(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 598 298**

(51) Int. CI.:

H04L 9/08 (2006.01)
H04L 9/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **21.12.2013 PCT/US2013/077348**

(87) Fecha y número de publicación internacional: **26.06.2014 WO14100788**

(96) Fecha de presentación y número de la solicitud europea: **21.12.2013 E 13821361 (6)**

(97) Fecha y número de publicación de la concesión europea: **20.07.2016 EP 2936731**

(54) Título: **Cálculos seguros gestionados, sobre datos cifrados**

(30) Prioridad:

21.12.2012 US 201213723879

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
26.01.2017

(73) Titular/es:

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

(72) Inventor/es:

LOFTUS, JACOB J.;
NAEHRIG, MICHAEL;
BOS, JOPPE WILLEM y
LAUTER, KRISTIN ESTELLA

(74) Agente/Representante:

DE ELZABURU MÁRQUEZ, Alberto

DESCRIPCIÓN

Cálculos seguros gestionados, sobre datos cifrados

Antecedentes

El mantenimiento de la confidencialidad de los datos es una preocupación importante para todos los usuarios de dispositivos informáticos con independencia de la(s) tarea(s) llevada(s) a cabo. Los esquemas de cifrado representan una forma de tecnología enfocada a la protección de datos cuando estos se almacenan en memoria y/o se transmiten a través de redes. El cifrado totalmente homomórfico (FHE) hace referencia a un esquema de cifrado que permite que un servidor no fiable ejecute un cálculo arbitrario sobre datos cifrados, en nombre de un dispositivo informático, al que se hace referencia normalmente como cliente. El documento WO2012/149395 A1 describe un esquema de cifrado totalmente homomórfico. CRAIG GENTRY et al. “Homomorphic Evaluation of the AES Circuit”, 19 de agosto de 2012, *ADVANCES IN CRYPTOLOGY CRYPTO* 2012, SPRINGER BERLIN HEIDELBERG, página(s) 850 a 867, ISBN: 978-3-642-32008-8, describen una implementación operativa de un cifrado homomórfico por niveles para un circuito de AES. CRAIG GENTRY et al. “Better Bootstrapping in Fully Homomorphic Encryption”, 21 de mayo de 2012, *PUBLIC KEY CRYPTOGRAPHY PKC 2012*, SPRINGER BERLIN HEIDELBERG, página(s) 1 a 16, describen una técnica de autoevaluación eficiente aplicable en el cifrado totalmente homomórfico.

Las soluciones convencionales para construir un esquema de FHE incurren normalmente en costes significativos debido a la dependencia de ciertos conceptos matemáticos (por ejemplo, retículos ideales) para evaluar cualquier función sobre datos cifrados. En la práctica, estas construcciones se pueden mejorar utilizando técnicas convencionales, tales como aquellas relacionadas con el cifrado a nivel de lotes o a nivel de bits, aunque normalmente permanecen como inviables por diversos motivos, por ejemplo, la necesidad de circuitos complicados, tales como aquellos que se basan en las normas de cifrado, y/o un espacio de almacenamiento sustancial para procesar textos cifrados. Aunque algunas construcciones de FHE tienen la capacidad de calcular textos cifrados en bloque homomórficamente, la evaluación de un único bloque utilizando estas construcciones es varios órdenes de magnitud más lenta, en términos de rendimiento y latencia, en comparación con la evaluación de un único bloque de manera no homomórfica. Dicha diferencia en el rendimiento realza la impractibilidad actual de implementación de esquemas de FHE utilizando estas soluciones convencionales.

Sumario

Este Sumario se proporciona para introducir una selección de conceptos representativos de una manera simplificada y que se describen adicionalmente más adelante en la Descripción Detallada. Este Sumario no está destinado a identificar características clave o características esenciales de la materia en cuestión reivindicada, ni está destinado a utilizarse de ninguna manera que limite el alcance de la materia en cuestión reivindicada.

De forma breve, varios aspectos de la materia en cuestión que se describe en la presente van dirigidos a la implementación de un esquema de cifrado totalmente homomórfico para evaluar un circuito convencional de cifrado. En uno de los aspectos, un sistema de cifrado de claves públicas basado en anillos es totalmente homomórfico cuando se construye utilizando problemas de retículos y/o textos cifrados reducidos, bien conocidos, con el fin de garantizar la corrección y la seguridad. En otro aspecto, el esquema de cifrado totalmente homomórfico codifica enteros fraccionando dichos enteros sobre la base de un parámetro de codificación, de tal manera que se realizan cálculos en paralelo.

Sobre la base de la entrada del usuario, el esquema de cifrado totalmente homomórfico se configura automáticamente para llevar a cabo dichos cálculos. En uno de los aspectos, uno o más componentes de bibliotecas de software/hardware seleccionan uno o más parámetros para configurar el esquema de cifrado con el fin de funcionar eficientemente. En uno de los aspectos, la restricción de un espacio de claves criptográficas permite un mayor rendimiento y un uso reducido de la memoria, al mismo tiempo que manteniendo la seguridad y la corrección. En uno de los aspectos, un dispositivo informático usa un componente de biblioteca para configurar el esquema de cifrado totalmente homomórfico con el fin de llevar a cabo los cálculos sobre los datos cifrados.

El componente de biblioteca se configura para proporcionar acceso a la funcionalidad homomórfica. Una de estas funciones lleva a cabo una operación matemática sobre uno o más elementos de datos cifrados. En uno de los aspectos, el componente de biblioteca se amplía con funciones homomórficas adicionales que se construyen sobre otras funciones homomórficas. El componente de biblioteca está configurado para procesar la entrada del usuario, fija cotas sobre operaciones computacionales y o bien ejecuta dichas operaciones o bien emite errores si no puede garantizarse la corrección. En uno de los aspectos, el componente de biblioteca informa al usuario cuando una salida descifrada proveniente de una función homomórfica no es igual al resultado de un mismo cálculo sobre la entrada si dicha entrada no hubiese estado cifrada.

Se pondrán de manifiesto otras ventajas a partir de la siguiente descripción detallada cuando la misma se considere en combinación con los dibujos.

Breve descripción de los dibujos

La presente invención se ilustra a título de ejemplo y no queda limitada en las figuras adjuntas en las cuales los números de referencia equivalentes indican elementos similares, y en las cuales:

la FIG. 1 es un diagrama de bloques que ilustra un sistema de ejemplo para gestionar cálculos seguros sobre datos cifrados, de acuerdo con una implementación ejemplificativa.

La FIG. 2 es un diagrama de flujo que ilustra etapas ejemplificativas para seleccionar automáticamente parámetros que dirigen la ejecución de operaciones computacionales, de acuerdo con una implementación ejemplificativa.

La FIG. 3 es un diagrama de flujo que ilustra etapas ejemplificativas para implementar un esquema de cifrado homomórfico por niveles, según una implementación ejemplificativa.

La FIG. 4 es un diagrama de flujo que ilustra etapas ejemplificativas para interaccionar con un servicio de cálculo por medio de uno o más componentes de biblioteca, de acuerdo con una implementación ejemplificativa.

La FIG. 5 es un diagrama de bloques que representa entornos en red no limitativos, de ejemplo, en los cuales se pueden implementar varias realizaciones descritas en la presente.

La FIG. 6 es un diagrama de bloques que representa un sistema informático o entorno operativo no limitativo, de ejemplo, en el cual se pueden implementar uno o más aspectos de varias realizaciones que se describen en la presente.

Descripción detallada

Varios aspectos de la tecnología que se describen en la presente van dirigidos en general a una biblioteca configurada para realizar cálculos sobre datos cifrados, a través de una red. El acceso a la funcionalidad de la biblioteca se puede proporcionar por medio de un recurso de red, tal como un servidor de red de área local o un entorno informático en la nube. El recurso de red puede ser no fiable por parte de otros dispositivos informáticos a los que se hace referencia en la presente como clientes. Por tanto, el cliente puede usar la biblioteca para evitar que cualquier dispositivo informático adversario descifre los datos cifrados durante la transmisión y/o mientras están almacenados en memoria en el recurso de red. De acuerdo con una implementación ejemplificativa, componentes de la biblioteca construyen un esquema de cifrado totalmente homomórfico, el cual se puede usar para externalizar de forma privada operaciones computacionales en el recurso de red, cuando se están cargando datos desde dispositivos informáticos distribuidos, al mismo tiempo que se logra un cierto nivel de corrección y seguridad de los datos.

Algunas realizaciones de la biblioteca implementan un esquema de cifrado totalmente homomórfico, por niveles, basado en problemas de retículos ideales, tales como el problema de aprendizaje con errores en anillos. Uno de los esquemas de cifrado ejemplificativo se puede basar en la dificultad cuántica de problemas del tipo vector más corto en retículos ideales. El esquema de cifrado totalmente homomórfico, por niveles, descrito en la presente, reduce el tamaño del texto cifrado y elimina la expansión del texto cifrado en la multiplicación homomórfica simplificando la funcionalidad de conmutación de clave. Dicho esquema también puede ser invariante a escala y, por lo tanto, anular el uso de la funcionalidad de conmutación del módulo.

El esquema de cifrado totalmente homomórfico, por niveles, también puede reducir la complejidad global en la que se incurre en el cálculo homomórfico utilizando módulos de texto plano pequeños, independientes, que posteriormente se combinan (por medio del Teorema chino del resto (CRT)) en un módulo del texto plano mayor. Dicha reducción puede dar como resultado tamaños más eficientes de texto plano, texto cifrado y una probabilidad menor de llegar a una cantidad máxima de cálculo y/o similares. En una implementación de ejemplo, en primer lugar se codifica un elemento de datos para producir un conjunto de valores codificados, y a continuación cada valor se cifra como un texto cifrado. Cálculos llevados a cabo sobre el elemento de datos pueden procesar cada texto cifrado individualmente, lo cual se puede facilitar a través de un tamaño de texto cifrado pequeño, y pueden combinar resultados del procesado en un texto cifrado de mayor tamaño.

Para ilustrar un ejemplo, la codificación de enteros utilizando una técnica basada en el CRT, que se describe en la presente, mejora la precisión de los cálculos puesto que cada entero se transforma en enteros de un tamaño más eficiente (por ejemplo, más pequeños). La técnica basada en el CRT también permite la codificación de enteros grandes reduciendo cada entero grande en enteros más pequeños que se procesan por separado y se combinan en un resultado correcto. Un entero hasta una cota B se puede codificar como un conjunto de enteros de entre los cuales cada entero se codifica hasta una cota t_i. Pueden llevarse a cabo correctamente operaciones computacionales sobre el conjunto de enteros, dado que cada módulo t_i es co-primo y el producto sobre todo t_i es mayor que la cota B. Cada entero x mod t_i del conjunto se puede cifrar y a continuación, se puede procesar en paralelo para devolver resultados cifrados los cuales seguidamente se descifran/descodifican con el fin de recuperar un entero original.

Debe entenderse que todos los ejemplos en la presente son no limitativos. Como tal, la presente invención no se

limita a ninguna realización, aspecto, concepto, estructura, funcionalidad o ejemplo particular que se describe en la presente. Por el contrario, todas las realizaciones, aspectos, conceptos, estructuras, funcionalidades o ejemplos que se describen en la presente son no limitativos, y la presente invención se puede utilizar de diversas maneras que proporcionan beneficios y ventajas en la computación y en general cálculos seguros.

La FIG. 1 es un diagrama de bloques que ilustra un sistema ejemplificativo para gestionar cálculos seguros sobre datos cifrados de acuerdo con una implementación ejemplificativa. Una pluralidad de dispositivos informáticos, representados en forma de una pluralidad de clientes 102 en la FIG. 2, utiliza un recurso 104 de red para llevar a cabo diversas tareas informáticas. Los componentes ejemplificativos del sistema de ejemplo pueden incluir además una parte 106 de presentación (*front end*) de biblioteca, que se ejecuta en cualquier cliente 102 de ejemplo, configurada para interaccionar con una parte interna (*back end*) 108 de biblioteca, que se ejecuta en el recurso 104 de red. La parte 106 de presentación de biblioteca se refiere en general a un conjunto de funciones (por ejemplo, una interfaz de programación de aplicaciones (API)) que comunican varios datos, instrucciones, órdenes y/o similares a un proceso configurado para gestionar dichas comunicaciones para el recurso 104 de red según se describe en la presente.

