(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 626 533**

(51) Int. CI.:

G06F 21/44 (2013.01)
G06F 21/60 (2013.01)
H04L 9/00 (2006.01)
H04L 29/06 (2006.01)
H04W 4/02 (2009.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **18.11.2014 PCT/US2014/066023**

(87) Fecha y número de publicación internacional: **04.06.2015 WO15080896**

(96) Fecha de presentación y número de la solicitud europea: **18.11.2014 E 14810066 (2)**

(97) Fecha y número de publicación de la concesión europea: **01.03.2017 EP 3075098**

(54) Título: **Intersección privada de conjuntos (PSI) asistida por servidor con transferencia de datos**

(30) Prioridad:

27.11.2013 US 201314091810

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
25.07.2017

(73) Titular/es:

MICROSOFT TECHNOLOGY LICENSING, LLC (100.0%)
One Microsoft Way
Redmond, WA 98052, US

(72) Inventor/es:

KAMARA, SENY

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Intersección privada de conjuntos (PSI) asistida por servidor con transferencia de datos

Antecedentes

La intersección privada de conjuntos (PSI) permite a dos partes encontrar la intersección de sus conjuntos sin revelar los elementos de los datos de sus conjuntos una a otra. PSI tiene numerosas aplicaciones en el mundo real incluyendo extracción de datos de preservación de privacidad, servicios basados en localización, y cálculos genómicos. Más específicamente, un protocolo PSI permite a dos partes P_1 y P_2 encontrar una intersección de dos conjuntos S_1 y S_2 de algún universo U sin tener que revelar los conjuntos una a otra. En otras palabras, con un protocolo PSI P_1 y P_2 pueden encontrar la intersección $I = S_1 \cap S_2$ de sus conjuntos sin averiguar ninguna información acerca del conjunto de la otra parte más allá de su tamaño.

Con un protocolo PSI asistida por servidor, las dos partes P_1 y P_2 pueden, además, externalizar algunos de sus cálculos a un servidor no de confianza – ejemplificado, por ejemplo, en la nube. Los protocolos PSI asistida por servidor son más eficientes para clientes que los protocolos PSI tradicionales en varios órdenes de magnitud.

El artículo “Outsourcing Multi-Party Computation” de S. Kamara et al., publicado el 25 de octubre de 2011, describe un estudio de cálculo seguro de múltiples partes en un ajuste asistido por servidor, y un protocolo asistido por servidor para intersección privada de conjuntos.

El articulo “Design and implementation of privacy-preserving reconciliation protocols” de G. Neugebauer et al., publicado el 18 de marzo de 2013, describe dos protocolos para reconciliación de preservación de privacidad, y su generalización.

El artículo “Fair Private Set Intersection with a Semi-trusted Arbiter” de D. Changyu et al., publicado el 15 de julio de 2013, describe un protocolo de intersección privada de conjuntos con un árbitro fuera de línea, de semi confianza.

Compendio

Este Compendio se proporciona para introducir una selección de conceptos de una forma simplificada que se describen además a continuación en la Descripción Detallada. Este Compendio no se pretende que identifique las características clave o las características esenciales de la materia objeto reivindicada, ni se pretende que sea usado para limitar el alcance de la materia objeto reivindicada.

La técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria proporciona un protocolo de intersección privada de conjuntos (PSI) asistida por servidor que soporta transferencias de datos. El protocolo PSI de la técnica puede operar en un ajuste asistido por servidor, donde las partes tienen acceso a un servidor no de confianza que pone a disposición sus recursos de cálculo como un servicio. En una realización, el protocolo requiere solamente un número lineal de invocaciones de cifrado de bloque (una permutación pseudoaleatoria), y la ejecución de un algoritmo de intersección de conjuntos estándar/de texto plano.

La técnica permite que dos clientes transfieran información acerca de alguno de sus elementos de datos sobre una red a través de un servidor. Para este fin, en una realización de la técnica, un primer cliente genera para cada elemento en el conjunto del primer cliente S_1: una etiqueta para el elemento, un identificador para el elemento, y una forma cifrada para la para los datos asociados con el elemento que se cifró usando una clave secreta de dos partes. El primer cliente envía a un servidor para cada elemento en el conjunto S_1, la etiqueta para cada elemento, el identificador para cada elemento, y la primera parte de la clave secreta de dos partes. El primer cliente también envía al segundo cliente, para cada elemento en el conjunto S_1, el identificador para cada elemento, los datos cifrados asociados con cada elemento y la segunda parte de la clave secreta de dos partes. Similar al primer cliente, el segundo cliente envía etiquetas generadas por los elementos del conjunto del segundo cliente S_2, al servidor. El servidor calcula la intersección de las etiquetas de los conjuntos recibidos, y envía a una segunda parte, para cada elemento en la intersección, la etiqueta, el identificador para el elemento y la primera parte de la clave de dos partes. El segundo cliente entonces puede descifrar los datos asociados con cada elemento en la intersección de los conjuntos usando las etiquetas, los identificadores y la primera y segunda partes de la clave secreta de dos partes. En virtud a esta técnica, ninguna parte de la transacción, ni los clientes ni el servidor, descubren ningún dato de los clientes que no deseen revelar.

Breve descripción de los dibujos

Las características, aspectos, y ventajas específicos de la descripción llegarán a ser entendidos mejor con respecto a la siguiente descripción, las reivindicaciones adjuntas, y los dibujos anexos en los que:

La FIG. 1 representa un diagrama de flujo de un protocolo de intersección privada de conjuntos asistida por servidor en la que no hay transferencia de datos entre las partes.

La FIG. 2 representa un diagrama de flujo de una realización ejemplar del protocolo de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria.

La FIG. 3 representa un diagrama de flujo de otra realización ejemplar de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria.

La FIG. 4 representa un diagrama de flujo del procesamiento del primer cliente de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la FIG. 3.

La FIG. 5 representa un diagrama de flujo del procesamiento del segundo cliente de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la FIG. 3.

La FIG. 6 representa un diagrama de flujo del procesamiento de de la tercera parte/del servidor de la técnica de la intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la FIG. 3.

La FIG. 7 representa una arquitectura para implementar una realización ejemplar de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria.

La FIG. 8 representa un diagrama de flujo de datos para implementar una realización ejemplar de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria.

La FIG. 9 es un esquema de un entorno informático ejemplar que se puede usarse para poner en práctica la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos.

Descripción detallada

En la siguiente descripción de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos, se hace referencia a los dibujos anexos, que forman una parte de la misma, y que muestran a modo de ilustración ejemplos mediante los cuales se puede poner en práctica la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria. Se tiene que entender que se pueden utilizar otras realizaciones y se pueden hacer cambios estructurales sin apartarse del alcance de la materia objeto reivindicada.

1.0 Técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos

Las siguientes secciones proporcionan una introducción, una visión de conjunto de cálculo de múltiples partes, una visión de conjunto de la notación usada en esta descripción, una descripción de la intersección privada de conjuntos asistida por servidor sin transferencia de datos, así como realizaciones ejemplares de técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria. También se describen un diagrama de flujo de datos ejemplar y escenarios ejemplares para poner en práctica la técnica.

Como materia preliminar, algunas de las figuras que siguen describen conceptos en el contexto de uno o más componentes estructurales, denominados de diversas maneras funcionalidad, módulos, características, elementos, etc. Los diversos componentes mostrados en las figuras se pueden implementar de cualquier manera. En un caso, la separación ilustrada de diversos componentes en las figuras en distintas unidades puede reflejar el uso de componentes distintos correspondientes en una implementación real. Alternativa o adicionalmente, cualquier componente único ilustrado en las figuras se puede implementar mediante componentes plurales reales. Alternativa o adicionalmente, la representación de cualesquiera dos o más componentes separados en las figuras puede reflejar diferentes funciones realizadas por un componente único real.

Otras figuras describen los conceptos en forma de diagrama de flujo. De esta forma, ciertas operaciones se describen como que constituyen bloques distintos realizados en un cierto orden. Tales implementaciones son ilustrativas y no limitativas. Ciertos bloques descritos en la presente memoria se pueden agrupar juntos y realizar en una operación única, ciertos bloques se pueden separar en bloques de componentes plurales, y ciertos bloques se pueden realizar en un orden que difiere del que se ilustra en la presente memoria (incluyendo una manera paralela de realización de los bloques). Los bloques mostrados en los diagramas de flujo se pueden implementar de cualquier manera.

1.1 Introducción

En el problema de la intersección privada de conjuntos (PSI), dos partes quieren averiguar la intersección de sus conjuntos sin revelar una a otra cualquier información acerca de sus conjuntos más allá de la intersección. PSI es un problema fundamental de seguridad y privacidad que aparece en muchos contextos diferentes. Consideremos, por ejemplo, el caso de dos o más instituciones que desean obtener una lista de clientes comunes con propósitos de extracción de datos; o una agencia gubernamental que quiere averiguar si cualquiera en su lista de exclusión aérea está en una lista de pasajeros del vuelo. PSI ha encontrado aplicaciones en una amplia gama de ajustes tales como cálculo genómico, servicios basados en localización, y detección colaborativa de red infectada.

1.1.1 Cálculo seguro de múltiples partes

PSI es un caso especial del problema más general de cálculo de múltiples partes (MPC) segura. En este problema, cada parte mantiene su propia entrada privada y la meta es calcular colectivamente una función de unión de las entradas de los participantes sin fuga de información adicional y al tiempo que se garantiza la exactitud de la salida. El diseño y la implementación de protocolos MPC prácticos ha sido un área activa de investigación durante la última década con numerosos esfuerzos para mejorar y optimizar las implementaciones de software y para desarrollar nuevos marcos de referencia. Una gran cantidad de trabajo, por lo tanto, se ha centrado en el diseño y la implementación de protocolos PSI de propósito especial eficientes.

1.1.2 Limitaciones de MPC

Como continúa la tendencia hacia más y más grandes bases de datos, gobiernos y organizaciones privadas a menudo gestionan bases de datos masivas que almacenan miles de millones de registros. Por lo tanto, para que cualquier solución PSI (y MPC en general) sea de interés práctico en tales ajustes, se necesitan procesar eficientemente conjuntos con decenas o cientos de millones de registros.

1.1.3 MPC asistido por servidor

Un planteamiento prometedor para abordar el cálculo de múltiples partes es MPC asistido por servidor o asistido por la nube. En esta variante de MPC, el ajuste estándar se aumenta con un pequeño conjunto de servidores que no tienen entradas al cálculo y que no reciben salida pero que ponen a disposición sus recursos de cálculo a las partes.

1.1.4 Preliminares y notación

Esta sección proporciona diversos preliminares y notación que son útiles en la comprensión de la siguiente descripción de la técnica de intersección privada de conjuntos asistida por servidor descrita en la presente memoria.

A lo largo de esta descripción, partes para un protocolo descrito que no son el servidor o terceras partes no de confianza se pueden denominar clientes. Los términos la tercera parte, la tercera parte no de confianza y el servidor también se pueden usar intercambiablemente.

Los protocolos descritos en la presente memoria son protocolos de una sola vuelta y tienen aproximadamente la siguiente estructura. Primero los clientes procesan sus conjuntos de entrada para generar un conjunto de etiquetas *T*, las cuales enviarán al servidor o a la tercera parte. El servidor entonces realiza una intersección sobre los conjuntos que recibe y devuelve los resultados. Para los protocolos seguros contra un servidor malicioso o no de confianza, los clientes entonces realizan algunas comprobaciones locales y extraen la intersección del mensaje del servidor. Con el propósito de esta descripción se usa una noción de no confabulación simplificada en donde dos partes P_1 y P_2 se consideran que no confabulan si no se dañan simultáneamente por la adversaria (por ejemplo, o bien P_1 es maliciosa o bien o P_2, pero no ambas).

En algunas realizaciones de la técnica, el servidor se trata de una manera conservativa como indigno de confianza (lo que significa, por ejemplo, que no se puede confiar en el servidor para mantener la confidencialidad de la información proporcionada al servidor). No obstante, en algunos escenarios, se supondrá que el servidor no se confabula con ningún modulo participante para eludir las provisiones de seguridad descritas en la presente memoria. Además, en algunos escenarios, se supondrá que las partes para el cálculo conjunto son entidades semi honestas en lo peor de los casos. Esto significa que se puede esperar de las entidades que sigan un protocolo de seguridad prescrito. Pero las entidades pueden tratar de aprovechar la información que descubren en el curso de este protocolo para destapar información adicional (a la cual no tienen derecho).

