Fórmula 184]

KeyGen(pk,sk,$\mathbb{S}$:= $(M,\rho),\{\tau,\vec{v}_i := \{(t,v_{i,t}) \mid t \in I_{\vec{v}_i}\} \mid i=1,\ldots,L\}$)

$$\vec{f} \xleftarrow{U} \mathbb{F}_q^r,\ \ \vec{\sigma}^T := (\sigma_1,\ldots,\sigma_L)^T := M \cdot \vec{f}^T,\ \ \sigma_0 := \vec{1} \cdot \vec{f}^T,$$

$$s_{i,t} \xleftarrow{U} \mathbb{F}_q,\ \ \sigma_i := \sum_{(t,v_{i,t}) \in \vec{v}_i} s_{i,t} \text{ para } \ i=1,\ldots,L;\ (t,v_{i,t}) \in \vec{v}_i,$$

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

$$k_0^* := (-\sigma_0, 0, 1, \eta_0, 0)_{\mathbb{B}_0^*},$$

para i=1,..., *L*

para $(t,v_{i,t}) \in \vec{v}_i$,

si $\rho(i) = (\tau, \vec{v}_i),\ \ \eta_{i,1},\ldots,\eta_{i,n_{\tau,t}} \xleftarrow{U} \mathbb{F}_q,$

$$k_{i,t}^* := (\overbrace{\delta_i v_{i,t},\ s_{i,t}}^{2},\ \overbrace{0^{n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{\eta_{i,1},\ldots,\eta_{i,n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{0,}^{1})_{\mathbb{B}_{\tau,t}^*}$$

$sk_{\mathbb{S}} := (\mathbb{S}, k_0^*, \{k_{i,t}^*\}_{i=1,\ldots,L;(t,v_{i,t}) \in \vec{v}_i})$.

devolver $sk_{\mathbb{S}}$.

Se describirá el algoritmo *Enc*.

Usando el dispositivo de procesado, la unidad 230 de generación de textos cifrados genera un elemento $c_{\tau,t}$ del texto cifrado ct_x para uno o más índices τ y cada índice t incluido en un conjunto $I_{xi\rightarrow}$, según se indica en la Fórmula 185.

[Fórmula 185]

$$c_{\tau,t} := (\overbrace{\omega_\tau x_{\tau,t},\ \omega_\tau}^{2},\ \overbrace{0^{n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{0^{n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{\phi_{\tau,t}}^{1})_{\mathbb{B}_{\tau,t}}$$

Resumiendo, el algoritmo *Enc* es tal como se indica en la Fórmula 186.

[Fórmula 186]

$$\mathrm{Enc}(\mathrm{pk}, m, \Gamma := \{(\tau, \vec{x_\tau} := \{(t, x_{\tau,t}) \mid t \in I_{\vec{x_t}}\} \mid 1 \le \tau \le d\})$$

$$\omega_\tau, \phi_0, \phi_{\tau,t}, \zeta \xleftarrow{U} \mathbb{F}_q \quad \text{para} \quad (\tau, \vec{x_\tau}) \in \Gamma; (t, x_{\tau,t}) \in \vec{x_\tau},$$

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

para $\quad (\tau, \vec{x_\tau}) \in \Gamma; (t, x_{\tau,t}) \in \vec{x_\tau}$

$$c_{\tau,t} := (\overbrace{\omega_\tau x_{\tau,t},\ \omega_\tau}^{2},\ \overbrace{0^{n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{0^{n_{\tau,t}},}^{n_{\tau,t}}\ \overbrace{\phi_{\tau,t}}^{1})_{\mathbb{B}_{\tau,t}},$$

$$c_{d+1} := g_T^{\zeta} m, \quad ct_\Gamma := (\Gamma, c_0, \{c_{\tau,t}\}_{(\tau, \vec{x_\tau}) \in \Gamma; (t, x_{\tau,t}) \in \vec{x_\tau}}, c_{d+1}).$$

devolver $\quad ct_\Gamma.$

Se describirá el algoritmo *Dec*.

Usando el dispositivo de procesado, la unidad 340 de operaciones de emparejamiento calcula la Fórmula 187, y genera así una clave de sesión $K = g_T^{\zeta}$ y calcula un mensaje $m' = c_{d+1}/K$.

[Fórmula 187]

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i) = (\tau, \vec{v_i}) \wedge t \in I_{\vec{v_i}}} e(c_{\tau,t}, k_{i,t}^*)^{\alpha_i}$$

Resumiendo, el algoritmo *Dec* es tal como se indica en la Fórmula 188.

[Fórmula 188]

$\mathrm{Dec}(\mathrm{pk}, \mathrm{sk}_{\mathbb{S}}, ct_{\Gamma})$

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

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

$$\sigma_0 = \sum_{i \in I} \alpha_i \sigma_i, \quad \text{y} \quad I \subseteq \{i \in \{1, \ldots, L\} \left| [\rho(i) = (\tau, \vec{v}_i) \wedge (\tau, \vec{x}_\tau) \in \Gamma \wedge \vec{v}_i \cdot \vec{x}_\tau = 0] \right\}.$$

$$K := e(c_0, k_0^*) \cdot \prod_{i \in I \wedge \rho(i) = (\tau, \vec{v}_i) \wedge t \in I_{\vec{v}_i}} e(c_{\tau,t}, k_{i,t}^*)^{\alpha_i},$$

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

devolver *m'*.

De esta manera, puede implementarse el esquema de cifrado funcional que presenta el esquema de cifrado por predicados de productos internos descrito en una de las anteriores realizaciones como estructura inferior.

En la anterior descripción, se ha descrito el esquema de cifrado funcional en el cual se requiere que el parámetro público se vuelva a emitir para añadir una categoría de atributo en una fase posterior. No obstante, tal como se describe en las Realizaciones 1 a 3, las partes de los índices se pueden proporcionar de manera que pueda añadirse una categoría de atributos sin volver a emitir el parámetro público.

En el esquema de cifrado funcional antes descrito, se utilizan la base $B_{\tau,t}$ y la base $B^*_{\tau,t}$. Así, es necesario proporcionar las partes de los índices para cada uno del índice τ y el índice t.

En este caso, el algoritmo G_{ob}, el algoritmo *Setup*, el algoritmo *KeyGen* y el algoritmo *Enc* del esquema de cifrado por predicados de productos internos descrito anteriormente son tal como se indica en la Fórmula 189 a la Fórmula 192. El algoritmo *Dec* es tal como se indica en la Fórmula 188, sin ningún cambio.

[Fórmula 189]

$\mathcal{G}_{\mathrm{ob}}(1^\lambda)$:

$$\mathrm{param}_{\mathbb{G}} := (q, \mathbb{G}, \mathbb{G}_T, g, e) \xleftarrow{R} \mathcal{G}_{\mathrm{bpg}}(1^\lambda), \quad \psi \xleftarrow{U} \mathbb{F}_q^\times,$$

$$N_0 := 5, \quad N_1 := 6 + 2n + 1,$$

para $t = 0,1$,

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

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

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

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

$$g_T := e(g, g)^\psi, \quad \mathrm{param} := (\{\mathrm{param}_{\mathbb{V}_t}\}_{t=0,1}, g_T)$$

devolver $(\mathrm{param}, \{\mathbb{B}_t, \mathbb{B}_t^*\}_{t=0,1})$.

[Fórmula 190]

$$\text{Setup}(1^{\lambda})$$

$$(\text{param},(\mathbb{B}_0,\mathbb{B}_0^*),(\mathbb{B},\mathbb{B}^*)) \xleftarrow{R} \mathcal{G}_{ob}(1^{\lambda}),$$

$$\hat{\mathbb{B}}_0 := (b_{0,1},b_{0,3},b_{0,5}),\ \hat{\mathbb{B}} := (b_1,\ldots,b_6,b_{6+2n+1}),$$

$$\hat{\mathbb{B}}_0^* := (b_{0,1}^*,b_{0,3}^*,b_{0,4}^*),\ \hat{\mathbb{B}}^* := (b_1^*,\ldots,b_6^*,b_{6+n+1}^*,\ldots,b_{6+2n}^*),$$

devolver $\text{pk} := (1^{\lambda}, \text{param}, \hat{\mathbb{B}}_0, \hat{\mathbb{B}}),\ \text{sk} := (\hat{\mathbb{B}}_0^*, \hat{\mathbb{B}}^*).$

[Fórmula 191]

$$\text{KeyGen}(\text{pk},\text{sk},\mathbb{S} := (M,\rho),\{\tau,\vec{v}_i := \{(t,v_{i,t}) \mid t \in I_{\vec{v}_i}\} \mid i=1,\ldots,L\})$$

$$\vec{f} \xleftarrow{U} \mathbb{F}_q^r,\ \ \vec{\sigma}^T := (\sigma_1,\ldots,\sigma_L)^T := M \cdot \vec{f}^T,\ \ \sigma_0 := \vec{1} \cdot \vec{f}^T,$$

$$s_{i,t} \xleftarrow{U} \mathbb{F}_q,\ \ \sigma_i := \sum_{(t,v_{i,t}) \in \vec{v}_i} s_{i,t} \text{ para } \quad i=1,\ldots,L;\ (t,v_{i,t}) \in \vec{v}_i,$$

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

$$k_0^* := (-\sigma_0,0,1,\eta_0,0)_{\mathbb{B}_0^*},$$

para i=1,.., L

para $(t,v_{i,t}) \in \vec{v}_i$,

si $\rho(i) = (\tau,\vec{v}_i),\ \ \mu_\tau,\mu_t \xleftarrow{U} \mathbb{F}_q,\ \ \eta_{i,1},\ldots,\eta_{i,n} \xleftarrow{U} \mathbb{F}_q,$

$$k_{i,t}^* := (\overbrace{\mu_\tau(\tau,\ -1,),\ \mu_t(t,\ -1,),\ \delta_i v_{i,t},\ s_{i,t}}^{6},$$

$$\overbrace{0^n,}^{n}\ \overbrace{\eta_{i,1},\ldots,\eta_{i,n},}^{n}\ \overbrace{0,}^{1})_{\mathbb{B}^*},$$

$$\text{sk}_{\mathbb{S}} := (\mathbb{S},k_0^*,\{k_{i,t}^*\}_{i=1,\ldots,L;(t,v_{i,t}) \in \vec{v}_i}).$$

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

[Fórmula 192]

$$\mathsf{Enc}(\mathsf{pk}, m, \Gamma := \{(\tau, \vec{x_\tau} := \{(t, x_{\tau,t}) \mid t \in I_{\vec{x_t}}\} \mid 1 \le \tau \le d\})$$

$$\omega_\tau, \phi_0, \phi_{\tau,t}, \zeta \xleftarrow{\cup} \mathbb{F}_q \quad \text{para} \quad (\tau, \vec{x_\tau}) \in \Gamma; (t, x_{\tau,t}) \in \vec{x_\tau},$$

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

para $(\tau, \vec{x_\tau}) \in \Gamma; (t, x_{\tau,t}) \in \vec{x_\tau}$

$$\sigma_\tau, \sigma_t \xleftarrow{\cup} \mathbb{F}_q,$$

$$c_{\tau,t} := (\overbrace{\sigma_\tau(1, \tau),\ \sigma_t(1, t),\ \omega_\tau x_{\tau,t},\ \omega_\tau}^{6},\ \underbrace{0^n,}_{n}\ \underbrace{0^n,}_{n}\ \underbrace{\phi_{\tau,t}}_{1})_{\mathbb{B}},$$

$$c_{d+1} := g_T^{\zeta} m, \quad ct_\Gamma := (\Gamma, c_0, \{c_{\tau,t}\}_{(\tau, \vec{x_\tau}) \in \Gamma; (t, x_{\tau,t}) \in \vec{x_\tau}}, c_{d+1}).$$

devolver ct_Γ.

En la anterior descripción, se ha descrito el caso en el que el esquema de cifrado por predicados de productos internos de la Realización 1 se aplica al esquema KP-FE descrito, por ejemplo, en la referencia bibliográfica 31, que no es documento de patente.

No obstante, de una manera similar, el esquema de cifrado por predicados de productos internos descrito en la Realización 1 se puede aplicar a un esquema de cifrado funcional con política de textos cifrados (esquema CP-FE) o un esquema de cifrado funcional con política unificada (esquema UP-FE) descrito en la referencia bibliográfica 31, no documento de patente. Alternativamente, el esquema de cifrado por predicados de productos internos descrito en la Realización 2 ó 3 se puede aplicar a cada uno de los esquemas de cifrado funcional descritos en la referencia bibliográfica 31, que no es documento de patente. Naturalmente, los esquemas de cifrado por predicados de productos internos, descritos en las anteriores realizaciones, se pueden aplicar a otros esquemas de cifrado funcional, no limitados a los esquemas de cifrado funcional descritos en la referencia bibliográfica 31, no documento de patente. De una manera similar, los esquemas de cifrado por predicados de productos internos, descritos en las realizaciones anteriores, se pueden aplicar a esquemas de firmas basadas en atributos, descritos en la referencia bibliográfica 32, no documento de patente, y otras referencias bibliográficas.

En cualquiera de los casos, los esquemas de cifrado por predicados de productos internos, descritos en las anteriores realizaciones, se pueden aplicar al cálculo del producto interno del vector de atributos $\vec{x}$ y el vector de predicados $\vec{v}$ para cada índice t.

Realización 5

En las realizaciones anteriores, se han descrito los métodos para implementar los procesos de las primitivas criptográficas en los espacios de vectores duales. En la Realización 5, se describirá un método para implementar los procesos de las primitivas criptográficas en grupos aditivos duales.

Más específicamente, en las realizaciones anteriores, los procesos de las primitivas criptográficas se implementan en el grupo cíclico del orden primo q. No obstante, cuando un anillo R se expresa usando un número compuesto M tal como se indica en la Fórmula 193, los procesos de las primitivas criptográficas descritos en las realizaciones anteriores también se pueden aplicar a un grupo aditivo que tenga el anillo R como coeficiente.

[Fórmula 193]

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

donde

$\mathbb{Z}$: entero, y

M: número compuesto.

Cambiando F_q por R en los algoritmos descritos en las realizaciones anteriores, pueden implementarse los procesos de las primitivas criptográficas en grupos aditivos duales.