Lo siguiente hace referencia a realizaciones en las cuales el recurso 104 de red facilita la provisión de servicios, tales como servicios de cálculo, al cliente 102 de ejemplo. Varios dispositivos informáticos, incluyendo una máquina física o máquina virtual a la que se hace referencia en la presente como servidor, pueden funcionar dentro del recurso 104 de red y llevar a cabo cálculos sobre conjuntos de datos almacenados, tales como un conjunto de datos ejemplificativo que comprende datos cifrados 110. La arquitectura ejemplificativa correspondiente al recurso 104 de red desacopla hardware dedicado, con respecto a software, de tal manera que cada componente de hardware/software se puede virtualizar en unidades, las cuales a continuación se pueden agrupar adicionalmente en torno a la funcionalidad. Cuando se lleva a cabo una tarea, el recurso 104 de red puede automatizar el aprovisionamiento y la configuración de algunas unidades, de tal manera que cada unidad lleva a cabo una parte de la tarea en paralelo.

El recurso 104 de red puede configurar uno o más componentes de la parte interna 108 de biblioteca para que funcionen en diversos entornos informáticos (por ejemplo, una agrupación de ordenadores locales, un entorno informático privado en la nube, un entorno informático público en la nube y/o un entorno informático híbrido). Al codificar/cifrar el conjunto de datos de ejemplo y al ejecutar diversas operaciones computacionales sobre el conjunto de datos de ejemplo, aunque manteniendo el conjunto de datos ejemplificativo en un estado cifrado, la parte interna 108 de biblioteca protege al conjunto de datos ejemplificativo con respecto un acceso no autorizado, una modificación y/o una mala aplicación, por ejemplo por parte de un recurso no fiable.

Según se describe de forma adicional posteriormente, la parte 108 de presentación de biblioteca proporciona acceso a un mecanismo 112 de cifrado que proporciona funcionalidad con respecto a un esquema 114 de cifrado homomórfico. Dicha funcionalidad permite que un módulo 116 de cálculo seleccione automáticamente datos 118 de parámetros para ejecutar dinámicamente cálculos seguros sobre los datos 110 de cifrado.

Por medio de una serie de implementaciones del esquema 114 de cifrado homomórfico, uno o más componentes de la parte interna 108 de biblioteca evalúan de manera correcta y segura un circuito de cifrado convencional de una profundidad/nivel particular. Por lo menos algunas implementaciones del esquema 114 de cifrado homomórfico incluyen un esquema totalmente homomórfico, por niveles, en el cual un usuario introduce una profundidad/nivel de circuito (por ejemplo, circuito de profundidad tres (3)/nivel cuatro (4) para un cifrado de 128 bits). En lugar de aplicar un procedimiento de autoevaluación o uno que utilice un cifrado a nivel de bits, el esquema de cifrado homomórfico, por niveles, es invariante a escala al limitar el aumento de ruido inherente y eliminar la técnica de conmutación del módulo.

La parte 106 de presentación de biblioteca se puede usar para construir un servicio de cálculo privado en la nube con el fin de externalizar cálculos sobre los datos cifrados 110. El módulo 116 de cálculo se puede configurar para establecer automáticamente cotas sobre el tamaño del conjunto de datos para los datos cifrados. Estas cotas se pueden determinar sobre la base de entradas de usuario que incluyen nivel de seguridad deseado, una o más operaciones computacionales, cantidad y tipo de datos a gestionar y/o similares. Cada operación computacional puede hacer referencia a una única función homomórfica o a una serie de dichas funciones. Una función homomórfica ejemplificativa se puede configurar para evaluar circuitos de cifrado, por ejemplo a través de la suma o multiplicación de textos cifrados. Una implementación ejemplificativa de la operación de cálculo combina estas funciones para proporcionar una funcionalidad adicional, por ejemplo con fines estadísticos, para modelado predictivo, aprendizaje automático y/o similares. Con fines ilustrativos, el usuario, por medio de la parte 106 de presentación de biblioteca, puede ordenar a la parte interna 108 de biblioteca que use las funciones de suma y/o multiplicación para calcular una media, una desviación típica, valores de regresión y otros datos estadísticos. La parte 106 de presentación de biblioteca se puede programar para usar funciones de evaluación homomórfica con el fin de inicializar y/o entrenar un clasificador lineal.

La configuración de una instancia del esquema 114 de cifrado homomórfico para garantizar seguridad y corrección se puede basar en suposiciones de dificultad computacional relacionadas con problemas bien conocidos de retículos, tales como el Problema del Vector Más Corto (SVP). Específicamente, implementaciones del esquema 114

de cifrado homomórfico basadas en el Aprendizaje con Errores en Anillos (RLWE) utilizan anillos de polinomios, en donde se representan polinomios como vectores en un retículo, y se truncan estos anillos de polinomios usando diversas técnicas. Las técnicas ejemplificativas incluyen la reducción de un tamaño/espacio de textos cifrados y/o un tamaño/espacio de mensajes de texto plano según un factor del módulo, la restricción de un espacio de claves criptográficas a una distribución acotada, la codificación de elementos de anillos de polinomios utilizando una técnica basada en el Teorema Chino del Resto (CRT) y otros. Una distribución acotada de claves criptográficas, a partir de la cual por ejemplo se muestrean elementos de anillos de polinomios que representan claves criptográficas, puede lograr la restricción del espacio de claves. La codificación de un elemento de datos enteros, incluyendo un entero grande, en forma de una colección de enteros más pequeños, permite cálculos eficientes sobre ese elemento de datos enteros.

En general, el supuesto del aprendizaje con errores en anillos (RLWE), que está relacionado con el supuesto del aprendizaje con errores (LWE), hace referencia a evitar que un adversario distinga una secuencia de muestras con respecto a pares aleatorios de elementos de anillos de polinomios. Cabe apreciar que los elementos de un anillo de polinomios pueden hacer referencia a cualquier tipo de polinomio, tal como un polinomio ciclotómico. Al no poder discernir un polinomio con respecto a otro, sin una potencia y un tiempo de cálculo sustanciales, el adversario no puede descodificar de manera razonable resultados de cálculo basándose en polinomios de entrada. Es bien entendible que supuesto del RLWE se puede reducir a la dificultad de problemas del vector más corto del peor caso en retículos ideales.

Aunque el esquema 114 de cifrado homomórfico lleva a cabo cálculos sobre elementos del anillo R, cada uno de los elementos se puede codificar utilizando un parámetro de codificación de tal manera que un vector codificado de coeficientes polinómicos es idéntico o invertible al polinomio original. De acuerdo con una implementación ejemplificativa, para todos los polinomios que se consideran como elementos del anillo R, el esquema 114 de cifrado homomórfico reduce coeficientes polinómicos en módulo q para producir el vector codificado. El mecanismo 112 de cifrado puede configurar el esquema 114 de cifrado homomórfico para mapear cada elemento de R con un entero dentro de un dominio de enteros de tamaño q. En general, dicho mapeo se puede expresar como una función $r_q(a)$ para indicar la reducción del elemento a al intervalo [0, q).

Usando el esquema 114 de cifrado homomórfico, el mecanismo 112 de cifrado muestrea claves criptográficas y/o polinomios de error aleatorios de distribuciones Gaussianas en las que cada distribución es de una anchura diferente y puede estar acotada a un intervalo específico. Una distribución Gaussiana acotada de claves criptográficas, por ejemplo, representa un espacio de claves restringido a partir del cual se general claves criptográficas eficientes. Un conjunto de claves criptográficas puede incluir una clave pública y una privada obtenidas a partir de una distribución acotada por B_{clave}; al mismo tiempo, a partir de una distribución acotada por B_{err} se deducen errores. Algunas implementaciones ejemplificativas del esquema 114 de cifrado homomórfico generan también otra clave criptográfica a la que se hace referencia como clave de evaluación.

Usando el esquema 114 de cifrado homomórfico, la parte interna 108 de biblioteca puede generar textos cifrados que únicamente se mapean con un solo elemento de anillo, por contraposición a los dos o más elementos que dictaminan en esquemas basados puramente en el aprendizaje con errores en anillos (RLWE). Además, la evaluación del esquema 114 de cifrado homomórfico da como resultado una expansión reducida o inexistente del texto cifrado mientras se ejecuta la multiplicación homomórfica. El mecanismo 112 de cifrado puede basarse en un supuesto de relación polinómica pequeña decisoria (*decisional small polynomial ratio*) (DSPR) para ampliar una construcción básica del esquema 114 de cifrado homomórfico. Por ejemplo, usando una técnica de producto tensorial, el esquema 114 de cifrado homomórfico garantiza que la distribución de claves públicas es estadísticamente similar a una distribución uniforme siempre que los elementos de las claves criptográficas se muestreen a partir de distribuciones Gaussianas de una anchura suficiente.

La FIG. 2 es un diagrama de flujo que ilustra etapas ejemplificativas para seleccionar automáticamente parámetros con el fin de configurar la ejecución de operaciones computacionales de acuerdo con una implementación de ejemplo. Uno o más componentes de hardware/software (por ejemplo, de la parte interna 108 de biblioteca de la FIG. 1) se pueden configurar para llevar a cabo las etapas ejemplificativas. Dichos componentes pueden formar al menos una parte de una biblioteca que facilita cálculos seguros de datos cifrados en nombre de un usuario de un dispositivo informático que está haciendo funcionar a un cliente.

Una implementación ejemplificativa incluye una representación estructural de un anillo R, tal como un anillo de polinomios. Sea d un entero positivo y defínase $R = \mathbb{Z}[X]/\phi_d(X)$ como el anillo de polinomios con coeficientes enteros módulo el polinomio ciclotómico d-ésimo $\phi_d(X) \in \mathbb{Z}[X]$. El grado ϕd es n = φ(d), donde φ es la función de Euler. Los elementos de R se pueden representar por todos los polinomios de Z[X] de grado inferior a n. Los elementos del anillo R son de módulo aritmético ϕd(X), lo cual es implícito siempre que en la presente se describan términos o igualdades que impliquen elementos de R.

Un vector de coeficientes puede representar un elemento arbitrario a $\in$ R como (a_1, a_2, a_3,..., a_{n-1}) donde $a_i \in \mathbb{Z}$. Por tanto, un polinomio para a se puede expresar de la manera siguiente:

$$a = \sum_{i=0}^{n-1} a_i X^i$$

En particular, el elemento a se puede interpretar como un elemento del espacio vectorial de $\mathbb{R}$. espacio $\mathbb{R}^n$. El componente de biblioteca de ejemplo selecciona una norma máxima sobre $\mathbb{R}^n$ para medir la dimensión de los elementos en R. Una implementación ejemplificativa de la norma máxima de a se puede calcular de la manera siguiente:

$$\|a\|_\infty = \max_i\{|a_i|\}$$

Sea χ una distribución de probabilidad sobre R de acuerdo con la cual se muestrean elementos de R. Usando la notación a $\leftarrow \chi$ para indicar que se muestrea a $\in$ R, la distribución χ sobre R está acotada por B para cierto B > 0 si, para todos los elementos a $\leftarrow \chi$, se cumple la desigualdad $\|a\|\infty < B$.