Un esquema de cifrado con clave privada, como el descrito en la presente memoria, es un conjunto de tres algoritmos de polinomio-tiempo (Gen, Enc, Dec) que funcionan como sigue. Gen es un algoritmo probabilístico que toma un parámetro de seguridad *k* en unario y devuelve una clave secreta *K*. Enc es un algoritmo probabilístico que toma una clave *K* y un mensaje *m* de *n* bits y devuelve el texto cifrado *C*. Dec es un algoritmo determinístico que toma la clave *K* y un texto cifrado *c* y devuelve *m* si *K* era la clave bajo la cual se produjo *c*.

En todos los protocolos descritos a continuación, *k* denota el parámetro de seguridad de cálculo (es decir, la longitud de clave para una Permutación Pseudoaleatoria (PRP)) mientras *s* denota un parámetro de seguridad estadístico. Para $\lambda \geq 1$, el conjunto S^{λ} se define como

$$S^{\lambda} = \{x \,||\, 1, \ldots, x \,||\, \lambda : x \in S\}$$

y $(S^{\lambda})^{\lambda} = S$. Si $F: U \rightarrow V$ es una función, la evaluación S de *F* es el conjunto $F(S)=\{F(s): s \in S\}$. F^{-1} se denota como la inversa de *F* donde $F^{-1}(F(S)) = S$. Si π: [ISI] → [ISI] es una permutación, entonces el set $\pi(S)$ es el conjunto que resulta de permutar los elementos de S según π (suponiendo una ordenación natural de los elementos). En otras palabras:

$$\pi(S) = \{x_{\pi(i)}: x_i \in S\}.$$

La unión y diferencia establecidas de dos conjuntos S_1 y S_2 se denota como $S_1 + S_2$ y $S_1 - S_2$, respectivamente.

1.2 PSI asistida por servidor semi honesto sin transferencia de datos

A modo de antecedentes, y antes de describir la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos, se describe primero un protocolo de intersección privada de conjuntos asistida por servidor sin transferencia de datos para un servidor semi honesto o cualquier confabulación de las partes maliciosas. El protocolo se muestra en la Tabla 1 y se describe en la FIG. 1.

Con referencia a la FIG. 1, permitamos que S_i sea el conjunto de la parte P_i. Las partes comienzan generando conjuntamente una clave secreta K de k bits para una Permutación Pseudoaleatoria (PRP) F, como se muestra en el bloque 102. Cada parte permuta aleatoriamente el conjunto $F_K(S_i)$ que consiste en etiquetas calculadas evaluando la PRP sobre los elementos del conjunto adecuado de la parte (como se muestra en el bloque 104), y envía el conjunto permutado al servidor (como se muestra en el bloque 106). El servidor entonces simplemente calcula y devuelve la intersección de las etiquetas $F_K(S_1)$ hasta $F_K(S_n)$ (véanse los bloques 108, 110). Cada parte puede conocer entonces qué elementos tienen en común con las otras partes.

Intuitivamente, la seguridad del protocolo sigue a partir del hecho de que las partes nunca reciben ningún mensaje una de otra, y su único comportamiento malicioso posible es cambiar sus propias etiquetas PRP lo cual simplemente se traduce en cambiar su conjunto de entrada. El servidor semi honesto solamente recibe etiquetas que debido a la pseudoaleatoriedad de la PRP no revela información acerca de los elementos del conjunto.

Configuración y entradas: Permitamos que F: $\{0,1\}^K \times U \rightarrow \{0, 1\}^{\geq K}$ sea una PRP. Cada parte P_i tiene un conjunto $S_i \subseteq U$ como entrada mientras que el servidor no tiene entrada:

1. P_1 muestrea una clave K de k bits aleatoria y la envía a P_i para $i \in [2,n]$;

2. Cada parte P_i para $i \in [n]$ envía $T_i = \pi_i(F_K(S_i))$ al servidor, donde π_i es una permutación aleatoria;

3. El servidor calcula $I = \cap^n_{i=1}{}^n T_i$ y la devuelve a todas las partes;

4. Cada parte P_i saca $F_K^{-1}(I)$.

Tabla 1. Un protocolo PSI asistida por servidor con un servidor semi honesto

El protocolo descrito en la Tabla 1 es seguro en presencia (1) de un servidor semi honesto y partes honestas o (2) un servidor honesto y cualquier confabulación de las partes maliciosas.

Cada P_i invoca la PRP, $|S_i|$ veces, mientras el servidor (en una realización) solamente realiza una intersección de conjuntos de “texto plano” y no operaciones criptográficas. No obstante, la técnica puede usar cualquiera de los algoritmos existentes para intersección de conjuntos. En una realización de la técnica se usa una inserción/búsqueda de tabla de generación de claves de folclore que funciona casi en tiempo lineal.

El protocolo se puede ejecutar asincrónicamente donde cada parte conecta en un tiempo diferente para presentar su mensaje al servidor y más tarde para obtener la salida.

1.3 PSI semi honesto con transferencia de datos

En muchas aplicaciones prácticas de PSI, una de las partes también desea transferir alguna información relacionada con los elementos en la intersección. Más precisamente, consideremos un escenario donde P_1 tiene, además de su conjunto S_1, una base de datos db que asocia con cada elemento x de S_1 algunos datos que se denotan como db(x). En tal escenario, P_1 puede desear transferir el conjunto $\{db(x): x \in S_1 \cap S_2\}$ a P_2, donde S_2 es el conjunto de P_2. Mientras el protocolo PSI descrito anteriormente no es suficiente para este escenario, se puede usar como un bloque de construcción en un protocolo más complejo para lograr tal transferencia de datos.

La FIG. 2 describe un proceso implementado por ordenador 200 para crear un protocolo para PSI asistida por servidor con transferencia de datos según la técnica. El proceso transfiere alguna información asociada con los elementos de datos en la intersección de dos conjuntos de datos mantenidos por dos partes sin revelar su conjunto de datos.

Como se muestra en el bloque 202, una primera parte genera un conjunto de triples para cada elemento en el conjunto S_1 que comprende: una etiqueta para el elemento, un identificador para el elemento, y una forma cifrada para los datos asociados con el elemento que se cifró usando una clave secreta de dos partes. Las etiquetas para cada elemento en S_1 se generan usando una clave compartida y una PRP.

Como se muestra en el bloque 204, la primera parte envía a la segunda parte, para cada elemento en el conjunto S_1, el identificador para cada elemento, los datos cifrados asociados con cada elemento y la segunda parte de la clave

secreta de dos partes. Adicionalmente, la primera parte envía a una tercera parte, para cada elemento en el conjunto S_1, la etiqueta para cada elemento, el identificador para los datos asociados con cada elemento, y la primera parte de la clave secreta de dos partes, como se muestra en el bloque 206. La segunda parte también envía etiquetas generadas para los elementos del conjunto S_2 a la tercera parte (bloque 208). Las etiquetas para los elementos del conjunto S_2 se generan usando la misma clave compartida y PRP que usó la primera parte.

Una vez que la tercera parte recibe los datos antes mencionados desde la primera y segunda partes, la tercera parte calcula la intersección de las etiquetas de los conjuntos y envía a la segunda parte, para cada elemento en la intersección, la etiqueta, el identificador asociado con el elemento y la primera parte de la clave secreta de dos partes, como se muestra en el bloque 210.

La segunda parte puede entonces descifrar los datos asociados con cada elemento en la intersección de los conjuntos usando las etiquetas, los identificadores y la primera y segunda parte de la clave secreta de dos partes, como se muestra en el bloque 212. Esto se puede hacer aplicando una función XOR a las partes para recuperar la clave y entonces usar la clave para descifrar los datos.

La Tabla 2 representa un protocolo PSI de dos partes con transferencia de datos que es seguro contra un servidor semi honesto y una P_1 y P_2 semi honestas. El protocolo se describe en mayor detalle con respecto a la FIG. 3.

Configuración y entradas: Permitamos que F: $\{0,1\}^K$ x U $\rightarrow$ $\{0, 1\}^{\geq K}$ sea una PRP. La parte P_1 tiene conjuntos $S_1 \subset U$ y db $\subseteq \{0,1\}^*$ como entradas y P_2 tiene un conjunto $S_2 \subseteq U$ como entrada. El servidor no tiene ninguna entrada.

1. P_1 muestrea tres claves de k bits K_e, K_l, K_1 y envía K_1 a P_2;

2. Para todo $x \in S_1$, P_1 calcula:

   (a) $z_{x,1} := F_{Ke}(x \,||\, 1)$ y $z_{x,2} := F_{Ke}(x \,||\, 2)$

   (b) $K_x := z_{x,1} \oplus z_{x,2}$

   (c) $c_x \leftarrow Enc_{Kx}(db(x))$

   (d) $id_x := F_{K1}(x)$

   (e) $l_x := F_{Kl}(x)$.

3. P_1 envía

$T_1 = \pi_1(\{(id_x, z_{x,1}, l_x)\})_{x \in S1})$

al servidor y

$D = \pi_1,(\{(id_x, z_{x,2}, c_x)\})_{x \in S1})$

a P_2, donde π_1 y π_1, son permutaciones aleatorias;

4. P_2 envía $T_2 = \pi_2(F_{Kl}(S_2))$ al servidor, donde π_2 es una permutación aleatoria;

5. El servidor envía a P_2

$I = \{(id, z_1) : (id, z_1, l) \in T_1 \wedge l \in T_2\}$;

6. P_2 saca

$P = \{Dec_{z1 \oplus z2}(c): (id, z_2, c) \in D \wedge (id, z_1) \in I\}$

Tabla 2. Un protocolo PSI de dos partes con transferencia de datos que es seguro contra partes semi honestas.

La FIG. 3 representa otro proceso 300 implementado por ordenador para transferir alguna información db(x) asociada con los elementos x_l de datos en la intersección de dos conjuntos de datos S_1 y S_2 mantenidos por dos partes P_1 y P_2 usando una tercera parte, por ejemplo, una tercera parte no de confianza, sin ninguna de las dos partes P_1 o P_2 que revela los datos en sus conjuntos. El proceso implementado por ordenador generalmente se refiere al protocolo mostrado en Tabla 2.

Como se muestra en el bloque 302, la primera parte P_1 genera una primera clave secreta K_e, una segunda clave secreta K_l, y una tercera clave secreta K_1; y envía la segunda clave secreta K_l a la segunda parte P_2. La clave K_e se usa para generar claves de cifrado para datos db(x) asociados con cada elemento del conjunto S_1, mientras que la clave K_l se usa para generar una etiqueta para un elemento x de un conjunto. La clave K_1 se usa para generar un identificador id_x para cada elemento x de un conjunto.

Como se muestra en el bloque 304, para cada elemento en el conjunto S_1 de P_1:

(1) Usando K_e, se genera una clave secreta de dos partes K_x. Esta clave secreta de dos partes K_x se compone de una primera parte $z_{x,1}$ y una segunda parte $z_{x,2}$. En una realización de la técnica esto se hace generando la primera parte $z_{x,1}$ aplicando una permutación pseudoaleatoria al elemento x, concatenada con una primera cadena usando la clave K_e. Del mismo modo la segunda parte $z_{x,2}$ se genera aplicando una permutación pseudoaleatoria al elemento *x*, concatenada con una segunda cadena usando la clave K_e. La clave K_x se genera entonces realizando una operación XOR en la primera parte $z_{x,1}$ y una segunda parte $z_{x,2}$. Es importante señalar que se generan una primera parte $z_{x,1}$ y una segunda parte $z_{x,2}$ de una clave para cada elemento de S_1. Además, también se podrían usar otros procedimientos para generar una clave de dos partes o incluso de múltiples partes.

(2) Los datos db(x) para el elemento se cifran usando la clave K_x para crear los datos cifrados C(db(x));

(3) Un identificador id_x para el elemento se crea aplicando una PRP al elemento usando la clave K_1; y

(4) La etiqueta l_x para el elemento se crea aplicando la clave secreta K_l y una PRP al elemento.

Como se muestra en el bloque 306, la primera parte P_1 envía a la tercera parte un conjunto de triples para cada elemento que comprende el identificador id_x, la primera parte de la clave $z_{x,1}$ y la etiqueta l_x. El orden de los triples se puede mezclar aleatoriamente anterior a enviarlos. La primera parte P_1 también envía a P_2 un conjunto de triples para cada elemento que comprende el identificador id_x, la segunda parte de la clave $z_{x,1}$ y los datos cifrados C(db(x)), como se muestra en el bloque 308. Este conjunto de triples también se puede mezclar aleatoriamente anterior a enviarlos a P_2.

