(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 774 104**

(51) Int. CI.:

G06F 21/12 (2013.01)
G06F 21/14 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **10.08.2017 E 17185746 (9)**

(97) Fecha y número de publicación de la concesión europea: **12.02.2020 EP 3441898**

(54) Título: **Procedimiento y equipo para proteger un software frente a una utilización no autorizada**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
16.07.2020

(73) Titular/es:

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

(72) Inventor/es:

ZWANZGER, JOHANNES

(74) Agente/Representante:

LOZANO GANDIA, José

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín Europeo de Patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre Concesión de Patentes Europeas).

DESCRIPCIÓN

Procedimiento y equipo para proteger un software frente a una utilización no autorizada

La presente invención se refiere a un procedimiento y un equipo para proteger un software frente a una utilización no autorizada.

Un software incluye a menudo un software ejecutable y una biblioteca del software, pudiendo acceder el software ejecutable, con argumentos predeterminados, a funciones predeterminadas de la biblioteca de software. Existe una necesidad de proteger el software completo frente a una utilización no autorizada, por ejemplo frente a una aplicación no autorizada de las funciones predeterminadas de la biblioteca de software.

Se conoce la práctica de parchear el software completo codificando el código ejecutable, para que el usuario no autorizado no pueda leerlo. El parcheo debe ejecutarse de nuevo cada vez que se modifique el software ejecutable y/o la biblioteca de software.

El documento EP 3 185 168 A1 da conocer un procedimiento y un sistema para aumentar la seguridad de un aparato de comunicación ejecutando una transacción con una aplicación de transacción del aparato de comunicación. En este contexto se proponen en particular medidas frente a code lifting (reutilización de código) y data lifting (reutilización de datos) para bibliotecas de software nativas conectadas dinámicamente entre sí.

El documento US 2012/0192283 A1 da a conocer un sistema y un procedimiento para modificar una aplicación de software de una forma original a una forma segura. Para ello se modifica la aplicación mediante transmutaciones. Transmutaciones son aquí modificaciones irreversibles de la aplicación. La aplicación de software modificada corresponde a la aplicación de software original, pero es resistente a ataques estáticos y/o dinámicos.

Partiendo de estos antecedentes, consiste un objetivo de la presente invención en lograr una protección mejorada de un software frente a una utilización no autorizada.

Este objetivo se logra mediante un procedimiento para proteger un software frente a una utilización no autorizada según la reivindicación 1 y mediante un equipo para proteger un software frente a una utilización no autorizada según la reivindicación 9. En las reivindicaciones secundarias se describen formas de ejecución del procedimiento.

Según un primer aspecto, se propone un procedimiento para proteger un software frente a una utilización no autorizada. El software incluye un software ejecutable y una biblioteca de software, que proporciona al software ejecutable funciones predeterminadas para argumentos predeterminados. El procedimiento incluye:

aplicación de una primera función a un argumento predeterminado de entre los argumentos predeterminados para generar un argumento protegido mediante el software ejecutable tal que el argumento predeterminado sólo puede averiguarse mediante aplicación de una primera función de inversión al argumento protegido y
aplicación de una segunda función a una función predeterminada de las funciones predeterminadas para generar una función protegida mediante la biblioteca de software tal que la función predeterminada sólo puede averiguarse aplicando una segunda función de inversión a la función protegida.

Para proteger el software pueden intercambiarse argumentos predeterminados y/o funciones predeterminadas, que se intercambian entre el software ejecutable y la biblioteca de software, de forma protegida, en particular codificada o firmada.

El software ejecutable es en particular un código fuente compilado. El software ejecutable puede proporcionar argumentos predeterminados, por ejemplo secuencias de números, a la biblioteca de software y obtener de la misma funciones predeterminadas como resultados. Los argumentos predeterminados pueden generarse o calcularse mediante el software ejecutable y/o ser parte del software ejecutable.

La biblioteca de software puede ser una biblioteca de software dinámica, como por ejemplo una DDL (“dynamic-link library”) bajo Windows o una “shared library” (biblioteca compartida) bajo sistemas operativos tipo Unix, como Linux. La biblioteca de software puede servir para extraer códigos funcionales para el software ejecutable. La biblioteca de software puede incluir varias funciones predeterminadas, que pueden descargarse mediante los argumentos predeterminados.

En determinadas formas de ejecución se desarrolla la biblioteca de software independientemente del software ejecutable. La biblioteca de software la aporta por ejemplo un suministrador al fabricante del software ejecutable. La biblioteca de software puede existir compilada y parcheada, con lo que la misma no puede modificarse.

El procedimiento descrito sirve en particular para proteger el software compuesto por el software ejecutable y la biblioteca de software frente a una utilización no autorizada. Una utilización no autorizada puede ser una aplicación no permitida del software, en particular de los argumentos predeterminados y/o de las funciones predeterminadas. Además puede incluir la utilización no autorizada también una manipulación y/o copiado del software o de partes del software, así como un ataque al software, en particular un ataque “code-lifting”.

Para proteger el software se proporciona a la biblioteca de software el argumento predeterminado protegido mediante el software ejecutable, en particular protegido criptográficamente o protegido con ayuda de un algoritmo. Para ello puede aplicarse a uno de los argumentos predeterminados una primera función, para generar un argumento protegido. La primera función puede entonces existir en el software ejecutable. En particular está distribuido un código de la primera función en toda la extensión del software ejecutable. La primera función es un algoritmo criptográfico. En determinadas formas de ejecución la primera función es un algoritmo invertible.

El argumento protegido incluye en particular el argumento predeterminado. En determinadas formas de ejecución se codifica criptográficamente el argumento predeterminado mediante la primera función, con lo que el argumento protegido corresponde al argumento predeterminado codificado.

El argumento predeterminado puede averiguarse o deducirse en particular sólo a partir del argumento protegido cuando se aplica al argumento protegido la primera función de inversión. La primera función de inversión puede por ejemplo modificar el argumento protegido tal que a partir de ello se recupera el argumento predeterminado. La primera función de inversión forma así en particular un elemento contrapuesto a la primera función. En determinadas formas de ejecución incluye la biblioteca de software la primera función de inversión.