A título de ejemplo de una distribución χ sobre R, una distribución Gaussiana discreta $\mathbb{D}_{\mathbb{Z},\sigma}$ se puede diseñar con una media cero(0) y una desviación típica σ con respecto al conjunto de enteros, lo cual asigna a cada elemento $\chi \in \mathbb{Z}$ una probabilidad proporcional a $\exp(-\pi|x|^2/\sigma)$. Cuando d es una potencia de dos (2) y $\phi d(X) = X^n + 1$, χ puede ser una Gaussiana discreta esférica $\chi = D_{\mathbb{Z}^n,\sigma}$ de tal manera que cada coeficiente de polinomio se muestrea de acuerdo con la distribución unidimensional $\mathbb{D}_{\mathbb{Z},\sigma}$.

La etapa 202 comienza con las etapas de ejemplo de la FIG. 2 y prosigue hacia la etapa 204 donde se proporcionan al cliente uno o más componentes de biblioteca para llevar a cabo operaciones computacionales sobre los datos cifrados. Para ilustrar una implementación ejemplificativa, un componente de biblioteca de ejemplo comprende instrucciones configuradas para calcular un conjunto de funciones homomórficas, incluyendo funciones de álgebra lineal (por ejemplo, producto escalar de vectores, multiplicación matricial y/o similares) y/o funciones matemáticas más complejas.

La etapa 206 hace referencia al procesado de entradas de usuario y a la generación de parámetros con respecto a la ejecución de un conjunto de operaciones computacionales. Cierta entrada de usuario puede indicar un nivel de seguridad deseado, un tamaño del conjunto de datos, una precisión del cálculo y/o similares. Sobre la base de dicha entrada de usuario, el componente de biblioteca ejemplificativo selecciona parámetros para configurar dinámicamente la ejecución del conjunto de operaciones computacionales al mismo tiempo que produciendo resultados de cálculo seguros y correctos. Un parámetro de ejemplo incluye un módulo específico, al que se hace referencia como módulo q, que tiene un tamaño variable o fijo (por ejemplo, 128 bits ó 1.024 bits) y que es una potencia de dos o, alternativamente, es un número primo de Mersenne. Otro parámetro ejemplificativo implica fijar un grado n de un polinomio de evaluación ϕd igual a $\varphi(d)$.

El componente de biblioteca ejemplificativo puede usar, alternativamente, datos de parámetros predeterminados, tales como un campo de base y la dimensión de retículos ideales. El componente de biblioteca ejemplificativo puede modificar los datos de los parámetros para adaptarse a cotas y estimaciones nuevas, garantizando corrección y seguridad. La selección automática de estos y/u otros parámetros evita que un sustancial ruido inherente provoque una evaluación imprecisa cuando se ejecutan las operaciones computacionales. La etapa 208 determina si una estimación de ruido para el conjunto de operaciones computacionales es aceptable.

Tal como se describe en la presente, la distribución χ se usa en muchos esquemas de cifrado totalmente homomórfico basados en el supuesto del problema de aprendizaje con errores en anillos (RLWE), por ejemplo para muestrear polinomios de error aleatorios que tienen coeficientes pequeños con una alta probabilidad. Dichos polinomios de error aleatorios son una parte significativa de los cálculos del término de ruido usados en el proceso de cifrado.

Puesto que un término de ruido inherente de norma pequeña permite la recuperación de texto plano (por ejemplo, datos descifrados) a partir de texto cifrado (por ejemplo, datos cifrados), una cota sobre el ruido inherente en un texto cifrado, suponiendo que las distribuciones de claves y de errores están acotadas, garantiza corrección en el esquema de cifrado homomórfico que se describe en la presente.

Para deducir cotas significativas sobre la magnitud y/o el aumento del ruido inherente durante funciones homomórficas, la distribución χ está acotada por B para cierta B. Para la distribución Gaussiana discreta χ, este supuesto es apropiado ya que los elementos muestreados tienden a ser de magnitud pequeña con una alta probabilidad. El acotamiento posibilita un muestreo a partir de una distribución Gaussiana truncada, que es estadísticamente próxima a la Gaussiana discreta verdadera χ si B se elige suficientemente grande. Por ejemplo, si la distribución truncada está acotada por $B = 6\sigma$, todas las muestras están acotadas por B con una probabilidad muy alta.

Calculando por lo menos una cota para la estimación de ruido, el componente de biblioteca de ejemplo establece un nivel particular de seguridad y corrección computacionales. Si la estimación del ruido no consigue cumplir la por lo menos una cota, la etapa 208 prosigue hacia la etapa 216 donde finalizan las etapas ejemplificativas que se representan en la FIG. 2. Si la estimación de ruido se ajusta a la cota de ruido inherente, la etapa 208 prosigue hacia la etapa 210. La etapa 210 va dirigida a una transformación entre cada operación computacional y funciones homomórficas estructuradas. Como ejemplo, el cálculo de un promedio o media en un conjunto implica una serie de sumas homomórficas. Como ejemplo alternativo, el cálculo de un producto escalar entre vectores de magnitud N implica N multiplicaciones homomórficas y N-1 sumas homomórficas. Todavía en otro ejemplo, la operación computacional puede implicar un número de productos escalares de vectores y, posiblemente, otras funciones homomórficas.

La etapa 212 ejecuta las funciones homomórficas estructuradas. Se aprecia que otras implementaciones pueden tener solamente una función homomórfica a ejecutar en la etapa 212. Sin embargo, durante dicha ejecución, el componente de biblioteca ejemplificativo actualiza la estimación de ruido y/o la cota del ruido inherente. Si la estimación del ruido supera la cota del ruido inherente, según una implementación ejemplificativa, el componente de biblioteca de ejemplo emite mensajes de error y/o reconfiguración, por ejemplo, cuando se alcanza una cantidad específica (por ejemplo, máxima) de cálculo o si se introduce un conjunto de datos considerablemente grande. Tras completar la ejecución de las funciones homomórficas estructuradas, el componente de biblioteca de ejemplo lleva a cabo la etapa 214 y comunica resultados cifrados al cliente. La etapa 216 finaliza las etapas de ejemplo descritas en la presente con respecto a la FIG. 2.

La FIG. 3 es un diagrama de flujo que ilustra etapas de ejemplo para implementar un esquema de cifrado homomórfico por niveles de acuerdo con una implementación ejemplificativa. Según se describe en la presente, dicho esquema de cifrado puede aportar a un usuario pruebas que se puede demostrar que son seguras.

Las etapas ejemplificativas pueden remitir al siguiente esquema de cifrado homomórfico por niveles, que se construye por medio de uno o más componentes de hardware/software (por ejemplo, de la parte interna 108 de biblioteca de la FIG. 1). Dicho esquema se parametriza con un módulo q y un módulo de texto plano t donde $1 < t < q$. Los textos cifrados producidos con este esquema son elementos de $R = \mathbb{Z}[X]/\phi_d(X)$, y los textos planos son elementos de R/tR. Las elecciones apropiadas de los módulos t y q y/o de un polinomio de evaluación ciclotómico $\phi d(X)$ que define R facilitan la confidencialidad de los datos y la corrección computacional.

La etapa 302 da inicio a las etapas ejemplificativas de la FIG. 3 y prosigue hacia la etapa 304 donde uno o más componentes de biblioteca acceden a un conjunto de datos y generan un conjunto de claves criptográficas sobre la base de parámetros seleccionados automáticamente. Por ejemplo, el módulo q se puede generalizar para una potencia específica de dos (por ejemplo, 128 bits ó 1.024 bits). Otro parámetro de ejemplo implica la fijación de un grado n del polinomio de evaluación ϕd igual a $\varphi(d)$ basándose en un parámetro de seguridad introducido por el usuario.

Otros parámetros de ejemplo incluyen cotas para distribuciones sobre R. Por ejemplo, una distribución Gaussiana se puede acotar con B_{clave} en un cierto número de desviaciones típicas. Para mostrar una realización ejemplificativa que hace uso de $B_{clave} = 1$, incluso cuando los polinomios f', g tienen coeficientes en {-1, 0, 1}, y la clave pública h es igual a $[tgf^{-1}]_q$, la clave pública h permanece indistinguible con respecto a una clave muestreada a partir de una distribución uniforme. La desviación típica de una distribución de errores χ_{err} sobre R acotada por B_{err} se puede fijar a σ=3:2. La cota de alta probabilidad sobre la magnitud de los coeficientes de errores extraídos a partir de distribuciones Gaussianas se puede seleccionar como 6σ.

Las siguientes instrucciones de ejemplo (que se indican a continuación como “INSTR”) se corresponden con una función homomórfica para generar una clave privada y una clave pública, a lo que se hace referencia como KeyGen(d, q, t, χ_{clave},χ_{err}):

INSTR(1): *Muestrear Polinomios f,g*$\leftarrow\chi_{clave}$ *y sea el Polinomio f=tf'* + 1;

INSTR(2): *Si Polinomio f* $\in$ *R es invertible módulo q,*

existe un Polinomio f^{-1} *tal que* $ff^{-1} = \tilde{1}$, donde $\tilde{1}(X) = \Sigma_i a_i X^i$,

con $a_0 = 1 \bmod q$ *y* $a_j = 0 \bmod q$ *f o toda* $j \neq 0$,

Si no, si Polinomio f no es invertible módulo q, seleccionar un f' nuevo y Repetir;

INSTR(3): *Calcular el* f^{-1} *inverso* $\in$ *R de Polinomio de f módulo q y fijar* $h = [tgf^{-1}]q$;

INSTR(4): *Dar salida a un par de Clave Pública y Clave Privada (pk, sk) = (h, f)* $\in R^2$;

La etapa 306 se refiere a la codificación de ciertos elementos de datos dentro del conjunto de datos. Un elemento de datos incluye datos de texto plano que se mapean con un conjunto de coeficientes enteros de un polinomio representativo en R. El componente de biblioteca ejemplificativo puede utilizar el ampliamente conocido Teorema

Chino del Resto para determinar un módulo q con el fin de codificar los datos de texto plano antes del cifrado. Puesto que el polinomio f descrito en la presente es invertible módulo q, los coeficientes del polinomio se pueden reducir según un entero módulo q. Así, un mapa $[\cdot]_q$ puede reducir un entero x módulo q a un resultado y representa ese resultado mediante un elemento en el intervalo (-q/2, q/2]. El mapa $[\cdot]_q$ se puede ampliar a polinomios en $\mathbb{Z}[X]$ y R aplicando por separado una entrada de mapa apropiada a cada coeficiente según se indica con lo siguiente:

$$[\cdot]q: R \to R, a = \sum_{i=0}^{n-1} a_i X^i \mapsto \sum_{i=0}^{n-1} [a_i]_q X^i$$

De manera similar, la anterior notación se puede modificar para vectores de polinomios aplicando mapeos a entradas de vectores por separado. Una implementación alternativa usa la reducción de entero x módulo q para representar cualquier coeficiente vectorial como un elemento en [0, q). Además del módulo q que se usa para reducir los coeficientes de los elementos que representan textos cifrados, existe un segundo módulo t < q que determina un espacio definido por R/tR (por ejemplo, al que se hace referencia como espacio de mensajes), que representa datos de texto plano/descifrados como polinomios en R módulo t.

