(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 691 258**

(51) Int. CI.:

H04L 9/08 (2006.01)
H04L 29/06 (2006.01)
G06F 12/14 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **07.09.2012 PCT/US2012/054108**

(87) Fecha y número de publicación internacional: **14.03.2013 WO13036733**

(96) Fecha de presentación y número de la solicitud europea: **07.09.2012 E 12830795 (6)**

(97) Fecha y número de publicación de la concesión europea: **18.07.2018 EP 2754062**

(54) Título: **Sistema y método para una comunicación anfitrión-esclavo segura**

(30) Prioridad:

08.09.2011 US 201161532527 P
30.11.2011 US 201113308363

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
26.11.2018

(73) Titular/es:

LEXMARK INTERNATIONAL, INC. (100.0%)
IP Law Department Bldg. 082-1 740 West New Circle Road
Lexington, KY 40550, US

(72) Inventor/es:

ADKINS, CHRISTOPHER, ALAN y
RADEMACHER, TIMOTHY, JOHN

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Sistema y método para una comunicación anfitrión-esclavo segura

Referencias cruzadas a las solicitudes relacionadas

La presente solicitud se relaciona con y reivindica prioridad de la solicitud de patente provisional U.S. 61/532,527, archivada el 8 de setiembre, 2011, titulada, “SISTEMA Y MÉTODO PARA UNA COMUNICACIÓN MAESTRO-ESCAVO SEGURA”.

Antecedentes

1. Campo de la descripción

Las realizaciones de ejemplo de la presente descripción se refieren de manera general a la comunicación maestro-esclavo segura, y más particularmente a un sistema y un método de comunicación en el que una clave de sesión es generada tanto por el dispositivo maestro como por el esclavo para usar tanto en el cifrado/descifrado como en la generación de direcciones de esclavo.

2. Descripción de la técnica relacionada

Los dispositivos de impresión son conocidos por usar esquemas de autenticación electrónicos asociados con sus elementos de suministros consumibles. Normalmente, el elemento de suministro reemplazable contiene un chip de circuito integrado que se comunica con el controlador ubicado en la impresora. En tal disposición, la impresora se configura como el dispositivo anfitrión y cada elemento de suministro como el dispositivo esclavo. El controlador en el anfitrión verifica la autenticidad de cada uno de los dispositivos esclavos mediante el envío de un reto a los mismos. La autenticidad es verificada por el anfitrión que recibe la respuesta correcta al reto desde el dispositivo esclavo.

En algunos esquemas existentes de autenticación de consumibles, los dispositivos anfitrión y esclavo se comunican a través del bus I^2C. El anfitrión envía comandos al esclavo usando la dirección de esclavo asignada al mismo, el esclavo ejecuta los comandos y envía las respuestas, según sea apropiado, de vuelta al anfitrión. Los comandos y los datos se envían sin verificación de datos.

Mientras las comunicaciones entre anfitriones y esclavos no están cifradas, dicho sistema utiliza una característica de cambio de dirección de esclavo única para hacer más difícil duplicar la función del dispositivo esclavo. La dirección de esclavo se cambia de manera regular a valores de dirección de esclavo determinados por un algoritmo que es conocido por tanto el anfitrión como el esclavo. Después de recibir un comando de cambio de dirección desde el anfitrión, el esclavo no responderá a los sondeos de direcciones desde el anfitrión hasta después de que un cierto comando se reciba en la nueva dirección. La dirección actual se almacena en una memoria no volátil de tanto el anfitrión como el esclavo por lo que la dirección actual junto con la posición en la secuencia de direcciones, se mantiene a lo largo de los ciclos de alimentación.

La característica de cambio de dirección hace que clonar el chip de circuito integrado del dispositivo esclavo sea más difícil porque el algoritmo para computar el siguiente valor de la dirección de esclavo utiliza el valor actual de la misma. El problema con esta característica es que el anfitrión y el esclavo pueden resultar estar no sincronizados en la secuencia de direcciones. Por ejemplo, esto ocurrirá al mover un elemento de suministro esclavo desde una impresora anfitriona a otra ya que la segunda impresora no sabrá donde está el dispositivo esclavo en la secuencia de direcciones. Para superar esto, se proporciona un medio para reconfigurar la secuencia, que sustancialmente debilita la seguridad del sistema.

