(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 665 987**

(51) Int. CI.:

H04L 9/00 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **14.11.2012 PCT/EP2012/072598**

(87) Fecha y número de publicación internacional: **13.06.2013 WO13083375**

(96) Fecha de presentación y número de la solicitud europea: **14.11.2012 E 12791733 (4)**

(97) Fecha y número de publicación de la concesión europea: **07.03.2018 EP 2742643**

(54) Título: **Dispositivo y procedimiento para la decodificación de datos**

(30) Prioridad:

06.12.2011 DE 102011087804

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
30.04.2018

(73) Titular/es:

SIEMENS AKTIENGESELLSCHAFT (100.0%)
Werner-von-Siemens-Straße 1
80333 München, DE

(72) Inventor/es:

MEYER, BERND y
SCHAFHEUTLE, MARCUS

(74) Agente/Representante:

LOZANO GANDIA, José

DISPOSITIVO Y PROCEDIMIENTO PARA LA DECODIFICACIÓN DE DATOS

DESCRIPCIÓN

La presente invención se refiere a la decodificación de datos mediante un dispositivo protegido de manera criptográfica y/o física.

En la criptografía teórica la seguridad de un procedimiento criptográfico se evalúa habitualmente mediante el comportamiento de entrada y salida. A este respecto, un atacante o *hacker* sólo tiene que ver la entrada y/o la salida de los datos procesados de manera criptográfica y entonces, conociendo el procedimiento utilizado, puede extraer conclusiones con respecto a la clave criptográfica utilizada.

En particular, las etapas de cálculo del procedimiento criptográfico tienen lugar en un entorno protegido al que el atacante no tiene acceso. En la práctica un entorno de cálculo protegido de este tipo puede reproducirse con un testigo de seguridad, por ejemplo con una tarjeta inteligente o una llave. El testigo de seguridad presenta múltiples medidas de protección en hardware y software, que permiten que, por un lado, sirva de memoria fiable y protegida frente a las manipulaciones para datos confidenciales, como por ejemplo material de claves, y por otro lado permiten la implementación de partes de la aplicación del sistema objetivo que se ha de proteger directamente en el entorno protegido del testigo. No obstante, todos los testigos de seguridad habituales en la práctica están muy limitados con respecto al tamaño de memoria de la memoria de datos y programas y con respecto a la potencia y al rendimiento de la capacidad de cálculo disponible. Por tanto, a menudo, en un testigo de seguridad sólo se implementan las funciones criptográficas principales. La mayor parte de la aplicación o del programa se ejecuta entonces en un sistema informático sin nivel de protección o con un nivel de protección claramente inferior.

También existen enfoques que permiten proteger físicamente los sistemas informáticos más potentes. No obstante, la inversión para estas medidas y los costes adicionales en la implementación técnica en relación con los niveles de protección alcanzables y las aplicaciones planificadas es tan elevada que tales procedimientos sólo se utilizan para aplicaciones con seguridad militar.

Cuando debe ejecutarse un algoritmo criptográfico que requiere de una información confidencial en un sistema que permite al atacante seguir el desarrollo del algoritmo, son necesarios mecanismos de protección adicionales para estar protegido frente a determinadas clases de ataques. Por ejemplo, si para el atacante es posible medir propiedades físicas del entorno de cálculo, como por ejemplo duración, consumo de corriente dinámico o radiación electromagnética durante la operación de cálculo, entonces son importantes medidas para defenderse de los denominados ataques de canal lateral.

En caso de que el entorno de cálculo para el procedimiento criptográfico esté compuesto por un sistema informático que no ofrece ninguna protección física, entonces el atacante tiene un control total sobre las etapas de cálculo realizadas y los datos procesados. En este caso el algoritmo debe implementarse de tal modo que el atacante, aunque pueda seguir la ejecución del algoritmo hasta el último detalle, no pueda entenderlo o extraer el secreto procesado. Una implementación de este tipo de un algoritmo se denomina ofuscada.

Los procedimientos para proporcionar un entorno de cálculo seguro van desde meras soluciones de software hasta hardware fabricado especialmente o combinaciones de software y hardware. A este respecto, para la implementación del entorno de cálculo seguro se utiliza al menos un módulo de seguridad que ofrece protección física y/o criptográfica.

Se utilizan meras soluciones de software cuando no están disponibles funciones de protección adicionales por el hardware del sistema informático o no se desea por motivos económicos. El programa, que se ejecutará en el sistema informático, puede modificarse mediante transformaciones de código adecuadas, denominadas ofuscación de código, de tal modo que se dificulte o en el mejor de los casos se evite la ingeniería inversa por parte de un atacante. Pueden utilizarse técnicas especiales, denominadas criptografía *White-Box*, para la protección de material de claves secreto en el software. El nivel de seguridad alcanzable es reducido en comparación con las técnicas con medidas de protección de hardware y normalmente la transformación del código va unida a pérdidas considerables de rendimiento y a una demanda de espacio de memoria y programa claramente superior.

En los enfoques de solución con soporte de hardware, en la mayoría de los casos de aplicación para proporcionar un sistema informático seguro se utiliza un testigo de seguridad. Un testigo de seguridad de este tipo contiene mecanismos de protección especiales en hardware, como por ejemplo sensores de temperatura, tensión de funcionamiento, ataques por medio de rayos láser, buses y memorias codificados, generadores de ruido, generadores de estados de espera aleatorios, protecciones frente al rastreo con agujas, estilos de diseño especiales para chips, etc. Los testigos de seguridad son sistemas informáticos sencillos compuestos por RAM, una memoria no volátil (habitualmente *flash* o EEPROM), una CPU e interfaces E/S (UART, SPI, USB, etc.) y ofrecen un nivel de seguridad comparativamente alto contra ataques. A menudo los testigos de seguridad contienen hardware adicional para el cálculo eficaz de procedimientos criptográficos (aceleradores DES, aceleradores AES, unidades de cálculo de números largos). La potencia de los testigos de seguridad es normalmente muy limitada con respecto al tamaño

de memoria, la velocidad de la CPU y el flujo de datos de las interfaces, de modo que sólo pueden realizarse partes pequeñas de una aplicación relevantes para la seguridad dentro del testigo.