Puesto que el argumento predeterminado se protege mediante el software ejecutable con la primera función, puede impedirse que el argumento predeterminado se capte y utilice maliciosamente. En particular sólo una biblioteca de software que incluya la primera función de inversión puede averiguar a partir del argumento protegido el argumento predeterminado. De esta manera puede por ejemplo impedirse que un usuario malicioso proporcione el argumento predeterminado a una biblioteca de software falseada, manipulando con ello el software. Además puede impedirse que se proporcionen a la biblioteca de software otros argumentos distintos a los argumentos predeterminados y que se utilicen sin autorización las funciones de la biblioteca de software. La biblioteca de software y el software ejecutable pueden colaborar conjuntamente para proteger el software. En particular puede aumentarse así la seguridad del software.

Adicionalmente, para proteger el software, se proporciona al software ejecutable la función predeterminada protegida mediante la biblioteca de software, en particular protegida criptográficamente o con ayuda de un algoritmo. Para ello puede aplicarse a una de las funciones predeterminadas una segunda función, para generar una función protegida. La segunda función puede aquí existir en la biblioteca de software. En particular está distribuido un código de la segunda función por toda la biblioteca de software. La segunda función es un algoritmo criptográfico. En determinadas formas de ejecución es la segunda función un algoritmo invertible.

La función protegida incluye en particular la función predeterminada. En determinadas formas se codifica la función predeterminada mediante la segunda función criptográficamente, con lo que la función protegida corresponde a la función codificada predeterminada.

La función predeterminada puede averiguarse o deducirse a partir de la función protegida en particular sólo cuando a la función protegida se aplique la segunda función de inversión. La segunda función de inversión puede modificar por ejemplo la función protegida tal que a partir de ella se recupere la función predeterminada. La segunda función de inversión constituye así en particular un elemento contrapuesto a la segunda función. En determinadas formas de ejecución incluye el software ejecutable la segunda función de inversión. En particular está distribuido un código de la segunda función de inversión por todo el software ejecutable.

Puesto que la función predeterminada se protege mediante la biblioteca de software con la segunda función, puede impedirse que la función predeterminada se capte y se utilice maliciosamente. En particular sólo un software ejecutable que incluya la segunda función de inversión puede determinar a partir de la función protegida la función predeterminada. De esta manera puede impedirse por ejemplo que un usuario malicioso aproveche sin autorización las funciones predeterminadas proporcionadas por la biblioteca de software, por ejemplo proporcionando el mismo las funciones predeterminadas a un software

ejecutable falseado. Además puede impedirse que al software ejecutable se le proporcionen otras funciones distintas a las funciones predeterminadas. En conjunto puede protegerse la seguridad del software.
Puede protegerse también el software completo cuando no exista ningún parche del software ejecutable y de la biblioteca de software como software total. Esto tiene la ventaja de que el software ejecutable y/o la biblioteca de software pueden modificarse a discreción, con lo que puede aumentarse la flexibilidad del software.

Según una forma de ejecución incluye el procedimiento además:

transmisión del argumento protegido a la biblioteca de software desde el software ejecutable;
aplicación de la primera función de inversión al argumento protegido mediante la biblioteca de software para averiguar el argumento predeterminado y
aportación de la función predeterminada para el argumento predeterminado que se ha averiguado al software ejecutable mediante la biblioteca de software.

La primera función de inversión puede existir en la biblioteca de software, con lo que la biblioteca de software puede averiguar el argumento predeterminado a partir del argumento protegido recibido. La biblioteca de software puede proporcionar al software ejecutable aquella función predeterminada que corresponde al argumento averiguado. En determinadas formas de ejecución se averigua la función predeterminada con ayuda de un algoritmo existente en la biblioteca de software para el argumento predeterminado y se proporciona como resultado al software ejecutable.

Según otra forma de ejecución incluye el procedimiento además:

transmisión de la función protegida al software ejecutable desde la biblioteca de software y
aplicación de la segunda función de inversión a la función protegida mediante el software ejecutable para averiguar la función predeterminada.

La segunda función de inversión puede existir en el software ejecutable, con lo que el software ejecutable puede averiguar la función predeterminada a partir de la función protegida recibida. El software ejecutable puede utilizar la función predeterminada por ejemplo para realizar cálculos y/o enviar la función predeterminada a un usuario.

Según otra forma de ejecución, codifica la primera función el argumento predeterminado con una primera clave de codificación, firma el argumento predeterminado con ayuda de una primera clave de firma con una primera firma digital y/o añade al argumento predeterminado un primer código de autentificación de mensajes.

El argumento protegido puede corresponder al argumento predeterminado codificado con la primera clave de codificación y/o al argumento predeterminado firmado con la primera firma digital. La primera clave de codificación puede ser entonces una clave criptográfica, como por ejemplo una clave simétrica o una clave pública asimétrica. La primera clave de firma puede entonces ser una clave criptográfica, como por ejemplo una clave asimétrica privada.

Además puede añadirse al argumento predeterminado un primer código de autentificación de mensajes, también denominado MAC (“message authentication code”). El primer código de autentificación de mensajes puede calcularse con una primera clave simétrica. El argumento protegido incluye en particular el argumento predeterminado, así como el primer código de autentificación de mensajes.

Según otra forma de ejecución, codifica la segunda función la función predeterminada con una segunda clave de codificación, firma la función predeterminada, con ayuda de una segunda clave de firma, con una segunda firma digital y/o añade a la función predeterminada un segundo código de autentificación de mensajes.

La función protegida puede corresponder a la función predeterminada codificada con la segunda clave de codificación y/o a la función predeterminada firmada con la segunda firma digital. La segunda clave de codificación puede entonces ser una clave criptográfica, como por ejemplo una clave simétrica o una clave asimétrica pública. La segunda clave de firma puede entonces ser una clave criptográfica, como por ejemplo una clave asimétrica privada.

Además puede añadirse al argumento predeterminado un segundo código de autentificación de mensajes (MAC). El segundo código de autentificación de mensajes puede calcularse con una segunda clave simétrica. La función protegida incluye en particular la función predeterminada, así como el segundo código de autentificación de mensajes.