La etapa 308 se refiere al cifrado de los elementos de datos codificados usando un esquema de cifrado homomórfico por niveles. Las siguientes instrucciones ejemplificativas (indicadas a continuación con “INSTR”) se corresponden con la función homomórfica a la que se hace referencia como Encrypt(pk,m) (Cifrar(pk,m)) para cifrar un mensaje de texto plano m donde un espacio de mensajes de texto plano se define como R/tR y hace referencia al anillo de polinomios en R módulo t:

INSTR(1): *Para un mensaje de texto plano m + tR, seleccionar $[m]_t$ como representante;*

INSTR(2): *Muestrear Polinomios s, e $\leftarrow \chi_{err}$ y Fijar clave pública h = pk = $[tg\ f^{-1}]_q$*

INSTR(3): *Calcular un mensaje de texto cifrado* $c = [\lfloor q/t \rfloor [m]_t + e + hs] \in R/qR$;

La siguiente ecuación se corresponde con la función homomórfica a la que se hace referencia como Decrypt(sk, c) (Descifrar(sk, c)) para descifrar un mensaje de texto cifrado c:

$$m = \left[\left\lfloor \frac{t}{q} \cdot [f_c]_q \right\rceil \right]_t \in R$$

Una de las realizaciones de ejemplo del esquema de cifrado homomórfico por niveles que se describe en la presente incluye una longitud de palabra w (por ejemplo, un entero positivo w > 1) usada para representar enteros en un sistema de base w. Definiendo $l_{q,w} = \lceil \log_w(q) \rceil$ donde w < q, la expresión $[a]_q = \sum_{i=0}^{l_{q,w}-1} [a_i]_w w^i$ se cumple para cada elemento a de R en el cual cada coeficiente a_i se mapea en un intervalo (-w/2, w/2]. La anterior expresión demuestra claramente que la norma de a_i en la suma es como mucho w/2 y, por lo tanto, dicha expresión se puede usar para determinar una clave de evaluación. Por consiguiente, los mensajes de texto cifrado se pueden fraccionar por el tamaño de palabra w, y después de aplicar una función homomórfica a cada parte, el esquema de cifrado homomórfico por niveles combina cada parte en un mensaje de texto cifrado resultante.

En relación con dicha realización, las siguientes instrucciones (indicadas a continuación con “INSTR”) se corresponden con una función de ejemplo a la que se hace referencia como KeyGen(d,q,t, χ_{clave},χ_{err}, w) que genera una clave privada, una clave pública y una clave de evaluación:

INSTR(1): *Muestrear Polinomios f',g$\leftarrow \chi_{clave}$ y sea el Polinomio f = tf'* + 1;

INSTR(2): *Si Polinomio f $\in$ R es invertible módulo q,*

existe un Polinomio f^{-1} tal que ff^{-1} = $\vec{1}$, donde $\vec{1}(X) = \Sigma_i a_i X^i$,

con a_0 = 1 mod q y a_j = 0 mod q f o toda j $\neq$ 0,

Si no, si Polinomio f no es invertible módulo q, seleccionar un f' nuevo y Repetir;

INSTR(3): *Calcular el f^{-1} inverso $\in$ R de Polinomio de f módulo q, Fijar h = $[tg\ f^{-1}]q$ y Fijar* $l_{q,w} = \lceil \log_w(q) \rceil$;

INSTR(4): *Muestrear* $e, s \leftarrow \chi_{err}^{l_{q,w}}$ y Calcular $\gamma=[P_{q,w}(f) + e + h \cdot s]_q$; INSTR(5): *Dar salida al Triplete de Clave Criptográfica (pk, sk, evk) = (h, f, γ) $\in R^2$*

Una alternativa a las implementaciones anteriores incluye un esquema de cifrado totalmente homomórfico, por niveles, que es seguro de manera demostrable bajo la dificultad cuántica supuesta de problemas de retículos convencionales del peor de los casos. La siguiente descripción se refiere a sistemas de cifrado basados en anillos, que se fundamentan en el supuesto de RLWE y un supuesto de seguridad circular considerando que se cumple el supuesto de DSPR para esta configuración de parámetros. El anillo de polinomios R viene dado por R =Z[x] = X^n + 1 donde n es una potencia de 2 y d = 2n. Además, gcd(*q,t*) = 1 y q = (*mod* d) de tal manera que X^n + 1 se divide en factores lineales diferenciados módulo q. La demostración de que se cumple el supuesto de DSPR requiere garantizar uniformidad de las claves públicas de acuerdo con cualquier técnica pertinente. Es necesario que la desviación típica de la distribución Gaussiana χ_{clave} sea mayor que √q, lo cual hace que la cota B_{clave} sea mayor que √q. Por tanto, la cota para el ruido inherente en un texto cifrado inicial √q. Después de ejecutar una función de multiplicación homomórfica, un método de producto tensorial puede evitar que la cota de ruido inherente total supere un umbral.

Dado un parámetro de seguridad como entrada de usuario, el componente de biblioteca de ejemplo da salida a un ajuste de precisión de datos módulo q, un tamaño de palabra w, una distribución de polinomios de error aleatorios, una distribución de claves criptográficas, un grado n del polinomio de evaluación ϕd igual a φ(d) y/o similares, donde d = 2n es la potencia de 2 que determina $\phi_d(X) = X^n + 1$ para definir R, q > t son módulos. $\mathbb{R}$ y $\mathbb{R}^n$ con desviaciones típicas σ_{err} y σ_{clave}, respectivamente, son distribuciones Gaussianas discretas sobre $\mathbb{Z}^n$ que muestrean elementos invertibles descartando los no invertibles. Con dicha configuración, el componente de biblioteca de ejemplo muestrea $a, s \in \chi_{err}^{\xi_w}$, calcula $\gamma = [f^{-1}P_{q,w}(D_{q,w}(f) \otimes D_{q,w}(f)) + e + \mathbb{Z}^n$ y da salida a una tripleta que comprende una clave pública (pk), una clave privada (sk) y una clave de evaluación (evk) iguales a los polinomios (h, f, y).

La etapa 310 representa una generación de función homográfica básica. El componente de biblioteca ejemplificativo usa datos de parámetros para completar la configuración de diversas especificaciones de la función homográfica. Una de las funciones homomórficas de ejemplo se refiere a una función Add(c_1, c_2) definida para calcular una suma de textos cifrados de entrada c1, c2 con la siguiente ecuación:

$$c_{add} = [c_1 + c_2]_q$$

Otra función homomórfica de ejemplo incluye una función Mult (c_1, c_2) que calcula una multiplicación de textos cifrados de entrada c_1, c_2 usando una clave de evaluación evk con la siguiente ecuación:

$$c_{mult} = \text{KeySwitch}\left(\left\lfloor \frac{t}{q} c_1 c_2 \right\rceil, \text{evk}\right)$$

Como otro ejemplo de función homomórfica, una función KeySwitch transforma el texto cifrado $\tilde{c}_{mult}$ que cifra el producto $[m_1 m_2]_t$ de texto plano m1 y m2, el cual es recuperable utilizando la clave de evaluación evk, en un texto cifrado c_{mult} que puede descifrarse con la clave privada original pk. Alternativamente, el componente de biblioteca de ejemplo utiliza una función de conmutación de clave conocida para construir funciones de multiplicación homomórfica.

El componente de biblioteca ejemplificativo puede utilizar otros componentes de software/hardware para llevar a cabo una reducción de módulo donde coeficientes polinómicos se reducen a escala según un factor, por ejemplo con una reducción según el módulo q. La función homomórfica de reducción de módulo se puede aplicar a cualquier conjunto de coeficientes polinómicos en el anillo R, incluyendo datos de texto plano (por ejemplo, descifrado) y/o datos de texto cifrado (por ejemplo, descifrado). Por consiguiente, un conjunto original de coeficientes y un conjunto codificado de coeficientes son congruentes entre sí módulo q, donde q puede ser igual a dos (2), una potencia de 2, un número primo y o similares. Dicha función se puede utilizar para codificar los datos de texto plano antes del cifrado, lo cual limita el tamaño del texto cifrado, reduce el ruido inherente (por ejemplo, la magnitud), mejora la latencia del cálculo y/o proporciona ventajas adicionales. En relación con las implementaciones ejemplificativas que implican codificación de datos, según se describe con respecto a la etapa 306, una congruencia entre un texto plano original (mensaje m) y un texto plano codificado (mensaje codificado m') se puede expresar por medio de la siguiente ecuación:

$$m' \equiv m \ (mod\ q)$$

La etapa 312 va dirigida a la construcción de otras funciones homográficas. El componente de biblioteca ejemplificativo puede utilizar componentes de software/hardware para llevar a cabo una función de álgebra lineal. Una de las funciones homomórficas ejemplificativas ejecuta una operación de producto escalar del mensaje de texto plano que comprende un conjunto de coeficientes de vectores en R con otro conjunto de coeficientes de vectores en R. En una de las implementaciones de ejemplo, el componente de biblioteca ejemplificativo construye sobre instancias de dicha función homomórfica de producto escalar para proporcionar una funcionalidad de multiplicación matricial homomórfica en la que cada entrada matricial comprende un conjunto de coeficientes de vectores en R. De

manera similar a las funciones de Mult() y Encrypt(), una función homomórfica de estimación de ruido puede estimar una magnitud de ruido inherente y proyectar la influencia de dicho ruido sobre los cálculos futuros de seguridad o corrección.

Todavía otra función homomórfica de ejemplo calcula una estimación de ruido que indica la magnitud de ruido cuando el texto plano se cifra en texto cifrado. Esta función puede acoplar la estimación de ruido para el texto cifrado con el anillo R. Opcionalmente, esta función calcula una estimación de ruido asociada a una ejecución de cualquier otra función homomórfica, incluyendo cualquiera de las funciones que se describen en la presente, tales como Add() o Mult(), u otras funciones homomórficas, tales como aquellas correspondientes a la implementación de otros esquemas de cifrado.

Usando la notación descrita en la presente, se pueden modelar términos de ruido inherente cuando se estiman aumentos de ruido para cualquier función homomórfica. La siguiente descripción incluye detalles sobre la determinación de cotas para dichos términos de ruido cuando la longitud de palabra se fija a w y $l_{q,w} = \lceil \log_w(q) \rceil$, y v_i en R indica un término de ruido inherente ejemplificativo en c_i. Por tanto, suponiendo que c_{add} almacena resultados a partir de ejecutar la función Add(c_1,c_2), un término de ruido correspondiente v_{add} se acota de acuerdo con la expresión $\|v_{add}\|_\infty \leq \|v_1\|_\infty + \|v_2\|_\infty + r_t(q)$. Si c_{mult} hace referencia a un producto cifrado de dos textos cifrados, la desigualdad subsiguiente acota un término de ruido correspondiente v_{mult}:

$$\|v_{mult}\|_\infty \leq \frac{1}{2}\delta\left(t\left(2+\delta t B_{key}\right)\left(\|v_1\|_\infty + \|v_2\|_\infty\right) + \left(1+r_t(q)\right)\min_i \|v_2\|_\infty\right)$$
$$+\frac{1}{2}\left(1+\left(\delta t B_{key}\right)^2 + \left(r_t(q)\delta t(3+\delta t B_{key})\right)\right) + \delta^2 t l_{q,w} w B_{key} B_{err}$$

Los términos de ruido descritos anteriormente se refieren a la suma y multiplicación homomórficas, pero dichos términos se pueden interpolar cuando se deducen cotas para operaciones computacionales que implican otras funciones homomórficas. A continuación, las cotas obtenidas se pueden usar para reducir parámetros adecuados con el fin de configurar automáticamente cálculos futuros de corrección y seguridad usando el esquema de cifrado totalmente homomórfico, por niveles, descrito en la presente.

De acuerdo con una implementación ejemplificativa en la cual una estructura, por niveles, de funciones de multiplicación representa un conjunto de operaciones computacionales, se supone que los textos cifrados en cada nivel presentan términos de ruido inherente sustancialmente iguales en cuanto a magnitud. Dicho supuesto puede constituir una aproximación a términos de ruido inherente no equilibrados, lo cual da como resultado estimaciones precisas de cotas de ruido.

Por tanto, el componente de biblioteca de ejemplo puede construir una estructura que comprende una serie de funciones Mult() que se ejecutan de manera iterativa. Una función de redondeo homomórfica puede conllevar un número considerable de multiplicaciones consecutivas. El esquema homomórfico por niveles puede utilizar un escalado por números racionales de tal manera que los polinomios resultantes presenten coeficientes racionales en lugar de coeficientes enteros. Cuando se aplica, la función de redondeo vuelve a convertir los coeficientes racionales en los coeficientes enteros correspondientes. Funciones homográficas ejemplificativas adicionales incluyen funciones basadas en normas de cifrado, tales como funciones AES (por ejemplo, AddKey, SubBytes, ShiftRows, MixColumns y/o similares) y/o similares.