Alternativamente en el mercado también están disponibles controladores monochip fabricados especialmente para aplicaciones de seguridad que, por ejemplo, pueden utilizarse en aplicaciones integradas. Estos sistemas informáticos son habitualmente algo más potentes que los testigos de seguridad, aunque encarecen considerablemente el diseño de un producto.

También es posible proteger todo el aparato con medidas especiales para el diseño de la carcasa. Estas medidas van desde conmutadores sencillos para reconocer la apertura de una carcasa, carcasas especiales, láminas de protección antitaladro, placas de circuito impreso especiales hasta técnicas de fabricación tales como encaje o sellado. Los aparatos pueden tener un sistema sensor activo para poder detectar intentos de manipulación y reaccionar a los mismos y para ello necesitan un suministro de corriente sin interrupciones por medio de una batería. Dentro de un aparato de este tipo puede utilizarse un sistema informático potente compuesto por componentes estándar. Sin embargo, las inversiones correspondientes para el desarrollo y la fabricación de tales sistemas son elevadas de modo que este tipo de medidas de protección habitualmente sólo se emplean en aplicaciones militares. Además son necesarias medidas de organización para el cambio regular de la batería para poder garantizar la disponibilidad de los aparatos.

Las funciones de codificación homomórficas han sido hasta hace pocos años un concepto discutido solamente en la criptografía teórica, que permitirá únicamente mediante operaciones de cálculo en datos codificados implementar un entorno de cálculo seguro para programas en sistemas informáticos no protegidos.

En el año 2009 Craig Gentry en su tesis doctoral describió por primera vez, teniendo en cuenta determinadas suposiciones de complejidad criptográficas, un método para la implementación de una función de codificación homomórfica con todas las propiedades necesarias para la realización de cualquier cálculo (véase [1]). Entretanto, las técnicas descritas se desarrollaron adicionalmente entre otros por Shai Halevi y Nigel Smart, y se mejoró su rendimiento (véase [2] a [4]).

El documento US 2011/0110525 A1 da a conocer un procedimiento que presenta la codificación de información según un esquema de codificación que utiliza una clave pública. El procedimiento presenta además la codificación de una pluralidad de instancias de una clave secreta, estando codificada cada una con al menos una instancia adicional de la clave pública. El procedimiento presenta además el envío de los datos codificados y de la pluralidad de instancias codificadas de la clave secreta a un destino, la recepción de un resultado codificado desde el destino y la decodificación del resultado codificado. Un procedimiento adicional comprende la recepción de una pluralidad de claves secretas codificadas e información que describen una función que se realiza en los datos. Este procedimiento presenta además la conversión de la información a un circuito que está configurado para realizar la función en los datos. El procedimiento presenta además la aplicación de los datos a las entradas del circuito y la evaluación de los datos utilizando a su vez la pluralidad de claves secretas codificadas.

El documento DE 101 20 288 A1 da a conocer un aparato reproductor para archivos de sonido en el que se generan datos de cuenta. El archivo de medios está codificado con claves aleatorias que a su vez están codificadas con la clave pública del usuario.

Por tanto, el objetivo de la presente invención es implementar de manera económica un entorno de cálculo seguro con seguridad más elevada a partir de los componentes estándar disponibles.

Por consiguiente se propone un dispositivo para la decodificación de datos que comprende una pluralidad de medios protegidos mediante al menos un módulo de protección con al menos un medio de recepción y un medio de decodificación. El medio de recepción está configurado para la recepción de datos de cálculo codificados por medio de una función de codificación homomórfica. El medio de decodificación está configurado para decodificar los datos de cálculo codificados mediante la realización de la inversa de la función de codificación homomórfica sobre los datos de cálculo codificados utilizando una clave privada asociada a la función de codificación homomórfica.

El módulo de seguridad sirve para la implementación de un entorno de cálculo seguro. A este respecto, el módulo de protección ofrece en particular protección física y/o criptográfica. El módulo de protección puede estar fabricado como mera solución de software, como hardware fabricado especialmente o como una combinación de software y hardware. A este respecto, el módulo de protección protege en particular contra ataques de canal lateral.

En el caso de los datos de cálculo codificados por medio de la función de codificación homomórfica puede tratarse de resultados de cálculos de un programa transformado por medio de una función de codificación homomórfica.

Mediante el uso de la función de codificación homomórfica para el cálculo de los datos de cálculo codificados es posible implementar el cálculo en un sistema informático no protegido.