A continuación se describirá una configuración de hardware del sistema 10 de procesado criptográfico (el dispositivo

100 de generación de claves, el dispositivo 200 de cifrado, el dispositivo 300 de descifrado) en las realizaciones.

La Fig. 16 es un diagrama que ilustra un ejemplo de la configuración de hardware del dispositivo 100 de generación de claves, el dispositivo 200 de cifrado, y el dispositivo 300 de descifrado.

Tal como se ilustra en la Fig. 16, cada uno del dispositivo 100 de generación de claves, el dispositivo 200 de cifrado, y el dispositivo 300 de descifrado tiene la CPU 911 (a la que se hace referencia también como Unidad de Procesado Central, dispositivo de procesado central, dispositivo de procesado, dispositivo aritmético, microprocesador, microordenador o procesador) que ejecuta programas. La CPU 911 está conectada, por medio de un bus 912, a la ROM 913, la RAM 914, la LCD 901 (Pantalla de Cristal Líquido), el teclado 902 (K/B), la placa 915 de comunicaciones, y el dispositivo 920 de disco magnético, y controla estos dispositivos de hardware. En lugar del dispositivo 920 de disco magnético (dispositivo de disco fijo), puede utilizarse un dispositivo de almacenamiento, tal como un dispositivo de disco óptico o un dispositivo de lectura/escritura de tarjetas de memoria. El dispositivo 920 de disco magnético se conecta por medio 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 comunicaciones son ejemplos del dispositivo de entrada. La placa 915 de comunicaciones es un ejemplo del dispositivo de comunicaciones. La LCD 901 es un ejemplo de un dispositivo de visualización.

El dispositivo 920 de disco magnético, la ROM 913, o similares, almacena un sistema operativo 921 (OS), un sistema 922 de ventanas, programas 923, y archivos 924. Los programas 923 son ejecutados por la CPU 911, el sistema operativo 921 y el sistema 922 de ventanas.

Los programas 923 almacenan software y programas que ejecutan las funciones descritas en la exposición anterior como la unidad 110 de generación de claves maestras, la unidad 120 de almacenamiento de claves maestras, la unidad 130 de introducción de información, la unidad 140 de generación de claves de descifrado, la unidad 150 de distribución de claves, la unidad 210 de adquisición de parámetros públicos, la unidad 220 de introducción de información, la unidad 230 de generación de textos cifrados, la unidad 240 de transmisión de datos, la unidad 310 de adquisición de claves de descifrado, la unidad 320 de almacenamiento de claves de descifrado, la unidad 330 de adquisición de textos cifrados, la unidad 340 de operaciones de emparejamiento, la unidad 350 de cálculo de mensajes, y similares. Los programas 923 también almacenan otros programas. Los programas son leídos y ejecutados por la CPU 911.

Los archivos 924 almacenan información, datos, valores de señales, valores de variables y parámetros, tales como el parámetro público pk, la clave secreta maestra sk, la clave de descifrado sk_v, el texto cifrado ct_x, el vector de predicados v, el vector de atributos x y el mensaje m en la descripción anterior, como elementos de un “archivo” y una “base de datos”. El “archivo” y la “base de datos” se almacenan en un soporte de grabación, tal como un disco o memoria. La información, datos, valores de señales, valores de variables y parámetros almacenados en el soporte de grabación, tal como el disco o la memoria, se leen a la memoria principal o memoria caché por medio de la CPU 911, a través de un circuito de lectura/escritura, y se usan para operaciones de la CPU 911, tales como extracción, búsqueda, consulta, comparación, cálculo, cómputo, procesado, salida, impresión y visualización. La información, datos, valores de señales, valores de variables y parámetros se almacenan temporalmente en la memoria principal, la memoria caché o la memoria intermedia durante las operaciones de la CPU 911 que incluyen extracción, búsqueda, consulta, comparación, cálculo, cómputo, procesado, salida, impresión y visualización.

Las flechas en los diagramas de flujo de la descripción anterior indican principalmente entrada/salida de datos y señales. Los datos y los valores de señales se almacenan en la memoria de la RAM 914, el soporte 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 señales, o cables, o por medio de ondas eléctricas.

Lo que se describe como “unidad” en la anterior descripción puede ser un “circuito”, un “dispositivo”, un “equipo”, unos “medios” o una “función”, y también puede ser una “etapa”, un “procedimiento”, o un “proceso”. Lo que se describe como “dispositivo” puede ser un “circuito”, un “equipo”, unos medios “medios”, o una “función”, y también puede ser una “etapa”, un “procedimiento”, o un “proceso”. Lo que se describe como un “proceso” puede ser una “etapa”. En otras palabras, lo que se describe como una “unidad” se puede realizar mediante microprogramas almacenados en la ROM 913. Alternativamente, lo que se describe como una “unidad” se puede implementar meramente mediante software, o meramente mediante hardware, tal como un elemento, un dispositivo, un sustrato, o una línea de cableado, o mediante una combinación de software y microprogramas, o mediante una combinación que incluye microprogramas. Los microprogramas y el software se almacenan como programas en el soporte de grabación, tal como la ROM 913. Los programas son leídos por la CPU 911 y son ejecutados por la CPU 911. Es decir, cada programa provoca que el ordenador o similares funcione como cada “unidad” antes descrita. Alternativamente, cada programa provoca que el ordenador o similares ejecute un procedimiento o un método de cada “unidad” descrita anteriormente.

Lista de referencias

100: dispositivo de generación de claves, 110: unidad de generación de claves maestras, 120: unidad de almacenamiento de claves maestras, 130: unidad de introducción de información, 140: unidad de generación de claves de descifrado, 150: unidad de distribución de claves, 200: dispositivo de cifrado, 210: unidad de adquisición de parámetros públicos, 220: unidad de introducción de información, 230: unidad de generación de textos cifrados, 240: unidad de transmisión de datos, 300: dispositivo de descifrado, 310: unidad de adquisición de claves de descifrado, 320: unidad de almacenamiento de claves de descifrado, 330: unidad de adquisición de textos cifrados, 340: unidad de operaciones de emparejamiento, 350: unidad de cálculo de mensajes

REIVINDICACIONES

1. Sistema criptográfico (10) que comprende un dispositivo (200) de cifrado y un dispositivo (300) de descifrado,

en donde el dispositivo (200) de cifrado incluye

una unidad (230) de generación de textos cifrados que está configurada para generar un texto cifrado que tiene un elemento c_0 en el cual un valor $\tilde{\omega}$ se fija como coeficiente de un vector de base $b_{0,r}$, y un elemento c_t en el cual información de atributos x_t se fija como coeficiente de un vector de base b_p y el valor $\tilde{\omega}$ se fija como coeficiente de un vector de base b_q, para cada índice t incluido en un conjunto $I_{x\rightarrow}$, y

en donde el dispositivo (300) de descifrado incluye

una unidad (320) de almacenamiento de claves de descifrado que está configurada para almacenar una clave de descifrado que tiene un elemento k_0 y un elemento k_t que se generan usando un valor s_t y un valor s_0 que es una suma del valor s_t para cada índice t incluido en un conjunto $I_{v\rightarrow}$, siendo el elemento k_0 un elemento en el cual un valor $-s_0$ se fija como coeficiente de un vector de base $b^*_{0,r}$ correspondiente al vector de base $b_{0,r}$, siendo el elemento k_t un elemento en el cual información de predicados v_t se fija como coeficiente de un vector de base b^*_p correspondiente al vector de base b_p y el valor s_t se fija como coeficiente de un vector de base b^*_q correspondiente al vector de base b_q, para cada índice t incluido en el conjunto $I_{v\rightarrow}$; y

una unidad (340, 350) de descifrado que está configurada para descifrar el texto cifrado generado por la unidad (230) de generación de textos cifrados con la clave de descifrado almacenada por la unidad (320) de almacenamiento de claves de descifrado, estando configurada la unidad (340, 350) de descifrado para descifrar el texto cifrado calculando un producto de operaciones de emparejamiento entre pares correspondientes de los vectores de base sobre el elemento c_0 y el elemento k_0 y sobre el elemento c_t y el elemento k_t para cada índice t incluido en el conjunto $I_{v\rightarrow}$.

2. Sistema criptográfico según la reivindicación 1,

en el que la unidad (230) de generación de textos cifrados está configurada además para generar un elemento c_t en el cual información J que se asigna de antemano al índice t se fija como coeficiente de un vector de base $b_{índice}$, y

en el que la unidad (320) de almacenamiento de claves de descifrado está configurada además para generar un elemento k_t en el cual información J' que tiene un producto interno de 0 con la información J que se asigna de antemano al índice t, se fija como coeficiente de un vector de base $b^*_{índice}$ correspondiente al vector de base $b_{índice}$.

3. Sistema criptográfico según la reivindicación 2,

en el que la unidad (230) de generación de textos cifrados está configurada para generar el elemento c_0 y el elemento c_t tal como se indica en la Fórmula 1,

en el que la unidad (320) de almacenamiento de claves de descifrado está configurada para almacenar el elemento k_0 y el elemento k_t tal como se indica en la Fórmula 2, y

en el que la unidad (340, 350) de descifrado está configurada para llevar a cabo un cálculo tal como se indica en la Fórmula 3,

[Fórmula 1]

$$c_0 := (\tilde{\omega},\ \overbrace{0^{u_0}}^{u_0},\ \zeta,\ \overbrace{0^{w_0}}^{w_0},\ \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbf{B}_0\boldsymbol{\theta}}$$

$$c_t = (\overbrace{\sigma_t(1,\ t),\ \omega x_t,\ \tilde{\omega}}^{4}\ \overbrace{0^{u}}^{u},\ \overbrace{0^{w}}^{w},\ \overbrace{\vec{\varphi}_t}^{z})_{\mathbf{B}}$$

donde

$\zeta, \sigma_t, \omega, \tilde{\omega}, \vec{\varphi}_0 := (\varphi_{0,1},...,\varphi_{0,z_0}), \vec{\varphi}_t := (\varphi_{t,1},...,\varphi_{t,z})$ son números aleatorios, y u_0, w_0, z_0, u, w, z son, cada uno de ellos, un entero de 0 ó mayor,

[Fórmula 2]

$$\boldsymbol{k}_0^* := (-s_0, \overbrace{0^{u_0},}^{u_0} \ 1, \ \overbrace{\vec{\eta}_0, \ \overbrace{0^{z_0},}^{z_0}}^{w_0})_{\mathbf{B}_0^*},$$

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t, \ -1,), \ \delta v_t, \ s_t}^{4} \ \overbrace{0^u,}^{u} \ \overbrace{\vec{\eta}_t,}^{w} \ \overbrace{0^z}^{z})_{\mathbf{B}^*}$$

donde

$\delta, \mu_t, \vec{\eta}_0 := (\eta_{0,1},...,\eta_{0,w_0}), \eta_t := (\eta_{t,1},...,\eta_{t,w})$ son números aleatorios, y u_0, w_0, z_0, u, w, z son, cada uno de ellos, un entero de 0 ó mayor,

[Fórmula 3]

$$K := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \prod_{t \in I_{\vec{v}}} e(\boldsymbol{c}_t, \boldsymbol{k}_t^*) \ .$$

4. Sistema criptográfico (10) que está configurado para realizar un proceso de una primitiva criptográfica determinando, para cada índice τ, si un producto interno es 0 entre un vector de atributos $x_\tau^{\rightarrow}$ que tiene, como elemento, información de atributos $x_{\tau,t}$ para cada índice t incluido en un conjunto $I_{\tau,x_\rightarrow}$, y un vector de predicados $v_\tau^{\rightarrow}$ que tiene, como elemento, información de atributos $v_{\tau,t}$ para cada índice t incluido en un conjunto $I_{\tau,v_\rightarrow}$, comprendiendo el sistema criptográfico un transmisor (200) y un receptor (300),

en donde el transmisor (200) incluye

una unidad (230) de generación de información de transmisión que está configurada para generar, para cada índice τ, información de transmisión que tiene un elemento $c_{\tau,t}$ en el cual información de atributos $x_{\tau,t}$ se fija como coeficiente de un vector de base b_p para cada índice t incluido en el conjunto $I_{\tau,x_\rightarrow}$ y un valor $\tilde{\omega}_\tau$ se fija como coeficiente de un vector de base b_q, y

en donde el receptor (300) incluye

una unidad (320) de almacenamiento de información de recepción que está configurada para almacenar, para cada índice τ, información de recepción que tiene un elemento $k_{\tau,t}$ en el cual información de atributos $v_{\tau,t}$ se fija como coeficiente de un vector de base b^*_p correspondiente al vector de base b_p y un valor $s_{\tau,t}$ se fija como coeficiente de un vector de base b^*_q correspondiente al vector de base b_q, para cada índice t incluido en el conjunto $I_{\tau,v_\rightarrow}$; y

una unidad (340) de operaciones de emparejamiento que está configurada para calcular, para cada índice τ, un producto de operaciones de emparejamiento entre pares correspondientes de los vectores de base sobre el elemento $c_{\tau,t}$ y el elemento $k_{\tau,t}$ para cada índice t incluido en el conjunto $I_{\tau,v_\rightarrow}$.

5. Sistema criptográfico (10) que comprende un dispositivo (200) de cifrado y un dispositivo (300) de descifrado,

en donde el dispositivo (200) de cifrado incluye

una unidad (230) de generación de textos cifrados que está configurada para generar un texto cifrado que tiene un elemento c_0 y un elemento c_t que se generan usando un valor f_t y un valor f_0 el cual es una suma del valor f_t para cada índice t incluido en un conjunto $I_{x_\rightarrow}$, siendo el elemento c_0 un elemento en el cual un valor $-f_0$ se fija como coeficiente de un vector de base $b_{0,r}$, siendo el elemento c_t un elemento en el cual información de atributos x_t se fija como coeficiente de un vector de base b_p y el valor f_t se fija como coeficiente de un vector de base b_q, para cada índice t incluido en el conjunto $I_{x_\rightarrow}$, y

en donde el dispositivo (300) de descifrado incluye

una unidad (320) de almacenamiento de claves de descifrado que está configurada para almacenar una clave de descifrado que tiene un elemento k_0 en el cual un valor $\tilde{\delta}$ se fija como coeficiente de un vector de base $b^*_{0,r}$ correspondiente al vector de base $b_{0,r}$, y un elemento k_t en el cual información de predicados v_t se fija como coeficiente de un vector de base b^*_p correspondiente al vector de base b_p y el valor $\tilde{\delta}$ se fija como coeficiente de un vector de base b^*_q correspondiente al vector de base b_q para cada índice t incluido en un conjunto $I_{v_\rightarrow}$; y

una unidad (340, 350) de descifrado que está configurada para descifrar el texto cifrado generado por la unidad (230) de generación de textos cifrados con la clave de descifrado almacenada por la unidad (320) de almacenamiento de claves de descifrado, estando configurada la unidad (340, 350) de descifrado para descifrar el texto cifrado mediante el cálculo de un producto de operaciones de emparejamiento entre pares correspondientes de los vectores de base sobre el elemento c_0 y el elemento k_0 y sobre el elemento c_t y el elemento k_t para cada índice t incluido en el conjunto $I_{x_\rightarrow}$.