En concreto, el sistema existente sufre de 1) una falta de verificación y corrección de datos; 2) comunicación no cifrada; y 3) secuencias de direcciones de esclavo reiniciables.

El funcionamiento en ambientes ruidosos puede provocar la corrupción de los datos en el bus, pero el sistema existente no tiene medios para detectar o corregir estos errores inducidos por el ruido. Esto es de cierta importancia ya que los elementos de suministro (dispositivos esclavos) se ubican a menudo dentro de la impresora anfitriona a una distancia relativamente larga del controlador del anfitrión y los cables del bus de comunicaciones se pueden enrutar cerca de fuentes de ruido agresivo, tales como motores. El envío de los comandos de forma no cifrada permite que un atacante aprenda los comandos y datos del sistema capturando el tráfico entre el controlador de la impresora y el elemento de suministro.

Basado en lo anterior, existe la necesidad de un sistema de comunicación anfitrión-esclavo mejorado. La descripción de la US 2003/093694 A1 puede ser útil para entender la presente invención, como describe este documento obteniendo una clave de sesión a partir de un identificador de semilla y de anfitrión.

Compendio

Las realizaciones de ejemplo superan las deficiencias con los esquemas de comunicación existentes y de este modo

satisfacen una necesidad significativa del dispositivo esclavo de comunicarse de manera segura con un anfitrión sobre un bus. El dispositivo esclavo puede incluir un procesador y una memoria acoplada a este que tiene almacenados dentro instrucciones de código de programa. Las instrucciones de código de programa, cuando son ejecutadas por el procesador, provocan al procesador a: después de que el dispositivo esclavo se reinicie, determinar un valor de semilla basado en un valor de semilla del dispositivo esclavo anterior a que el dispositivo esclavo se reiniciara; recibir un número de anfitrión desde un anfitrión que es sustancialmente aleatorio; determinar una clave de sesión basada en el valor de semilla determinado y el número de anfitrión, siendo la clave de sesión sustancialmente aleatoria; y usar la clave de sesión para realizar operaciones de cifrado y descifrado en los datos a transmitir y en los datos recibidos por el dispositivo esclavo, respectivamente, y a determinar un valor de dirección para el dispositivo esclavo para comunicarse con el anfitrión. En la presente memoria, una primera parte de la clave de sesión es usada por el procesador para las operaciones de cifrado y descifrado, en donde una segunda parte es usada para generar el valor de dirección. Creando una clave de sesión que no se comunica con el anfitrión y que se usa en el cifrado/descifrado así como en la generación de direcciones de esclavo, el dispositivo esclavo coopera con el anfitrión para comunicarse de manera segura con este.

Breve descripción de los dibujos

Las características anteriormente mencionadas y otras y las ventajas de las diversas realizaciones, y la manera de alcanzarlas, resultarán más claras y se entenderán mejor con referencia a los dibujos adjuntos.

La Figura 1 es un diagrama de bloques de un sistema de comunicación que incluye un dispositivo anfitrión y al menos un dispositivo esclavo; y

La Figura 2 es un diagrama de flujo que ilustra el funcionamiento del dispositivo esclavo de la Figura 1 según una realización de ejemplo,

Descripción detallada

Se ha de entender que la invención no se limita en su aplicación a los detalles de construcción y disposición de los componentes enunciados en la siguiente descripción o ilustrados en los dibujos. La invención es capaz de otras realizaciones y de ser puesta en práctica o ser llevada a cabo de diversas maneras. También, se ha de entender que la fraseología y la terminología usadas en la presente memoria son por el propósito de la descripción y no debería considerarse como limitante. El uso de “incluye,” “comprende,” o “tiene” y las variaciones de los mismos están destinadas en la presente memoria a abarcar los elementos listados después y los equivalentes de los mismos así como elementos adicionales. A menos que estén limitados de otro modo, los términos “conectado,” “acoplado,” y las variaciones de los mismos se usan en la presente memoria ampliamente y abarcan las conexiones y acoplamientos directos e indirectos. Además, los términos “conectado,” “acoplado,” y las variaciones de los mismos no se restringen a las conexiones o acoplamientos físicos o mecánicos. Además, y como se describe en los párrafos siguientes, las configuraciones mecánicas específicas ilustradas en los dibujos están destinadas a ejemplificar las realizaciones de la invención y son posibles otras configuraciones mecánicas alternativas.