Los datos de cálculo codificados se calculan por un módulo, por ejemplo un ordenador no protegido a partir de

componentes estándar. A este respecto, el ordenador no protegido aplica el programa transformado por medio de la función de codificación homomórfica a datos de entrada para proporcionar los datos de cálculo codificados. Entonces, los datos de cálculo codificados están codificados por medio de la función de codificación homomórfica. Ventajosamente, mediante las propiedades de la función de codificación homomórfica es posible realizar la parte del programa que requiere mucho cálculo en el sistema informático u ordenador no protegido de componentes estándar. De este modo, esta parte de la ejecución del programa que requiere mucho cálculo no tiene que ejecutarse en el dispositivo de decodificación con el entorno de cálculo seguro configurado para la decodificación. La seguridad del programa o de la aplicación se garantiza mediante las propiedades criptográficas de la función de codificación homomórfica y no requiere de medidas de protección de hardware especiales mediante el sistema informático. En particular no existen las limitaciones de potencia habituales para el dispositivo de decodificación, por ejemplo un testigo de seguridad, por el tamaño de la memoria y el programa o la potencia de la CPU del dispositivo de decodificación. El programa transformado realiza todos los cálculos de manera codificada. Los datos de cálculo codificados se transfieren al dispositivo de decodificación y a este respecto se reciben por el medio de recepción del dispositivo de decodificación. La decodificación final se realiza en el dispositivo de decodificación protegido, porque el módulo no protegido, en este caso por ejemplo el sistema informático no protegido, no puede poner a disposición medidas suficientes para proteger la clave secreta de la función de codificación homomórfica que se requiere para la decodificación.

La presente solución es muy económica porque para el entorno de cálculo seguro, en este caso el dispositivo de decodificación, sólo son necesarios recursos reducidos, en particular potencia de cálculo y capacidad de almacenamiento en comparación con el módulo para la aplicación del programa transformado. A este respecto, las tareas que requieren mucho cálculo de la aplicación del programa transformado se realizan por el módulo no protegido o poco protegido, que está dotado de mayores recursos, pero con medidas de seguridad menos costosas. Por el contrario el dispositivo de decodificación tiene un nivel de seguridad superior, aunque para la etapa restante de la decodificación necesita claramente menos recursos que el módulo para la aplicación del programa transformado.

Una función de codificación homomórfica de este tipo permite sumar, restar y multiplicar los valores presentes de manera codificada de una estructura matemática adecuada, de modo que el resultado de los cálculos esté presente a su vez de manera codificada. Durante el cálculo en ningún momento hay resultados intermedios u otra información sobre los valores combinados sin codificar de manera que un atacante pueda tener acceso y la realización de estas etapas de cálculo tampoco requiere del conocimiento de material de claves secreto. Es decir, los cálculos pueden realizarse en un sistema informático no protegido, sin poner en riesgo o dar a conocer los datos procesados.

Con ayuda de las operaciones matemáticas de suma, resta y multiplicación es posible calcular funciones polinomiales de valores codificados de manera segura y secreta. Entonces, mediante las funciones polinomiales, a su vez puede describirse cualquier cálculo de sistemas informáticos. Cuando el resultado del cálculo está presente finalmente como valor codificado, sólo el receptor legítimo que tiene la clave privada de la función de codificación homomórfica puede decodificar los valores calculados.

En una forma de realización los medios protegidos por medio del al menos un módulo de protección comprenden un medio de almacenamiento. El medio de almacenamiento está configurado para almacenar un programa para el cálculo de la inversa de la función de codificación homomórfica y para proporcionar el programa almacenado de la inversa de la función de codificación homomórfica al medio de decodificación. El medio de almacenamiento es por ejemplo una memoria *flash* o una memoria RAM.

En otra forma de realización el medio de almacenamiento está configurado además para almacenar la clave privada asociada a la función de codificación homomórfica y proporcionar la clave privada almacenada al medio de decodificación.

En otra forma de realización el medio de decodificación está configurado de manera cableada. Por ejemplo el medio de decodificación está configurado como circuito integrado (IC o ASIC) o como disposición de puertas programables en campo (FPGA).

En otra forma de realización el dispositivo es un testigo de seguridad, un testigo USB, una tarjeta inteligente, un servidor protegido o un ordenador protegido.

Además se propone un sistema con un módulo que presenta un medio de aplicación para aplicar el programa transformado por medio de la función de codificación homomórfica a datos de entrada para proporcionar los datos de cálculo codificados y un medio de transmisión para transmitir los datos de cálculo codificados, y con un dispositivo como se explicó anteriormente para decodificar los datos de cálculo codificados transmitidos.

En una forma de realización del sistema el módulo es un ordenador no protegido o un servidor no protegido.

En otra forma de realización el medio de transmisión del módulo y el medio de recepción del dispositivo están configurados para ejecutar un protocolo de interfaz predeterminado para la transmisión de los datos de cálculo

codificados.

Mediante el protocolo de interfaz predeterminado los datos de cálculo codificados pueden transmitirse de manera controlada y predeterminada del módulo al dispositivo de decodificación.

En otra forma de realización el módulo está dotado de recursos para ejecutar completamente el programa transformado en un periodo de tiempo predeterminado en el que el dispositivo no puede ejecutar completamente el programa transformado.

A este respecto, los recursos comprenden en particular la potencia de cálculo proporcionada de la CPU así como la capacidad de almacenamiento de la memoria o de las memorias del módulo.

En otra forma de realización el módulo presenta un medio de protección que está configurado para proteger el programa transformado frente a una manipulación. El medio de protección puede garantizar una ejecución libre de manipulaciones del programa transformado.

En otra forma de realización el medio de protección está configurado para establecer una prueba de la ejecución correcta del programa transformado mediante el medio de aplicación, en particular por medio de pruebas verificables de manera probabilística (*Probabilistically-Checkable-Proofs* (PCP)) (véase [5] a [7]).

En esta forma de realización el entorno de protección del programa en el módulo potente que realiza los cálculos codificados de manera homomórfica puede ampliarse en tal medida que el programa se proteja frente a manipulaciones. Para ello puede ampliarse el programa en el módulo potente pero por lo demás no protegido, por ejemplo, el sistema informático no protegido, en particular de tal modo que al aplicar el programa al mismo tiempo se establezcan pruebas de la realización correcta del cálculo. Para ello, en particular, se utilizan pruebas verificables de manera probabilística (PCP). Las pruebas PCP tienen la propiedad de que pueden comprobarse de manera muy eficaz. Entonces la comprobación de la prueba del cálculo realizado correctamente en el módulo puede producirse a su vez en el dispositivo de decodificación. Preferiblemente sólo cuando se acepte la prueba del cálculo como válida el dispositivo de decodificación decodificará los datos de cálculo codificados de manera homomórfica.