Según otra forma de ejecución, decodifica la primera función de inversión el argumento protegido con una primera clave de decodificación, comprueba una firma digital del argumento protegido con una primera

clave de verificación y/o comprueba un primer código de autentificación de mensajes del argumento protegido.

La primera clave de decodificación puede formar con la primera clave de codificación un par de claves criptográficas. La primera clave de decodificación puede ser entonces una clave criptográfica, por ejemplo una clave simétrica o una clave asimétrica privada, que es adecuada para decodificar un argumento protegido, que se codificó con una clave simétrica o asimétrica pública como primera clave de codificación.

La primera clave de verificación puede formar con la primera clave de firma un par de claves criptográficas. La primera clave de verificación puede ser entonces una clave criptográfica, por ejemplo una clave asimétrica pública, que es adecuada para verificar una firma que se creó con una clave asimétrica privada como primera clave de firma.

La primera función de inversión puede comprobar también una firma digital del argumento protegido. Se comprueba en particular si el argumento protegido se ha firmado digitalmente con la primera clave de firma, que es propia del software ejecutable. Si es éste el caso puede proporcionar la biblioteca de software la función predeterminada. Con la firma digital puede comprobarse además un origen del argumento protegido.

El primer código de autentificación de mensajes puede comprobarse generando la biblioteca de software para el argumento predeterminado protegido un primer código de autentificación de mensajes a comparar y compara el primer código de autentificación de mensajes a comparar con el primer código de autentificación de mensaje recibido. El primer código de autentificación de mensajes a comparar puede calcularse por ejemplo con la primera clave simétrica. Si el primer código de autentificación de mensajes a comparar coincide con el primer código de autentificación de mensajes recibido, proporciona la biblioteca de software en particular la función predeterminada. Con el código de autentificación de mensajes puede además comprobarse un origen del argumento protegido.

Según otra forma de ejecución, decodifica la segunda función de inversión la función protegida con una segunda clave de decodificación, comprueba una firma digital de la función protegida con una segunda clave de verificación y/o comprueba un segundo código de autentificación de mensajes de la función protegida.

La segunda clave de decodificación puede formar con la segunda clave de codificación un par de claves criptográficas. La segunda clave de decodificación puede ser entonces una clave criptográfica, por ejemplo una clave simétrica o una clave asimétrica privada, que es adecuada para decodificar un argumento protegido que se había codificado con una clave simétrica o asimétrica pública como segunda clave de codificación.

La segunda clave de verificación puede formar con la segunda clave de firma un par de claves criptográficas. La segunda clave de verificación puede ser entonces una clave criptográfica, por ejemplo una clave asimétrica pública, que es adecuada para verificar una firma que se había confeccionado con una clave asimétrica privada como segunda clave de firma.

La segunda función de inversión puede comprobar también una firma digital de la función protegida. Se comprueba en particular si la función protegida se ha firmado digitalmente con la segunda clave de firma digital, que pertenece a la biblioteca de software. Si es éste el caso, puede utilizar el software ejecutable la función predeterminada proporcionada. Con la firma digital puede además comprobarse un origen de la función protegida.

El segundo código de autentificación de mensajes puede comprobarse generando el software ejecutable para la función predeterminada recibida un segundo código de autentificación de mensajes a comparar y compara el segundo código de autentificación de mensajes a comparar con el segundo código de autentificación de mensajes recibido. El segundo código de autentificación de mensajes a comparar puede calcularse por ejemplo con la segunda clave simétrica. Si el segundo código de autentificación de mensajes a comparar coincide con el segundo código de autentificación de mensajes recibido, utiliza el software ejecutable en particular la función predeterminada recibida. Con el código de autentificación de mensajes puede además comprobarse un origen de la función protegida.

Según otra forma de ejecución, incluye el procedimiento además:

averiguación mediante la biblioteca de software de si un argumento recibido a través de la biblioteca de software ha sido protegido con la primera función y

si se averigua que el argumento recibido a través de la biblioteca de software no se ha protegido con la primera función, se impide la aportación de la función predeterminada al software ejecutable.

Si se proporciona a la biblioteca de software un argumento falseado, puede detectar la biblioteca de software esto al no haberse protegido el argumento falseado con la primera función, por ejemplo porque el argumento falseado no está firmado digitalmente. En particular no puede averiguar la biblioteca de software con la primera función de inversión ningún argumento predeterminado a partir del argumento falseado. Un argumento falseado puede detectarse en determinadas formas de ejecución mediante la biblioteca de software, con lo que puede impedirse una manipulación del software.

Si se averigua que el argumento recibido no ha sido protegido con la primera función, puede emitirse además una señal de alarma. Además, en este caso pueden borrarse o inutilizarse las funciones predeterminadas de la biblioteca de software. También pueden borrarse o inutilizarse otros datos del software.

En determinadas formas de ejecución se proporciona la función predeterminada al software ejecutable en el caso de que se averigüe que el argumento recibido ha sido protegido con la primera función. En este caso el argumento recibido es un argumento protegido.

Según otra forma de ejecución, incluye el procedimiento además:

averiguación mediante el software ejecutable de si una función recibida a través del software ejecutable se ha protegido con la segunda función y
si se averigua que la función recibida a través del software ejecutable no se ha protegido con la segunda función, emisión de un aviso de falta y/o borrado de al menos una parte del software.

En el caso de que se proporcione al software ejecutable una función falseada, puede detectar esto el software ejecutable en base a que la función falseada no ha sido protegida con la segunda función, por ejemplo porque la función falseada no está firmada digitalmente. En particular no puede averiguar el software ejecutable con la segunda función de inversión ninguna función predeterminada a partir de la función falseada. Una función falseada puede detectarse en determinadas formas de ejecución mediante el software ejecutable, con lo que puede impedirse una manipulación del software.

Si se averigua que la función recibida no se ha protegido con la segunda función, puede emitirse un aviso de falta, en particular una señal de alarma. Además puede borrarse o inutilizarse en este caso al menos una parte del software.

En determinadas formas de ejecución sigue utilizando el software ejecutable la función predeterminada, si se averigua que la función recibida ha sido protegida con la segunda función. En este caso la función recibida es una función protegida.

Según otra forma de ejecución, las funciones predeterminadas se encuentran codificadas en la biblioteca de software. Para codificar las funciones predeterminadas puede parchearse la biblioteca de software.