La segunda parte P_2 genera etiquetas para los elementos en el conjunto S_2 de P_2 permutando los elementos usando la clave K_l y una PRP, y envía las etiquetas para cada elemento en el conjunto S_2 a la tercera parte no de confianza (véase el bloque 310).

La tercera parte compara el conjunto de etiquetas recibidas desde P_1 y el conjunto de etiquetas recibido desde P_2 para encontrar la intersección *I* de las etiquetas permutadas de P_1 y P_2, y envía a P_2 todos los triples que recibe desde P_1 que tienen una etiqueta dentro de la intersección (bloque 312). P_2 puede descifrar entonces los triples que P_2 recibió de P_1 que comprenden datos cifrados de P_1 C(db(x)) para los elementos que P_1 y P_2 mantienen en la intersección de los conjuntos S_1 y S_2 usando ambas partes de la clave secreta de dos partes (block 312).

Cada P_i invoca la PRP, $|S_i|$ veces, mientras el servidor (en una realización) solamente realiza una intersección de conjuntos de “texto plano” y no operaciones criptográficas. No obstante, la técnica puede usar cualquiera de los algoritmos existentes para la intersección de conjuntos. En una realización de la técnica se usa una inserción/búsqueda de tabla de generación de claves de folclore que funciona en tiempo casi lineal.

El protocolo se puede ejecutar asincrónicamente donde cada parte conecta en un tiempo diferente para presentar su mensaje al servidor y más tarde para obtener la salida.

1.4 Procesamiento servidor-cliente

Como se mencionó anteriormente, aunque las descripciones de los protocolos se refieren a las partes que mantienen conjuntos de datos (por ejemplo, una primera y segunda partes por ejemplo) y una tercera parte que calcula la intersección de los conjuntos de datos, los expertos en la técnica comprenderán que las partes mencionadas pueden operar en una configuración servidor-cliente. Las partes para un protocolo descrito que no se referencian como el servidor o la tercera parte no de confianza se pueden denominar por lo tanto clientes. Además, los términos la tercera parte, la tercera parte no de confianza y el servidor también se pueden usar intercambiablemente. La siguiente descripción describe el protocolo descrito en FIG. 3 en términos de procesamiento del lado de cliente y de servidor. Se debería señalar que el servidor puede ser realmente más de un servidor o entidad informática trabajando en una nube informática. Del mismo modo, aunque la descripción se refiere a un primer y segundo clientes, muchos más clientes pueden transferir datos acerca de sus conjuntos usando el servidor.

1.4.1 Procesamiento del lado de cliente

La FIG. 4 representa un procesador implementado por ordenador 400 que muestra el procesamiento del primer cliente con respecto al protocolo descrito en la FIG. 3. Las operaciones principales durante el paso de procesamiento de cliente son la aplicación de una PRP para generar etiquetas y la aplicación de una permutación aleatoria para mezclar aleatoriamente los datos antes de su transferencia. Hay muchas posibilidades para aplicar una PRP para generar las etiquetas. Por ejemplo, en una realización de la técnica las permutaciones aleatorias se ejemplificaron usando el Estándar de Cifrado Avanzado (AES) (por ejemplo, el modo contador (CTR), por ejemplo, si los elementos son mayores de 128 bits). Se puede usar cualquier otro cifrado de bloque, no obstante.

Con referencia a FIG. 4, como se muestra en el bloque 402, el primer cliente P_1 genera una primera clave secreta K_e, una segunda clave secreta K_l, y una tercera clave secreta K_1; y envía la segunda clave secreta K_l al segundo cliente P_2.

Como se muestra en el bloque 404, para cada elemento en el conjunto S_1 de P_1:

(1) usando K_e, se genera una clave secreta de dos partes K_x. Esta clave secreta de dos partes K_x comprende una primera parte $z_{x,1}$ y una segunda parte $z_{x,2}$.

(2) Los datos db(x) para el elemento se cifran usando la clave K_x para crear los datos cifrados C(db(x));

(3) Se crea un identificador id_x para los datos db(x) asociados con el elemento; y

(4) Una etiqueta l_x para el elemento se crea aplicando la clave secreta K_l y una PRP aleatoria al elemento.

Como se muestra en el bloque 406, el primer cliente P_1 envía al servidor un conjunto de triples para cada elemento que comprende el identificador id_x, la primera parte de la clave $z_{x,1}$ y la etiqueta l_x. El primer cliente P_1 envía a P_2 un conjunto de triples para cada elemento que comprende el identificador id_x, la segunda parte de la clave $z_{x,1}$ y los datos cifrados C(db(x)), como se muestra en el bloque 408. El conjunto de triples en ambos de estos ejemplos se mezclar aleatoriamente anterior a enviar.

La FIG. 5 proporciona un diagrama de flujo de un procesador implementado por ordenador 500 que representa solamente el procesamiento del segundo cliente del protocolo tratado en la FIG. 3. Como se muestra en el bloque 502, el segundo cliente P_2 recibe la clave secreta K_l. El cliente P_2 también recibe un conjunto de triples para cada elemento que comprende el identificador id_x, la segunda parte de la clave $z_{x,1}$ y los datos cifrados C(db(x)), como se muestra en el bloque 504. El segundo cliente P_2 genera etiquetas para los elementos en el conjunto S_2 de P_2 permutando los elementos usando la clave K_l y una PRP, y envía las etiquetas para cada elemento en el conjunto S_2 al servidor, como se muestra en el bloque 506. P_2 recibe todos los triples de P_1 que tienen una etiqueta dentro de la intersección (bloque 508). P_2 entonces puede descifrar los triples que P_2 recibió desde P_1 que comprenden los datos cifrados de P_1 C(db(x)) por los elementos que P_1 y P_2 mantienen en la intersección de los conjuntos S_1 y S_2 usando ambas partes de la clave secreta de dos partes (bloque 510).

1.4.2 Procesamiento del lado de servidor

La FIG. 6 proporciona un diagrama de flujo que representa solamente el procesamiento de servidor (por ejemplo, una tercera parte) 600 mostrado en la FIG. 3. Una de las funciones del servidor es calcular la intersección de los conjuntos de las partes usando las etiquetas. En una realización de la técnica, se realiza una intersección de texto simple en las etiquetas con el fin de determinar qué elementos de datos tienen en común los clientes. Esto no implica operaciones criptográficas en la parte del servidor (por ejemplo, una tercera parte no de confianza). En otra realización de la técnica, el planteamiento de comparación trivial por pares para calcular la intersección de conjuntos tiene una complejidad cuadrática y no escala a grandes conjuntos. Por lo tanto, en una realización de la técnica, se implementó un algoritmo de intersección de conjuntos de folclore basado en tablas de generación de claves. En esta realización, el servidor genera claves de los elementos del primer conjunto en una tabla de generación de claves, y entonces intenta buscar los elementos del segundo conjunto en la misma tabla. Cualquier elemento con una búsqueda exitosa se añade a la intersección. El servidor entonces saca un vector booleano que indica qué elementos del segundo conjunto están en la intersección y cuáles no.

Con referencia de nuevo a la FIG. 6, como se muestra en el bloque 602, el servidor recibe desde el primer cliente P_1 un conjunto de triples para cada elemento que comprende el identificador id_x, la primera parte de la clave $z_{x,1}$ y la etiqueta l_x. El servidor recibe desde la segunda parte P_2 las etiquetas para cada elemento en el conjunto S_2 (véase el bloque 604).

El servidor compara el conjunto de etiquetas recibidas desde P_1 y el conjunto de etiquetas recibidas desde P_2 para encontrar la intersección I de las etiquetas permutadas de P_1 y P_2 (bloque 606) y envía a P_2 todos los triples que recibió desde P_1 que tienen una etiqueta dentro de la intersección (bloque 608). Como se trató anteriormente, la intersección de las etiquetas se puede encontrar de una variedad de formas, tales como, por ejemplo, calculando una intersección de texto plano en las etiquetas.

1.5 Arquitectura ejemplar para poner en práctica la técnica

Se han proporcionado procesos ejemplares para poner en práctica la técnica, la siguiente sección proporciona una discusión de una arquitectura ejemplar para poner en práctica la técnica.

La FIG. 7 muestra una vista general de un sistema 700 ilustrativo que incluye un módulo de servidor 702 para realizar una tarea de procesamiento en nombre de uno o más módulos cliente. En este ejemplo, la FIG. 7 muestra dos módulos cliente, esto es el módulo de cliente P_1 704 y el módulo de cliente P_2 706. No obstante, el módulo de servidor 702 puede proporcionar servicios a cualquier número de módulos participantes, incluyendo un módulo participante, o más de dos módulos participantes.

El módulo de servidor 702 puede representar cualquier tipo de funcionalidad de cálculo. En un caso, corresponde a un servidor informático que incluye funcionalidad de procesamiento, funcionalidad de entrada, funcionalidad de salida, funcionalidad de almacenamiento, etc. En un escenario, el módulo de servidor 702 puede representar un recurso de procesamiento en un sistema informático en la nube, tal como un centro de datos que proporciona un servicio informático en la nube. El módulo de servidor 702 puede representar un único recurso proporcionado en una única ubicación o un recurso distribuido que se distribuye en ubicaciones plurales. Por ejemplo, el módulo de servidor 702 puede corresponder a una única máquina física; alternativamente, el módulo de servidor 702 puede representar un módulo de servidor virtual que se corresponde con el hardware informático subyacente correspondiente de cualquier manera.

Cada módulo de cliente 704, 706 puede representar del mismo modo cualquier tipo de funcionalidad que incluye funcionalidad de procesamiento, funcionalidad de entrada, funcionalidad de salida, funcionalidad de almacenamiento, etc. En ejemplos concretos ilustrativos, cualquier módulo de cliente puede corresponder a un dispositivo informático personal estacionario, un dispositivo informático de ordenador portátil o miniordenador portátil, un dispositivo informático de asistente digital personal (PDA), un dispositivo informático de tipo lápiz, un dispositivo de teléfono móvil, una consola de juegos, un receptor multimedia digital, etc.

El módulo de servidor 702 está conectado a los módulos de cliente P_1 704 y P_2 706 a través de cualquier tipo de red 708. La red 708 puede representar cualquier tipo de mecanismo de acoplamiento punto a punto o multipunto. En una implementación, la red 708 puede corresponder a una red de área extensa (por ejemplo, Internet), una red de área local, o una combinación de las mismas. La red 708 puede incluir cualquier combinación de enlaces inalámbricos, enlaces cableados, encaminadores, pasarelas, etc., que se gobiernan por cualquier protocolo o combinación de protocolos. El módulo de servidor 702 puede representar un recurso remoto o local en relación con cualquiera de los módulos participantes.

En una implementación de la técnica, usando la arquitectura 700 mostrada en la FIG. 7, cada uno de los clientes 704, 706 aplica una permutación pseudoaleatoria (PRP) compartida a los elementos de sus conjuntos de datos usando una clave secreta compartida para crear etiquetas de los elementos de su conjunto y enviar estas etiquetas al servidor 702. El primer cliente 704 genera además: un identificador para cada uno de los elementos del conjunto de la primera parte; una clave secreta de múltiples partes para cada uno de los elementos del conjunto de la primera parte (que podría ser, por ejemplo, una clave de dos partes); y datos cifrados asociados con cada uno de los elementos del conjunto del primer cliente usando la clave secreta de múltiples partes. El primer cliente envía los datos cifrados para todos los elementos en el conjunto del primer cliente a las otras partes, junto con una parte de la clave secreta de múltiples partes, y el identificador para cada elemento del conjunto del primer cliente. El primer cliente envía al servidor las partes restantes de la clave de múltiples partes y los identificadores para cada elemento. Cada cliente 704, 706 recibe para cada uno de los elementos en la intersección, las etiquetas para la intersección de los elementos de datos, los identificadores, y las partes restantes de la clave de múltiples partes desde el servidor 702. Cada cliente 704, 706 entonces puede descifrar los datos asociados cifrados para cada uno de los elementos en la intersección de los conjuntos usando las etiquetas, los identificadores y todas las partes de la clave de múltiples partes para cada elemento.