La etapa 314 hace referencia a la provisión de acceso a las funciones homográficas a una pluralidad de clientes (por ejemplo, dispositivos informáticos). Por medio del componente de biblioteca de ejemplo, cualquier cliente puede interaccionar con y usar realizaciones de las otras funciones homomórficas a través de mecanismos de hardware (por ejemplo, conjuntos de instrucciones de microprocesador) y/o mecanismos de software (por ejemplo, bibliotecas de software basadas en controladores). La etapa 316 finaliza las etapas de ejemplo de la FIG. 3.

La FIG. 4 es un diagrama de flujo que ilustra etapas de ejemplo para interaccionar con un servicio de cálculo por medio de uno o más componentes de biblioteca de acuerdo con una implementación ejemplificativa. El servicio de cálculo proporciona acceso a cierta funcionalidad basada en software/hardware, disponible en un recurso de red (por ejemplo, un recurso informático privado en la nube). Un usuario que hace funcionar un dispositivo informático, al que se hace referencia en la presente como cliente, puede usar el servicio de cálculo para configurar automáticamente un esquema de cifrado totalmente homomórfico, por niveles, con el fin de efectuar estos cálculos seguros.

Según se describe en la presente, un componente de biblioteca de ejemplo (por ejemplo, una parte 106 de presentación de biblioteca de la FIG. 1) se puede configurar para realizar por lo menos algunas de las etapas de ejemplo iniciando cálculos seguros sobre datos cifrados y/o analizando todos los resultados de dichos cálculos. La etapa 302 inicia las etapas de ejemplo y prosigue a la etapa 304 donde se establecen parámetros de seguridad. Tal como se indica en la presente exposición, el parámetro de seguridad define ciertos ajustes deseados, incluyendo un nivel de profundidad para un circuito de cifrado normalizado. El usuario, por ejemplo, puede solicitar un cifrado totalmente homomórfico de 128 bits en un circuito de AES de nivel de profundidad N (por ejemplo, 3). Introduciendo el parámetro de seguridad como entrada en el servicio de cálculo, el usuario puede indicar un nivel de corrección,

por ejemplo, en términos de precisión, y/o seguridad, por ejemplo, en términos de indistinguibilidad para un adversario. El servicio de cálculo devuelve un conjunto de parámetros adicionales para configurar el esquema de cifrado totalmente homomórfico, por niveles, de una manera que satisfaga sustancialmente el parámetro de seguridad.

La etapa 406 se refiere al uso del servicio de cálculo para cifrara elementos de datos de un conjunto de datos. Tal como se describe en la presente, usando fundamentos que se encuentran en el Teorema Chino del Resto (CRT), el componente de biblioteca de ejemplo puede ordenar al servicio de cálculo que codifique cada elemento de datos cifrado, de tal manera que el adversario no pueda distinguir entre ese elemento de datos y otro elemento de datos aleatorio.

La etapa 408 selecciona una operación computacional para que sea llevada a cabo por el servicio de cálculo. Por consiguiente, la etapa 408 procede a emitir una orden por medio de la etapa 410, la etapa 412 y/o la etapa 414. Si la operación de cálculo se refiere a una función de suma homomórfica, la etapa 408 prosigue hacia la etapa 410. Si la orden va dirigida a una función de multiplicación homomórfica, la etapa 408 prosigue hacia la etapa 412. Si el usuario desea resultados de otra función homomórfica, la etapa 408 prosigue hacia la etapa 414. Aunque algunas operaciones de cálculo pueden incluir una única función homomórfica, otras operaciones computacionales implican la ejecución de funciones homomórficas estructuradas en las cuales una disposición de llamadas a funciones lleva a cabo cualquier tipo de cálculo matemático. Por ejemplo, una serie de llamadas a funciones se puede configurar para entrenar un clasificador lineal.

La etapa 416 determina si procesar resultados de cálculos o llevar a cabo otra operación computacional. La etapa 416 vuelve a la etapa 408 si, por ejemplo, el cliente está ejecutando un proceso ocupado en un cálculo continuo, tal como una aplicación en línea de aprendizaje automático. No obstante, la etapa 416 prosigue hacia la etapa 418 cuando los resultados de los cálculos van a ser analizados. Puesto que los resultados tanto cifrados como de los cálculos permanecen en un estado cifrado durante toda la operación computacional, el esquema de cifrado evita que un recurso no fiable descifre los datos cifrados durante la transmisión y/o el almacenamiento. Un adversario en un servidor anfitrión no fiable o conectado de otra manera al cliente, por ejemplo, no puede distinguir entre dos sumas de pares de elementos de datos aleatorios. La etapa 418 hace referencia al componente de biblioteca de ejemplo que usa el servicio de cálculo para descifrar los resultados de cálculo cifrados. La etapa 420 finaliza las etapas de ejemplo representadas en la FIG. 4.

Ejemplos de entornos en red y distribuidos

Aquellos con conocimientos habituales en la materia pueden apreciar que las diversas realizaciones y métodos descritos en la presente se pueden implementar en relación con cualquier ordenador u otro cliente o dispositivo servidor, que se puede desplegar como parte de una red de ordenadores o en un entorno informático distribuido, y se puede conectar a cualquier tipo de medio o medios de almacenamiento de datos. A este respecto, las diversas realizaciones descritas en la presente se pueden implementar en cualquier entorno o sistema de ordenadores que tenga un número cualquiera de unidades de memoria o almacenamiento, y un número cualquiera de aplicaciones y procesos que discurran sobre un número cualquiera de unidades de almacenamiento. Esto incluye, aunque sin carácter limitativo, un entorno con ordenadores de servidor y ordenador de cliente desplegados en un entorno de red o un entorno informático distribuido, con medios de almacenamiento remotos o locales.

La informática distribuida proporciona compartición de recursos y servicios de ordenador mediante intercambio comunicativo entre dispositivos y sistemas informáticos. Estos recursos y servicios incluyen el intercambio de información, almacenamiento en memoria caché y almacenamiento en disco para objetos, tales como archivos. Estos recursos y servicios incluyen también la compartición de capacidad de procesado sobre múltiples unidades de procesado para el equilibrado de la carga, la expansión de recursos, la especialización del procesado, y similares. La informática distribuida saca provecho de la conectividad en red, permitiendo que los clientes hagan uso de su capacidad de conjunto para beneficiar al proyecto en su totalidad. A este respecto, una variedad de dispositivos puede tener aplicaciones, objetos o recursos que pueden participar en los mecanismos de gestión de recursos según se ha descrito para varias realizaciones de la exposición en cuestión.

La FIG. 5 proporciona un diagrama esquemático de un ejemplo de entorno informático en red o distribuido. El entorno informático distribuido comprende objetos informáticos 510, 512, etcétera, y objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, los cuales pueden incluir programas, métodos, medios de almacenamiento de datos, lógica programable, etcétera, según se representa mediante las aplicaciones ejemplificativas 530, 532, 534, 536, 538. Puede apreciarse que los objetos informáticos 510, 512, etcétera, y los objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, pueden comprender diferentes dispositivos, tales como asistentes personales digitales (PDAs), dispositivos de audio/vídeo, teléfonos móviles, reproductores de MP3, ordenadores personales, ordenadores portátiles.

Cada objeto informático 510, 512, etcétera, y los objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, se pueden comunicar con otro u otros objetos informáticos 510, 512, etcétera, y objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, por medio de la red 540 de comunicaciones, de manera o bien directa o indirecta. Aunque se ilustra como un único elemento en la FIG. 5, la red 540 de comunicaciones puede

comprender otros objetos informáticos y dispositivos informáticos que proporcionen servicios al sistema de la FIG. 5, y/o puede representar múltiples redes interconectadas, las cuales no se muestran. Cada objeto informático 510, 512, etcétera, u objeto o dispositivo informático 520, 522, 524, 526, 528, etcétera, también puede contener una aplicación, tal como las aplicaciones 530, 532, 534, 536, 538, que podría hacer uso de una API, u otro objeto, software, microprograma y/o hardware, adecuado para la comunicación con o la implementación de la aplicación proporcionada de acuerdo con varias realizaciones de la exposición en cuestión.

Existe una variedad de sistemas, componentes, y configuraciones en red que soportan entornos informáticos distribuidos. Por ejemplo, los sistemas informáticos se pueden conectar entre sí a través de sistemas por cable o inalámbricos, por redes locales o redes ampliamente distribuidas. En la actualidad, muchas redes están acopladas a Internet, lo cual proporciona una infraestructura para informática ampliamente distribuida y abarca muchas redes diferentes, aunque puede usarse cualquier infraestructura de red, por ejemplo, las comunicaciones que sean inherentes a los sistemas que se describen en diversas realizaciones.

Así, puede utilizarse una multitud de topologías de red e infraestructuras de red, tales como cliente/servidor, entre entidades pares, o arquitecturas híbridas. “Cliente” es un miembro de una clase o grupo que usa los servicios de otra clase o grupo con el cual no está relacionado. Un cliente puede ser un proceso, por ejemplo, en términos generales un conjunto de instrucciones o tareas, que solicita un servicio proporcionado por otro programa o proceso. El proceso de cliente utiliza el servicio solicitado sin necesidad de “conocer” ningún detalle de funcionamiento sobre el otro programa o el propio servicio.

En una arquitectura de cliente/servidor, particularmente un sistema en red, un cliente es habitualmente un ordenador que accede a recursos de red compartidos y proporcionados por otro ordenador, por ejemplo, un servidor. En la ilustración de la FIG. 5, como ejemplo no limitativo, los objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, se pueden considerar como clientes, y los objetos informáticos 510, 512, etcétera, se pueden considerar como servidores en donde los objetos informáticos 510, 512, etcétera, que actúan como servidores proporcionan servicios de datos, tales como la recepción de datos desde objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, de cliente, almacenamiento de datos, procesado de datos, transmisión de datos a objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, de cliente, aunque cualquier ordenador puede considerarse un cliente, un servidor, o ambas opciones, en función de las circunstancias.

Un servidor es típicamente un sistema de ordenador remoto accesible a través de una red remota o local, tal como Internet o infraestructuras de red inalámbricas. El proceso de cliente puede estar activo en un primer sistema de ordenador, y el proceso de servidor puede estar activo en un segundo sistema de ordenador, comunicándose entre sí a través de un medio de comunicaciones, lo cual proporciona una funcionalidad distribuida y permite que múltiples clientes saquen provecho de las capacidades de recopilación de información del servidor.

En un entorno de red en el cual la red 540 de comunicaciones o bus es Internet, por ejemplo, los objetos informáticos 510, 512, etcétera, pueden ser servidores web con los cuales se comunican otros objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, por medio de cualquiera de entre una serie de protocolo conocidos, tales como el protocolo de transferencia de hipertexto (HTTP). Los objetos informáticos 510, 512, etcétera, que actúan como servidores, también pueden servir de clientes, por ejemplo, los objetos o dispositivos informáticos 520, 522, 524, 526, 528, etcétera, tal como puede ser característico de un entorno informático distribuido.

Dispositivo informático de ejemplo

Tal como se ha mencionado, de forma ventajosa, las técnicas descritas en la presente se pueden aplicar a cualquier dispositivo. Puede entenderse, por lo tanto, que para su uso en relación con las diversas realizaciones se contemplan dispositivos informáticos y objetos informáticos de mano, portátiles y otros de cualquier tipo. Por consiguiente, el ordenador remoto de propósito general que se describe a continuación en la FIG. 6 no es más que un ejemplo de un dispositivo informático.