El respectivo medio, por ejemplo el medio de recepción, el medio de decodificación o el medio de protección, puede estar implementado con técnica de hardware o software. En el caso de una implementación con técnica de hardware el respectivo medio puede estar configurado como dispositivo o como parte de un dispositivo, por ejemplo como ordenador o como microprocesador o como IC, ASIC o FPGA. En el caso de una implementación con técnica de software el respectivo medio puede estar configurado como producto de programa informático, como una función, como una rutina, como parte de un código de programa o como objeto ejecutable.

Además se propone un procedimiento para la decodificación de datos por medio de un dispositivo protegido mediante al menos un módulo de seguridad que presenta las etapas siguientes:

- recibir datos de cálculo codificados por medio de una función de codificación homomórfica, y

- decodificar los datos de cálculo codificados mediante la realización de la inversa de la función de codificación homomórfica sobre los datos de cálculo codificados utilizando una clave privada asociada a la función de codificación homomórfica.

Además se propone un producto de programa informático que en un módulo controlado por programa da lugar a la realización del procedimiento como se explicó anteriormente para la decodificación de datos por medio de un dispositivo protegido mediante al menos un módulo de seguridad. El módulo controlado por programa es en particular un dispositivo de decodificación, como un testigo de seguridad.

Un producto de programa informático como un medio de programa informático puede proporcionarse o suministrarse por ejemplo como medio de almacenamiento, como tarjeta de almacenamiento, lápiz USB, CD-ROM, DVD o también en forma de archivo descargable de un servidor en una red. Esto puede producirse por ejemplo en una red de comunicación inalámbrica mediante la transmisión de un archivo correspondiente con el producto de programa informático o el medio de programa informático.

Las propiedades, características y ventajas de esta invención descritas anteriormente así como la manera en que se consiguen se entenderán mejor y de manera más clara junto con la siguiente descripción de los ejemplos de realización, que se explican en más detalle junto con los dibujos.

A este respecto muestran:

la figura 1, un diagrama de bloques de un primer ejemplo de realización de un dispositivo de decodificación;

la figura 2, un diagrama de bloques de un segundo ejemplo de realización de un dispositivo de decodificación;

la figura 3, un diagrama de bloques de un primer ejemplo de realización de un sistema con un módulo de cálculo y un dispositivo de decodificación;

la figura 4, un diagrama de bloques de un segundo ejemplo de realización de un sistema con un módulo de cálculo y un dispositivo de decodificación;

la figura 5, un diagrama de bloques de un tercer ejemplo de realización de un sistema con un módulo de cálculo y un dispositivo de decodificación;

la figura 6, un diagrama de flujo de un primer ejemplo de realización de un procedimiento para la decodificación de datos; y

la figura 7, un diagrama de flujo de un segundo ejemplo de realización de un procedimiento para la decodificación de datos.

En las figuras los elementos iguales o con la misma función están dotados de los mismos números de referencia siempre que no se indique lo contrario.

En la figura 1 se representa un diagrama de bloques de un primer ejemplo de realización de un dispositivo 1 de decodificación protegido de manera criptográfica y/o física.

El dispositivo 1 de decodificación es por ejemplo un testigo de seguridad, un testigo USB, una tarjeta inteligente, un servidor protegido o un ordenador protegido.

Los medios 3, 4 del dispositivo 1 están protegidos mediante al menos un módulo 2 de protección contra manipulación, espionaje y/o ataques de *hackers*.

El dispositivo 1 de decodificación tiene un medio 3 de recepción y un medio 4 de decodificación. El medio 3 de recepción es adecuado para recibir datos de cálculo B codificados, codificados por medio de una función de codificación f homomórfica.

En el caso de los datos de cálculo B codificados por medio de la función de codificación f homomórfica puede tratarse de resultados de cálculos de un programa P transformado por medio de la función de codificación f homomórfica.

Dicho de otro modo puede transformarse un programa S utilizando la función de codificación f homomórfica en un programa P transformado. Los datos de cálculo B calculados mediante el programa P transformado están codificados, concretamente mediante la función de codificación f homomórfica.

El medio 4 de decodificación es adecuado para decodificar los datos de cálculo B codificados recibidos mediante la realización de la inversa I de la función de codificación f homomórfica sobre los datos de cálculo B codificados utilizando una clave privada k1 asociada a la función de codificación f homomórfica.

En conjunto a la función de codificación f homomórfica está asociado un par de claves de la clave privada k1 y una clave pública k2. La clave pública k2 se utiliza por ejemplo para la transformación de un programa S (véase para ello la figura 5). El medio 4 de decodificación emite en el lado de salida los datos de cálculo K decodificados como texto abierto. Los datos de cálculo K decodificados pueden indicarse por ejemplo en un dispositivo de visualización (no mostrado) o utilizarse para otro programa o una aplicación.

Alternativamente el medio 4 de decodificación también puede estar configurado de manera cableada.

La figura 2 muestra un diagrama de bloques de un segundo ejemplo de realización de un dispositivo 1 de decodificación. El dispositivo 1 de decodificación de la figura 2 se basa en el dispositivo 1 de decodificación de la figura 1 y presenta todas las características de la figura 1. Además el dispositivo 1 de decodificación de la figura 2 tiene un medio 5 de almacenamiento que está configurado para almacenar un programa para el cálculo de la inversa I de la función de codificación f homomórfica y proporcionárselo al medio 4 de decodificación. Además el medio 5 de almacenamiento almacena la clave privada k1 asociada a la función de codificación f homomórfica y se la proporciona al medio 4 de decodificación.