6. Sistema criptográfico según la reivindicación 5,

en el que la unidad (230) de generación de textos cifrados está configurada además para generar un elemento c_t en el cual información J que se asigna de antemano al índice t se fija como coeficiente de un vector de base $b_{índice}$, y

en el que la unidad (320) de almacenamiento de claves de descifrado está configurada además para generar un elemento k_t en el cual información J' que presenta un producto interno de 0 con la información J que se asigna de antemano al índice t, se fija como coeficiente de un vector de base $b^*_{índice}$ correspondiente al vector de base $b_{índice}$.

7. Sistema criptográfico según la reivindicación 6,

en el que la unidad (230) de generación de textos cifrados está configurada para generar el elemento c_0 y el elemento c_t según se indica en la Fórmula 4,

en el que la unidad (320) de almacenamiento de claves de descifrado está configurada para almacenar el elemento k_0 y el elemento k_t según se indica en la Fórmula 5, y

en el que la unidad (340, 350) de descifrado está configurada para realizar un cálculo según se indica en la Fórmula 6,

[Fórmula 4]

$$c_0 := (-f_0, \overbrace{0^{u_0}}^{u_0}, \zeta, \overbrace{0^{w_0}}^{w_0}, \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbf{B}_0\boldsymbol{\theta}}$$

$$\boldsymbol{c}_t = (\overbrace{\sigma_t(1,\ t),\ \omega x_t, f_t}^{4}\ \overbrace{0^{u}}^{u},\ \overbrace{0^{w}}^{w},\ \overbrace{\vec{\varphi}_t}^{z})_{\mathbf{B}}$$

donde

$\zeta, \sigma_t, \omega, \vec{\varphi}_0 := (\varphi_{0,1},...,\varphi_{0,z_0}), \vec{\varphi}_t := (\varphi_{t,1},...,\varphi_{t,z})$ son números aleatorios, y u_0, w_0, z_0, u, w, z son, cada uno de ellos, un entero de 0 ó mayor,

[Fórmula 5]

$$\boldsymbol{k}_0^* := (\tilde{\delta}, \overbrace{0^{u_0}}^{u_0},\ 1,\ \overbrace{\vec{\eta}_0}^{w_0},\ \overbrace{0^{z_0}}^{z_0})_{\mathbf{B}_0^*},$$

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t,\ -1,),\ \delta v_t,\ \tilde{\delta}}^{4}\ \overbrace{0^{u}}^{u},\ \overbrace{\vec{\eta}_t}^{w},\ \overbrace{0^{z}}^{z})_{\mathbf{B}^*}$$

donde

$\delta, \tilde{\delta}, \mu_t, \vec{\eta}_0 := (\eta_{0,1},...,\eta_{0,w_0}), \eta_t := (\eta_{t,1},...,\eta_{t,w})$ son números aleatorios, y u_0, w_0, z_0, u, w, z son, cada uno de ellos, un entero de 0 ó mayor,

[Fórmula 6]

$$K := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \prod_{t \in I_{\vec{x}}} e(\boldsymbol{c}_t, \boldsymbol{k}_t^*).$$

8. Sistema criptográfico (10) que está configurado para realizar un proceso de una primitiva criptográfica determinando, para cada índice τ, si un producto interno es 0 entre un vector de atributos $x_\tau^{\rightarrow}$ que tiene, como elemento, información de atributos $x_{\tau,t}$ para cada índice t incluido en un conjunto $I_{\tau,x_\rightarrow}$, y un vector de predicados $v_\tau^{\rightarrow}$ que tiene, como elemento, información de atributos $v_{\tau,t}$ para cada índice t incluido en un conjunto $I_{\tau,v_\rightarrow}$, comprendiendo el sistema criptográfico un transmisor (200) y un receptor (300),

en donde el transmisor (200) incluye

una unidad (230) de generación de información de transmisión que está configurada para generar, para cada índice τ, información de transmisión que tiene un elemento $c_{\tau,t}$ en el cual información de atributos $x_{\tau,t}$ se fija como coeficiente de un vector de base b_p y un valor $f_{\tau,t}$ se fija como coeficiente de un vector de base b_q, para cada índice t incluido en el conjunto $I_{\tau,x_\rightarrow}$, y

en donde el receptor (300) incluye

una unidad (320) de almacenamiento de información de recepción que está configurada para almacenar, para cada índice τ, información de recepción que tiene un elemento $k_{\tau,t}$ en el cual información de atributos $v_{\tau,t}$ se fija como coeficiente de un vector de base b^*_p correspondiente al vector de base b_p y un valor $\delta^{\sim}_{\tau}$ se fija como coeficiente de un vector de base b^*_q correspondiente al vector de base b_q, para cada índice t incluido en el conjunto $I_{\tau,v_{\rightarrow}}$; y

una unidad (340) de operaciones de emparejamiento que está configurada para calcular, para cada índice τ, un producto de operaciones de emparejamiento entre pares correspondientes de los vectores de base sobre el elemento $c_{\tau,t}$ y el elemento $k_{\tau,t}$ para cada índice t incluido en el conjunto $I_{\tau,x_{\rightarrow}}$.

9. Sistema criptográfico (10) que comprende un dispositivo (200) de cifrado y un dispositivo (300) de descifrado,

en el que el dispositivo (200) de cifrado incluye

una unidad (230) de generación de textos cifrados que está configurada para generar un texto cifrado que tiene un elemento c_0 y un elemento c_t que se generan usando un valor f_t y un valor f_0 el cual es una suma del valor f_t para cada índice t de t = 1, ..., n (siendo n un entero de 1 ó mayor), siendo el elemento c_0 un elemento en el cual un valor $\omega^{\sim}$ se fija como coeficiente de un vector de base $b_{0,r}$ y un valor $-f_0$ se fija como coeficiente de un vector de base $b_{0,r'}$, siendo el elemento c_t un elemento en el cual información de atributos x_t se fija como coeficiente de un vector de base b_p para cada índice t incluido en un conjunto I_x, el valor $\omega^{\sim}$ se fija como coeficiente de un vector de base b_q, y el valor f_t se fija como coeficiente de un vector de base $b_{q'}$, y

en el que el dispositivo (300) de descifrado incluye

una unidad (320) de almacenamiento de claves de descifrado que está configurada para almacenar una clave de descifrado que tiene un elemento k_0 y un elemento k_t que se generan usando un valor s_t y un valor s_0 el cual es una suma del valor s_t para cada índice t, siendo el elemento k_0 un elemento en el cual un valor $-s_0$ se fija como coeficiente de un vector de base $b^*_{0,r}$ correspondiente al vector de base $b_{0,r}$ y un valor $\delta^{\sim}$ se fija como coeficiente de un vector de base $b^*_{0,r'}$ correspondiente al vector de base $b_{0,r'}$, siendo el elemento k_t un elemento en el cual información de predicados v_t se fija como coeficiente de un vector de base b^*_p correspondiente al vector de base b_p, el valor s_t se fija como coeficiente de un vector de base b^*_q correspondiente al vector de base b_q, y el valor $\delta^{\sim}$ se fija como coeficiente de un vector de base $b^*_{q'}$ correspondiente al vector de base $b_{q'}$; y

una unidad (340, 350) de descifrado que está configurada para descifrar el texto cifrado generado por la unidad (230) de generación de textos cifrados con la clave de descifrado almacenada por la unidad (320) de almacenamiento de claves de descifrado, estando configurada la unidad (340, 350) de descifrado para descifrar el texto cifrado mediante el cálculo de un producto de operaciones de emparejamiento entre pares correspondientes de los vectores de base sobre el elemento c_0 y el elemento k_0 y sobre el elemento c_t y el elemento k_t para cada índice t.

10. Sistema criptográfico según la reivindicación 9,

en el que la unidad (230) de generación de textos cifrados está configurada además para generar un elemento c_t en el cual información J que se asigna de antemano al índice t se fija como coeficiente de un vector de base $b_{índice}$, y

en el que la unidad (320) de almacenamiento de claves de descifrado está configurada además para generar un elemento k_t en el cual información J' que tiene un producto interno de 0 con la información J que se asigna de antemano al índice t, se fija como coeficiente de un vector de base $b^*_{índice}$ correspondiente al vector de base $b_{índice}$.

11. Sistema criptográfico según la reivindicación 10,

en el que la unidad (230) de generación de textos cifrados está configurada para generar el elemento c_0 y el elemento c_t tal como se indica en la Fórmula 7,

en el que la unidad (320) de almacenamiento de claves de descifrado está configurada para almacenar el elemento k_0 y el elemento k_t tal como se indica en la Fórmula 8, y

en el que la unidad (340, 350) de descifrado está configurada para llevar a cabo un cálculo tal como se indica en la Fórmula 9,

[Fórmula 7]

$$c_0 := (\tilde{\omega},\ -f_0\ \overbrace{0^{u_0},}^{u_0}\ \zeta,\ \overbrace{0^{w_0},}^{w_0}\ \overbrace{\vec{\varphi}_0}^{z_0})_{\mathbf{B}_0\boldsymbol{\theta}}$$

$$\boldsymbol{c}_t = (\overbrace{\sigma_t(1,\ t),\ \omega x_t,\ \tilde{\omega},\ f_t}^{5}\ \overbrace{0^u,}^{u}\ \overbrace{0^w,}^{w}\ \overbrace{\vec{\varphi}_t}^{z})_{\mathbf{B}}$$

donde

$\zeta, \sigma_t, \omega, \tilde{\omega}, \vec{\varphi}_0 := (\varphi_{0,1}, ..., \varphi_{0,z_0}), \vec{\varphi}_t := (\varphi_{t,1}, ..., \varphi_{t,z})$ son números aleatorios, y u_0, w_0, z_0, u, w, z son, cada uno de ellos, un entero de 0 ó mayor,

[Fórmula 8]

$$\boldsymbol{k}_0^* := (-s_0,\ \tilde{\delta},\ \overbrace{0^{u_0},}^{u_0}\ 1,\ \overbrace{\vec{\eta}_0,}^{w_0}\ \overbrace{0^{z_0},}^{z_0})_{\mathbf{B}_0^*},$$

$$\boldsymbol{k}_t^* := (\overbrace{\mu_t(t,\ -1,),\ \delta v_t,\ s_t,\ \tilde{\delta}}^{5}\ \overbrace{0^{u},}^{u}\ \overbrace{\vec{\eta}_t,}^{w}\ \overbrace{0^{z}}^{z})_{\mathbf{B}^*}$$

donde

$\delta, \tilde{\delta}, \mu_t, \vec{\eta}_0 := (\eta_{0,1}, ..., \eta_{0,w_0}), \eta_t := (\eta_{t,1}, ..., \eta_{t,w})$ son números aleatorios, y u_0, w_0, z_0, u, w, z son, cada uno de ellos, un entero de 0 ó mayor,

[Fórmula 9]

$$K := e(\boldsymbol{c}_0, \boldsymbol{k}_0^*) \prod_{t=1}^{n} e(\boldsymbol{c}_t, \boldsymbol{k}_t^*)\ .$$

12. Sistema criptográfico (10) que está configurado para realizar un proceso de una primitiva criptográfica determinando para cada índice τ si un producto interno es 0 entre un vector de atributos $x_\tau^{\rightarrow}$ que tiene, como elemento, información de atributos $x_{\tau,t}$ para cada índice t de t = 1, ..., n (siendo n un entero de 1 ó mayor), y un vector de predicados $v_\tau^{\rightarrow}$ que tiene, como elemento, información de atributos $v_{\tau,t}$ para cada índice t, comprendiendo el sistema criptográfico un transmisor (200) y un receptor (300),

en donde el transmisor (200) incluye

una unidad (230) de generación de información de transmisión que está configurada para generar, para cada índice τ, información de transmisión que tiene un elemento $c_{\tau,t}$ en el cual información de atributos $x_{\tau,t}$ se fija como coeficiente de un vector de base b_p, un valor $\tilde{\omega}_\tau$ se fija como coeficiente de un vector de base b_q, y un valor f_t se fija como coeficiente de un vector de base $b_{q'}$, para cada índice t incluido en un conjunto $I_{\tau,x_\rightarrow}$, y

en donde el receptor (300) incluye

una unidad (320) de almacenamiento de información de recepción que está configurada para almacenar, para cada índice τ, información de recepción que tiene un elemento $k_{\tau,t}$ en el cual información de atributos $v_{\tau,t}$ se fija como coeficiente de un vector de base b^*_p correspondiente al vector de base b_p, un valor $s_{\tau,t}$ se fija como coeficiente de un vector de base b^*_q correspondiente al vector de base b_q, y un valor $\tilde{\delta}$ se fija como coeficiente de un vector de base $b^*_{q'}$ correspondiente al vector de base $b_{q'}$, para cada índice t incluido en un conjunto $I_{\tau,v_\rightarrow}$; y

una unidad (340) de operaciones de emparejamiento que está configurada para calcular un producto de operaciones de emparejamiento entre pares correspondientes de los vectores de base sobre el elemento $c_{\tau,t}$ y el elemento $k_{\tau,t}$ para cada índice τ y cada índice t.