Según otra forma de ejecución incluye el procedimiento además una autentificación de la biblioteca de software en el software ejecutable y/o una autentificación del software ejecutable en la biblioteca de software.

Para autentificar el software ejecutable en la biblioteca de software, puede generar la biblioteca de software datos de comprobación y transmitirlos al software ejecutable. Los datos de comprobación son por ejemplo datos aleatorios, que se generan con un generador de números aleatorios auténtico.

El software ejecutable puede codificar los datos de comprobación por ejemplo con una clave de codificación simétrica o asimétrica. Los datos de comprobación codificados pueden transmitirse a la biblioteca de software, que realiza la decodificación con una clave de decodificación correspondiente simétrica o asimétrica, que corresponde a la clave de codificación simétrica o bien que forma con la clave de codificación asimétrica un par de claves. Los datos decodificados pueden compararse con los datos de comprobación generados mediante la biblioteca de software. Si hay coincidencia, puede autentificarse el software ejecutable con éxito en la biblioteca de software.

Además puede firmar el software ejecutable los datos de comprobación con ayuda de una clave privada con una firma digital y transmitir los datos de comprobación firmados a la biblioteca de software. La biblioteca de software puede verificar la firma con ayuda de una clave pública correspondiente, que con la clave privada forma un par de claves. Si se averigua que la firma es correcta, puede autentificarse el software ejecutable con éxito en la biblioteca de software.

Además, el software ejecutable puede asociar a los datos de comprobación un código de autentificación de mensaje (MAC), que se calcula con una clave simétrica y transmitir los datos de comprobación con el código de autentificación de mensajes a la biblioteca de software. La biblioteca de software genera para los datos de comprobación recibidos, independientemente, un código de autentificación de mensajes a

comparar y compara el mismo con el código de autentificación de mensajes recibido. Si existe coincidencia, puede autentificarse el software ejecutable con éxito en la biblioteca de software.

Como consecuencia de la autentificación del software ejecutable en la biblioteca de software, puede permitirse que se proporcionen las funciones predeterminadas al software ejecutable.

La autentificación de la biblioteca de software en el software ejecutable se realiza de la misma manera.

Según un segundo aspecto, se propone un programa de computadora que sobre un equipo controlado por programa origina la ejecución del procedimiento según el primer aspecto o bien de una forma de ejecución del primer aspecto.

Un producto de programa de computadora, como por ejemplo un medio de programa de computadora, puede proporcionarse o suministrarse por ejemplo como medio de memoria, como por ejemplo tarjeta de memoria, lápiz USB, CD-ROM, DVD o también en forma de un fichero que puede descargarse desde un servidor en una red. Esto puede realizarse por ejemplo en una red de comunicación inalámbrica mediante la transmisión de un fichero correspondiente con el producto de programa de computadora o el medio de programa de computadora.

Según un tercer aspecto, se propone un equipo para proteger un software frente a una utilización no autorizada, que se define en la reivindicación 9.

La primera y/o segunda unidad puede estar implementada según técnica de hardware y/o también técnica de software. En una implementación según técnica de hardware, puede estar constituida la correspondiente unidad como equipo o como parte de un equipo, por ejemplo como computadora o como microprocesador o como ordenador de control de un vehículo. En una implementación según técnica de software puede estar constituida la correspondiente unidad como producto de programa de computadora, como una función, como una rutina, como parte de un código de programa o como objeto ejecutable.

Las formas de ejecución y características descritas para el procedimiento propuesto son válidas correspondientemente para el equipo propuesto.

Otras posibles implementaciones de la invención incluyen también combinaciones no citadas explícitamente de características o formas de ejecución antes descritas o descritas a continuación en relación con los ejemplos de ejecución. Al respecto añadirá el especialista también aspectos individuales como mejoras o complementos relativos a la correspondiente forma básica de la invención.

Otras ventajosas variantes de ejecución y aspectos ventajosos de la invención son objeto de las reivindicaciones secundarias, así como de los ejemplos de ejecución descritos a continuación. En lo que sigue se describirán más en detalle el procedimiento y el equipo en base a formas de ejecución preferidas, con referencia a las figuras adjuntas.

La figura 1 muestra un equipo para proteger un software frente a una utilización no autorizada según una primera forma de ejecución;
la figura 2 muestra un procedimiento para proteger un software frente a una utilización no autorizada según una primera forma de ejecución;
la figura 3 muestra un procedimiento para proteger un software frente a una utilización no autorizada según una segunda forma de ejecución;
la figura 4 muestra un procedimiento para proteger un software frente a una utilización no autorizada según una tercera forma de ejecución y
la figura 5 muestra un equipo para proteger un software frente a una utilización no autorizada según una segunda forma de ejecución.

En las figuras se han dotado los mismos elementos o elementos que tienen la misma función de las mismas referencias, siempre que no se haya indicado otra cosa.

La figura 1 muestra un equipo 4 para proteger un software 3 frente a una utilización no autorizada según una primera forma de ejecución. El equipo 4 está implementado según técnica de software y está contenido en el software 3.

El software 3 incluye un software ejecutable 1 y una biblioteca de software 2. El software ejecutable 1 se obtuvo compilando un código fuente no representado. La biblioteca de software 2 es un código generado y parcheado mediante un suministrador externo.

El software ejecutable 1 incluye N argumentos predeterminados X_1, X_2, …, X_N. Los argumentos predeterminados X_1, X_2, …, X_N son números. El software ejecutable 1 incluye además una primera función P y una segunda función de inversión V.

La biblioteca de software 2 incluye funciones predeterminadas $F(X_1)$, $F(X_2)$, …, $F(X_N)$, que pueden descargarse en cada caso con los argumentos predeterminados X_1, X_2, …, X_N mediante el software ejecutable 1.

Además incluye la biblioteca de software 2 una primera función de inversión Q, así como una segunda función U. La primera función de inversión Q es la función inversa de la primera función P y está definida tal que Q(P(A)) = A para cualquier argumento A. Además, la segunda función de inversión V es la función inversa de la segunda función U y está definida tal que V(U(A)) = A para cualquier argumento A.