En la figura 3 se representa un diagrama de bloques de un primer ejemplo de realización de un sistema 6 con un módulo 7 de cálculo y un dispositivo 1 de decodificación. Por ejemplo el módulo 7 de cálculo es un sistema informático no protegido, mientras que el dispositivo 1 de decodificación es un testigo de seguridad protegido de manera criptográfica y/o física.

El módulo 7 de cálculo tiene un medio 8 de aplicación, que está configurado para aplicar el programa P

transformado a datos de entrada A para proporcionar los datos de cálculo B codificados. El medio 8 de aplicación proporciona los datos de cálculo B calculados a un medio 9 de transmisión del dispositivo 7 de cálculo. El medio 9 de transmisión está configurado para proporcionar los datos de cálculo B codificados al dispositivo 1 de decodificación. El medio 9 de transmisión del módulo 7 y el medio 3 de recepción del dispositivo 1 están configurados para ejecutar un protocolo de interfaz predeterminado para la transmisión de los datos de cálculo B codificados.

El dispositivo 1 de decodificación corresponde al dispositivo 1 de la figura 2. Por consiguiente el medio 8 de recepción recibe los datos de cálculo B codificados transmitidos y se los proporciona al medio 4 de decodificación.

El módulo 7 de cálculo, al contrario que el dispositivo 1 de decodificación, está configurado para ejecutar completamente el programa P transformado en un periodo de tiempo predeterminado. El módulo 7 de cálculo tiene una potencia de cálculo claramente superior así como una capacidad de almacenamiento claramente superior al dispositivo 1 de decodificación. Por consiguiente, en el caso del sistema 6 se aprovecha la potencia de cálculo del módulo 7 de cálculo para la aplicación del programa P transformado, mientras que en cuanto a la seguridad se utiliza el dispositivo 1 de decodificación.

En la figura 4 se representa un diagrama de bloques de un segundo ejemplo de realización de un sistema 6 con un módulo 7 de cálculo y un dispositivo 1 de decodificación. El segundo ejemplo de realización de la figura 4 se basa en el primer ejemplo de realización de la figura 3 y presenta todas las características del primer ejemplo de realización de la figura 3. Además el módulo 7 de cálculo de la figura 4 tiene un medio 10 de protección. El medio 10 de protección está configurado para proteger el programa P transformado frente a manipulaciones. A este respecto, el medio 10 de protección puede establecer en particular una prueba de la ejecución correcta del programa P transformado mediante el medio 8 de aplicación. En este sentido se utilizan preferiblemente pruebas verificables de manera probabilística (PCP).

La figura 5 muestra un diagrama de bloques de un tercer ejemplo de realización de un sistema 6 con un módulo 7 de cálculo y un dispositivo 1 de decodificación. El tercer ejemplo de realización de la figura 5 se basa en el segundo ejemplo de realización de la figura 4 y presenta todas las características del segundo ejemplo de realización de la figura 4. Además el sistema 6 de la figura 5 tiene un medio 11 de transformación. El medio 11 de transformación está configurado para transformar un programa S para el cálculo de los datos de cálculo en el programa P transformado que se utiliza por el medio 8 de aplicación. Para esta transformación el medio 11 de transformación utiliza la función de codificación f homomórfica utilizando la clave pública k2.

La figura 6 ilustra un diagrama de flujo de un primer ejemplo de realización de un procedimiento para la decodificación de datos por medio de un dispositivo 1 protegido, que también puede denominarse dispositivo 1 de decodificación. El dispositivo 1 de decodificación está configurado por ejemplo según la figura 1 o 2.

En la etapa 601 se reciben datos de cálculo B codificados por el dispositivo 1 de decodificación. Los datos de cálculo B codificados están codificados mediante una función de codificación f homomórfica.

En la etapa 602 se decodifican los datos de cálculo B codificados mediante la realización de la inversa I de la función de codificación f homomórfica sobre los datos de cálculo B codificados utilizando una clave privada k1 asociada a la función de codificación f homomórfica.

En la figura 7 se representa un diagrama de flujo de un segundo ejemplo de realización de un procedimiento para la decodificación de datos mediante un dispositivo 1 de decodificación protegido.

En la etapa 701 se transforma un programa S para el cálculo de los datos de cálculo por medio de una función de codificación f homomórfica utilizando una clave pública k2 asociada a la función de codificación f homomórfica en un programa P transformado.

En la etapa 702 se aplica el programa P transformado a datos de entrada A para proporcionar datos de cálculo B codificados.

En la etapa 703 se transmiten los datos de cálculo B codificados al dispositivo 1 de decodificación.

En la etapa 704 se decodifican los datos de cálculo B codificados por medio del dispositivo 1 de decodificación. Para ello se aplica la inversa I de la función de codificación f homomórfica sobre los datos de cálculo B codificados utilizando la clave privada k1 asociada a la función de codificación f homomórfica. Los datos de cálculo B decodificados están disponibles entonces como datos de cálculo K decodificados en texto abierto.

Aunque la invención se ha ilustrado y descrito en detalle mediante el ejemplo de realización preferido, la invención no está limitada por los ejemplos dados a conocer y el experto puede deducir otras variaciones sin alejarse del alcance de protección de la invención.