1.6 Diagrama de flujo de datos ejemplar para poner en práctica la técnica

La FIG. 8 representa un diagrama de flujo de datos 800 para transferir alguna información db(x) 802 asociada con los elementos de datos en la intersección de dos conjuntos de datos S_1 804 y S_2 806 mantenida por dos partes P_1 808 y P_2 810 usando una tercera parte 812 no de confianza, sin que cualquiera de las dos partes P_1 o P_2 revele los datos en sus conjuntos. La primera parte P_1 808 genera una primera clave secreta K_e 812, una segunda clave secreta K_l 814, y una tercera clave secreta K_1 816 y envía la segunda clave secreta K_l 814 a la segunda parte P_2 810. Para cada elemento en el conjunto S_1 de P_1: (1) usando K_e 812, se genera una clave secreta de dos partes K_x. Esta clave secreta de dos partes K_x comprende una primera parte $z_{x,1}$ 822a y una segunda parte de $z_{x,2}$, 822b; (2) los datos db(x) 802 para el elemento se cifran usando la clave K_x para crear los datos cifrados C(db(x)) 820; (3) se crea un identificador id_x 824 para el elemento; y (4) se crea una etiqueta l_x 826 para el elemento aplicando la clave secreta K_l 814 y una PRP al elemento. La primera parte P_1 808 envía a la tercera parte 812 un conjunto de triples mezclados aleatoriamente para cada elemento que comprende el identificador id_x 824, la primera parte de la clave $z_{x,1}$ 822a y la etiqueta l_x 826. La primera parte P_1 808 envía a P_2 810 un conjunto de triples para cada elemento que comprende el identificador id_x 824, la segunda parte de la clave $z_{x,1}$ 822b y los datos cifrados C(db(x)) 820. La segunda parte P_2 810 genera etiquetas 828 para los elementos en el conjunto S_2 de P_2 806 permutando los elementos usando la clave K_l 814 y una PRP, y enviando las etiquetas 828 para cada elemento en el conjunto S_2 806 a la tercera parte no de confianza 812. La tercera parte 812 compara el conjunto de etiquetas 826 recibidas desde P_1 y el conjunto de etiquetas 828 recibidas desde P_2 810 para encontrar la intersección I 832 de las etiquetas permutadas de P_1 y P_2, enviando a P_2 810 todos los triples recibidos desde P_1 808 que tienen una etiqueta dentro de la intersección I 832. P_2 810 puede descifrar entonces los triples que P_2 recibió desde P_1 808 que comprenden los datos cifrados de P_1 C(db(x)) 820 para los elementos que P_1 y P_2 mantienen en la intersección I 832 de los conjuntos S_1 y S_2.

2.0 Aplicaciones ejemplares

La técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos se puede aplicar a muchas aplicaciones del mundo real, tales como, por ejemplo, con respecto a la transferencia de datos médicos de pacientes, la verificación de que los pasajeros de un vuelo no están en una lista de exclusión aérea, o la transferencia de algunos, pero no todos, los datos de usuario para aplicaciones en línea. Estas aplicaciones se describen en breve a continuación, pero hay muchas, muchas otras aplicaciones que pueden emplear de manera rentable la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos.

2.1 Transferencia de datos médicos

Hay muchos escenarios en los que un hospital puede no desear divulgar los detalles acerca de los registros de pacientes individuales o bien a otro hospital o bien a un servidor. Por ejemplo, en el caso en el que dos hospitales hayan tratado a algunos de los mismos pacientes, puede ser deseable para un hospital transferir los datos de paciente acerca de los pacientes que ambos han tratado desde un hospital a otro, mientras que no sería deseable transferir datos acerca de todos los pacientes. En este caso, la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos se puede usar para transferir datos de paciente acerca de los pacientes que ambos hospitales hayan tratado para que cada hospital tenga un registro completo de las historias médicas de estos pacientes.

2.2 Determinación de pasajeros programados que están en una lista de exclusión aérea

También hay muchos escenarios en los que una aerolínea podría no querer divulgar los detalles acerca de los viajeros en sus vuelos a una agencia gubernamental o a otro país. Por ejemplo, una aerolínea podría no querer divulgar su lista de pasajeros entera a una agencia gubernamental o a un país extranjero para que la agencia gubernamental o el país extranjero determine si alguno de los pasajeros está en una lista de exclusión aérea. En este caso, la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos se puede usar para determinar qué pasajeros programados en un vuelo están en la lista de exclusión aérea sin revelar datos acerca de los otros pasajeros programados.

2.3 Compartición de datos de usuario para aplicaciones en línea o en la nube

También hay muchos escenarios en los que un proveedor de un servicio o aplicación en línea, por ejemplo, un servicio o una aplicación de juegos, podría querer compartir información acerca de los usuarios que el servicio/aplicación en línea tiene en común con otras aplicaciones o servicios en línea. En este caso, probablemente no querría proporcionar información acerca de todos sus usuarios. En este caso, la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos se puede usar para proporcionar información acerca de los usuarios que los dos servicios/aplicaciones en línea tienen en común, sin revelar datos acerca de otros usuarios.

3.0 Entorno de operación ejemplar:

La técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria está operativa dentro de numerosos tipos de entornos o configuraciones de sistemas informáticos de propósito general o propósito especial. La FIG. 9 ilustra un ejemplo simplificado de un sistema informático de propósito general en el que se pueden implementar diversas realizaciones y elementos de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos, como se describe en la presente memoria. Se debería señalar que cualesquiera recuadros que se representan por líneas de trazos o discontinuas en la FIG. 9 representan realizaciones alternativas del dispositivo informático simplificado, y que cualquiera de o todas estas realizaciones alternativas, como se describe a continuación, se pueden usar en combinación con otras realizaciones alternativas que se describen a lo largo de este documento.

Por ejemplo, la FIG. 9 muestra un diagrama de sistema general que muestra un dispositivo 900 informático simplificado. Tales dispositivos informáticos se pueden encontrar típicamente en dispositivos que tienen al menos alguna capacidad de cálculo mínima, incluyendo, pero no limitada a, ordenadores personales, ordenadores servidores, dispositivos informáticos de mano, ordenadores portátiles o móviles, dispositivos de comunicaciones tales como teléfonos celulares y PDA, sistemas multiprocesador, sistemas basados en microprocesador, receptores multimedia digitales, electrónica de consumo programable, PC en red, miniordenadores, ordenadores centrales, reproductores de medios de audio o vídeo, etc.

Para permitir a un dispositivo implementar la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos, el dispositivo debería tener una capacidad de cálculo y una memoria de sistema suficientes para permitir operaciones de cálculo básicas. En particular, como se ilustra por la FIG. 9, la capacidad de cálculo se ilustra de manera general por una o más unidades de procesamiento 910, y puede incluir también una o más GPU 915, cualquiera de las dos o ambas en comunicación con la memoria de sistema 920. Señalar que la(s) unidad(es) de procesamiento 910 del dispositivo informático general puede(n) ser microprocesadores especializados, tales como un DSP, un VLIW, u otro microcontrolador, o puede(n) ser CPU convencional(es) que tiene(n) uno o más núcleos de procesamiento, incluyendo núcleos especializados basados en GPU en una CPU de múltiples núcleos. Cuando se usa en dispositivos de propósito especial tal como la técnica de intersección privada de conjuntos

asistida por servidor con transferencia de datos, el dispositivo informático se puede implementar como un ASIC o una FPGA, por ejemplo.

Además, el dispositivo informático simplificado de la FIG. 9 puede incluir también otros componentes, tales como, por ejemplo, una interfaz de comunicaciones 930. El dispositivo informático simplificado de la FIG. 9 puede incluir también uno o más dispositivos de entrada informáticos convencionales 940 (por ejemplo, dispositivos de apuntamiento, teclados, dispositivos de entrada de audio y habla, dispositivos de entrada de vídeo, dispositivos de entrada hápticos, dispositivos para recibir transmisiones de datos cableadas o inalámbricas, etc.). El dispositivo informático simplificado de la FIG. 9 puede incluir también otros componentes opcionales, tales como, por ejemplo, uno o más dispositivos de salida informáticos convencionales 950 (por ejemplo, dispositivo(s) de visualización 955, dispositivos de salida de audio, dispositivos de salida de vídeo, dispositivos para transmitir transmisiones de datos cableadas o inalámbricas, etc.). Señalar que las interfaces de comunicaciones típicas 930, los dispositivos de entrada 940, los dispositivos de salida 950, y los dispositivos de almacenamiento 960 para ordenadores de propósito general son bien conocidos por los expertos en la técnica, y no se describirán en detalle en la presente memoria.

El dispositivo informático simplificado de la FIG. 9 puede incluir también una variedad de medios legibles por ordenador. Los medios legibles por ordenador pueden ser cualquier medio disponible al que se pueda acceder por el ordenador 900 a través de los dispositivos de almacenamiento 960 e incluye tanto medios volátiles como no volátiles que son o bien extraíbles 970 y/o bien no extraíbles 980, para almacenamiento de información tal como instrucciones legibles por ordenador o ejecutables por ordenador, estructuras de datos, módulos de programa, u otros datos. Los medios legibles por ordenador pueden comprender medios de almacenamiento informático y medios de comunicación. Los medios de almacenamiento informático se refieren a medios legibles por ordenador o máquina tangibles o dispositivos de almacenamiento, tales como DVD, CD, discos flexibles, unidades de cinta, discos duros, unidades ópticas, dispositivos de memoria de estado sólido, RAM, ROM, EEPROM, memoria rápida u otra tecnología de memoria, casetes magnéticos, cintas magnéticas, almacenamiento en disco magnético, u otros dispositivos de almacenamiento magnético, o cualquier otro dispositivo que se pueda usar para almacenar la información deseada y al que se pueda acceder mediante uno o más dispositivos informáticos.

El almacenamiento de información tal como instrucciones legibles por ordenador o ejecutables por ordenador, estructuras de datos, módulos de programa, etc., también se puede consumar usando cualquiera de una variedad de los medios de comunicación antes mencionados para codificar una o más señales de datos moduladas u ondas portadoras, u otros mecanismos de transporte o protocolos de comunicaciones, e incluye cualquier mecanismo de entrega de información cableado o inalámbrico. Señalar que los términos “señal de datos modulada” u “onda portadora” se refieren generalmente a una señal que tiene una o más de sus características establecidas o cambiadas de tal manera en cuanto a codificar información en la señal. Por ejemplo, los medios de comunicación incluyen medios cableados tales como una red cableada o una conexión directa cableada que lleva una o más señales de datos modulados, y medios inalámbricos tales como medios acústicos, de RF, infrarrojos, láser, y otros inalámbricos para transmitir y/o recibir una o más señales de datos modulados u ondas portadoras. Las combinaciones de cualquiera de las anteriores también se deberían incluir dentro del alcance de los medios de comunicación.

Además, se pueden almacenar, recibir, transmitir, o leer software, programas, y/o productos de programas informáticos que incorporen algunas de o todas las diversas realizaciones de la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria, o partes de la misma a partir de cualquier combinación deseada de medios legibles por ordenador o por máquina o dispositivos de almacenamiento y medios de comunicación en forma de instrucciones ejecutables por ordenador u otras estructuras de datos.

Finalmente, la técnica de intersección privada de conjuntos asistida por servidor con transferencia de datos descrita en la presente memoria se puede describir además en el contexto general de instrucciones ejecutables por ordenador, tales como módulos de programa, que se ejecutan por un dispositivo informático. En general, los módulos de programa incluyen rutinas, programas, objetos, componentes, estructuras de datos, etc., que realizan tareas particulares o implementan tipos de datos abstractos particulares. Las realizaciones descritas en la presente memoria también se pueden poner en práctica en entornos informáticos distribuidos en los que las tareas se realizan por uno o más dispositivos de procesamiento remoto, o dentro de una nube de uno o más dispositivos, que están enlazados a través de una o más redes de comunicaciones. En un entorno informático distribuido, los módulos de programa se pueden situar tanto en medios de almacenamiento informático local como remoto, incluyendo dispositivos de almacenamiento de medios. Aún más, las instrucciones antes mencionadas se pueden implementar, en parte o en su totalidad, como circuitos lógicos de hardware, que pueden incluir o no un procesador.

También se debería señalar que cualquiera de o todas las realizaciones alternativas antes mencionadas descritas en la presente memoria se pueden usar en cualquier combinación deseada para formar realizaciones híbridas adicionales. Aunque la materia objeto se ha descrito en un lenguaje específico a características estructurales y/o actos metodológicos, se tiene que entender que la materia objeto definida en las reivindicaciones adjuntas no está necesariamente limitada a las características o actos específicos descritos anteriormente. Las características y los actos específicos descritos anteriormente se describen como formas de ejemplo de implementación de las reivindicaciones.