La protección del software 3 frente a la utilización no autorizada se realiza según un procedimiento representado en la figura 2 para proteger un software 3 frente a una utilización no autorizada. A continuación se describe la protección del software con referencia a las figuras 1 y 2.

En una etapa S1 aplica el software ejecutable 1 la primera función P a un argumento predeterminado X_i de los argumentos predeterminados X_1, X_2, …, X_N y genera así un argumento protegido $P(X_i)$. La primera función P es una primera función de codificación, con la que se codifica el argumento predeterminado X_i. Para codificar el argumento predeterminado X_i utiliza la primera función P una primera clave de codificación, que es una clave criptográfica simétrica. En formas de ejecución alternativas puede estar definida la primera función P también mediante cualquier algoritmo de transformación invertible. El argumento protegido $P(X_i)$ es el argumento predeterminado X_i codificado con la primera función P.

El argumento protegido $P(X_i)$ incluye el argumento predeterminado X_i tal que el argumento predeterminado X_i sólo puede averiguarse aplicando la primera función de inversión Q. Un usuario maligno no puede por lo tanto captar el argumento predeterminado $P(X_i)$.

En una etapa S11 se transmite el argumento protegido $P(X_i)$ a la biblioteca de software 2, tal como se representa en la figura 1 mediante la flecha izquierda.

En una etapa S12 se aplica en la biblioteca de software 2 la primera función de inversión Q al argumento protegido $P(X_i)$. Puesto que la primera función de inversión Q es la función inversa de la primera función P, se recupera en la etapa S12 el argumento predeterminado X_i a partir del argumento protegido $P(X_i)$. La primera función de inversión Q utiliza para ello la primera clave de codificación simétrica también utilizada para la codificación.

La biblioteca de software 2 recibe así el argumento predeterminado X_i del software ejecutable 1, sin que el mismo pueda ser manipulado.

La biblioteca de software 2 proporciona al software ejecutable 1 en una etapa S20 la función predeterminada $F(X_i)$, que corresponde al argumento predeterminado X_i averiguado. Este se transmite al software ejecutable 1.

La figura 3 muestra un procedimiento para proteger un software 3 frente a una utilización no autorizada según una segunda forma de ejecución. Para proteger el equipo 4 de la figura 1 puede ejecutarse el procedimiento de la figura 3.

Las etapas S1, S 11 y S 12 se describieron ya con referencia la figura 2. En una etapa S2 que sigue a la etapa S 12, se aplica según el procedimiento a la segunda forma de ejecución en la biblioteca de software 2 la segunda función U a la función predeterminada $F(X_i)$, con lo que se genera una función protegida U(F(Xi)).

La segunda función U es una segunda función de codificación, con la que puede codificarse la función predeterminada $F(X_i)$, que corresponde al argumento predeterminado X_i. Para codificar la función predeterminada $F(X_i)$ utiliza la segunda función U una segunda clave de codificación, que es una clave criptográfica simétrica. La función protegida U(F(Xi)) es la función predeterminada $F(X_i)$ codificada con la segunda función U.

La función protegida U(F(Xi)) incluye la función predeterminada $F(X_i)$ tal que la función predeterminada $F(X_i)$ sólo puede averiguarse aplicando la segunda función de inversión V. El usuario maligno no puede por lo tanto captar la función predeterminada $F(X_i)$.

En una etapa S21 se transmite la función protegida U(F(Xi)) al software ejecutable 1, tal como se representa en la figura 1 mediante la flecha derecha. Las etapas S2 y S21 forman conjuntamente la etapa S20 antes descrita.

En una etapa S22 aplica el software ejecutable 1 la segunda función de inversión V a la función protegida U(F(Xi)), con lo que se recupera la función predeterminada $F(X_i)$. La función predeterminada $F(X_i)$ se utiliza y/o emite a continuación mediante el software ejecutable 1.

La figura 4 muestra un procedimiento para proteger un software 3 frente a una utilización no autorizada según una tercera forma de ejecución. El procedimiento representado en la figura 4 es una ampliación del procedimiento de la figura 3. El equipo 4 representado en la figura 1 puede ejecutar el procedimiento representado en la figura 4.

En las etapas S 31 a S 37 se autentifica el software ejecutable 1 en la biblioteca de software 2. Para ello genera la biblioteca de software 2 en una etapa S 31 datos de comprobación. Los datos de comprobación se generan con un generador de números aleatorios.

En una etapa S 32 se transmiten los datos de comprobación al software ejecutable 1. El software ejecutable 1 codifica los datos de comprobación en una etapa S 33 con una clave criptográfica y genera así datos de comprobación codificados, que se transmiten en una etapa S 34 a la biblioteca de software 2.

En una etapa S 35 decodifica la biblioteca de software 2 los datos de comprobación codificados recibidos y genera así datos codificados de comparación. Los datos codificados de comparación se comparan en una etapa S 36 con los datos de comprobación. Si en la etapa S 36 se averigua que los datos de comprobación coinciden con los datos codificados de comparación, se autentifica el software ejecutable 1 con éxito en la biblioteca de software 2 y se permite que se proporcionen las funciones predeterminadas $F(X_1)$, $F(X_2)$, …, $F(X_N)$ al software ejecutable 1 en una etapa S37.

Pero si en la etapa S36 se determina que los datos de comprobación no coinciden con los datos codificados de comparación, no se permite que se proporcionen las funciones predeterminadas $F(X_1)$, $F(X_2)$, …, $F(X_N)$ al software ejecutable 1 en una etapa S38. En la etapa S 38 se emite un aviso de falta.

Tras la autentificación con éxito del software ejecutable 1 en la biblioteca de software 2 en la etapa 37, pueden ejecutarse las etapas S1 y S11 ya descritas. Cuando se recibe el argumento protegido $P(X_i)$, averigua la biblioteca de software 2 en una etapa S14 si el argumento recibido $P(X_i)$ ha sido efectivamente protegido, en particular codificado, con la primera función P. Si se averigua que el argumento recibido $P(X_i)$ ha sido protegido con la primera función P, se ejecutan las etapas S12, S2, S21 y S22 ya descritas.