Las realizaciones se pueden implementar parcialmente por medio de un sistema operativo, para ser usado por un desarrollador de servicios para un dispositivo u objeto, y/o se pueden incluir dentro de software de aplicación que funciona de manera que ejecuta uno o más aspectos funcionales de las diversas realizaciones que se describen en la presente. El software se puede describir en el contexto general de instrucciones ejecutables por ordenador, tales como módulos de programas, que son ejecutados por uno o más ordenadores, tales como estaciones de trabajo de cliente, servidores u otros dispositivos. Aquellos versados en la materia apreciarán que los sistemas de ordenador presentan una variedad de configuraciones y protocolos que se pueden utilizar para comunicar datos, y por lo tanto, ninguna configuración o protocolo particular se considera como limitativo.

Así, la FIG. 6 ilustra un ejemplo de un entorno adecuado 600 de sistema informático en el cual pueden implementarse uno o más aspectos de las realizaciones descritas en la presente, aunque tal como se ha puesto de manifiesto anteriormente, el entorno 600 de sistema informático es solamente un ejemplo de entorno informático adecuado y no pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad. Adicionalmente, el entorno 600 de sistema informático no está destinado a interpretarse de manera que presente dependencia alguna en relación con un componente cualquiera o una combinación de componentes cualquiera ilustrados en el entorno

ejemplificativo 600 de sistema informático.

En referencia a la FIG. 6, un dispositivo remoto de ejemplo para implementar una o más realizaciones incluye un dispositivo informático de propósito general en forma de un ordenador 610. Los componentes del ordenador 610 pueden incluir, aunque sin carácter limitativo, una unidad 620 de procesado, una memoria 630 de sistema, y un bus 622 de sistema que acopla varios componentes del sistema incluyendo la memoria de sistema a la unidad 620 de procesado.

Típicamente, el ordenador 610 incluye una variedad de soportes legibles por ordenador y puede ser cualesquiera soportes disponibles a los que se pueda acceder por medio del ordenador 610. La memoria 630 de sistema puede incluir soportes de almacenamiento de ordenador en forma de memoria volátil y/o no volátil, tal como memoria de solo lectura (ROM) y/o memoria de acceso aleatorio (RAM). A título de ejemplo, y sin carácter limitativo, la memoria 630 del sistema también puede incluir un sistema operativo, programas de aplicación, otros módulos de programa, y datos de programa.

Un usuario puede introducir órdenes e información en el ordenador 610 a través de dispositivos 640 de entrada. Un monitor u otro tipo de dispositivo de visualización está conectado también al bus 622 del sistema por medio de una interfaz, tal como la interfaz 650 de salida. Además de un monitor, los ordenadores también pueden incluir otros dispositivos de salida periféricos, tales como altavoces y una impresora, los cuales se pueden conectar a través de la interfaz 650 de salida.

El ordenador 610 puede funcionar en un entorno de red o distribuido usando conexiones lógicas con otro u otros ordenadores remotos, tales como el ordenador remoto 670. El ordenador remoto 670 puede ser un ordenador personal, un servidor, un encaminador, un PC de red, un dispositivo par u otro nodo de red común, o cualquier otro dispositivo remoto de transmisión o que haga uso de los medios, y puede incluir cualquiera o la totalidad de los elementos antes descritos en relación con el ordenador 610. Las conexiones lógicas representadas en la Fig. 6 incluyen una red 672, tal como una red área local (LAN) o una red de área extensa (WAN), pero también pueden incluir otras redes/buses. Dichos entornos de red son habituales en viviendas, oficinas, redes de ordenadores a nivel de empresa, intranets e Internet.

Tal como se ha mencionado anteriormente, aunque se han descrito realizaciones de ejemplo en relación con diversos dispositivos informáticos y arquitecturas de red, los conceptos subyacentes se pueden aplicar a cualquier sistema de red y cualquier dispositivo o sistema informático en el cual resulte deseable mejorar la eficiencia del uso de recursos.

Además, existen múltiples maneras de implementar la misma funcionalidad o una similar, por ejemplo, una API apropiada, un kit de herramientas, código de controladores, un sistema operativo, un objeto de software de control, autónomo o descargable, etcétera, que permita que las aplicaciones y los servicios saquen provecho de las técnicas proporcionadas en la presente. Así, las realizaciones de la presente se contemplan desde el punto de vista de una API (u otro objeto de software), así como desde un objeto de software o hardware que implemente una o más realizaciones que se han descrito en la presente. Por lo tanto, varias realizaciones descritas en el presente documento pueden tener aspectos que estén en su totalidad en hardware, parcialmente en hardware y parcialmente en software, así como en software.

El término “ejemplificativo” se usa en la presente para significar que sirve como ejemplo, instancia o ilustración. Para evitar dudas, la materia en cuestión que se da a conocer en la presente no queda limitada por dichos ejemplos. Adicionalmente, ningún aspecto o diseño descrito en la presente como “ejemplificativo” debe considerarse necesariamente como preferido o ventajoso con respecto a otros aspectos o diseños, ni el mismo pretende excluir estructuras y técnicas ejemplificativas equivalentes y conocidas para aquellos con conocimientos habituales en la materia. Además, en la medida en la que se usan los términos “incluye”, “tiene”, “contiene”, y otros vocablos similares, para evitar dudas, dichos términos pretenden ser inclusivos de una manera similar a la expresión “que comprende”, como vocablo de transición abierta sin excluir ningún elemento adicional o alternativo cuando se utilice en una reivindicación.

Tal como se ha mencionado, las diversas técnicas descritas en la presente se pueden implementar en relación con hardware o software o, cuando resulte apropiado, con una combinación de ambas opciones. Tal como se usa en la presente, los términos “componente”, “módulo”, “sistema” y similares están destinados así mismo a referirse a una entidad relacionada con un ordenador, o bien hardware, o bien una combinación de hardware y software, o bien software, o bien software en ejecución. Por ejemplo, un componente puede ser, aunque sin carácter limitativo, un proceso que se ejecute en un procesador, un procesador, un objeto, un ejecutable, un hilo de ejecución, un programa y/o un ordenador. A título ilustrativo, tanto una aplicación que se ejecuta en ordenador como el ordenador pueden ser un componente. Uno o más componentes pueden residir dentro de un proceso y/o de un hilo de ejecución, y un componente puede estar localizado en un ordenador y/o puede estar distribuido entre dos o más ordenadores.

Los sistemas antes mencionados se han descrito con respecto a la interacción entre varios componentes. Puede apreciarse que dichos sistemas y componentes pueden incluir esos componentes o sub-componentes especificados,

algunos de los componentes o sub-componentes especificados, y/o componentes adicionales, y en concordancia con diversas permutaciones y combinaciones de lo anterior. Los sub-componentes también se pueden implementar como componentes acoplados comunicativamente a otros componentes, en lugar de incluidos dentro de componentes parentales (jerárquicamente). Adicionalmente, puede indicarse que uno o más componentes se pueden combinar en un único componente que proporcione una funcionalidad compuesta o se pueden dividir en varios sub-componentes independientes, y que se pueden proporcionar una o más capas centrales cualesquiera, tales como una capa de gestión, para acoplarse comunicativamente a dichos sub-componentes con el fin de proporcionar una funcionalidad integrada. Todos los componentes descritos en la presente también pueden interaccionar con otro u otros componentes no descritos específicamente en la presente, aunque conocidos en general por aquellos versados en la materia.

A la vista de los sistemas ejemplificativos descritos en la presente, también pueden valorarse metodologías que se puedan implementar de acuerdo con la materia en cuestión descrita, en referencia a los diagramas de flujo de las diversas figuras. Aunque con fines de simplificar la explicación, las metodologías se muestran y describen en forma de una serie de bloques, debe entenderse y apreciarse que las diversas realizaciones no quedan limitadas por el orden de los bloques, en la medida en la que algunos bloques pueden producirse en un orden diferente y/o de manera simultánea con otros bloques, con respecto a lo que se representa y describe en el presente documento. Cuando por medio de un diagrama de flujo se ilustre un flujo no secuencial, o ramificado, puede apreciarse que se pueden implementar otras diversas ramas, trayectos de flujo, y órdenes de los bloques, los cuales logren el mismo resultado o uno similar. Por otra parte, algunos bloques ilustrados son opcionales en la implementación de las metodologías descritas en la presente en lo sucesivo.

Conclusión

Aunque la invención es susceptible de experimentar varias modificaciones y construcciones alternativas, en los dibujos se muestran ciertas realizaciones ilustradas de ellas y las mismas se han descrito anteriormente de manera detallada. No obstante, debe entenderse que no hay intención alguna de limitar la invención a las formas específicas dadas a conocer, sino que, por el contrario, la intención es abarcar todas las modificaciones, construcciones alternativas, y equivalentes que se sitúen dentro del espíritu y alcance de la invención.

Además de las diversas realizaciones descritas en la presente, debe entenderse que pueden usarse otras realizaciones similares o se pueden aplicar modificaciones y adiciones sobre la(s) realización(es) descrita(s) para llevar a cabo la misma función o una equivalente de la(s) realización(es) correspondiente(s) sin desviarse con respecto a las mismas. Todavía adicionalmente, múltiples chips de procesado o múltiples dispositivos pueden compartir la ejecución de una o más funciones descritas en la presente, y de manera similar, el almacenamiento puede materializarse a través de una pluralidad de dispositivos. Por consiguiente, la invención no debe limitarse a ninguna realización individual, sino que más bien debe considerarse en concordancia con las reivindicaciones adjuntas en cuanto a cobertura, espíritu y alcance.

REIVINDICACIONES

1. Método ejecutado al menos parcialmente en por lo menos un procesador de un recurso (104) de red para gestionar cálculos seguros sobre datos cifrados, comprendiendo el método:

procesar un conjunto de datos que comprende los datos cifrados sobre la base de un esquema (114) de cifrado homomórfico que define un único elemento de un anillo de polinomios para cada elemento de datos del conjunto de datos, procesar una entrada (206) de usuario correspondiente a un conjunto de operaciones computacionales sobre el conjunto de datos, seleccionar automáticamente parámetros (206 a 210) correspondientes al conjunto de operaciones computacionales, y configurar la ejecución (212) del conjunto de operaciones computacionales sobre los datos cifrados.

2. Método de la reivindicación 1, en el que el procesado del conjunto de datos comprende además codificar elementos de datos del conjunto de datos para producir datos codificados usando una técnica de codificación basada en el CRT, y usar el esquema (114) de cifrado homomórfico para cifrar los datos codificados y producir los datos cifrados.

3. Método de la reivindicación 1, en el que la selección automática de los parámetros comprende además calcular una cota (208) de ruido para el conjunto de operaciones computacionales.

4. Método de la reivindicación 3, que comprende además determinar por lo menos uno de entre un parámetro de codificación, un grado de un polinomio ciclotómico, o una distribución de claves criptográficas, o determinar los parámetros que minimizan la cota de ruido.

5. Método de la reivindicación 1, que comprende además transformar (210) el conjunto de operaciones computacionales y los parámetros en funciones homomórficas estructuradas.

6. Recurso (104) de red configurado para proporcionar a por lo menos un dispositivo informático (102_1 a 102_N) servicios de cálculo para un conjunto de datos, comprendiendo el recurso (104) de red: un mecanismo (112) de cifrado configurado para generar un conjunto de 5 claves criptográficas a partir de un anillo de polinomios truncado que representa un esquema (114) de cifrado homomórfico por niveles, en donde el esquema (114) de cifrado homomórfico por niveles define un único elemento de un anillo de polinomios para cada elemento de datos del conjunto de datos, y para proporcionar al por lo menos un dispositivo informático (102_1 a 102_N) acceso a por lo menos una función homomórfica configurada para llevar a cabo cálculos sobre el conjunto de datos.