REIVINDICACIONES

1. Uno o más medios legibles por ordenador (960) que comprenden instrucciones ejecutables por ordenador incorporadas en los mismos que, cuando se ejecutan por un dispositivo informático (702, 900) que representa una tercera parte (812), realizan un proceso (600) para transferir alguna información asociada con elementos de un conjunto de datos mantenido por una primera parte (808) a una segunda parte (810) sin revelar el conjunto de datos, que comprende:

recibir (602) desde la primera parte (808), para cada elemento en un conjunto S_1: una etiqueta para cada elemento, un identificador para el elemento, y la primera parte de una clave secreta de dos partes;

recibir (604) desde la segunda parte (810) etiquetas generadas por los elementos de un segundo conjunto S_2;

calcular (606) la intersección de las etiquetas de los conjuntos S_1 y S_2;

enviar (608) a la segunda parte (810), para cada elemento en la intersección de las etiquetas: la etiqueta, el identificador para el elemento y la primera parte de la clave secreta de dos partes.

2. Los medios de la reivindicación 1, en donde la primera parte (808) genera (202) triples para cada elemento en el conjunto S_1 que comprende:

una etiqueta para el elemento,

un identificador para el elemento, y

una forma cifrada para los datos asociados con el elemento que se cifró usando la clave secreta de dos partes.

3. Los medios de la reivindicación 1, en donde la primera parte (808) envía (204) a la segunda parte (810), para cada elemento en el conjunto S_1: el identificador para cada elemento, los datos cifrados asociados con cada elemento y la segunda parte de la clave secreta de dos partes.

4. Los medios de la reivindicación 1, que además comprenden:

la segunda parte (810) que descifra (212) los datos asociados con cada elemento en la intersección de los conjuntos usando los identificadores y la primera y una segunda parte de la clave secreta de dos partes.

5. Los medios de la reivindicación 1, en donde la primera parte (808) y la segunda parte (810) generan etiquetas para su conjunto aplicando una permutación pseudoaleatoria (PRP) compartida a los elementos de sus conjuntos de datos usando una clave compartida.

6. Los medios de la reivindicación 1, en donde la primera parte (808) genera la clave secreta de dos partes usando otra clave secreta.

7. Los medios de la reivindicación 1, en donde la tercera parte (812) es una tercera parte (812) no de confianza.

8. Los medios de la reivindicación 4, en donde la primera (808) y la segunda parte (810) son clientes (704, 706) y la tercera parte (812) es uno o más servidores (702) en una nube informática.

9. Los medios de la reivindicación 1, en donde la tercera parte (812) realiza una intersección de conjuntos de texto plano en las etiquetas.

10. Un proceso implementado por ordenador (300) para transferir alguna información db(x) asociada con los elementos de datos en la intersección de dos conjuntos de datos S_1 y S_2 mantenidos por dos partes P_1 (808) y P_2 (810) usando una tercera parte (812) no de confianza, sin que cualquiera de las dos partes P_1 o P_2 revele los datos en sus conjuntos, que comprende:

la primera parte P_1 que genera (302) una primera clave secreta K_e, una segunda clave secreta K_l, y una tercera clave secreta K_1 y que envía la segunda clave secreta K_l a la segunda parte P_2;

para cada elemento x en el conjunto S_1 de P_1 (304):

usar K_e, que genera una clave secreta de dos partes K_x que comprende una primera parte $z_{x,1}$ y una segunda parte $z_{x,2}$,

cifrar datos db(x) asociados con el elemento usando la clave K_x,

crear un identificador id_x para el elemento,

crear una etiqueta l_x para el elemento aplicando una permutación pseudoaleatoria al elemento usando la clave secreta K_l;

la primera parte P_1 que envía (306) a la tercera parte no de confianza un conjunto de triples para cada elemento en el conjunto S_1 que comprende: el identificador id_x, la primera parte de la clave $z_{x,1}$ y la etiqueta l_x;

la primera parte P_1 que envía (308) a la segunda parte P_2 un conjunto de triples para cada elemento que comprende: el identificador id_x, la segunda parte de la clave $z_{x,2}$ y los datos cifrados $C(db(x))$;

la segunda parte P_2 que genera (310) etiquetas para los elementos en el conjunto S_2 de P_2 aplicando una permutación a los elementos usando la clave K_l, y que envía las etiquetas para cada elemento en el conjunto S_2 a la tercera parte no de confianza;

la tercera parte no de confianza que compara (312) el conjunto de etiquetas recibidas desde P_1 y el conjunto de etiquetas recibidas desde P_2 para encontrar la intersección de las etiquetas de P_1 y P_2; y

la tercera parte no de confianza que envía a P_2 todos los triples que la tercera parte recibió desde P_1 que tienen una etiqueta dentro de la intersección.

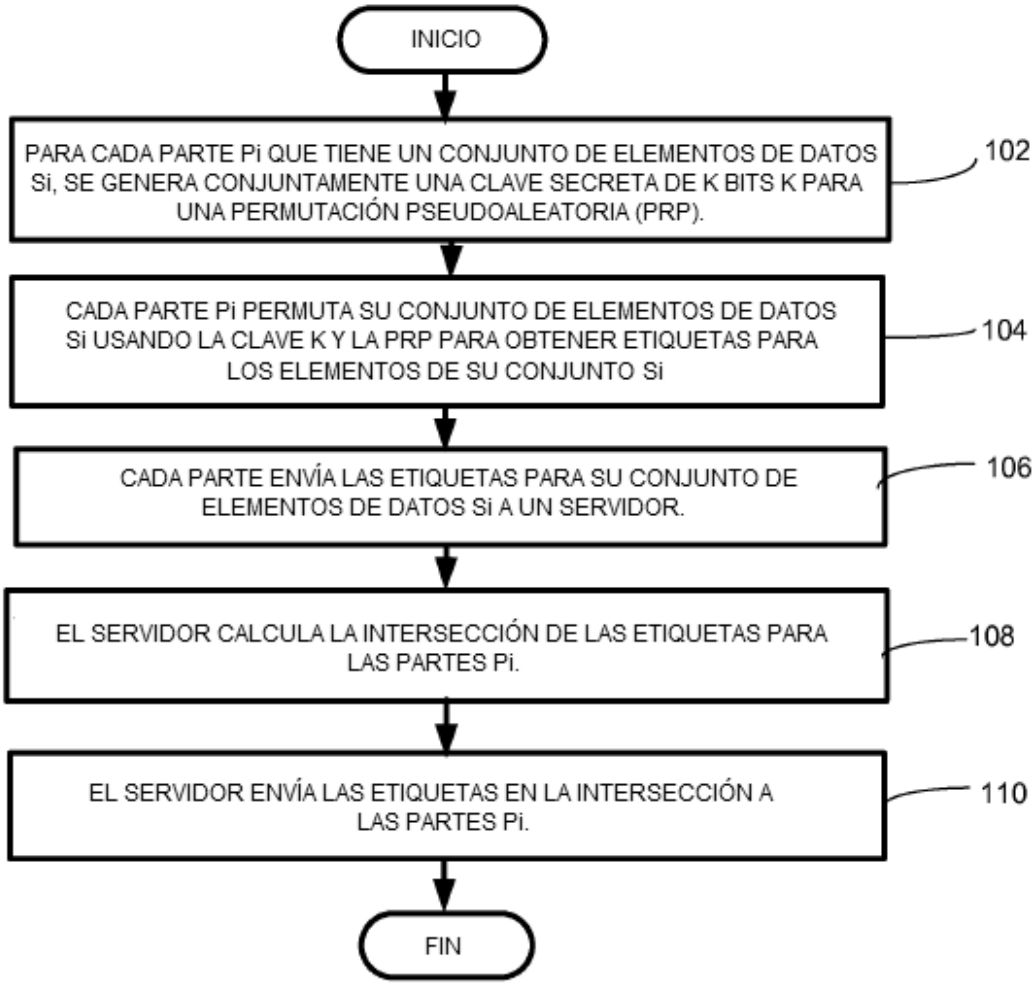
INICIO
PARA CADA PARTE Pi QUE TIENE UN CONJUNTO DE ELEMENTOS DE DATOS Si, SE GENERA CONJUNTAMENTE UNA CLAVE SECRETA DE K BITS K PARA UNA PERMUTACIÓN PSEUDOALEATORIA (PRP).
102
CADA PARTE Pi PERMUTA SU CONJUNTO DE ELEMENTOS DE DATOS Si USANDO LA CLAVE K Y LA PRP PARA OBTENER ETIQUETAS PARA LOS ELEMENTOS DE SU CONJUNTO Si
104
CADA PARTE ENVÍA LAS ETIQUETAS PARA SU CONJUNTO DE ELEMENTOS DE DATOS Si A UN SERVIDOR.
106
EL SERVIDOR CALCULA LA INTERSECCIÓN DE LAS ETIQUETAS PARA LAS PARTES Pi.
108
EL SERVIDOR ENVÍA LAS ETIQUETAS EN LA INTERSECCIÓN A LAS PARTES Pi.
110
FIN

FIG. 1

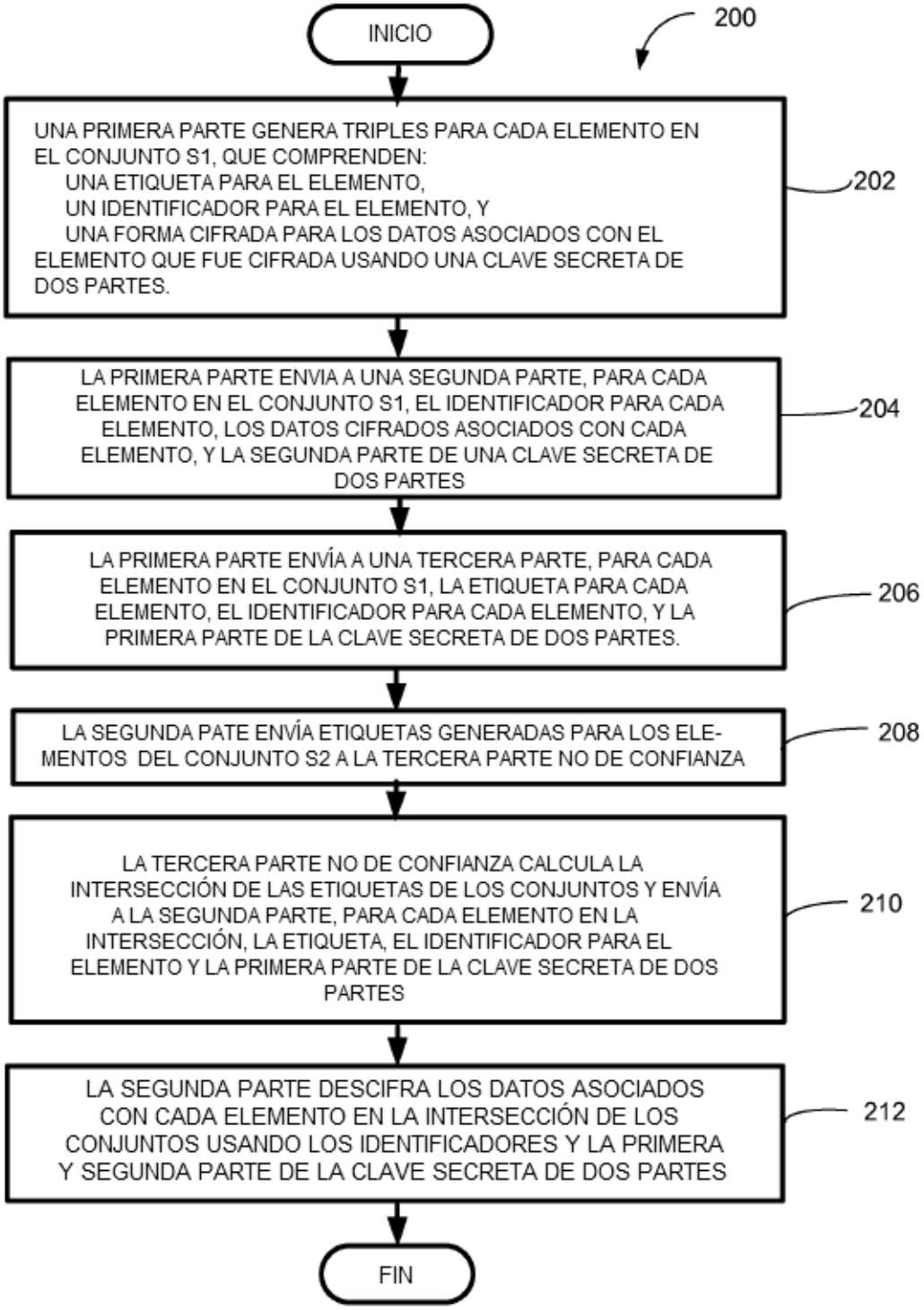
INICIO
200
UNA PRIMERA PARTE GENERA TRIPLES PARA CADA ELEMENTO EN EL CONJUNTO S1, QUE COMPRENDEN:
UNA ETIQUETA PARA EL ELEMENTO,
UN IDENTIFICADOR PARA EL ELEMENTO, Y
UNA FORMA CIFRADA PARA LOS DATOS ASOCIADOS CON EL ELEMENTO QUE FUE CIFRADA USANDO UNA CLAVE SECRETA DE DOS PARTES.
202
LA PRIMERA PARTE ENVIA A UNA SEGUNDA PARTE, PARA CADA ELEMENTO EN EL CONJUNTO S1, EL IDENTIFICADOR PARA CADA ELEMENTO, LOS DATOS CIFRADOS ASOCIADOS CON CADA ELEMENTO, Y LA SEGUNDA PARTE DE UNA CLAVE SECRETA DE DOS PARTES
204
LA PRIMERA PARTE ENVÍA A UNA TERCERA PARTE, PARA CADA ELEMENTO EN EL CONJUNTO S1, LA ETIQUETA PARA CADA ELEMENTO, EL IDENTIFICADOR PARA CADA ELEMENTO, Y LA PRIMERA PARTE DE LA CLAVE SECRETA DE DOS PARTES.
206
LA SEGUNDA PATE ENVÍA ETIQUETAS GENERADAS PARA LOS ELEMENTOS DEL CONJUNTO S2 A LA TERCERA PARTE NO DE CONFIANZA
208
LA TERCERA PARTE NO DE CONFIANZA CALCULA LA INTERSECCIÓN DE LAS ETIQUETAS DE LOS CONJUNTOS Y ENVÍA A LA SEGUNDA PARTE, PARA CADA ELEMENTO EN LA INTERSECCIÓN, LA ETIQUETA, EL IDENTIFICADOR PARA EL ELEMENTO Y LA PRIMERA PARTE DE LA CLAVE SECRETA DE DOS PARTES
210
LA SEGUNDA PARTE DESCIFRA LOS DATOS ASOCIADOS CON CADA ELEMENTO EN LA INTERSECCIÓN DE LOS CONJUNTOS USANDO LOS IDENTIFICADORES Y LA PRIMERA Y SEGUNDA PARTE DE LA CLAVE SECRETA DE DOS PARTES
212
FIN