Evidentemente, si se averigua que el argumento recibido $P(X_i)$ no ha sido protegido con la primera función P, se impide en una etapa S15 que se proporcione la función predeterminada $F(X_i)$. En la etapa S15 se indica además a un usuario un aviso de falta, que señaliza un ataque al software 3.

La figura 5 muestra un equipo 14 para proteger un software frente a una utilización no autorizada según una segunda forma de ejecución. El equipo 14 incluye una primera unidad 15 y una segunda unidad 16, que aquí son procesadores. El equipo 14 es adecuado para ejecutar los procedimientos para proteger un software según las figuras 2 a 4. Al respecto se ejecuta la etapa S1 ya descrita mediante la primera unidad 15 y la etapa S2 se ejecuta mediante la segunda unidad 16.

Aún cuando la presente invención se ha descrito en base a ejemplos de ejecución, puede modificarse la misma de diversas formas. Por ejemplo puede ejecutar la primera y/o segunda función P, U cualquier operación sobre el argumento predeterminado X_i y/o sobre la función predeterminada $F(X_i)$. En particular puede la primera y/o segunda función P, U firmar digitalmente el argumento predeterminado X_i y/o la función predeterminada $F(X_i)$ y/o añadir a la misma un código de autentificación de mensajes. La primera y/o segunda función de inversión Q, V pueden adaptarse a la primera y/o segunda función P, U. El software 3 puede incluir también varias bibliotecas de software 2. Además puede proporcionar la biblioteca de software 2 para un único argumento X varias funciones $F(X_i)$. Distintos argumentos predeterminados X_i pueden protegerse con una misma primera función P; pero también pueden estar previstas distintas primeras funciones P para los distintos argumentos predeterminados X_i. Lo mismo es válido para la segunda función U.

La autentificación del software ejecutable 1 en la biblioteca de software 2 puede realizarse además también en un procedimiento de firma, en el que los datos de comprobación se firman con una firma digital y/o un procedimiento de autentificación de mensajes en el que a los datos de comprobación se les asocia un código de autentificación de mensajes. Además puede también autentificarse la biblioteca de software 2 en un procedimiento de autentificación similar en el software ejecutable 1.

REIVINDICACIONES

1. Procedimiento para proteger un software (3) frente a una utilización no autorizada, en el que el software (3) incluye un software ejecutable (1) y una biblioteca de software (2), que proporciona al software ejecutable (1) funciones predeterminadas ($F(X_1)$, $F(X_2)$, …, $F(X_N)$) para argumentos predeterminados (X_1, X_2, …, X_N), que incluye:
aplicación (S1) de una primera función (P) a un argumento predeterminado (X_i) de entre los argumentos predeterminados (X_1, X_2, …, X_N) para generar un argumento protegido ($P(X_i)$) mediante el software ejecutable (1) tal que el argumento predeterminado (X_i) sólo puede averiguarse aplicando una primera función de inversión (Q) al argumento protegido ($P(X_i)$) y en el que la primera función (P) es un algoritmo criptográfico;
transmisión (S11) del argumento protegido $P(X_i)$ a la biblioteca de software (2) desde el software ejecutable,
aplicación (S12) de la primera función de inversión (Q_i) al argumento protegido ($P(X_i)$) mediante la biblioteca de software (2) para averiguar el argumento predeterminado (X_i); y
aportación (S13) de la función predeterminada ($F(X_i)$) para el argumento predeterminado (X_i) que se ha averiguado al software ejecutable (1) mediante la biblioteca de software (2);
aplicación (S2) de una segunda función (U) a la función predeterminada ($F(X_i)$) de entre las funciones predeterminadas ($F(X_1)$, $F(X_2)$, …, $F(X_N)$) para generar una función protegida (U(F(Xi)) .mediante la biblioteca de software (2) tal que la función predeterminada ($F(X_i)$) sólo puede averiguarse aplicando una segunda función de inversión (V) a la función protegida (U(F(Xi)), siendo la segunda función (U) un algoritmo criptográfico;
transmisión (S21) de la función protegida (U(F(Xi)) al software ejecutable (1) desde la biblioteca de software (2) y
aplicación (S22) de la segunda función de inversión (V) a la función protegida (U(F(Xi)) mediante el software ejecutable (1) para averiguar la función predeterminada ($F(X_i)$).

2. Procedimiento según la reivindicación 1,
en el que la primera función (P) codifica el argumento predeterminado (X_i) con una primera clave de codificación, firma el argumento predeterminado (X_i), con ayuda de una primera clave de firma, con una primera firma digital y/o añade al argumento predeterminado (X_i) un primer código de autentificación de mensajes y/o
en el que la segunda función (U) codifica la función predeterminada ($F(X_i)$) con una segunda clave de codificación, firma la función predeterminada ($F(X_i)$), con ayuda de una segunda clave de firma, con una segunda firma digital y/o añade a la función predeterminada ($F(X_i)$) un segundo código de autentificación de mensajes.

3. Procedimiento según la reivindicación 1 ó 2,
en el que la primera función de inversión (Q) decodifica el argumento protegido $P(X_i)$ con una primera clave de decodificación, comprueba una firma digital del argumento protegido $P(X_i)$ con una primera clave de verificación y/o comprueba un primer código de autentificación de mensajes del argumento protegido y/o
en el que la segunda función de inversión (V) decodifica la función protegida (U(F(Xi)) con una segunda clave de decodificación, y/o comprueba una firma digital de la función protegida (U(F(Xi)) y/o comprueba un segundo código de autentificación de mensajes de la función protegida.

4. Procedimiento según una de las reivindicaciones 1 a 3,
que incluye además:
averiguación (S14) mediante la biblioteca de software (2) de si un argumento recibido a través de la biblioteca de software (2) ha sido protegido con la primera función (P) y
si se averigua que el argumento recibido a través de la biblioteca de software (2) no se ha protegido con la primera función (P), se impide (S15) la aportación de la función predeterminada ($F(X_i)$) al software ejecutable (1).

5. Procedimiento según una de las reivindicaciones 1 a 4,
que incluye además:
averiguación mediante el software ejecutable (1) de si una función recibida a través del software ejecutable (1) se ha protegido con la segunda función (U) y
si se averigua que la función recibida a través del software ejecutable (1) no se ha protegido con la segunda función (U), emisión de un aviso de falta y/o borrado de al menos una parte del software (3).