7. Recurso de red de la reivindicación 6, que comprende además un módulo (116) de cálculo configurado para fraccionar un elemento de datos cifrados en una pluralidad de partes de tamaño fijo, ejecutar por lo menos una de las funciones homomórficas sobre cada parte y combinar cada fracción resultante en un elemento resultante de datos cifrados, en donde el módulo (116) de cálculo está configurado además para transformar un conjunto de operaciones computacionales en una estructura que comprende llamadas a funciones homomórficas, en donde el módulo (116) de cálculo usa la estructura para evaluar un circuito de cifrado convencional, en donde el módulo (116) de cálculo está configurado además para seleccionar el circuito de cifrado convencional sobre la base de una entrada de usuario que comprende un parámetro de seguridad, en donde el módulo (116) de cálculo está configurado además para generar automáticamente datos de parámetros con el fin de evaluar el circuito de cifrado convencional con respecto a la ejecución del conjunto de operaciones computacionales.

8. Recurso de red de la reivindicación 6, en el que el mecanismo (112) de cifrado está configurado además para codificar un elemento de datos como un conjunto de valores usando un conjunto de módulos co-primos y construir otra función homomórfica usando la por lo menos una función homomórfica, o en donde el mecanismo (112) de cifrado está configurado además para reducir el tamaño de un texto cifrado según un factor de módulo, o en donde el mecanismo (112) de cifrado está destinado además a restringir un espacio de claves criptográficas muestreando elementos invertibles del anillo de polinomios truncado, o en donde el mecanismo (112) de cifrado está configurado además para implementar el esquema (114) de cifrado homomórfico por niveles basándose en el problema del aprendizaje decisorio con errores en anillos, RLWE, y el problema de la relación polinómica pequeña decisoria, DSPR, en donde el mecanismo (112) de cifrado lleva a cabo una reducción cuántica en retículos ideales.

9. Método para un recurso de red con el fin de proporcionar a por lo menos un dispositivo informático (102_1 a 102_N) servicios de cálculo para un conjunto de datos, comprendiendo el método:

generar un conjunto de claves criptográficas a partir de un anillo de polinomios truncado que representa un esquema (114) de cifrado homomórfico por niveles, en donde el esquema (114) de cifrado homomórfico por niveles define un único elemento del anillo de polinomios para cada elemento de datos del conjunto de datos; y

proporcionar al por lo menos un dispositivo informático (102_1 a 102_N) acceso a por lo menos una función homomórfica configurada para llevar a cabo cálculos sobre el conjunto de datos.

10. Método según la reivindicación 9, que comprende además una o más de las etapas de:

fraccionar un elemento de datos cifrados en una pluralidad de partes de tamaño fijo, ejecutar por lo menos una de las funciones homomórficas sobre cada parte y combinar cada fracción resultante en un elemento resultante de datos cifrados; o transformar un conjunto de operaciones computacionales en una estructura que comprende llamadas a funciones homomórficas y usar la estructura para evaluar un circuito de cifrado; o seleccionar un circuito de cifrado sobre la base de una entrada de usuario que comprende un parámetro de seguridad; o generar automáticamente datos de parámetros con el fin de evaluar un circuito de cifrado con respecto a la ejecución del conjunto de operaciones computacionales; o

codificar un elemento de datos como un conjunto de valores usando un conjunto de módulos co-primos y construir otra función homomórfica usando la por lo menos una función homomórfica; o reducir el tamaño de un texto cifrado según un factor de módulo; o restringir un espacio de claves criptográficas muestreando elementos invertibles del anillo de polinomios truncado; o implementar el esquema (114) de cifrado homomórfico por niveles basándose en el problema del aprendizaje decisorio con errores en anillos, RLWE, y el problema de la relación polinómica pequeña decisoria, DSPR; o llevar a cabo una reducción cuántica en retículos ideales.

11. Método que comprende:

acceder a un servicio de cálculo en un recurso (104) de red a través de una red;

establecer un parámetro de seguridad con el servicio (108) de cálculo que configura un esquema (114) de cifrado homomórfico por niveles con el fin de proporcionar cálculos seguros de un conjunto de datos;

por medio de una biblioteca (108) asociada al servicio de cálculo, cifrar elementos de datos del conjunto de datos de acuerdo con el esquema (114) de cifrado homomórfico por niveles, definiendo el esquema (114) de cifrado homomórfico por niveles un único elemento de un anillo de polinomios para cada elemento de datos del conjunto de datos, y calcular una estimación de ruido inherente para cada elemento de datos; y usar por lo menos un componente de la biblioteca para seleccionar una operación computacional para que sea llevada a cabo por el servicio de cálculo.

12. Método de la reivindicación 11, que comprende además:

usar el por lo menos un componente de la biblioteca para descifrar resultados de cálculo a través de la red.

13. Soporte legible por ordenador que comprende instrucciones ejecutables por ordenador, las cuales, cuando se ejecutan en un procesador (620), provocan que el procesador (620) lleve a cabo el método de acuerdo con cualquiera de las reivindicaciones 1 a 5 ó 9 a 10.

14. Uno o más soportes legibles por ordenador que tienen instrucciones ejecutables por ordenador, las cuales, cuando se ejecutan en un procesador (620), provocan que el procesador (620) lleve a cabo el método de acuerdo con cualquiera de las reivindicaciones 11 ó 12.

15. Dispositivo informático (600) que comprende un procesador (620) acoplado a uno más soportes legibles (630) por ordenador, de acuerdo con la reivindicación 14.

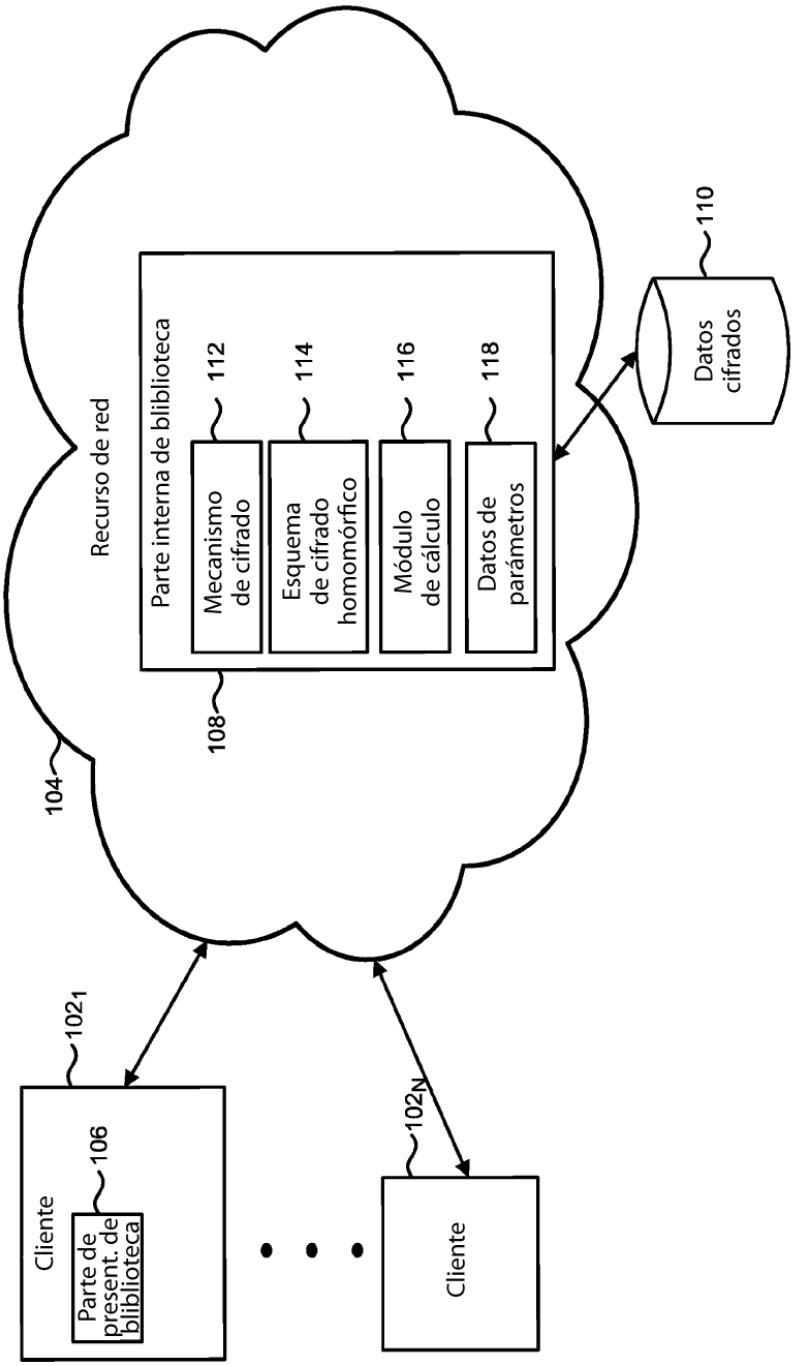
Recurso de red
Parte interna de bliblioteca
Mecanismo de cifrado
Esquema de cifrado homomórfico
Módulo de cálculo
Datos de parámetros
112
114
116
118
108
104
110
Datos cifrados
Cliente
Parte de present. de biblioteca
106
102_1
102_N
Cliente

FIG. 1

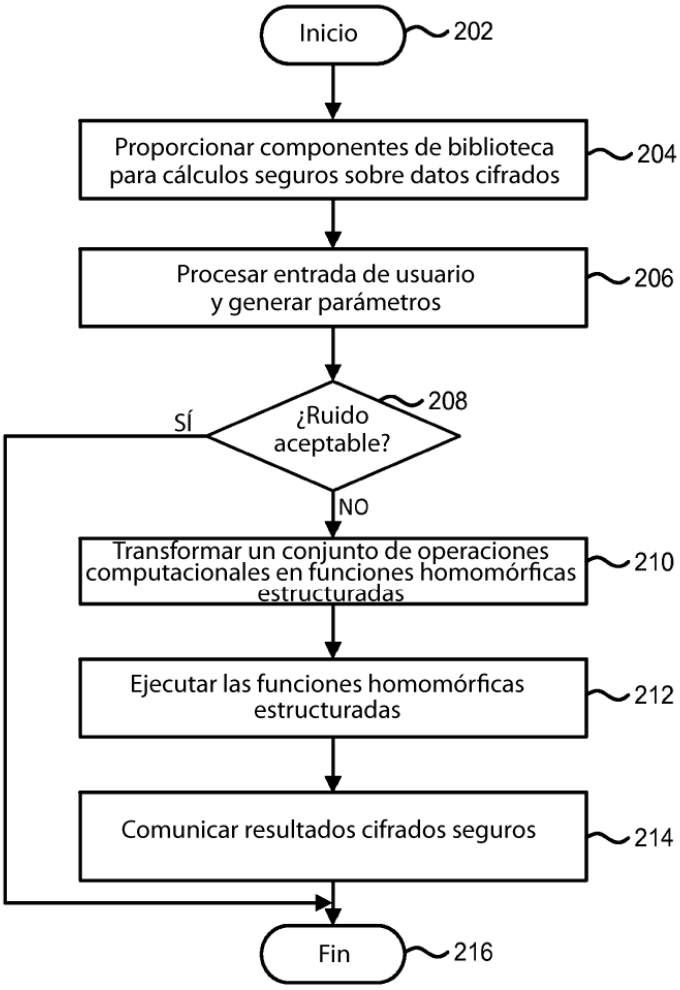
Inicio
202
Proporcionar componentes de biblioteca para cálculos seguros sobre datos cifrados
204
Procesar entrada de usuario y generar parámetros
206
¿Ruido aceptable?
208
SÍ
NO
Transformar un conjunto de operaciones computacionales en funciones homomórficas estructuradas
210
Ejecutar las funciones homomórficas estructuradas
212
Comunicar resultados cifrados seguros
214
Fin
216