FIG. 2

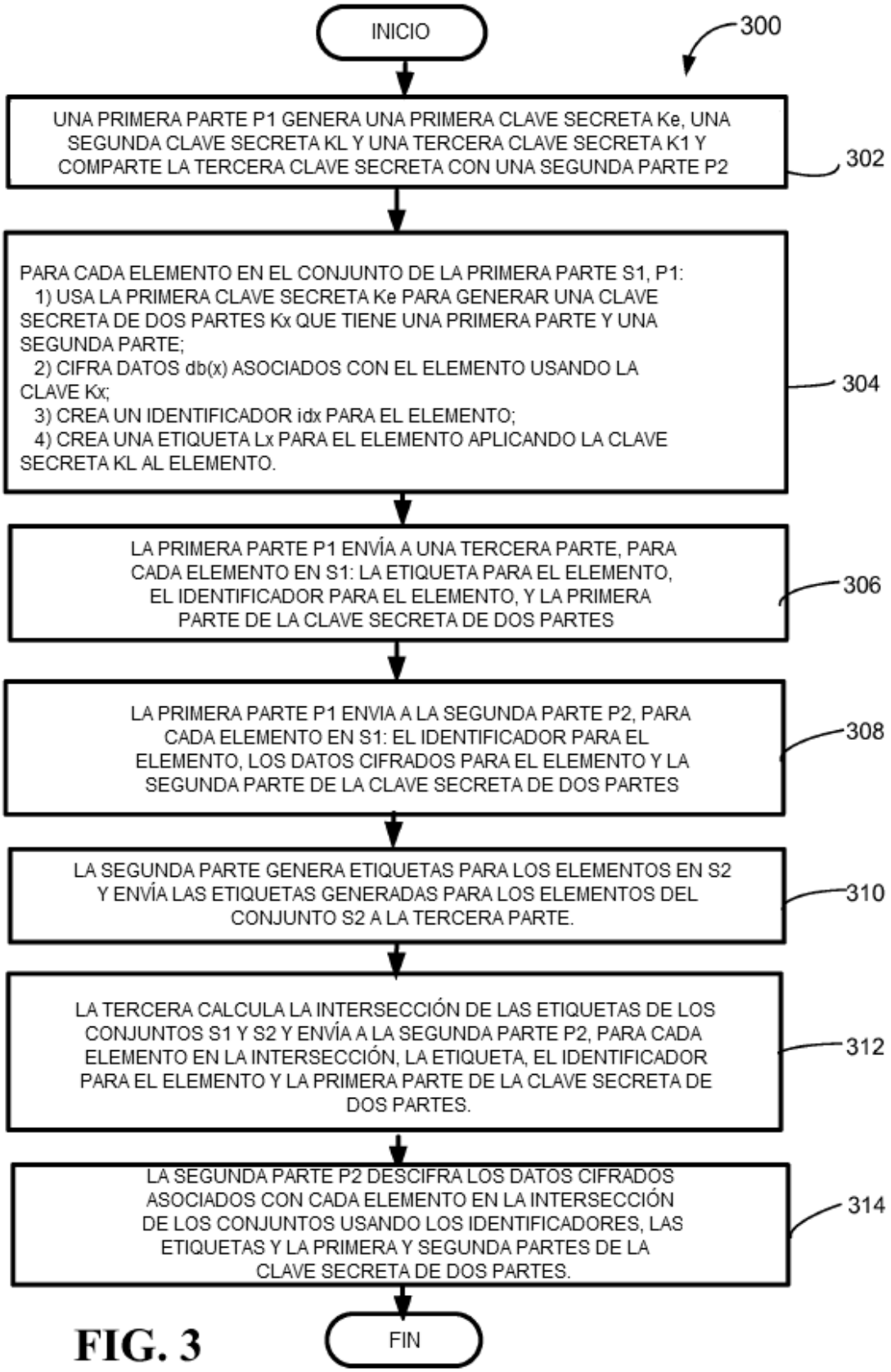
INICIO
300
UNA PRIMERA PARTE P1 GENERA UNA PRIMERA CLAVE SECRETA Ke, UNA SEGUNDA CLAVE SECRETA KL Y UNA TERCERA CLAVE SECRETA K1 Y COMPARTE LA TERCERA CLAVE SECRETA CON UNA SEGUNDA PARTE P2
302
PARA CADA ELEMENTO EN EL CONJUNTO DE LA PRIMERA PARTE S1, P1:
1) USA LA PRIMERA CLAVE SECRETA Ke PARA GENERAR UNA CLAVE SECRETA DE DOS PARTES Kx QUE TIENE UNA PRIMERA PARTE Y UNA SEGUNDA PARTE;
2) CIFRA DATOS db(x) ASOCIADOS CON EL ELEMENTO USANDO LA CLAVE Kx;
3) CREA UN IDENTIFICADOR idx PARA EL ELEMENTO;
4) CREA UNA ETIQUETA Lx PARA EL ELEMENTO APLICANDO LA CLAVE SECRETA KL AL ELEMENTO.
304
LA PRIMERA PARTE P1 ENVÍA A UNA TERCERA PARTE, PARA CADA ELEMENTO EN S1: LA ETIQUETA PARA EL ELEMENTO, EL IDENTIFICADOR PARA EL ELEMENTO, Y LA PRIMERA PARTE DE LA CLAVE SECRETA DE DOS PARTES
306
LA PRIMERA PARTE P1 ENVIA A LA SEGUNDA PARTE P2, PARA CADA ELEMENTO EN S1: EL IDENTIFICADOR PARA EL ELEMENTO, LOS DATOS CIFRADOS PARA EL ELEMENTO Y LA SEGUNDA PARTE DE LA CLAVE SECRETA DE DOS PARTES
308
LA SEGUNDA PARTE GENERA ETIQUETAS PARA LOS ELEMENTOS EN S2 Y ENVÍA LAS ETIQUETAS GENERADAS PARA LOS ELEMENTOS DEL CONJUNTO S2 A LA TERCERA PARTE.
310
LA TERCERA CALCULA LA INTERSECCIÓN DE LAS ETIQUETAS DE LOS CONJUNTOS S1 Y S2 Y ENVÍA A LA SEGUNDA PARTE P2, PARA CADA ELEMENTO EN LA INTERSECCIÓN, LA ETIQUETA, EL IDENTIFICADOR PARA EL ELEMENTO Y LA PRIMERA PARTE DE LA CLAVE SECRETA DE DOS PARTES.
312
LA SEGUNDA PARTE P2 DESCIFRA LOS DATOS CIFRADOS ASOCIADOS CON CADA ELEMENTO EN LA INTERSECCIÓN DE LOS CONJUNTOS USANDO LOS IDENTIFICADORES, LAS ETIQUETAS Y LA PRIMERA Y SEGUNDA PARTES DE LA CLAVE SECRETA DE DOS PARTES.
314
FIN

FIG. 3

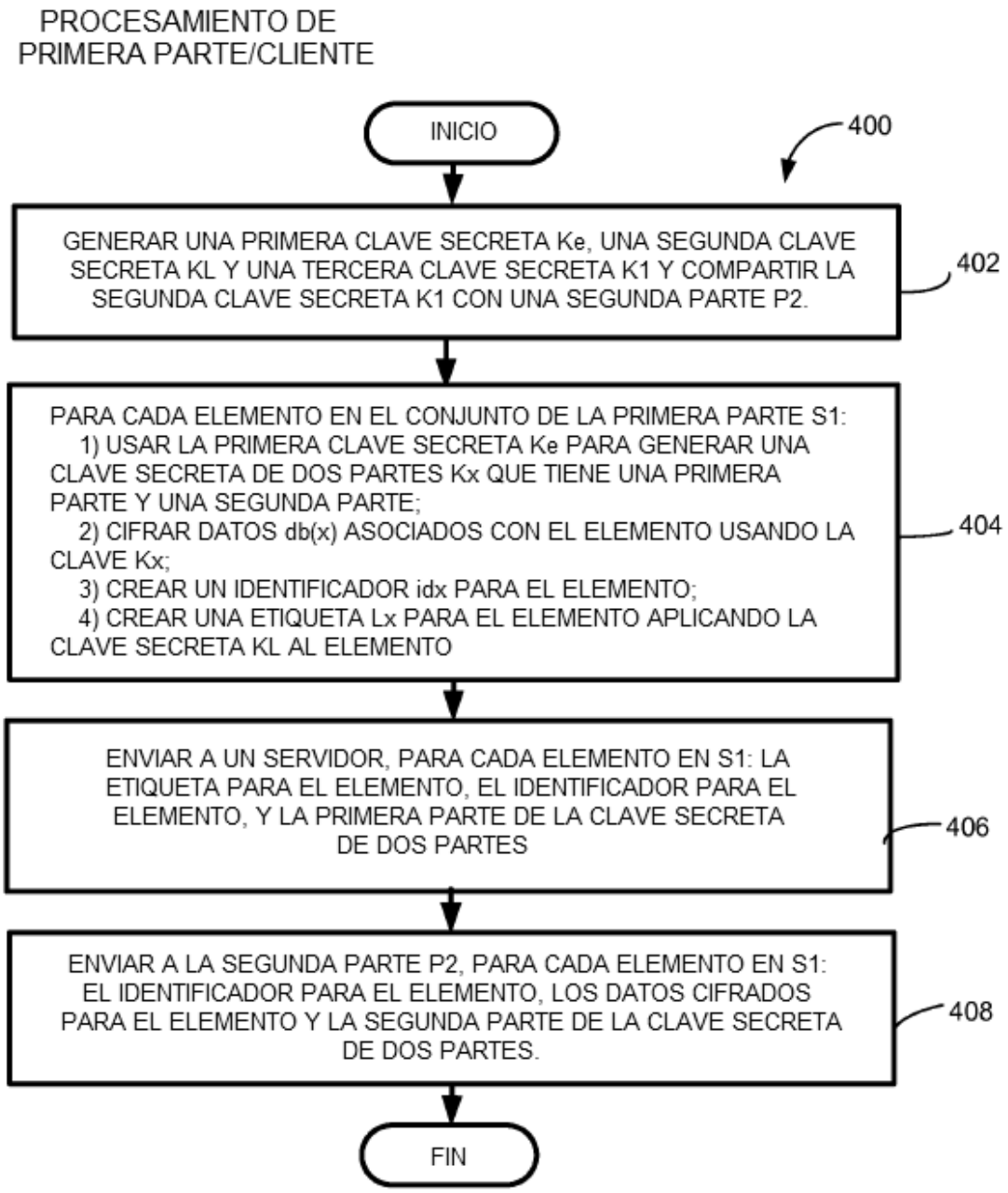
PROCESAMIENTO DE PRIMERA PARTE/CLIENTE
INICIO
400
GENERAR UNA PRIMERA CLAVE SECRETA Ke, UNA SEGUNDA CLAVE SECRETA KL Y UNA TERCERA CLAVE SECRETA K1 Y COMPARTIR LA SEGUNDA CLAVE SECRETA K1 CON UNA SEGUNDA PARTE P2.
402
PARA CADA ELEMENTO EN EL CONJUNTO DE LA PRIMERA PARTE S1:
1) USAR LA PRIMERA CLAVE SECRETA Ke PARA GENERAR UNA CLAVE SECRETA DE DOS PARTES Kx QUE TIENE UNA PRIMERA PARTE Y UNA SEGUNDA PARTE;
2) CIFRAR DATOS db(x) ASOCIADOS CON EL ELEMENTO USANDO LA CLAVE Kx;
3) CREAR UN IDENTIFICADOR idx PARA EL ELEMENTO;
4) CREAR UNA ETIQUETA Lx PARA EL ELEMENTO APLICANDO LA CLAVE SECRETA KL AL ELEMENTO
404
ENVIAR A UN SERVIDOR, PARA CADA ELEMENTO EN S1: LA ETIQUETA PARA EL ELEMENTO, EL IDENTIFICADOR PARA EL ELEMENTO, Y LA PRIMERA PARTE DE LA CLAVE SECRETA DE DOS PARTES
406
ENVIAR A LA SEGUNDA PARTE P2, PARA CADA ELEMENTO EN S1: EL IDENTIFICADOR PARA EL ELEMENTO, LOS DATOS CIFRADOS PARA EL ELEMENTO Y LA SEGUNDA PARTE DE LA CLAVE SECRETA DE DOS PARTES.
408
FIN