6. Procedimiento según una de las reivindicaciones 1 a 5,
en el que las funciones predeterminadas ($F(X_1)$, $F(X_2)$, …, $F(X_N)$) se encuentran codificadas en la biblioteca de software (2).

7. Procedimiento según una de las reivindicaciones 1 a 6,
que incluye además:

autentificación (S31-S37) de la biblioteca de software (2) en el software ejecutable (1) y/o una autentificación del software ejecutable (1) en la biblioteca de software (2).

8. Producto de programa de computadora, que origina sobre un equipo controlado por programa la ejecución del procedimiento según una de las reivindicaciones 1 a 7.

9. Equipo (4, 14) para proteger un software (3) frente a una utilización no autorizada, estando preparado el equipo para ejecutar el procedimiento según una de las reivindicaciones 1 a 7, incluyendo el software (3) un software ejecutable (1) y una biblioteca de software (2), que proporciona al software ejecutable (1) funciones predeterminadas ($F(X_1)$, $F(X_2)$, …, $F(X_N)$) para argumentos predeterminados (X_1, X_2, …, X_N), que incluye:
una primera unidad (15) para aplicar la primera función (P) según la reivindicación 1;
y una segunda unidad (16) para aplicar la segunda función (U) según la reivindicación 1.

FIG 1

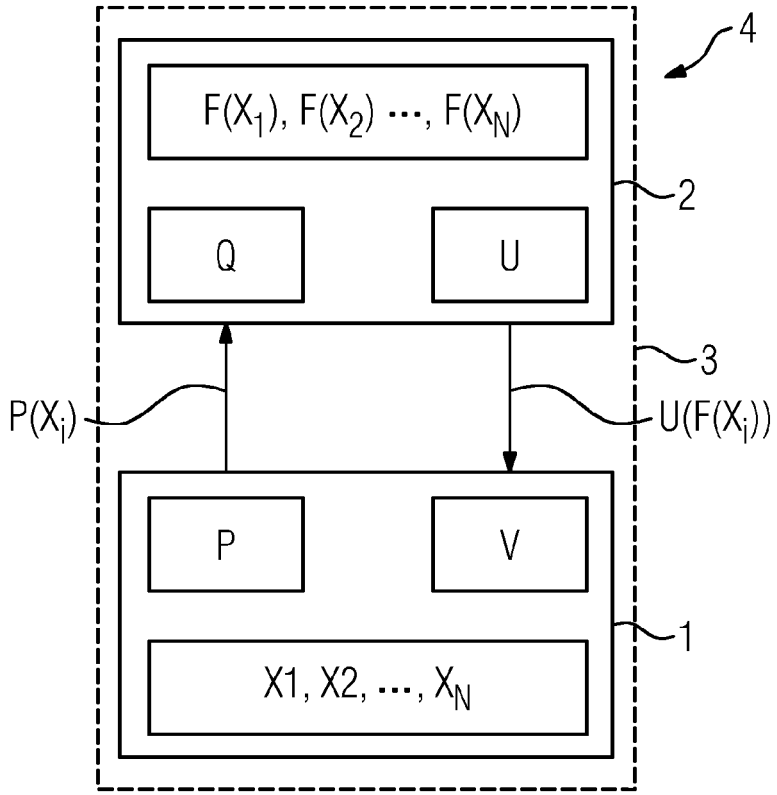

FIG 2

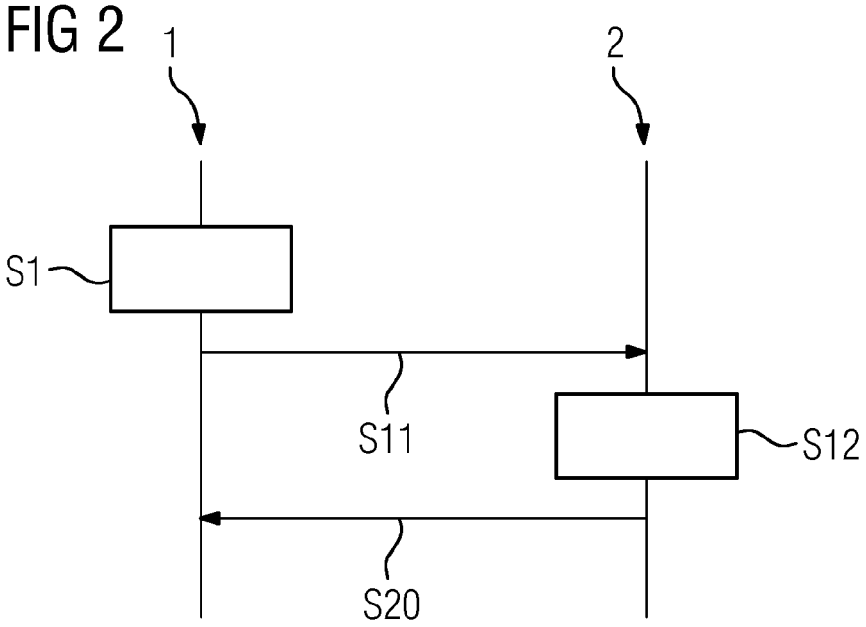

FIG 3

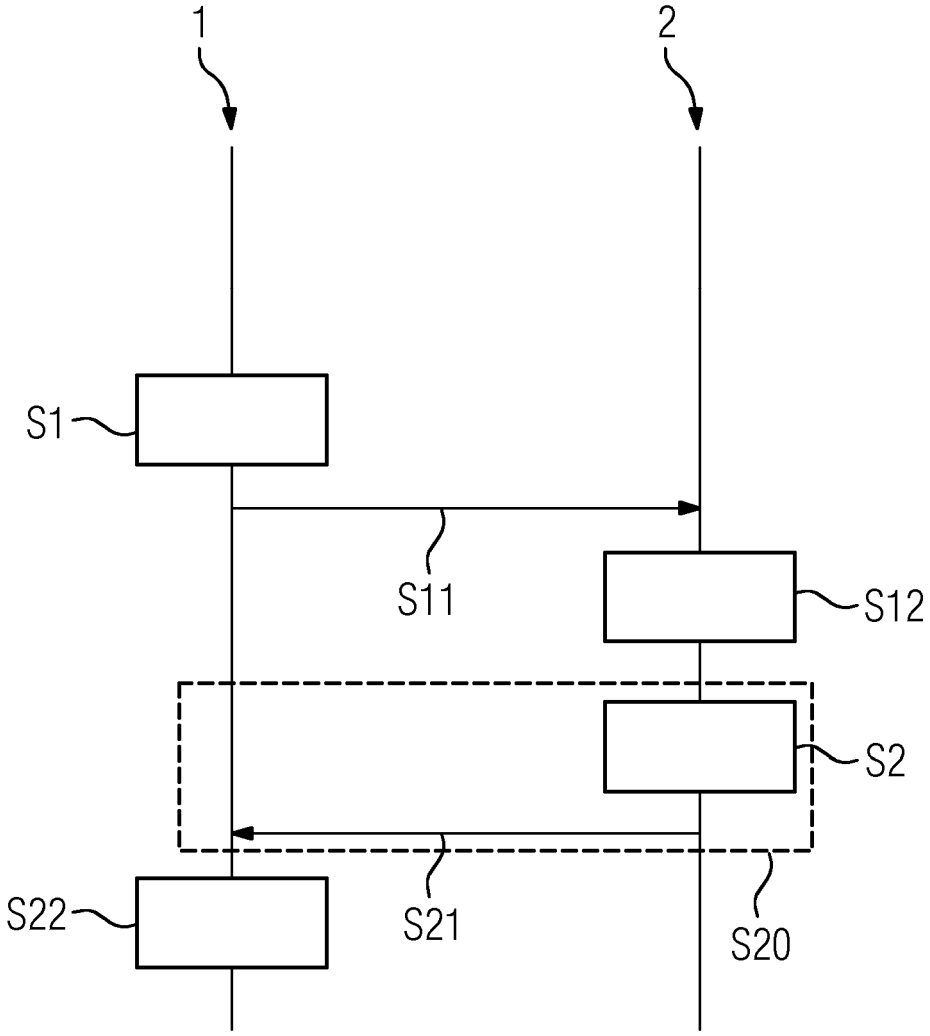

FIG 4

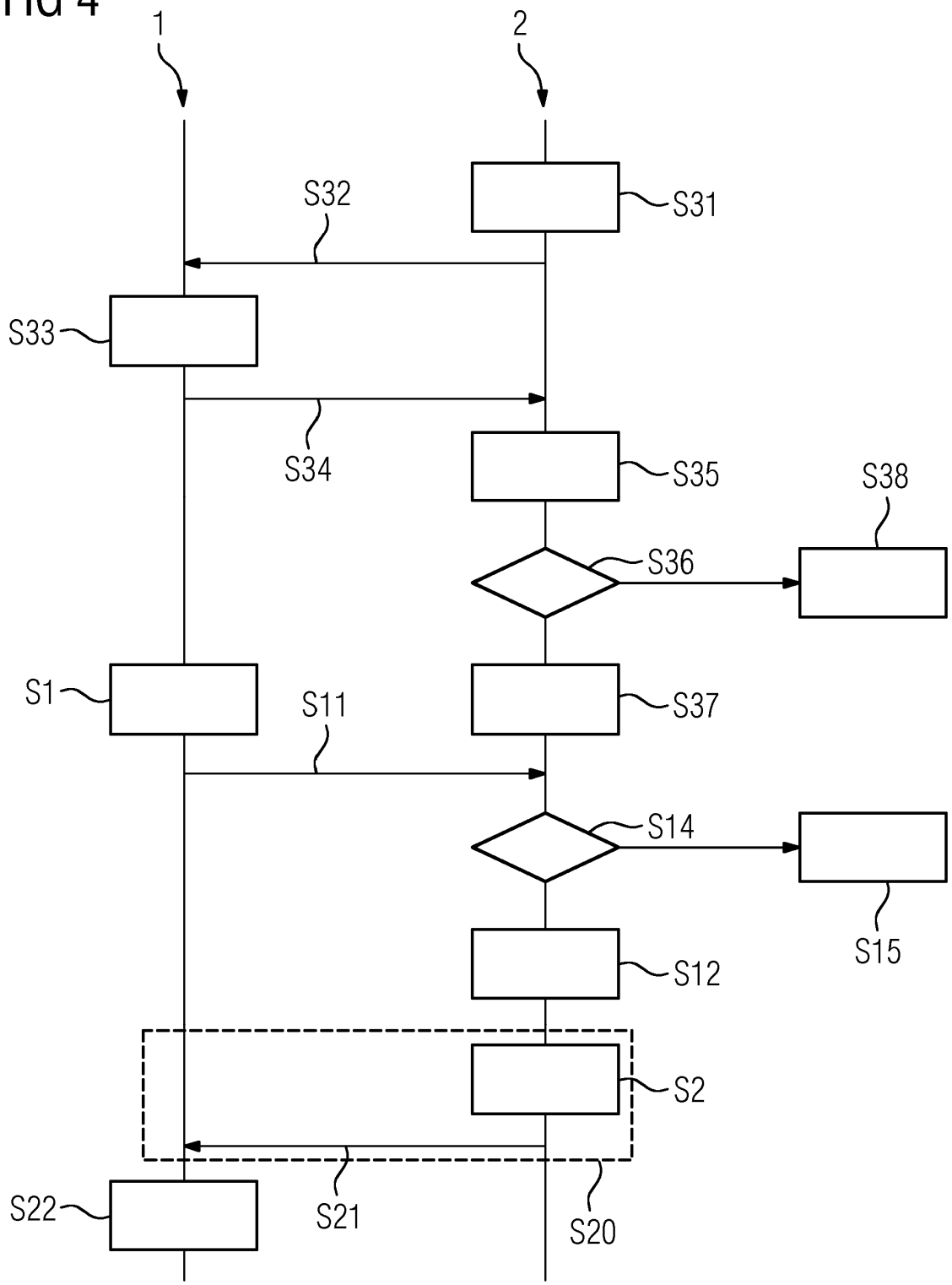
1
2
S32
S31
S33
S34
S35
S38
S36
S1
S11
S37
S14
S15
S12
S2
S22
S21
S20

FIG 5