Las realizaciones de ejemplo de la presente descripción están dirigidas a la comunicación entre un dispositivo 100 anfitrión y uno o más dispositivos 110 esclavos, como se muestra en la Fig. 1. El dispositivo 100 anfitrión y el dispositivo 110 esclavo se comunican a través de un bus 120. En una realización de ejemplo, el dispositivo 100 anfitrión es un dispositivo de impresión y el dispositivo 110 esclavo es un elemento de suministro reemplazable. En concreto, el dispositivo 100 anfitrión puede incluir componentes y módulos utilizados normalmente en impresoras, que incluyen un motor 130 de impresión para presentar una imagen en una hoja de medios. Por ejemplo, el motor 130 de impresión puede ser un motor de impresión para una impresora láser o para una impresora de tinta. Se entiende que el motor 130 de impresión puede ser cualquier motor usado para crear una imagen en una hoja de medios. El dispositivo 100 anfitrión puede incluir además un sistema 140 de escáner para capturar una imagen que aparece en una hoja de medios para su uso posterior en una operación de impresión, comunicación por correo electrónico o similar Se puede incluir un sistema 150 alimentador de medios en el dispositivo 100 anfitrión para mover sucesivamente las hojas de medios desde una pila de entrada (no mostrada) al motor 130 de impresión para realizar una operación de impresión después de la cual la hoja impresa se puede mover a un área de salida del dispositivo 100 anfitrión (no mostrada). Los detalles del motor 130 de impresión, el sistema 140 de escáner y el sistema 150 alimentador de medios son bien conocidos y no se describirán en la presente memoria por razones de simplicidad.

El dispositivo 100 anfitrión puede incluir además una interfaz 160 de usuario que permite la comunicación entre el dispositivo 100 anfitrión y un usuario del mismo. La interfaz 160 de usuario puede ser cualquier interfaz para facilitar la comunicación entre el dispositivo 100 anfitrión y el usuario, tal como, por ejemplo, una pantalla táctil.

El dispositivo 100 anfitrión puede incluir además un puerto 170 de la interfaz para comunicarse con uno o más dispositivos 110 esclavos a través del bus 120. El dispositivo 100 anfitrión puede incluir además un controlador 180 para controlar los diferentes componentes del dispositivo 100 anfitrión. En el contexto en el que el dispositivo 100 anfitrión es un dispositivo de impresión, el controlador 180 puede controlar el funcionamiento del motor 130 de impresión, el sistema 140 de escáner, el sistema 150 alimentador de medios, la interfaz 160 y la interfaz 170 de

usuario. El controlador 180 puede ejecutar instrucciones almacenadas en la memoria 190 para controlar los diversos componentes del dispositivo 100 anfitrión.

En una realización en la que el dispositivo 100 anfitrión es un dispositivo de impresión, el dispositivo 110 esclavo puede ser un cartucho o envase de tinta o tóner, por ejemplo. Además, o como alternativa, el dispositivo 110 esclavo puede ser otro componente reemplazable de una impresora láser anfitriona, tal como una unidad de revelado de un motor 130 de impresión o una unidad de fusor.

El dispositivo esclavo puede incluir un procesador 200 para, entre otras cosas, cooperar con el dispositivo 100 anfitrión a realizar la autenticación del esclavo para permitir sólo a los esclavos autorizados comunicarse con el dispositivo 100 anfitrión y de este modo evitar ataques o daños al dispositivo 100 anfitrión. El procesador 200 se acopla con la memoria 210 que tiene instrucciones almacenadas dentro para su ejecución por el procesador 200. El procesador 200 y la memoria 210 se pueden crear en un chip 230 de circuito integrado. En una realización alternativa, el procesador 200 y l memoria 210 residen en chips de circuitos integrados separados. En aún otra realización alternativa, el dispositivo 110 esclavo puede incluir una circuitería, tal como una circuitería basada en una máquina de estados, para cooperar con el dispositivo 100 anfitrión para realizar la autenticación del esclavo.