Fig. 1

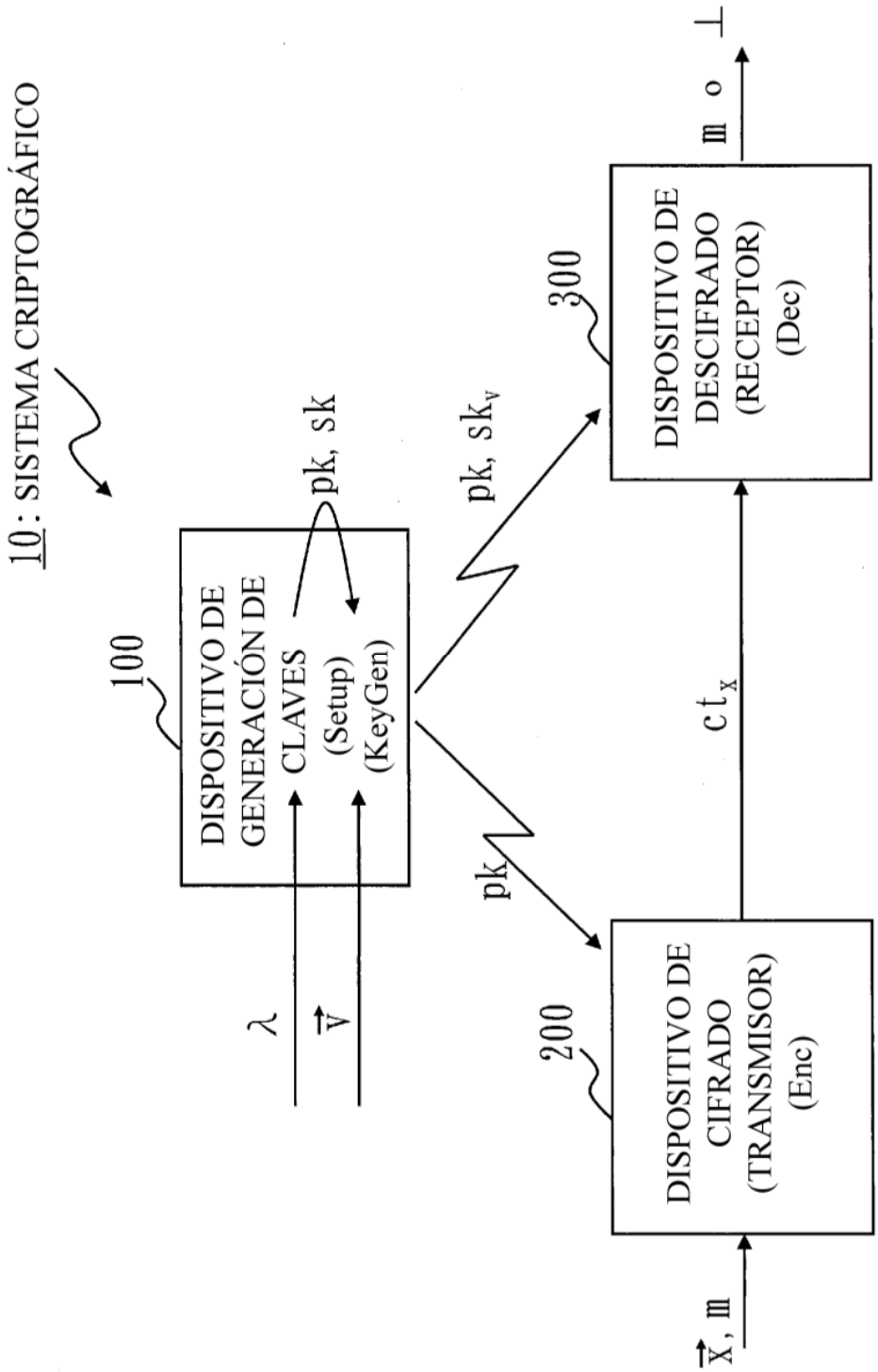

10: SISTEMA CRIPTOGRÁFICO
100
DISPOSITIVO DE GENERACIÓN DE CLAVES (Setup) (KeyGen)
λ
$\vec{v}$
pk, sk
pk, sk_v
pk
200
DISPOSITIVO DE CIFRADO (TRANSMISOR) (Enc)
$\vec{x}$, m
ct_x
300
DISPOSITIVO DE DESCIFRADO (RECEPTOR) (Dec)
m o ⊥

Fig. 2

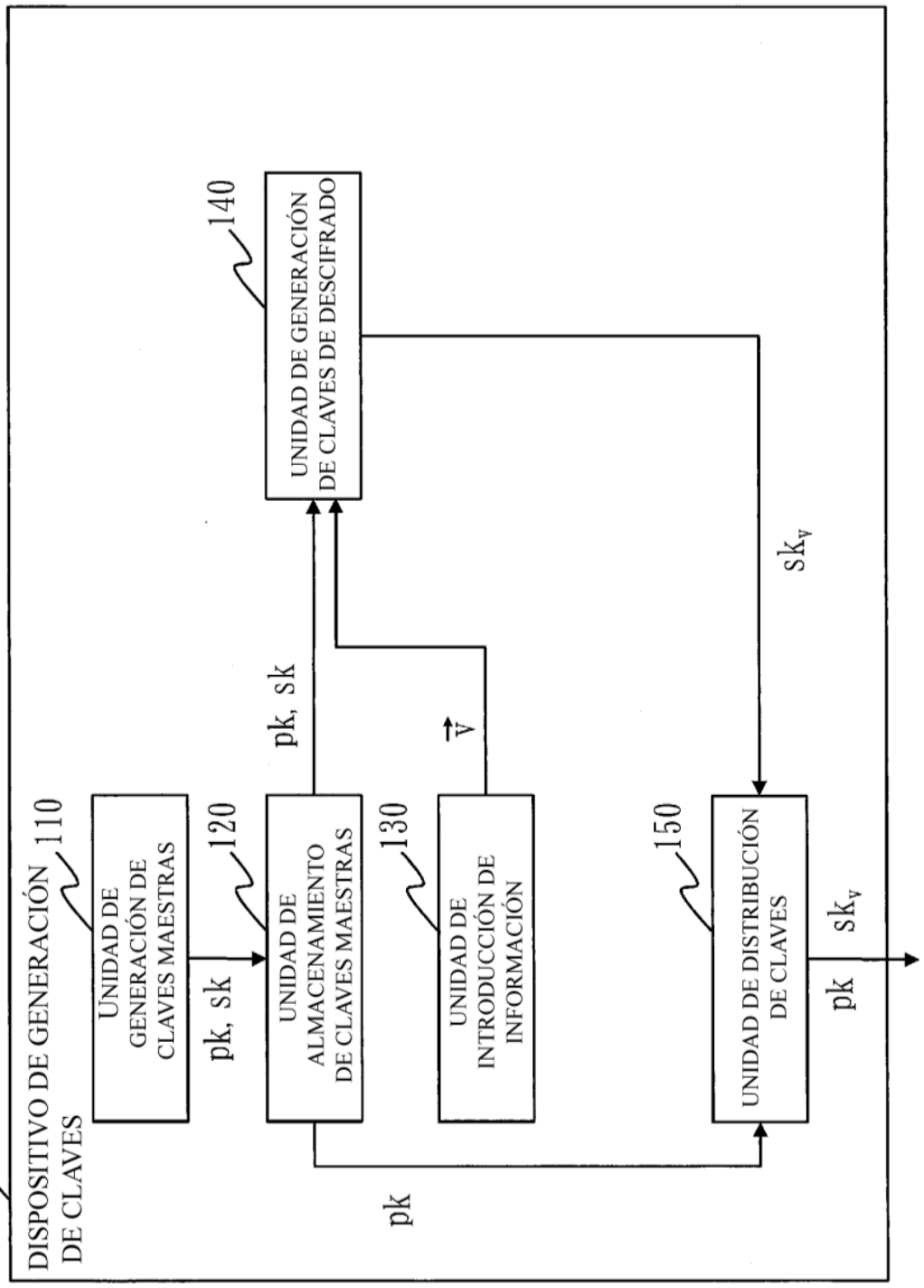

100
DISPOSITIVO DE GENERACIÓN DE CLAVES
110
UNIDAD DE GENERACIÓN DE CLAVES MAESTRAS
120
UNIDAD DE ALMACENAMIENTO DE CLAVES MAESTRAS
130
UNIDAD DE INTRODUCCIÓN DE INFORMACIÓN
140
UNIDAD DE GENERACIÓN DE CLAVES DE DESCIFRADO
150
UNIDAD DE DISTRIBUCIÓN DE CLAVES
pk, sk
pk, sk
$\vec{v}$
pk
sk_v
pk
sk_v

Fig. 3

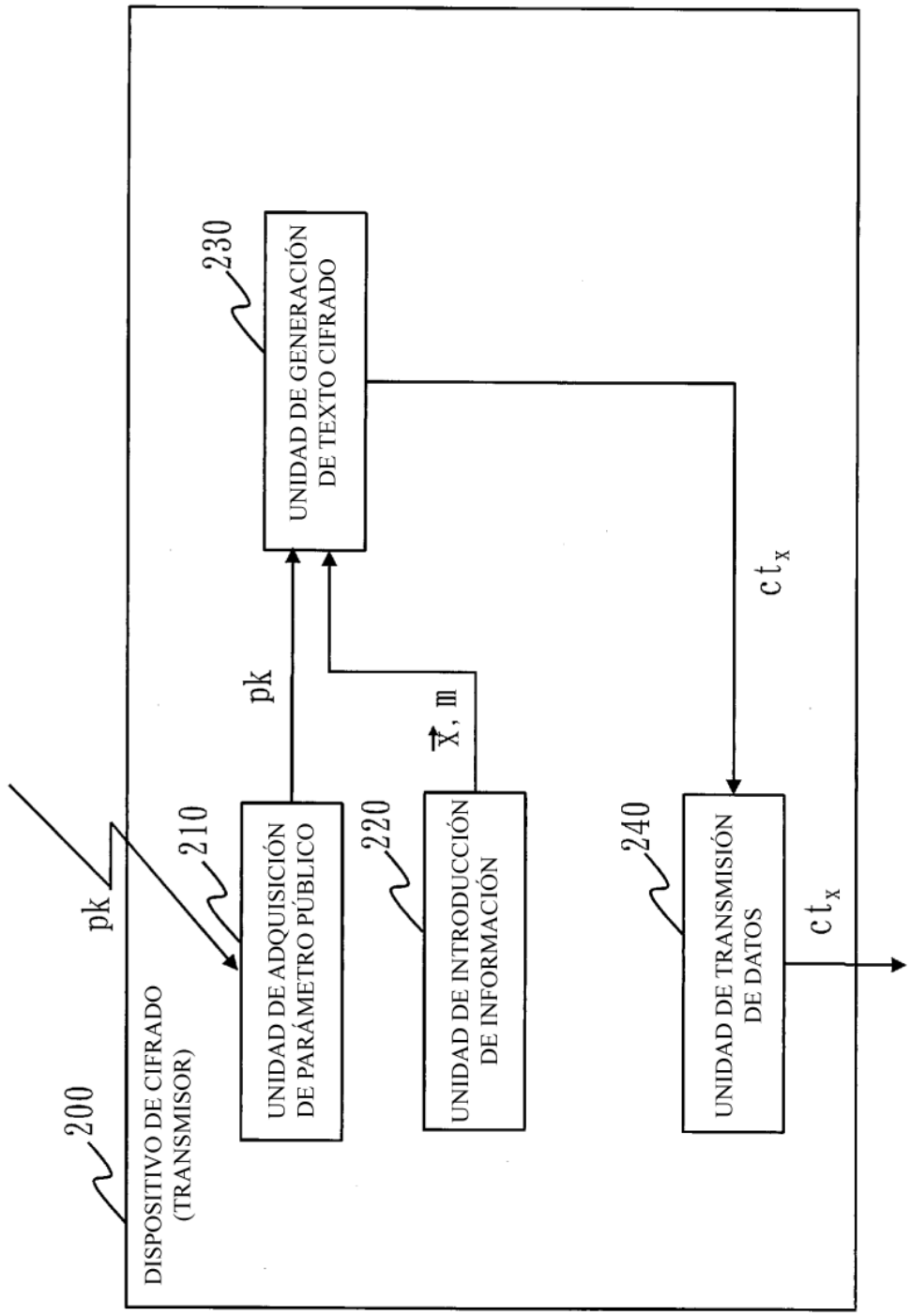

200
DISPOSITIVO DE CIFRADO
(TRANSMISOR)
pk
210
UNIDAD DE ADQUISICIÓN
DE PARÁMETRO PÚBLICO
pk
220
UNIDAD DE INTRODUCCIÓN
DE INFORMACIÓN
$\vec{x}$, m
230
UNIDAD DE GENERACIÓN
DE TEXTO CIFRADO
ct_x
240
UNIDAD DE TRANSMISIÓN
DE DATOS
ct_x