FIG. 4

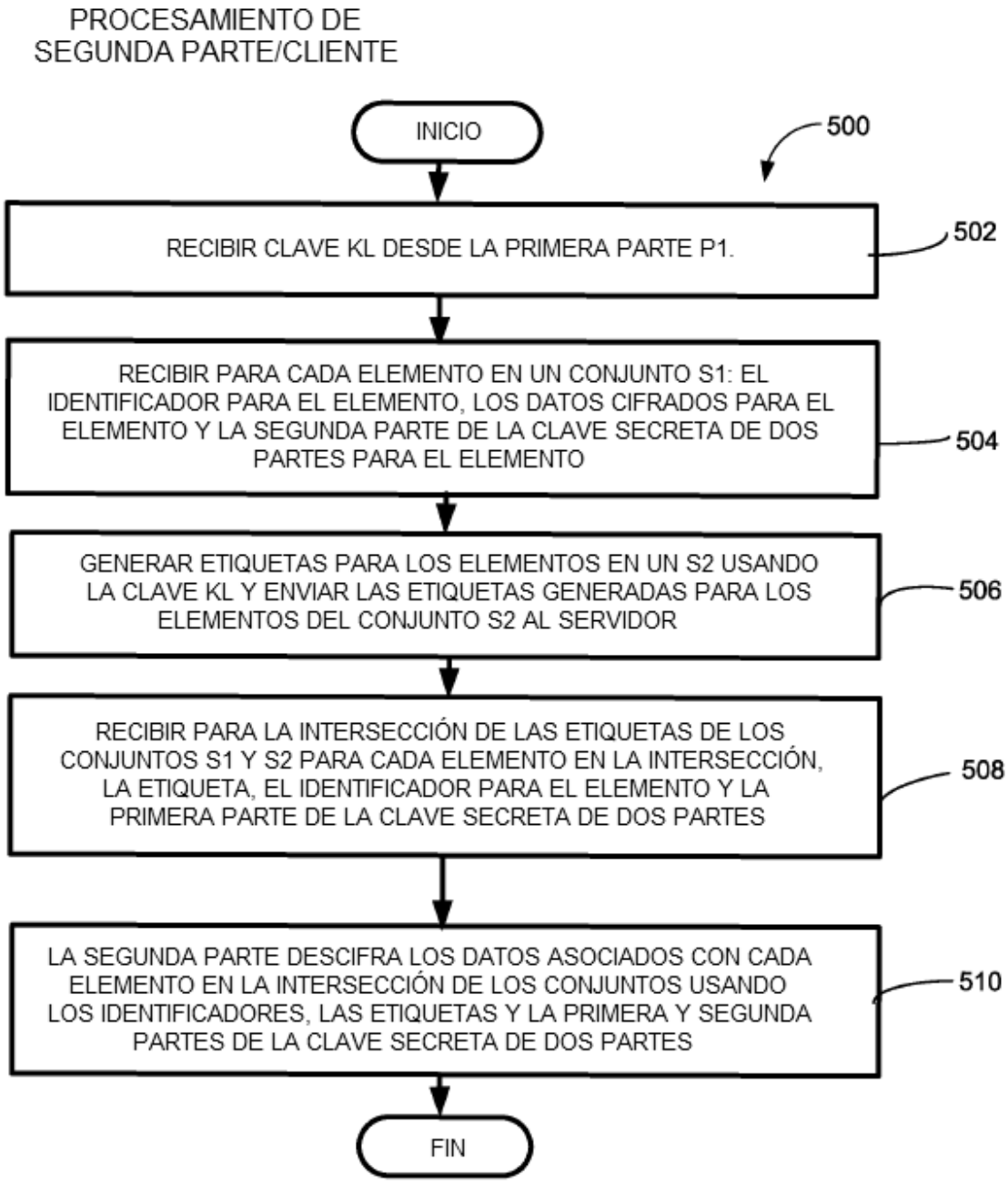
PROCESAMIENTO DE SEGUNDA PARTE/CLIENTE
INICIO
500
RECIBIR CLAVE KL DESDE LA PRIMERA PARTE P1.
502
RECIBIR PARA CADA ELEMENTO EN UN CONJUNTO S1: EL IDENTIFICADOR PARA EL ELEMENTO, LOS DATOS CIFRADOS PARA EL ELEMENTO Y LA SEGUNDA PARTE DE LA CLAVE SECRETA DE DOS PARTES PARA EL ELEMENTO
504
GENERAR ETIQUETAS PARA LOS ELEMENTOS EN UN S2 USANDO LA CLAVE KL Y ENVIAR LAS ETIQUETAS GENERADAS PARA LOS ELEMENTOS DEL CONJUNTO S2 AL SERVIDOR
506
RECIBIR PARA LA INTERSECCIÓN DE LAS ETIQUETAS DE LOS CONJUNTOS S1 Y S2 PARA CADA ELEMENTO EN LA INTERSECCIÓN, LA ETIQUETA, EL IDENTIFICADOR PARA EL ELEMENTO Y LA PRIMERA PARTE DE LA CLAVE SECRETA DE DOS PARTES
508
LA SEGUNDA PARTE DESCIFRA LOS DATOS ASOCIADOS CON CADA ELEMENTO EN LA INTERSECCIÓN DE LOS CONJUNTOS USANDO LOS IDENTIFICADORES, LAS ETIQUETAS Y LA PRIMERA Y SEGUNDA PARTES DE LA CLAVE SECRETA DE DOS PARTES
510
FIN

FIG. 5

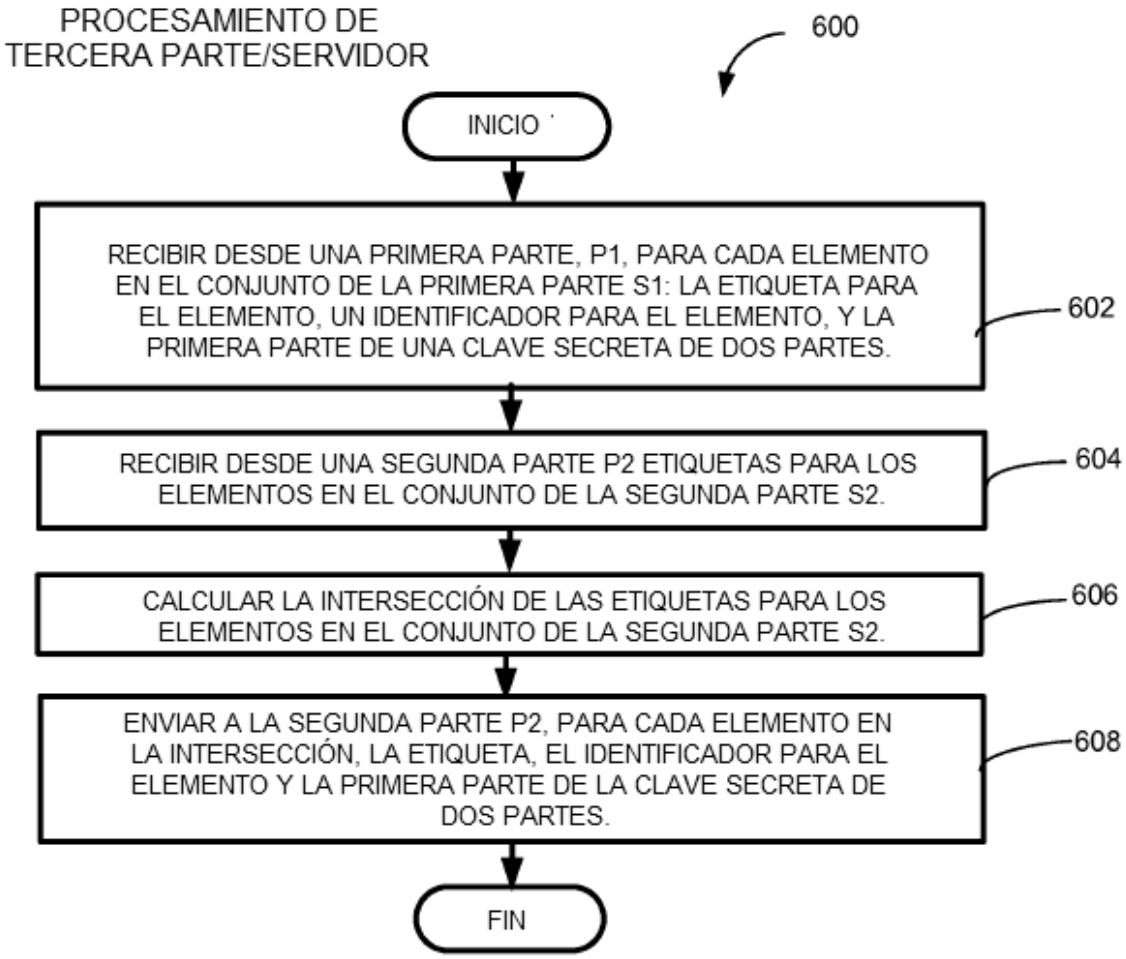
PROCESAMIENTO DE TERCERA PARTE/SERVIDOR
600
INICIO
RECIBIR DESDE UNA PRIMERA PARTE, P1, PARA CADA ELEMENTO EN EL CONJUNTO DE LA PRIMERA PARTE S1: LA ETIQUETA PARA EL ELEMENTO, UN IDENTIFICADOR PARA EL ELEMENTO, Y LA PRIMERA PARTE DE UNA CLAVE SECRETA DE DOS PARTES.
602
RECIBIR DESDE UNA SEGUNDA PARTE P2 ETIQUETAS PARA LOS ELEMENTOS EN EL CONJUNTO DE LA SEGUNDA PARTE S2.
604
CALCULAR LA INTERSECCIÓN DE LAS ETIQUETAS PARA LOS ELEMENTOS EN EL CONJUNTO DE LA SEGUNDA PARTE S2.
606
ENVIAR A LA SEGUNDA PARTE P2, PARA CADA ELEMENTO EN LA INTERSECCIÓN, LA ETIQUETA, EL IDENTIFICADOR PARA EL ELEMENTO Y LA PRIMERA PARTE DE LA CLAVE SECRETA DE DOS PARTES.
608
FIN