Bibliografía

[1] Craig Gentry: A Fully Homomorphic Encryption Scheme, Dissertation, Stanford University, septiembre de 2009

[2] Nigel P. Smart, Frederik Vercauteren: Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes, Public Key Cryptography 2010, Lecture Notes in Computer Science 6056, págs. 420-443, Springer 2010

[3] Marten van Dijk, Craig Gentry, Shai Halevi, Vinod Vaikuntanathan: Fully Homomorphic Encryption over the Integers, Advances in Cryptography, EUROCRYPT 2010, Lecture Notes in Computer Science 6110, págs. 24-43, Springer 2010

[4] Craig Gentry, Shai Halevi: Fully Homomorphic Encryption without Squashing Using Depth-3 Arithemtic Circuits, FOCS 2011

[5] Sanjeev Arora, Shmuel Safra: Probabilistic Checking of Proofs: A New Characterization of NP, Journal of the ACM, 45(1):70-122, 1998

[6] Ingrid Biehl, Bernd Meyer, Susanne Wetzel: Ensuring the Integrity of Agent-Based Computation by Short Proofs, Mobile Agents 1998, Lecture Notes in Computer Science 1477, págs. 183-194, Springer 1999

[7] William Aiello, Sandeep N. Bhatt, Rafail Ostrovsky, Sivaramakrishnan Rajagopalan: Fast Verification of Any Remote Procedure Call: Short Witness- Indistinguishable One-Round Proofs for NP, Automata, Languages and Programming, International Colloquium ICALP 2000, Lecture Notes in Computer Science 1853, págs. 463-474, Springer 2000

REIVINDICACIONES

1. Sistema (6), con:

un módulo (7), que presenta un medio (8) de aplicación para aplicar un programa (P) transformado por medio de una función de codificación (f) homomórfica a datos de entrada (A) para proporcionar datos de cálculo (B) codificados y un medio (9) de transmisión para transmitir los datos de cálculo (B) codificados, siendo el módulo (7) un ordenador no protegido o un servidor no protegido, y

un dispositivo (1) para la decodificación de datos que presenta una pluralidad de medios (3-5) protegidos mediante al menos un módulo (2) de protección, que comprenden un medio (3) de recepción y un medio (4) de decodificación,

estando configurado el medio (3) de recepción para recibir los datos de cálculo (B) codificados por medio de la función de codificación (f) homomórfica, y

estando configurado el medio (4) de decodificación para decodificar los datos de cálculo (B) codificados mediante la realización de la inversa (I) de la función de codificación (f) homomórfica sobre los datos de cálculo (B) codificados utilizando una clave privada (k1) asociada a la función de codificación (f) homomórfica, siendo el dispositivo (1) un testigo de seguridad, un testigo USB o una tarjeta inteligente,

estando dotado el módulo (7) de recursos para ejecutar completamente el programa (P) transformado en un periodo de tiempo predeterminado, en el que el dispositivo (1) no puede ejecutar completamente el programa (P) transformado, teniendo el dispositivo (1) una potencia de cálculo inferior y una capacidad de almacenamiento inferior al módulo (7).

2. Sistema según la reivindicación 1, caracterizado porque el medio (9) de transmisión del módulo (7) y el medio (3) de recepción del dispositivo (1) están configurados para ejecutar un protocolo de interfaz predeterminado para la transmisión de los datos de cálculo (B) codificados.

3. Sistema según la reivindicación 1, caracterizado porque el módulo (7) presenta un medio (10) de protección, que está configurado para proteger el programa (P) transformado frente a una manipulación.

4. Sistema según la reivindicación 3, caracterizado porque el medio (10) de protección está configurado para establecer una prueba de la ejecución correcta del programa (P) transformado mediante el medio (8) de aplicación, en particular por medio de pruebas verificables de manera probabilística (PCP).

5. Sistema según una de las reivindicaciones 1 a 4, caracterizado porque está previsto un medio (11) de transformación para la transformación de un programa (S) para el cálculo de los datos de cálculo por medio de la función de codificación (f) homomórfica utilizando una clave pública (k2) asociada a la función de codificación (f) homomórfica en el programa (P) transformado.