Fig. 4

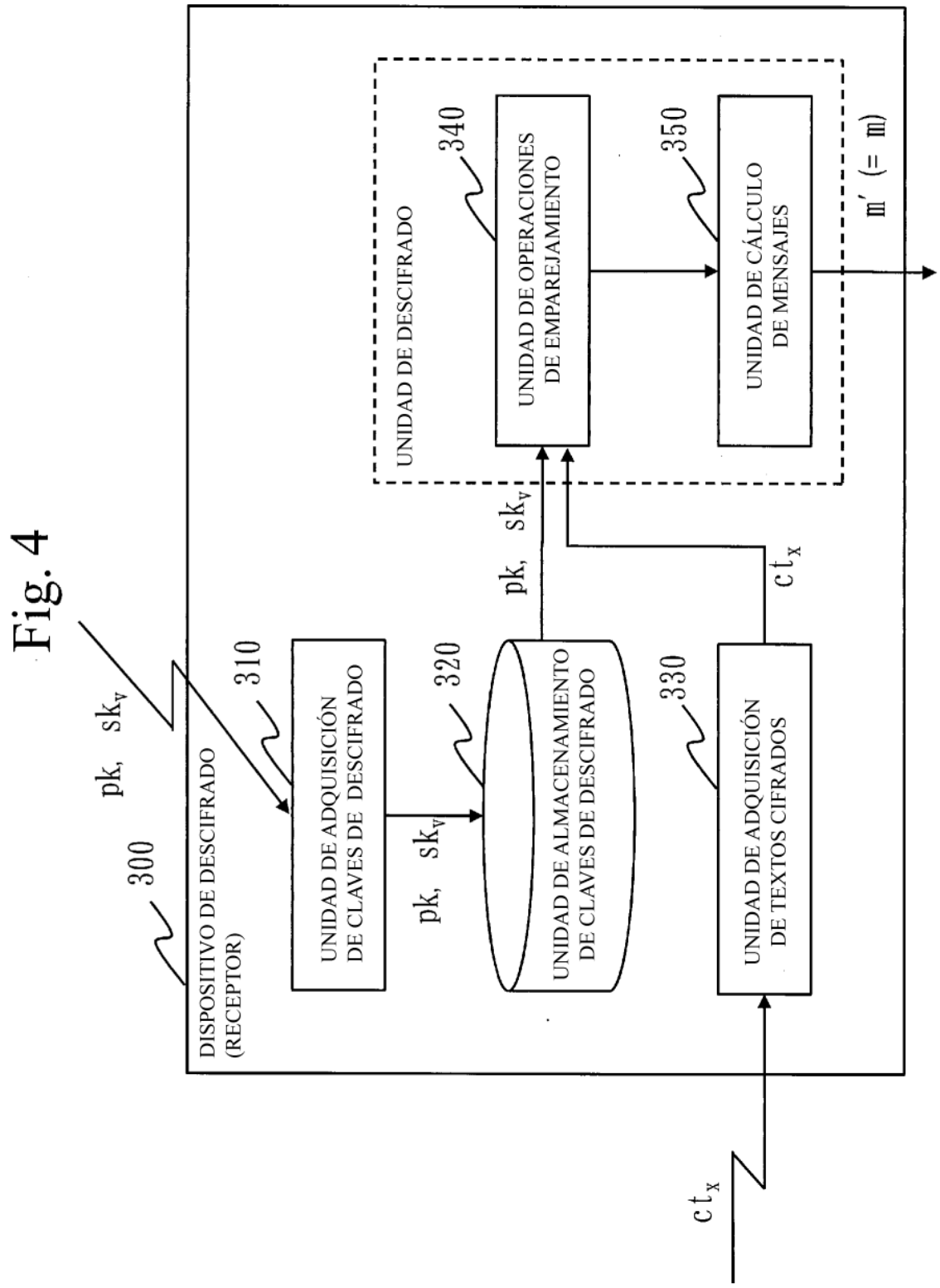

DISPOSITIVO DE DESCIFRADO
(RECEPTOR)
300
pk, sk$_v$
UNIDAD DE ADQUISICIÓN
DE CLAVES DE DESCIFRADO
310
pk, sk$_v$
UNIDAD DE ALMACENAMIENTO
DE CLAVES DE DESCIFRADO
320
pk, sk$_v$
UNIDAD DE ADQUISICIÓN
DE TEXTOS CIFRADOS
330
ct$_x$
ct$_x$
UNIDAD DE DESCIFRADO
UNIDAD DE OPERACIONES
DE EMPAREJAMIENTO
340
UNIDAD DE CÁLCULO
DE MENSAJES
350
m′ (= m)

Fig. 5

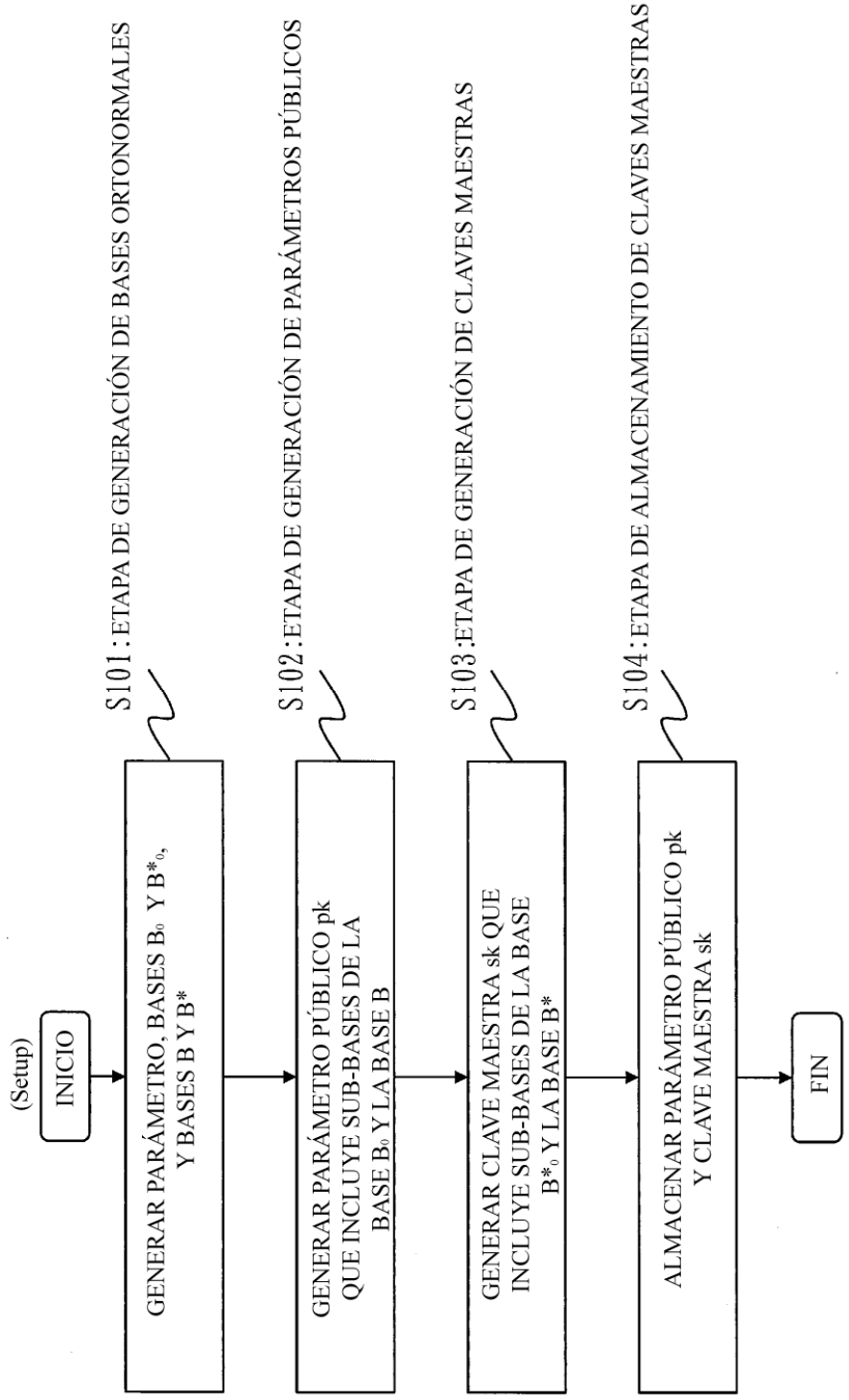

(Setup)
INICIO
S101 : ETAPA DE GENERACIÓN DE BASES ORTONORMALES
GENERAR PARÁMETRO, BASES B_0 Y B^*_0, Y BASES B Y B*
S102 : ETAPA DE GENERACIÓN DE PARÁMETROS PÚBLICOS
GENERAR PARÁMETRO PÚBLICO pk QUE INCLUYE SUB-BASES DE LA BASE B_0 Y LA BASE B
S103 : ETAPA DE GENERACIÓN DE CLAVES MAESTRAS
GENERAR CLAVE MAESTRA sk QUE INCLUYE SUB-BASES DE LA BASE B^*_0 Y LA BASE B*
S104 : ETAPA DE ALMACENAMIENTO DE CLAVES MAESTRAS
ALMACENAR PARÁMETRO PÚBLICO pk Y CLAVE MAESTRA sk
FIN

Fig. 6

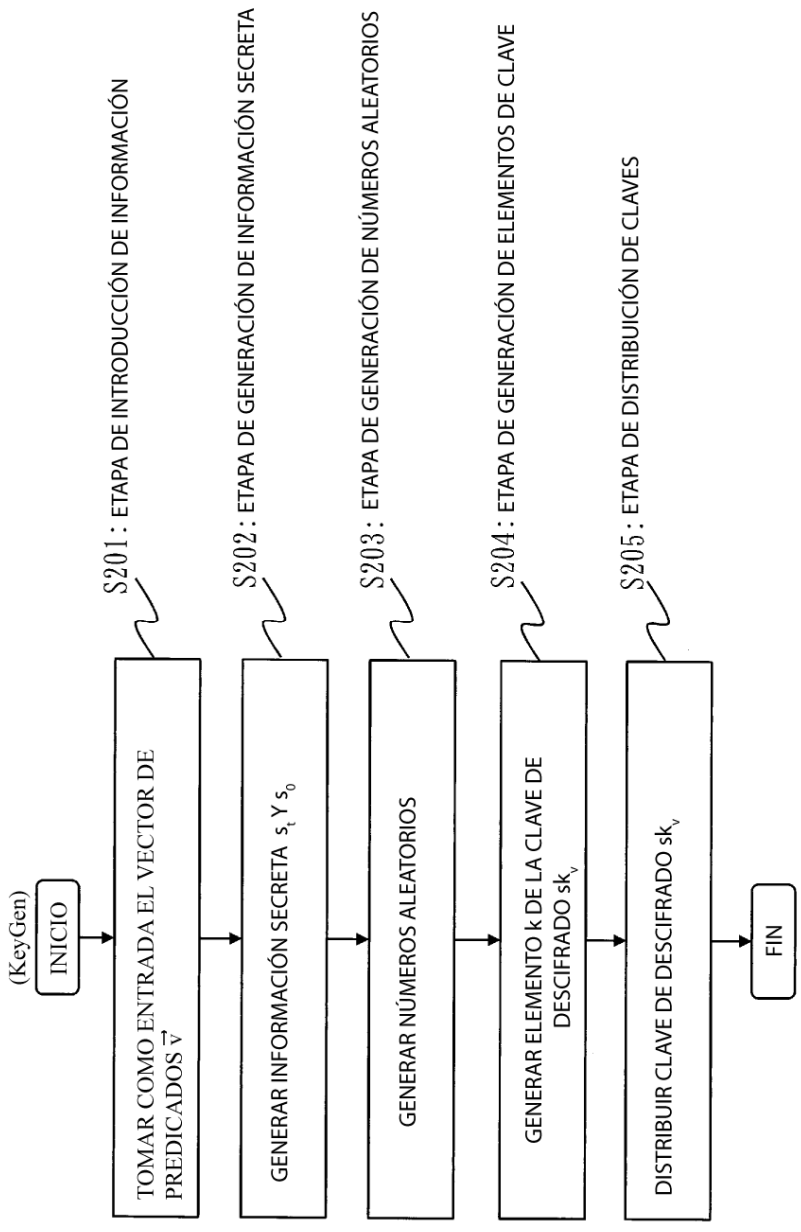

(KeyGen)
INICIO
S201: ETAPA DE INTRODUCCIÓN DE INFORMACIÓN
TOMAR COMO ENTRADA EL VECTOR DE PREDICADOS $\vec{v}$
S202: ETAPA DE GENERACIÓN DE INFORMACIÓN SECRETA
GENERAR INFORMACIÓN SECRETA s_t Y s_0
S203: ETAPA DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMEROS ALEATORIOS
S204: ETAPA DE GENERACIÓN DE ELEMENTOS DE CLAVE
GENERAR ELEMENTO k DE LA CLAVE DE DESCIFRADO sk_v
S205: ETAPA DE DISTRIBUICIÓN DE CLAVES
DISTRIBUIR CLAVE DE DESCIFRADO sk_v
FIN

Fig. 7

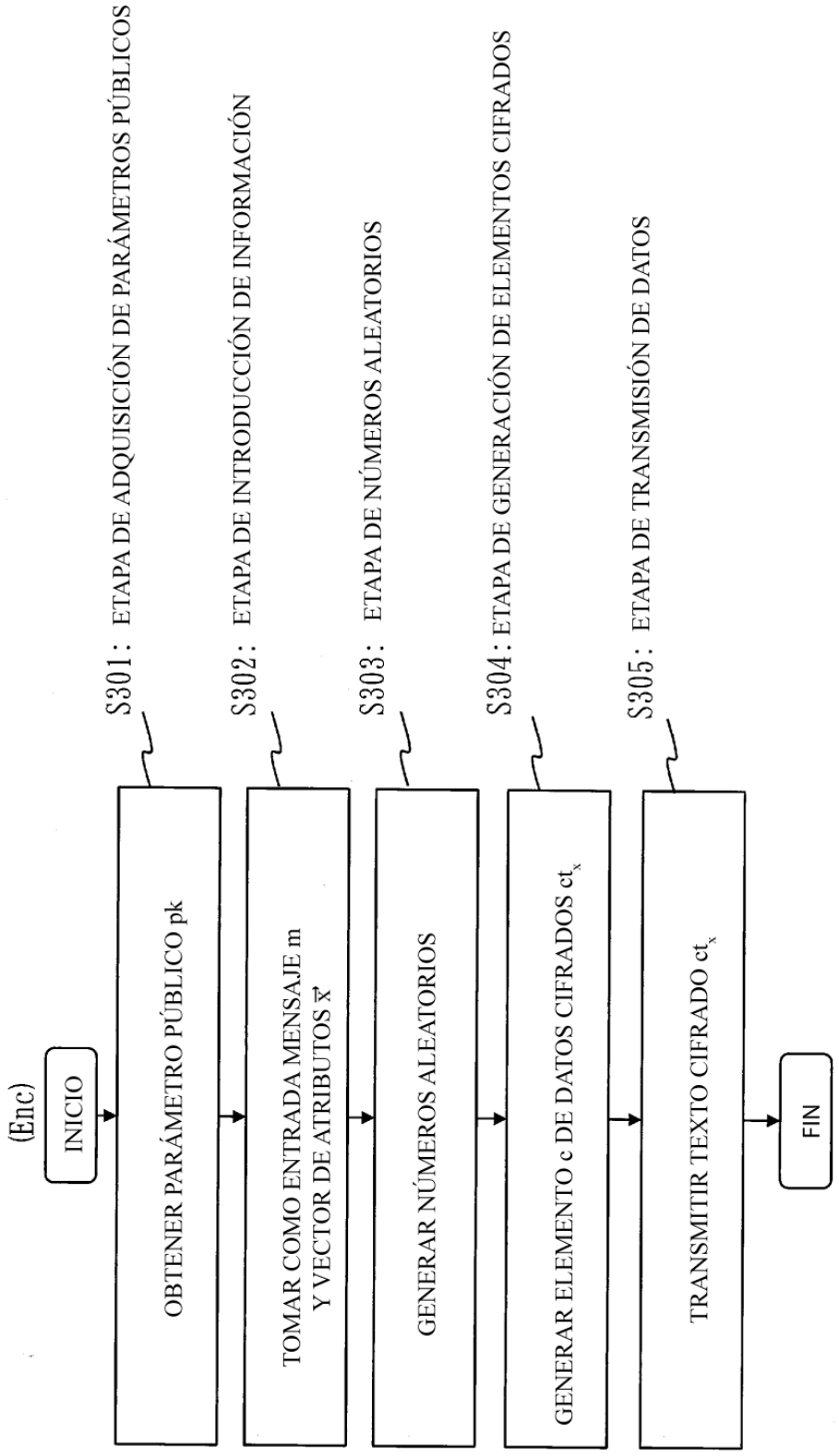

(Enc)
INICIO
S301: ETAPA DE ADQUISICIÓN DE PARÁMETROS PÚBLICOS
OBTENER PARÁMETRO PÚBLICO pk
S302: ETAPA DE INTRODUCCIÓN DE INFORMACIÓN
TOMAR COMO ENTRADA MENSAJE m
Y VECTOR DE ATRIBUTOS $\vec{x}$
S303: ETAPA DE NÚMEROS ALEATORIOS
GENERAR NÚMEROS ALEATORIOS
S304: ETAPA DE GENERACIÓN DE ELEMENTOS CIFRADOS
GENERAR ELEMENTO c DE DATOS CIFRADOS ct_x
S305: ETAPA DE TRANSMISIÓN DE DATOS
TRANSMITIR TEXTO CIFRADO ct_x
FIN