FIG. 2

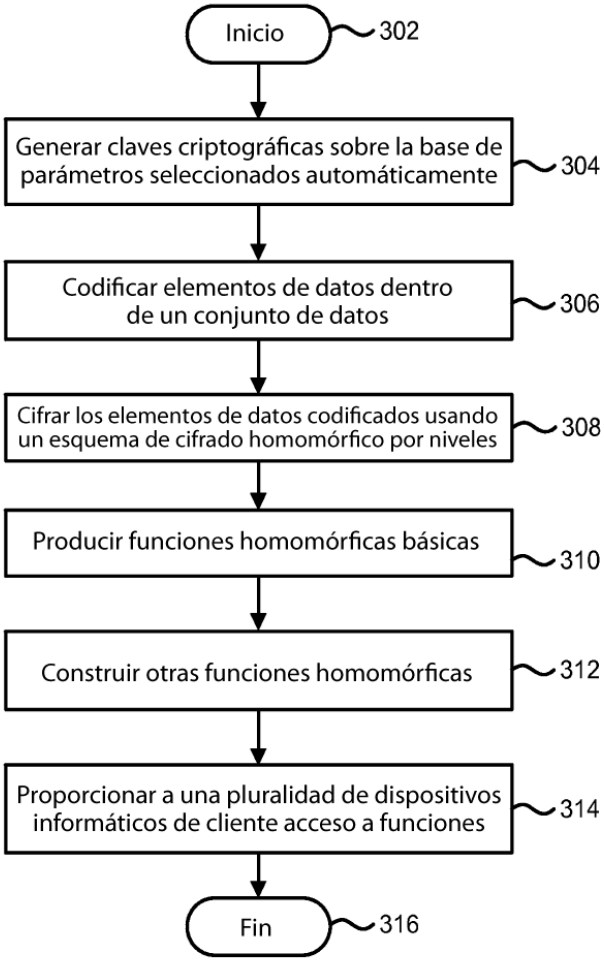
Inicio
302
Generar claves criptográficas sobre la base de parámetros seleccionados automáticamente
304
Codificar elementos de datos dentro de un conjunto de datos
306
Cifrar los elementos de datos codificados usando un esquema de cifrado homomórfico por niveles
308
Producir funciones homomórficas básicas
310
Construir otras funciones homomórficas
312
Proporcionar a una pluralidad de dispositivos informáticos de cliente acceso a funciones
314
Fin
316

FIG. 3

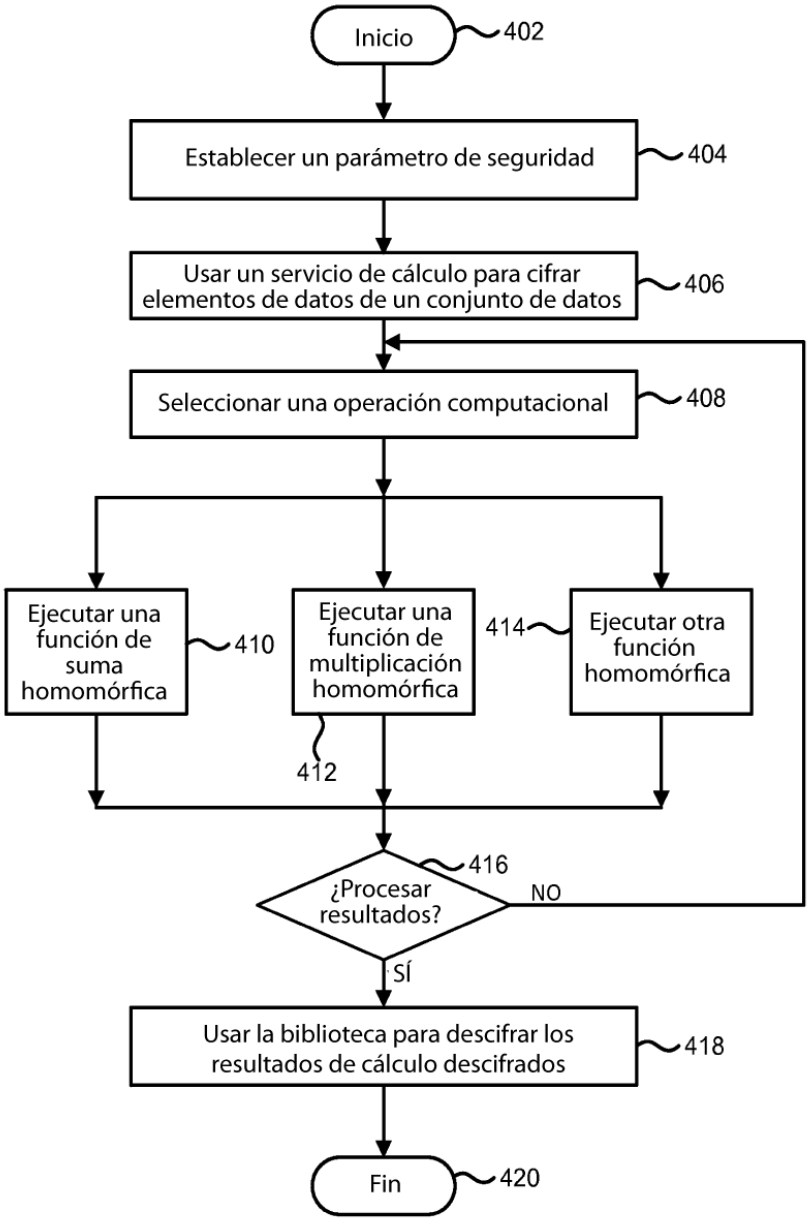
Inicio
402
Establecer un parámetro de seguridad
404
Usar un servicio de cálculo para cifrar elementos de datos de un conjunto de datos
406
Seleccionar una operación computacional
408
Ejecutar una función de suma homomórfica
410
Ejecutar una función de multiplicación homomórfica
412
414
Ejecutar otra función homomórfica
¿Procesar resultados?
416
NO
SÍ
Usar la biblioteca para descifrar los resultados de cálculo descifrados
418
Fin
420

FIG. 4

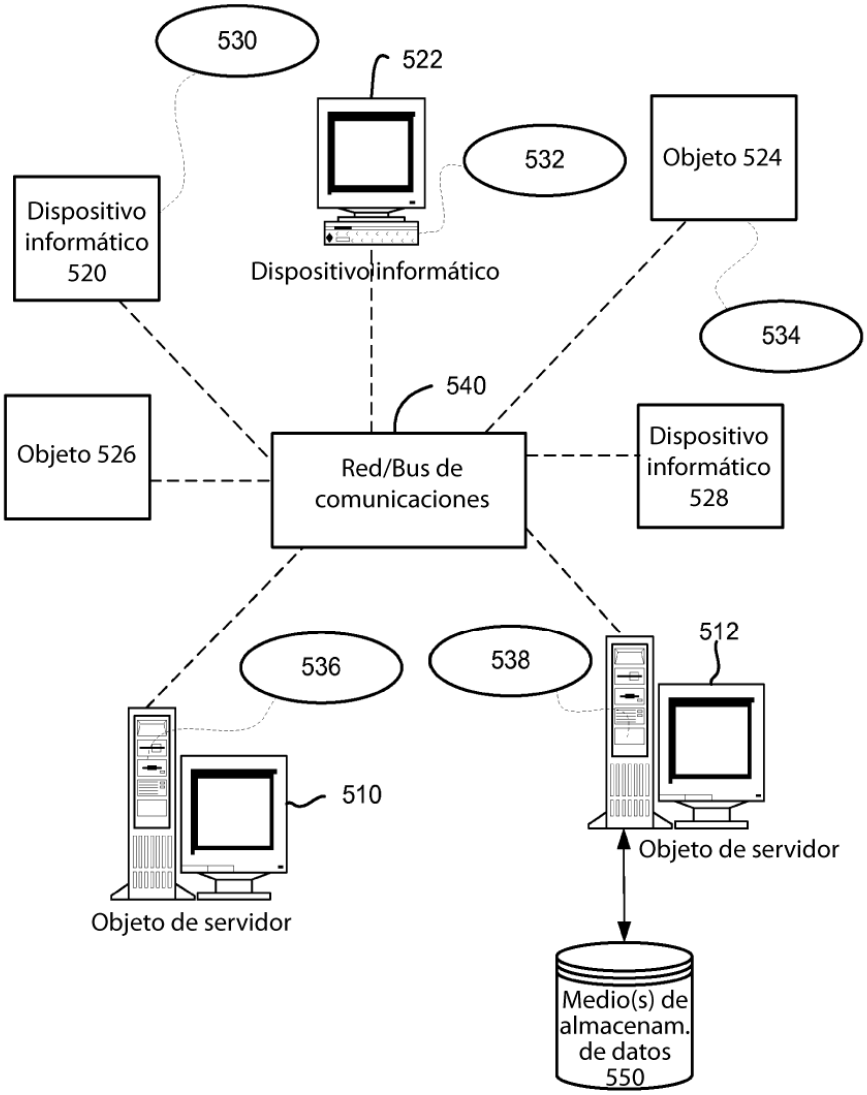
530
522
532
Objeto 524
Dispositivo informático 520
Dispositivo informático
534
540
Objeto 526
Red/Bus de comunicaciones
Dispositivo informático 528
536
538
512
510
Objeto de servidor
Objeto de servidor
Medio(s) de almacenam. de datos 550

FIG. 5

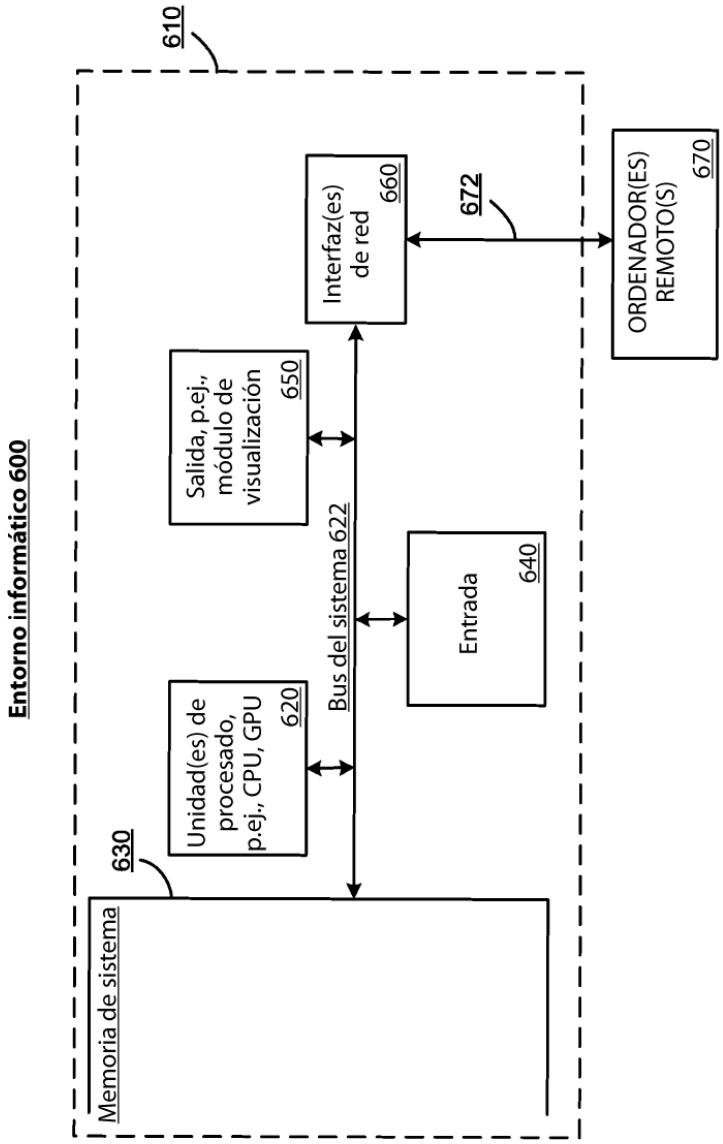
Entorno informático 600
610
Memoria de sistema
630
Unidad(es) de procesado, p.ej., CPU, GPU
620
Bus del sistema 622
Entrada
640
Salida, p.ej., módulo de visualización
650
Interfaz(es) de red
660
672
ORDENADOR(ES) REMOTO(S)
670

FIG. 6