FIG 1

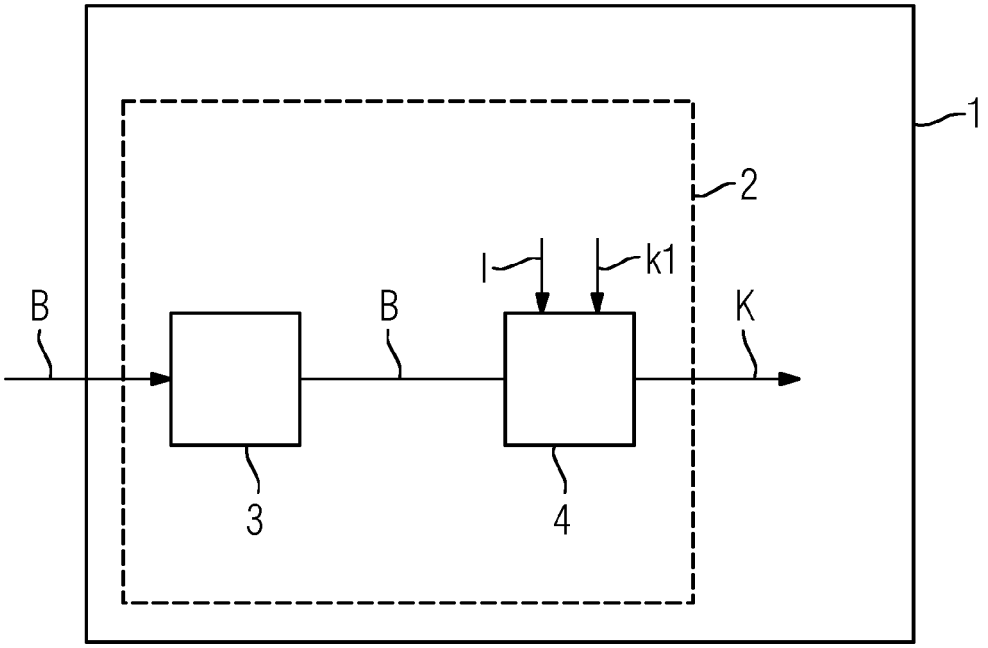

FIG 2

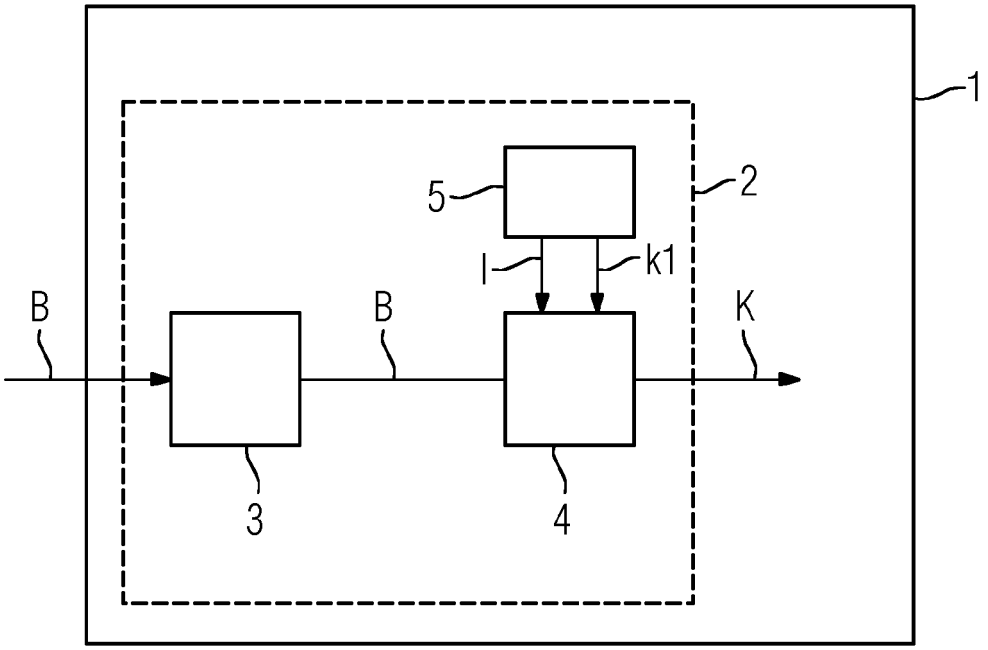

FIG 3

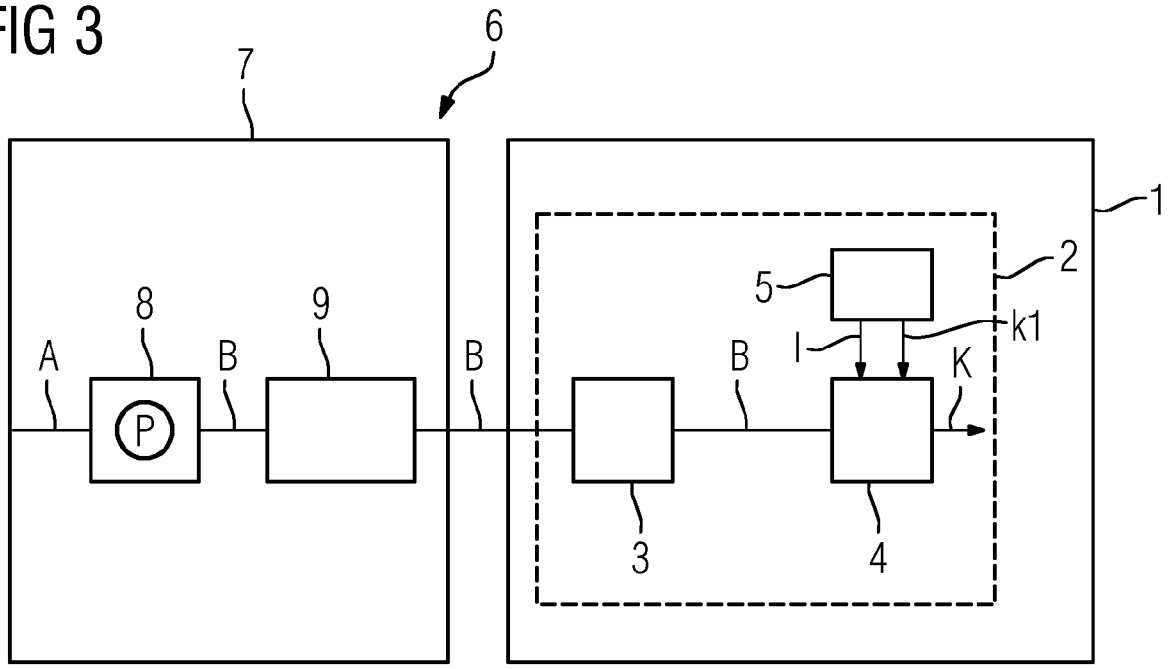