Fig. 8

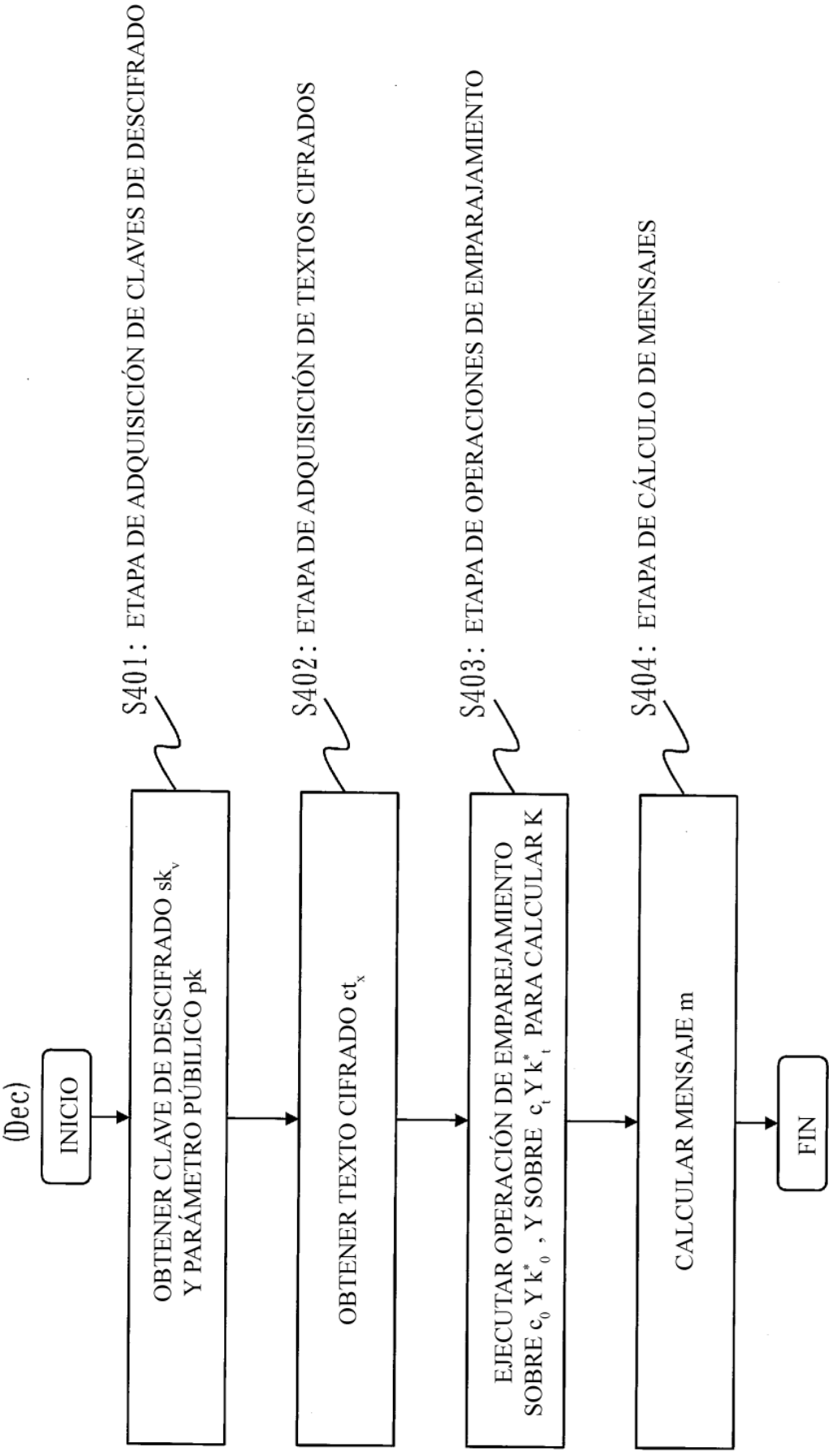

(Dec)
INICIO
S401: ETAPA DE ADQUISICIÓN DE CLAVES DE DESCIFRADO
OBTENER CLAVE DE DESCIFRADO sk_v Y PARÁMETRO PÚBLICO pk
S402: ETAPA DE ADQUISICIÓN DE TEXTOS CIFRADOS
OBTENER TEXTO CIFRADO ct_x
S403: ETAPA DE OPERACIONES DE EMPARAJAMIENTO
EJECUTAR OPERACIÓN DE EMPAREJAMIENTO SOBRE c_0 Y k^*_0 , Y SOBRE c_t Y k^*_t PARA CALCULAR K
S404: ETAPA DE CÁLCULO DE MENSAJES
CALCULAR MENSAJE m
FIN

Fig. 9

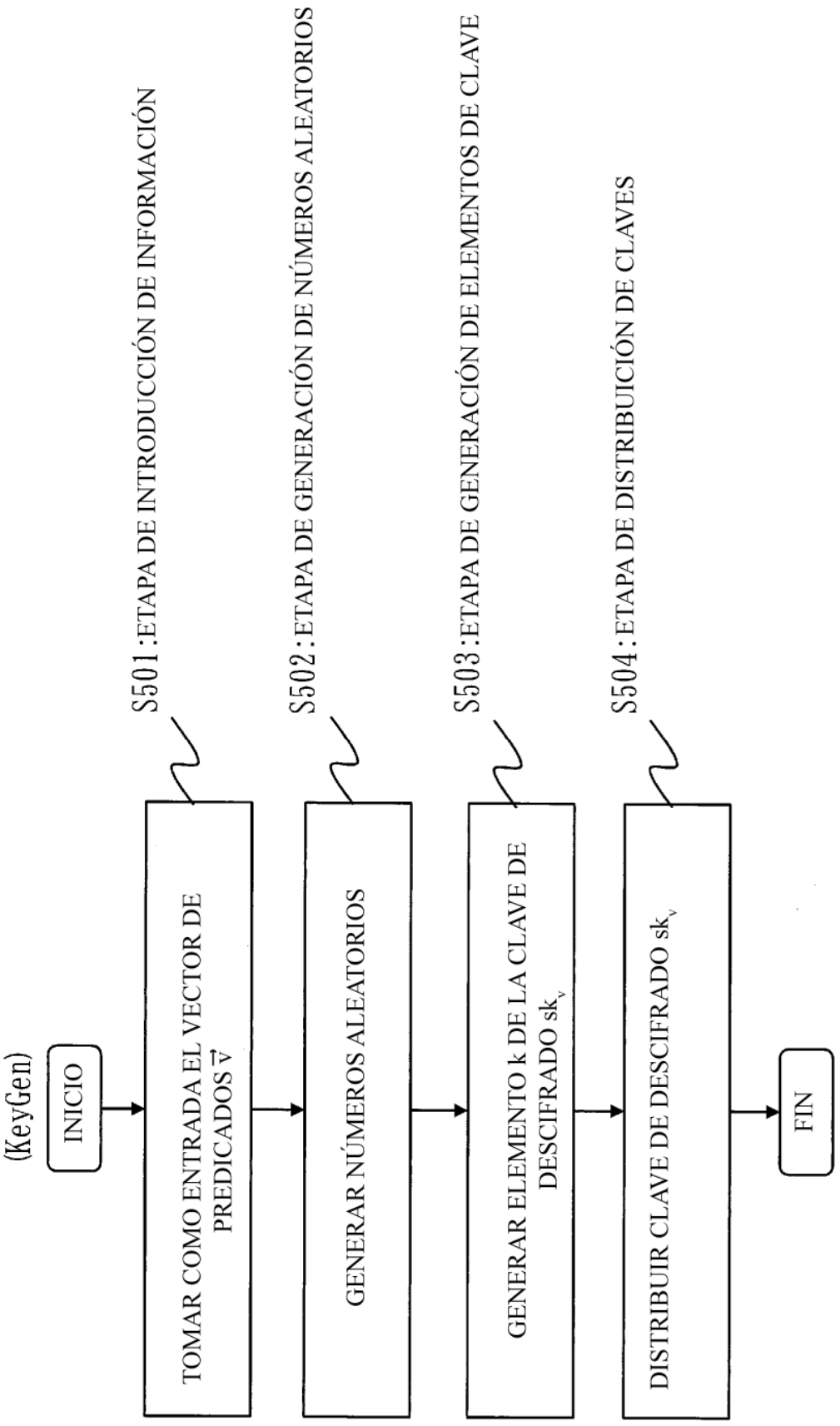

(KeyGen)
INICIO
S501 : ETAPA DE INTRODUCCIÓN DE INFORMACIÓN
TOMAR COMO ENTRADA EL VECTOR DE PREDICADOS $\vec{v}$
S502 : ETAPA DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMEROS ALEATORIOS
S503 : ETAPA DE GENERACIÓN DE ELEMENTOS DE CLAVE
GENERAR ELEMENTO k DE LA CLAVE DE DESCIFRADO sk_v
S504 : ETAPA DE DISTRIBUICIÓN DE CLAVES
DISTRIBUIR CLAVE DE DESCIFRADO sk_v
FIN

Fig. 10

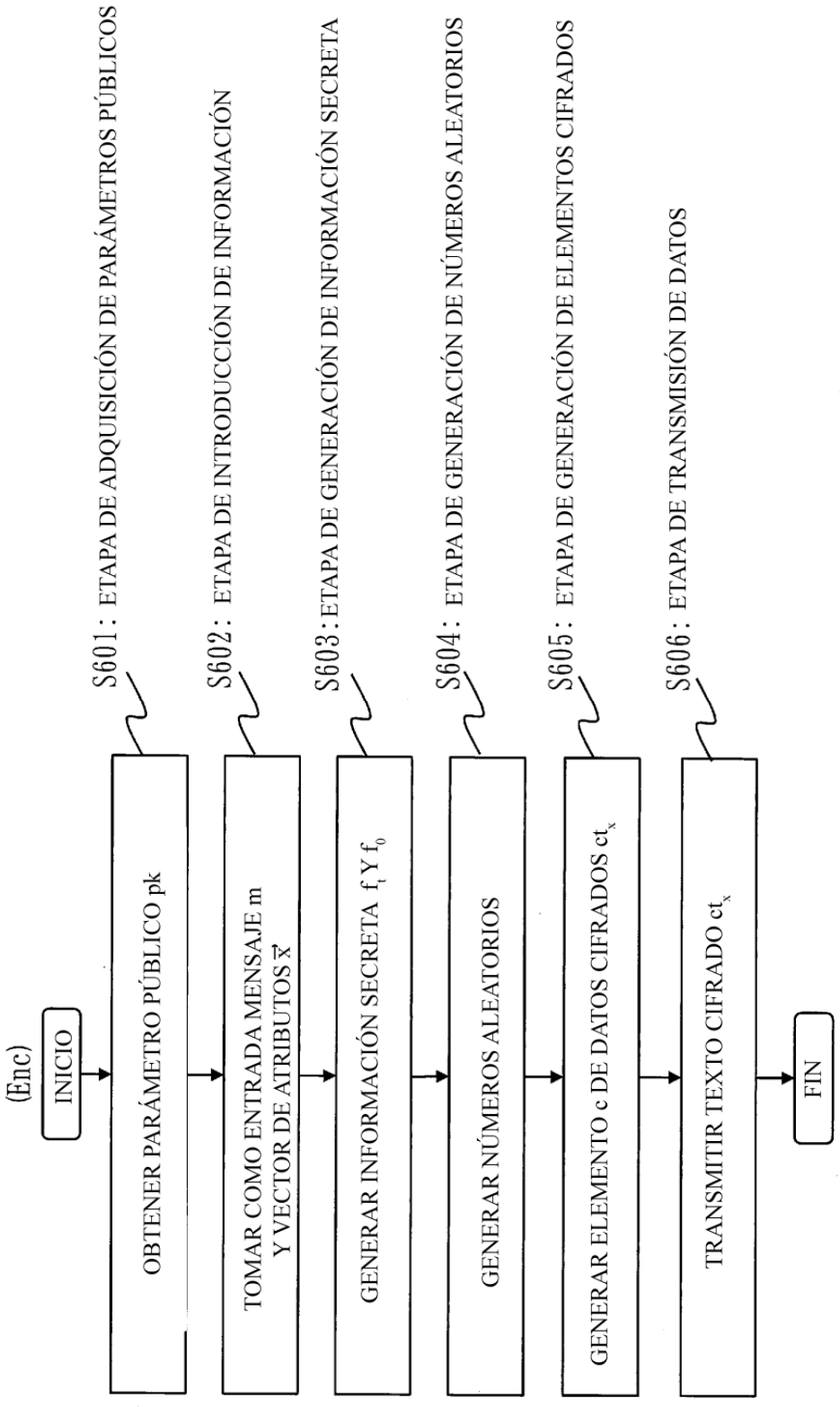

(Enc)
INICIO
S601: ETAPA DE ADQUISICIÓN DE PARÁMETROS PÚBLICOS
OBTENER PARÁMETRO PÚBLICO pk
S602: ETAPA DE INTRODUCCIÓN DE INFORMACIÓN
TOMAR COMO ENTRADA MENSAJE m
Y VECTOR DE ATRIBUTOS $\vec{x}$
S603: ETAPA DE GENERACIÓN DE INFORMACIÓN SECRETA
GENERAR INFORMACIÓN SECRETA f_t Y f_0
S604: ETAPA DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMEROS ALEATORIOS
S605: ETAPA DE GENERACIÓN DE ELEMENTOS CIFRADOS
GENERAR ELEMENTO c DE DATOS CIFRADOS ct_x
S606: ETAPA DE TRANSMISIÓN DE DATOS
TRANSMITIR TEXTO CIFRADO ct_x
FIN