FIG. 6

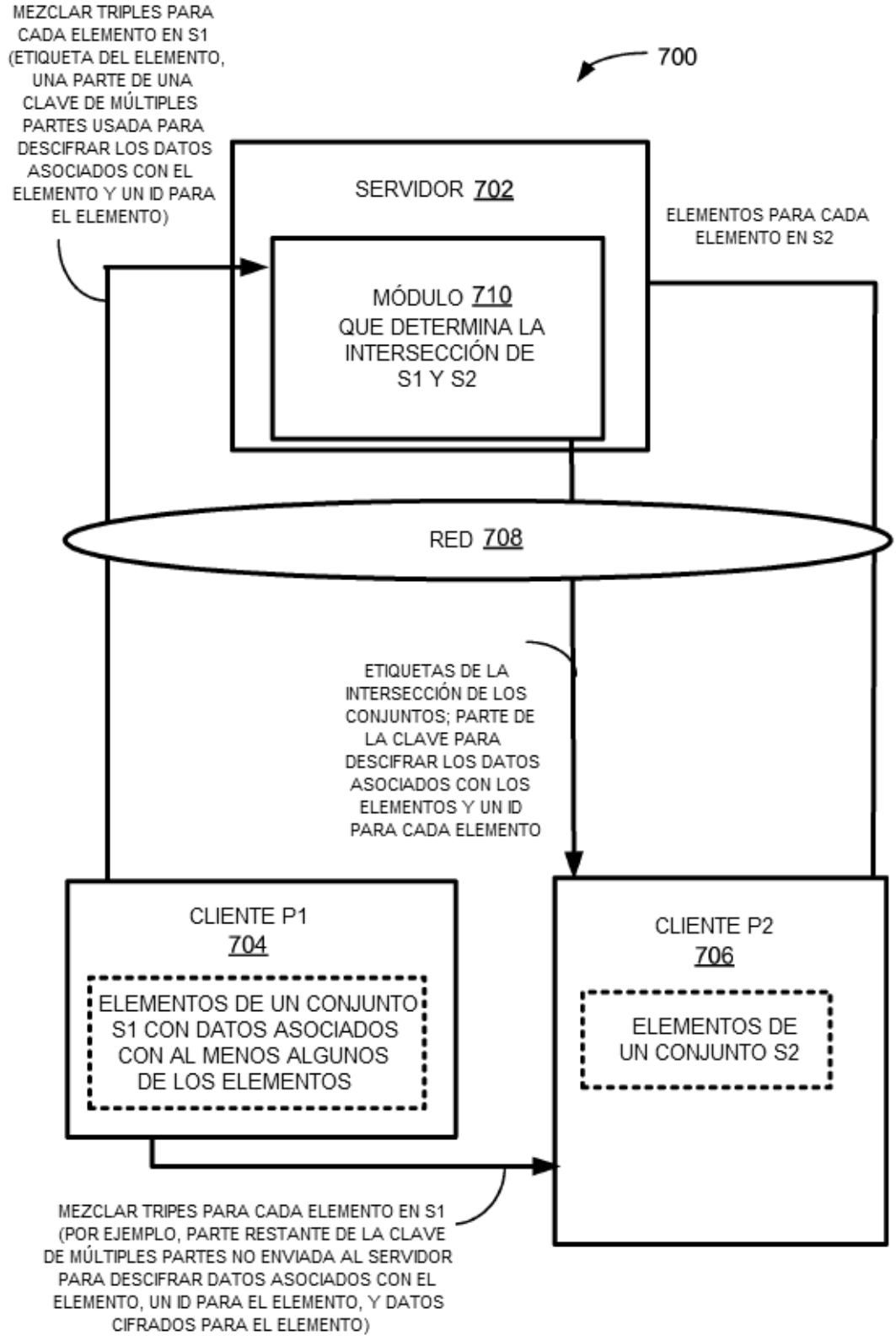
MEZCLAR TRIPLES PARA CADA ELEMENTO EN S1 (ETIQUETA DEL ELEMENTO, UNA PARTE DE UNA CLAVE DE MÚLTIPLES PARTES USADA PARA DESCIFRAR LOS DATOS ASOCIADOS CON EL ELEMENTO Y UN ID PARA EL ELEMENTO)
700
SERVIDOR 702
ELEMENTOS PARA CADA ELEMENTO EN S2
MÓDULO 710 QUE DETERMINA LA INTERSECCIÓN DE S1 Y S2
RED 708
ETIQUETAS DE LA INTERSECCIÓN DE LOS CONJUNTOS; PARTE DE LA CLAVE PARA DESCIFRAR LOS DATOS ASOCIADOS CON LOS ELEMENTOS Y UN ID PARA CADA ELEMENTO
CLIENTE P1 704
ELEMENTOS DE UN CONJUNTO S1 CON DATOS ASOCIADOS CON AL MENOS ALGUNOS DE LOS ELEMENTOS
CLIENTE P2 706
ELEMENTOS DE UN CONJUNTO S2
MEZCLAR TRIPES PARA CADA ELEMENTO EN S1 (POR EJEMPLO, PARTE RESTANTE DE LA CLAVE DE MÚLTIPLES PARTES NO ENVIADA AL SERVIDOR PARA DESCIFRAR DATOS ASOCIADOS CON EL ELEMENTO, UN ID PARA EL ELEMENTO, Y DATOS CIFRADOS PARA EL ELEMENTO)

FIG. 7

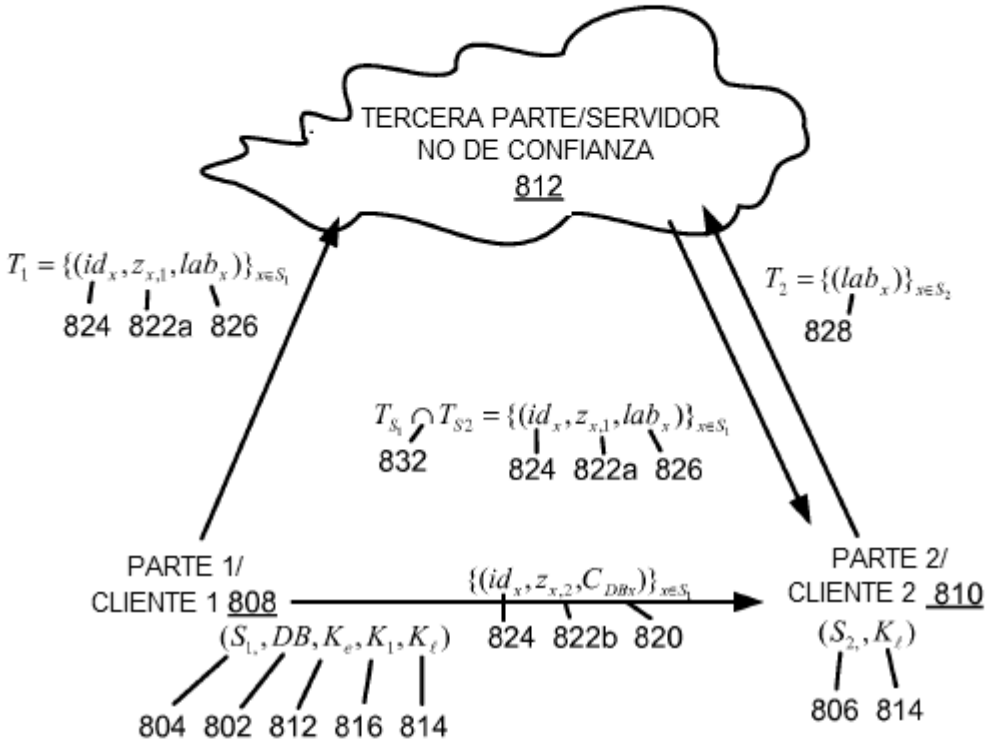
TERCERA PARTE/SERVIDOR
NO DE CONFIANZA
812
$T_1 = \{(id_x, z_{x,1}, lab_x)\}_{x \in S_1}$
824 822a 826
$T_2 = \{(lab_x)\}_{x \in S_2}$
828
$T_{S_1} \cap T_{S2} = \{(id_x, z_{x,1}, lab_x)\}_{x \in S_1}$
832
824 822a 826
PARTE 1/
CLIENTE 1 808
$(S_1, DB, K_e, K_1, K_\ell)$
804 802 812 816 814
$\{(id_x, z_{x,2}, C_{DBx})\}_{x \in S_1}$
824 822b 820
PARTE 2/
CLIENTE 2 810
(S_2, K_ℓ)
806 814

FIG. 8

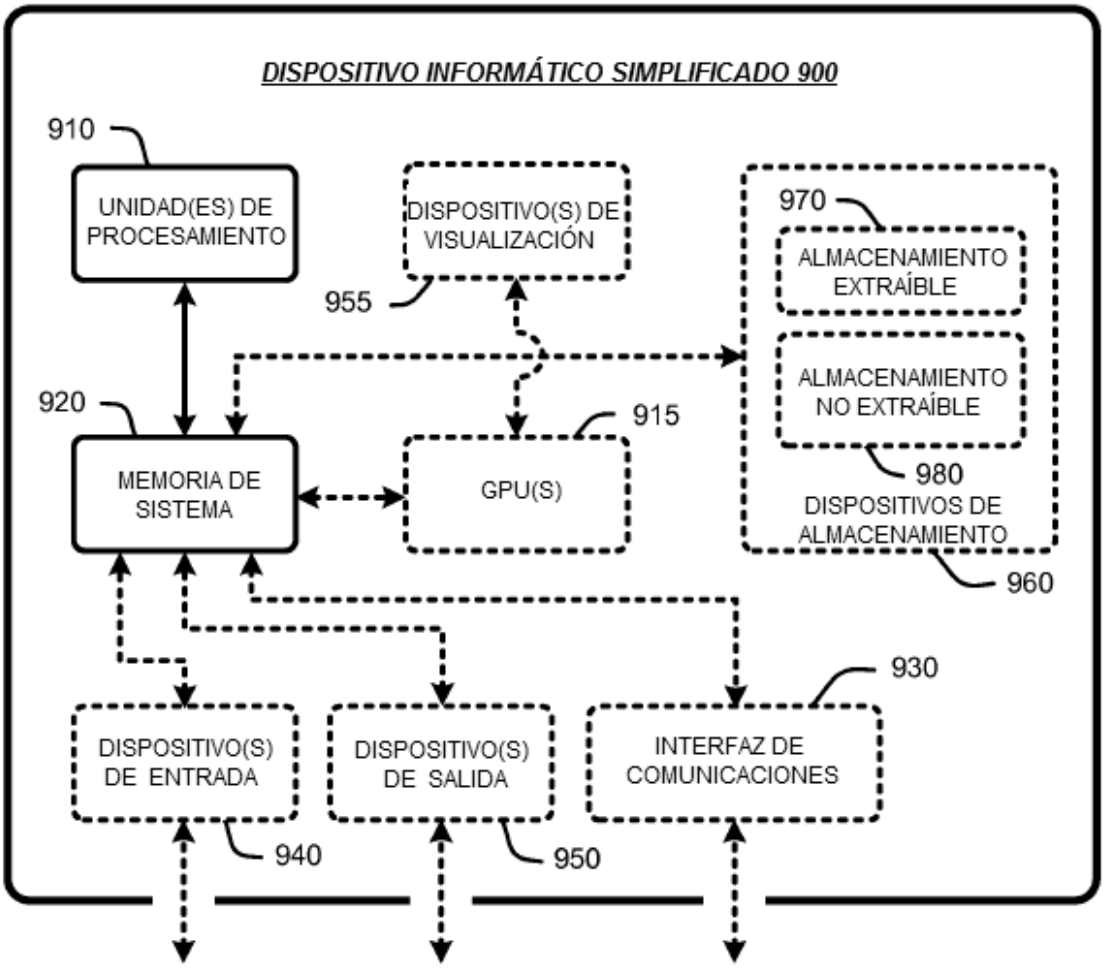
DISPOSITIVO INFORMÁTICO SIMPLIFICADO 900
910
UNIDAD(ES) DE PROCESAMIENTO
DISPOSITIVO(S) DE VISUALIZACIÓN
955
970
ALMACENAMIENTO EXTRAÍBLE
ALMACENAMIENTO NO EXTRAÍBLE
980
DISPOSITIVOS DE ALMACENAMIENTO
960
920
MEMORIA DE SISTEMA
915
GPU(S)
930
DISPOSITIVO(S) DE ENTRADA
DISPOSITIVO(S) DE SALIDA
INTERFAZ DE COMUNICACIONES
940
950

FIG. 9