FIG 4

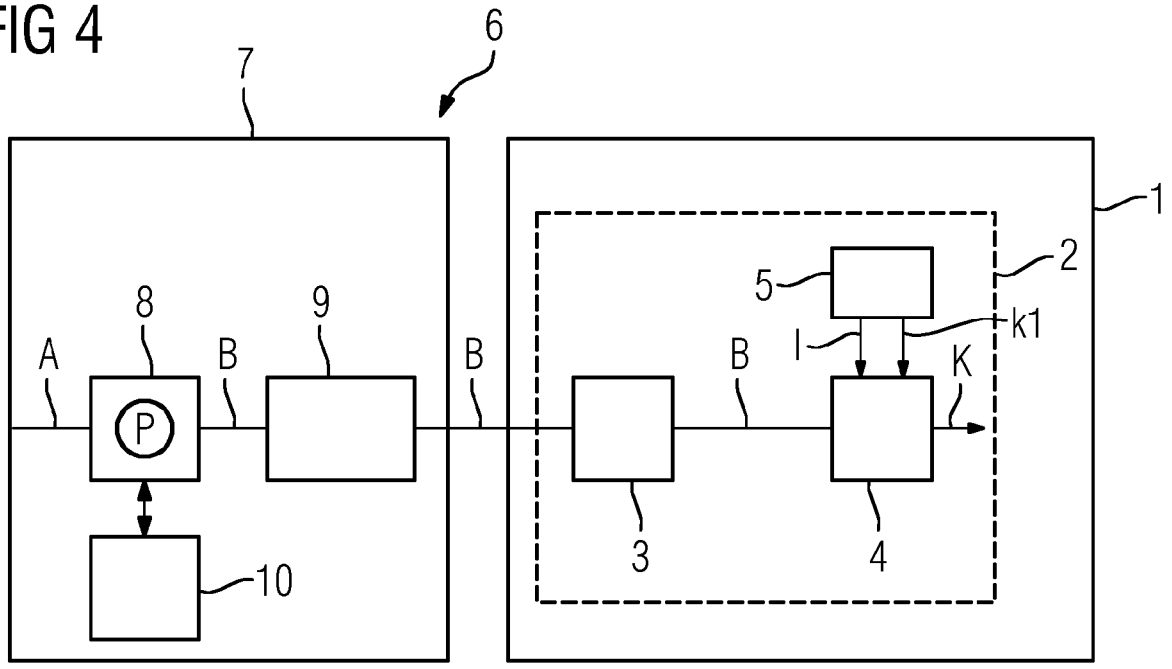

FIG 5

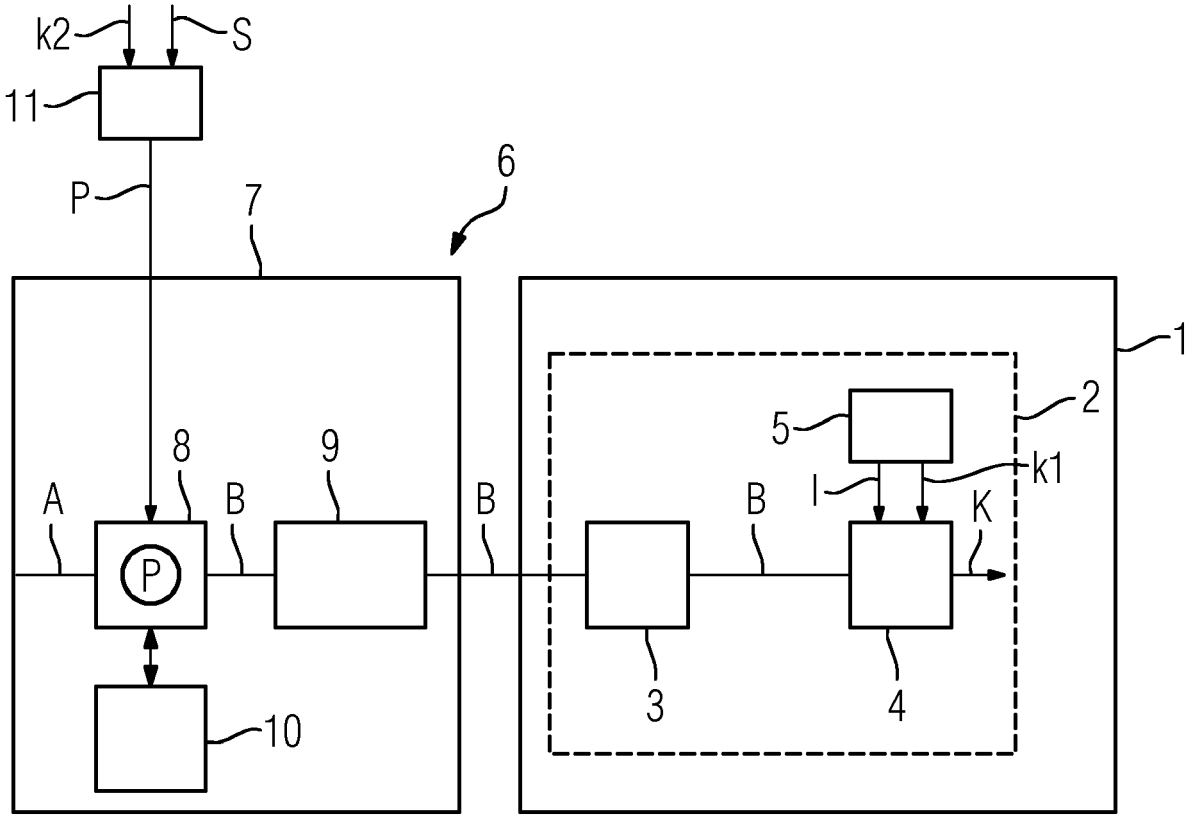

FIG 6

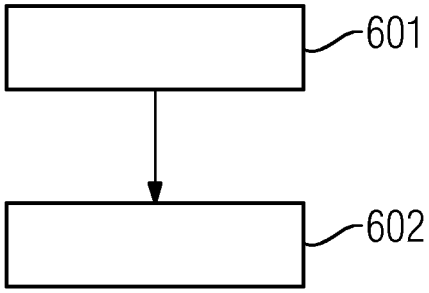

FIG 7