Fig. 11

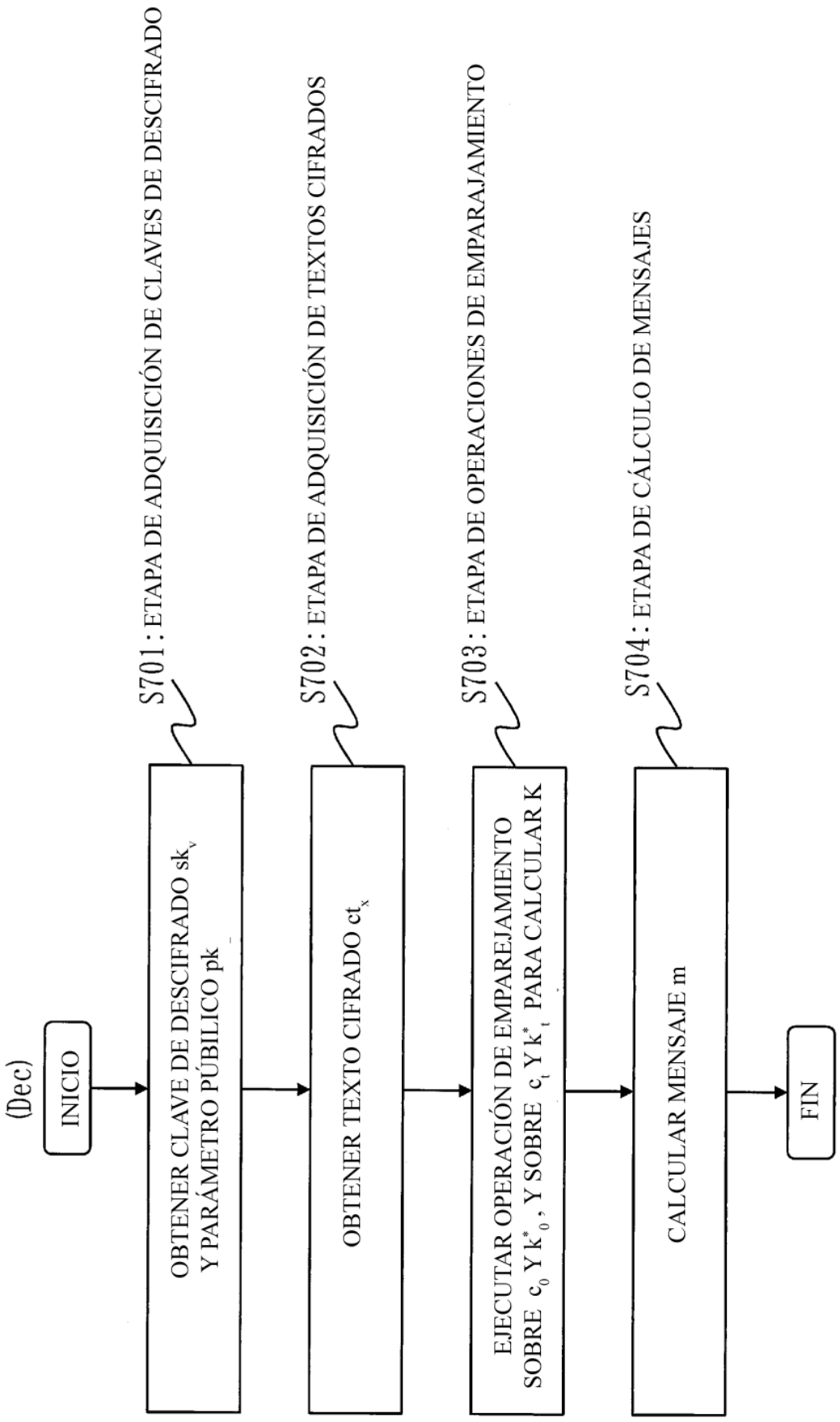

(Dec)
INICIO
S701 : ETAPA DE ADQUISICIÓN DE CLAVES DE DESCIFRADO
OBTENER CLAVE DE DESCIFRADO sk_v Y PARÁMETRO PÚBILICO pk
S702 : ETAPA DE ADQUISICIÓN DE TEXTOS CIFRADOS
OBTENER TEXTO CIFRADO ct_x
S703 : ETAPA DE OPERACIONES DE EMPARAJAMIENTO
EJECUTAR OPERACIÓN DE EMPAREJAMIENTO SOBRE c_0 Y k^*_0 , Y SOBRE c_t Y k^*_t PARA CALCULAR K
S704 : ETAPA DE CÁLCULO DE MENSAJES
CALCULAR MENSAJE m
FIN

Fig. 12

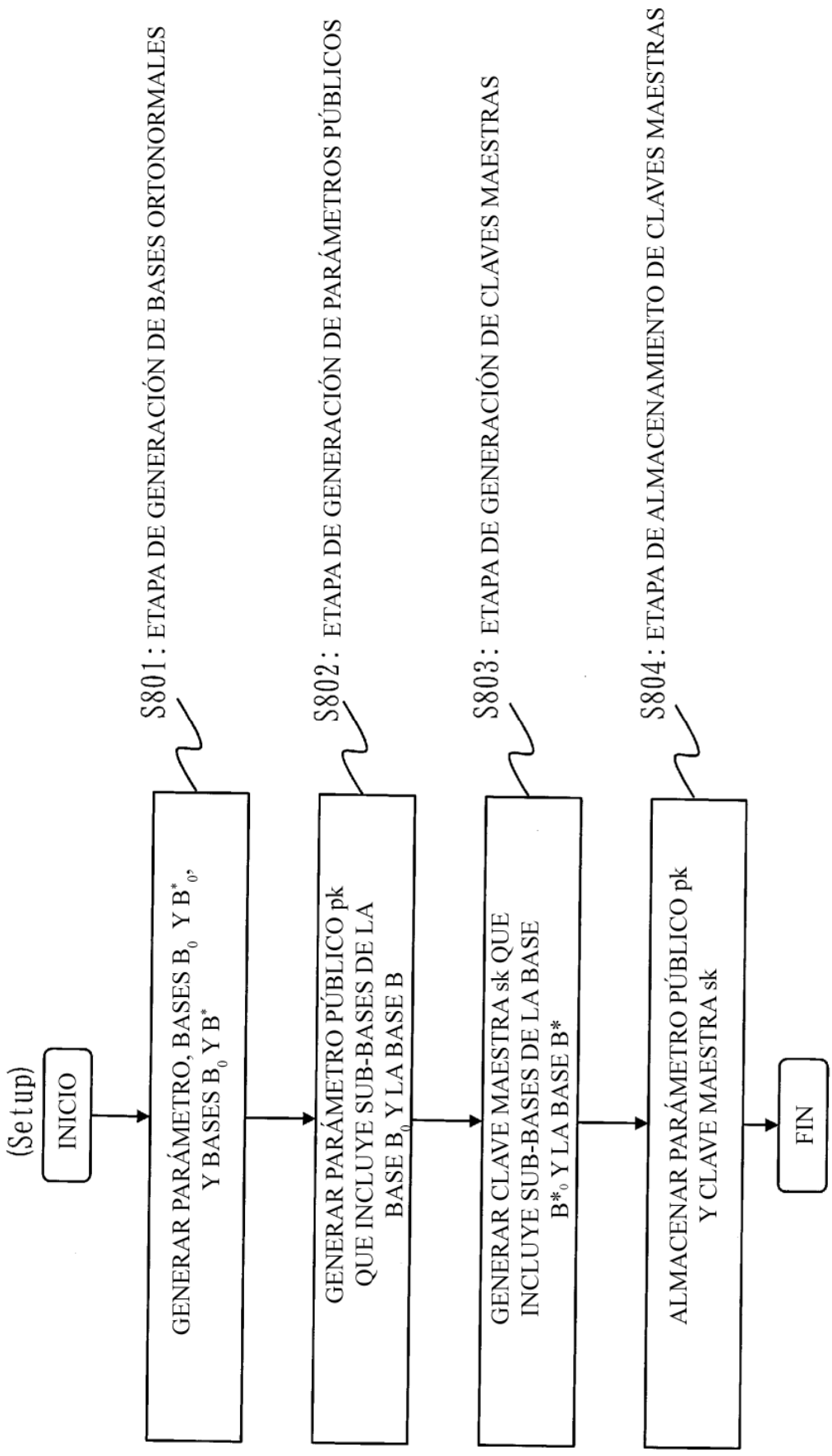

(Setup)
INICIO
S801: ETAPA DE GENERACIÓN DE BASES ORTONORMALES
GENERAR PARÁMETRO, BASES B_0 Y B^*_0, Y BASES B_0 Y B^*
S802: ETAPA DE GENERACIÓN DE PARÁMETROS PÚBLICOS
GENERAR PARÁMETRO PÚBLICO pk QUE INCLUYE SUB-BASES DE LA BASE B_0 Y LA BASE B
S803: ETAPA DE GENERACIÓN DE CLAVES MAESTRAS
GENERAR CLAVE MAESTRA sk QUE INCLUYE SUB-BASES DE LA BASE B^*_0 Y LA BASE B^*
S804: ETAPA DE ALMACENAMIENTO DE CLAVES MAESTRAS
ALMACENAR PARÁMETRO PÚBLICO pk Y CLAVE MAESTRA sk
FIN

Fig. 13

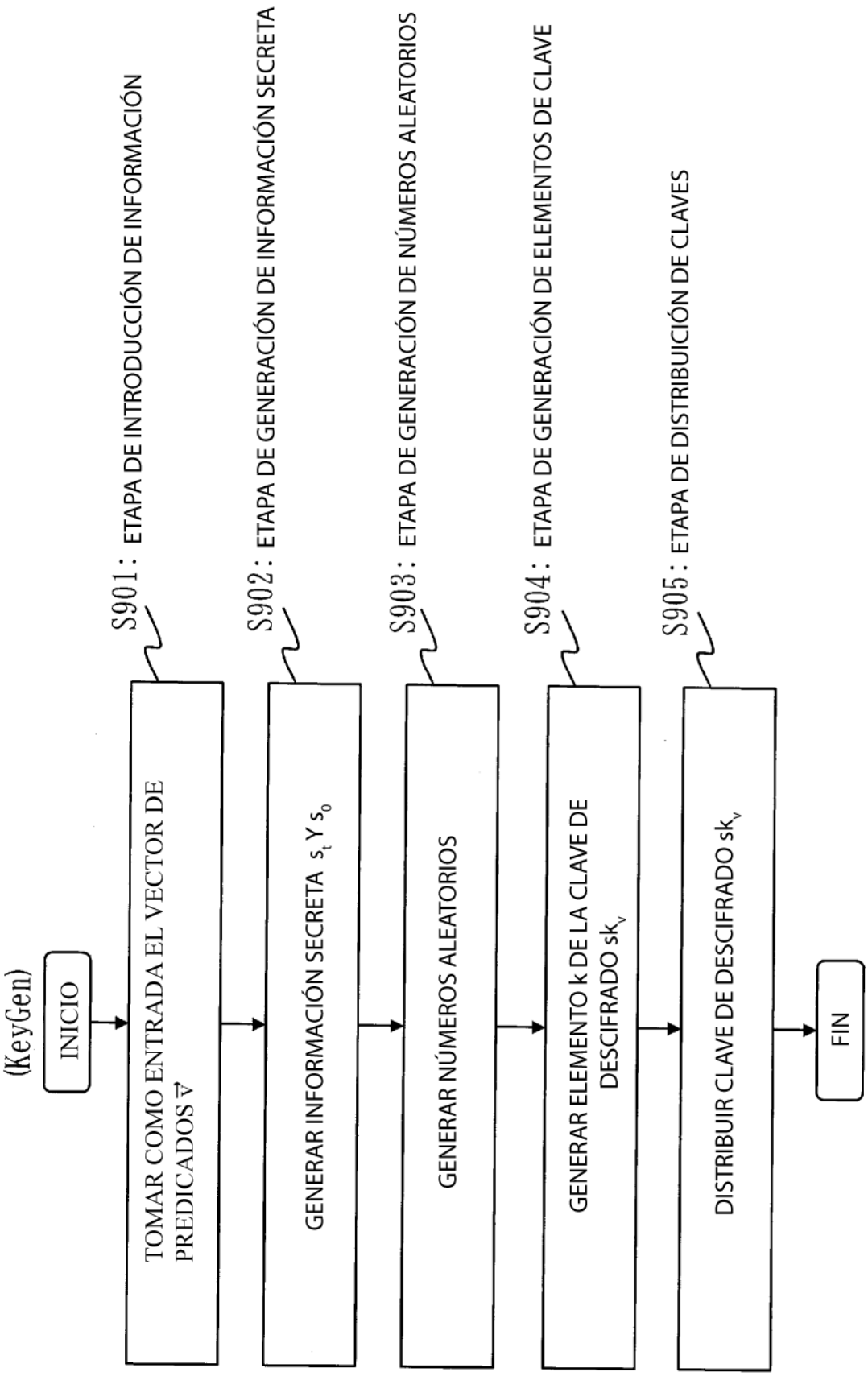

(KeyGen)
INICIO
S901: ETAPA DE INTRODUCCIÓN DE INFORMACIÓN
TOMAR COMO ENTRADA EL VECTOR DE PREDICADOS $\vec{v}$
S902: ETAPA DE GENERACIÓN DE INFORMACIÓN SECRETA
GENERAR INFORMACIÓN SECRETA s_t Y s_0
S903: ETAPA DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMEROS ALEATORIOS
S904: ETAPA DE GENERACIÓN DE ELEMENTOS DE CLAVE
GENERAR ELEMENTO k DE LA CLAVE DE DESCIFRADO sk_v
S905: ETAPA DE DISTRIBUICIÓN DE CLAVES
DISTRIBUIR CLAVE DE DESCIFRADO sk_v
FIN

Fig. 14

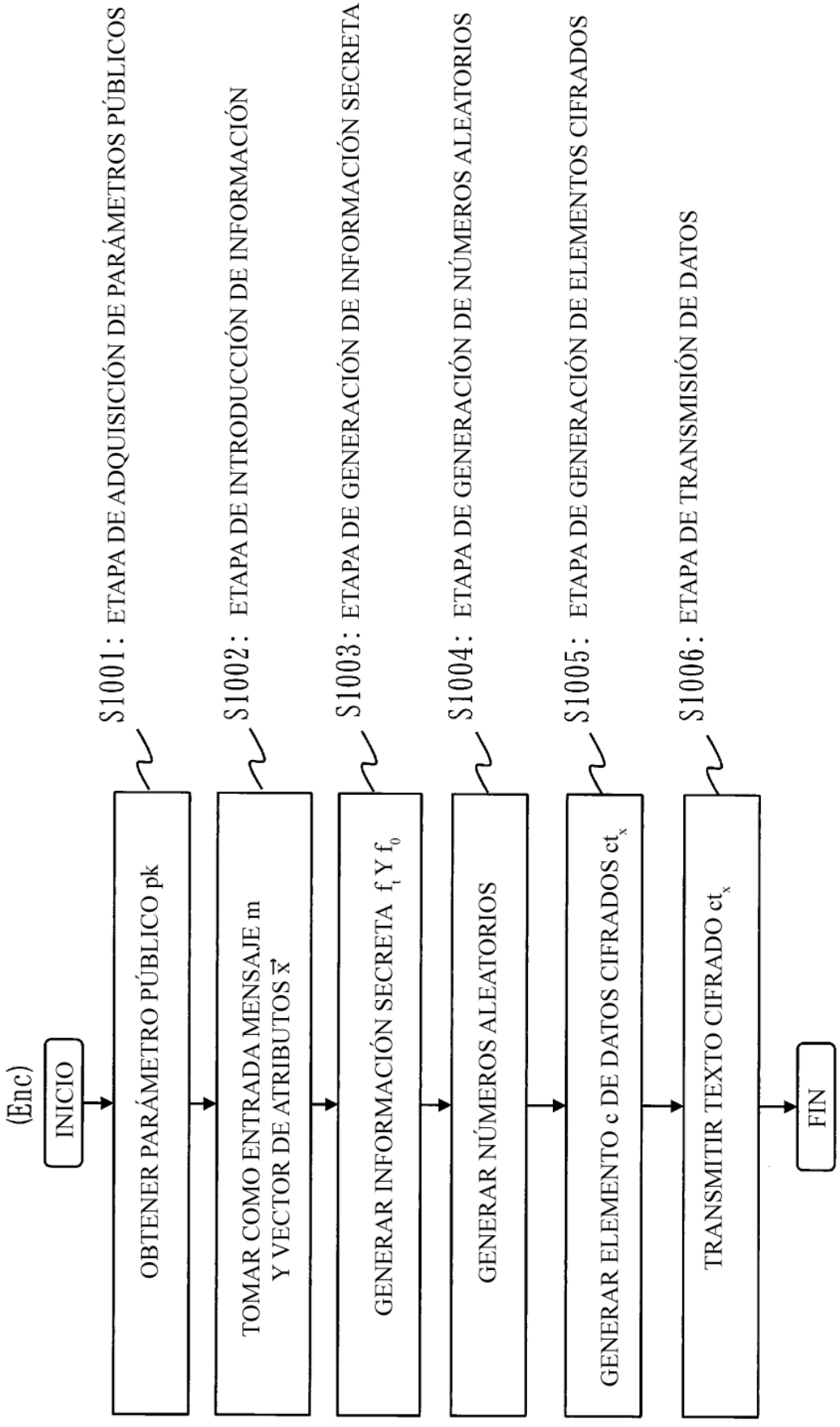

(Enc)
INICIO
S1001: ETAPA DE ADQUISICIÓN DE PARÁMETROS PÚBLICOS
OBTENER PARÁMETRO PÚBLICO pk
S1002: ETAPA DE INTRODUCCIÓN DE INFORMACIÓN
TOMAR COMO ENTRADA MENSAJE m
Y VECTOR DE ATRIBUTOS $\vec{x}$
S1003: ETAPA DE GENERACIÓN DE INFORMACIÓN SECRETA
GENERAR INFORMACIÓN SECRETA f_t Y f_0
S1004: ETAPA DE GENERACIÓN DE NÚMEROS ALEATORIOS
GENERAR NÚMEROS ALEATORIOS
S1005: ETAPA DE GENERACIÓN DE ELEMENTOS CIFRADOS
GENERAR ELEMENTO c DE DATOS CIFRADOS ct_x
S1006: ETAPA DE TRANSMISIÓN DE DATOS
TRANSMITIR TEXTO CIFRADO ct_x
FIN

Fig. 15

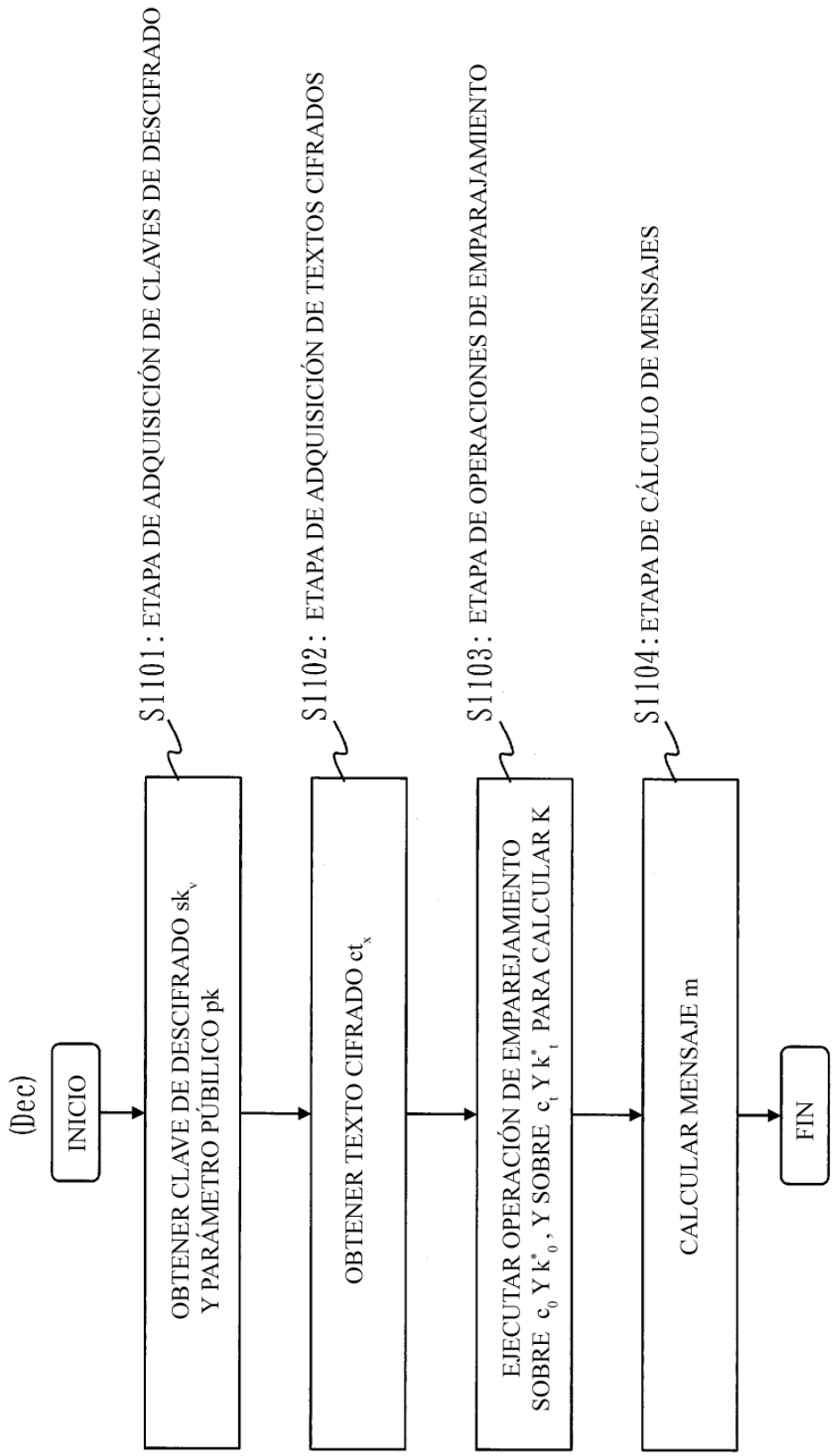
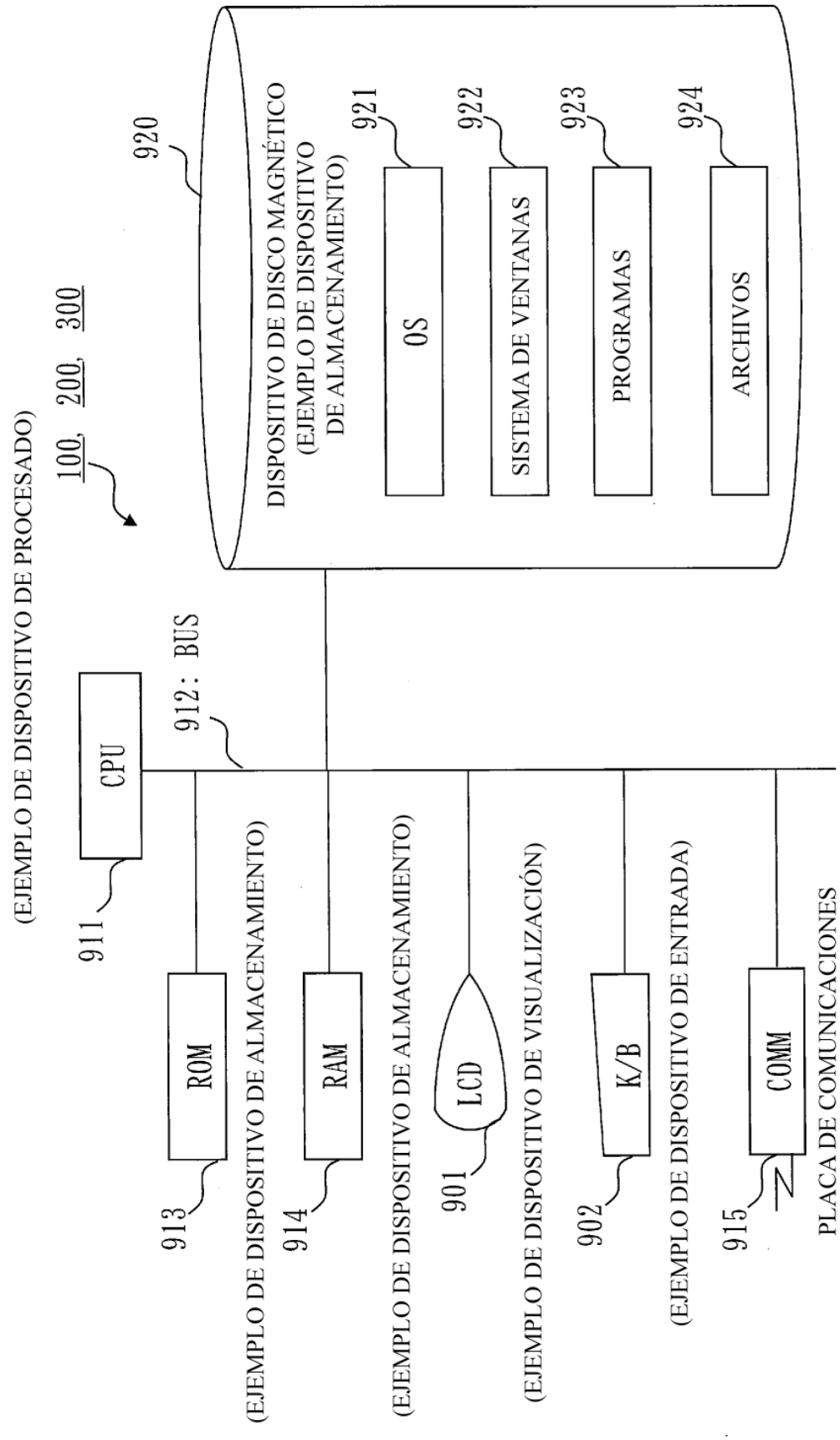

(Dec)
INICIO
S1101: ETAPA DE ADQUISICIÓN DE CLAVES DE DESCIFRADO
OBTENER CLAVE DE DESCIFRADO sk_v Y PARÁMETRO PÚBLICO pk
S1102: ETAPA DE ADQUISICIÓN DE TEXTOS CIFRADOS
OBTENER TEXTO CIFRADO ct_x
S1103: ETAPA DE OPERACIONES DE EMPARAJAMIENTO
EJECUTAR OPERACIÓN DE EMPAREJAMIENTO SOBRE c_0 Y k^*_0, Y SOBRE c_t Y k^*_t PARA CALCULAR K
S1104: ETAPA DE CÁLCULO DE MENSAJES
CALCULAR MENSAJE m
FIN

Fig. 16


(EJEMPLO DE DISPOSITIVO DE PROCESADO)
100, 200, 300
911
CPU
912: BUS
920
DISPOSITIVO DE DISCO MAGNÉTICO
(EJEMPLO DE DISPOSITIVO
DE ALMACENAMIENTO)
921
OS
922
SISTEMA DE VENTANAS
923
PROGRAMAS
924
ARCHIVOS
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
COMM
PLACA DE COMUNICACIONES