Se entiende que el dispositivo 100 anfitrión no se limita a un dispositivo de impresión y puede ser virtualmente cualquier dispositivo electrónico con el que un elemento extraíble y/o reemplazable se pueda comunicar a través de un bus 120. Se entiende de manera similar que el dispositivo 110 esclavo puede ser virtualmente cualquier elemento reemplazable que se comunica con el dispositivo 100 anfitrión, incluyendo dispositivos esclavos que estén acoplados de manera comunicativa con éste de forma temporal.

El bus 120 puede ser cualquier bus que soporte un protocolo de bus en el que un anfitrión 100 y uno o más dispositivos 110 esclavos se comuniquen los unos con los otros. Según una realización de ejemplo, el bus 120 puede ser un bus de Circuito inter-Integrado (I^2C). En un bus I^2C, un cable del bus 120 compartido lleva los datos de una manera bidireccional, y otro cable lleva las señales de reloj desde el dispositivo 100 anfitrión hasta los dispositivos 110 esclavos. También, aunque el bus 120 compartido se ilustra como un bus en serie de dos cables, se pueden utilizar estructuras de bus en paralelo compartidos.

Según al menos algunas realizaciones, incluyendo las realizaciones en las que el bus 120 es un bus I^2C, el bus 120 es un bus maestro-esclavo, con el dispositivo 100 anfitrión sirviendo como el maestro del bus y los dispositivos 110 esclavos como los esclavos del bus. Al usar el protocolo I^2C, el dispositivo 100 anfitrión inicia todas las comunicaciones con los respectivos dispositivos 110 esclavos. Los dispositivos 110 esclavos sólo responden a las solicitudes del dispositivo 100 anfitrión. En el caso de que un impostor se conecte al bus 120 compartido y emplee una dirección de esclavo válida, entonces el dispositivo impostor puede recibir una comunicación dirigida a él desde el dispositivo 100 anfitrión. Cuando se pasa información sensible a través del bus 120 a los dispositivos 110 esclavos, el dispositivo impostor puede recibir esta misma de una manera no autorizada, desconocida para el dispositivo 100 anfitrión. Esto puede ocurrir si un dispositivo 110 esclavo autorizado fuera desconectado del bus 120 compartido y el dispositivo impostor fuera conectado en el mismo y programado o cableado para asumir la dirección del dispositivo 110 esclavo que fue desconectado. Si los dispositivos 110 esclavos fueran equipados todos con direcciones fijas, lo que ha sido la práctica establecida, entonces no es excesivamente complicado acoplar un dispositivo impostor al bus 120 compartido y recibir las comunicaciones sensibles de una manera no autorizada desconocida para el dispositivo 100 anfitrión. Como resultado, los dispositivos 110 esclavos cambian de manera ocasional sus direcciones de esclavo en respuesta a una solicitud del dispositivo 100 anfitrión.

En una realización de ejemplo, el anfitrión 100 y el esclavo 110 se comunican usando comandos y datos cifrados usando un cifrado de flujo u otro esquema de cifrado. El establecimiento de una sesión de cifrado es realizado mediante el intercambio de valores entre el anfitrión 100 y el esclavo 110. Entonces el anfitrión 100 y el esclavo 110 calculan cada uno de manera independiente una clave de sesión a partir de los valores intercambiados y un secreto que es conocido por ambos. La clave de sesión se usa después para inicializar el cifrado de flujo (o cualquier otro esquema de cifrado) y la función de dirección de esclavo.

Específicamente, la tabla a continuación muestra los valores en el esquema de cifrado entre el anfitrión 100 y el esclavo 110, incluyendo los tamaños de ejemplo para cada valor.

Datos	Descripción	Tamaño
SN	Número de serie del esclavo	4 bytes
EK	Clave de cifrado secreta	16 bytes
SEMILLA	Semilla del número aleatorio de esclavo	20 bytes
SID	Identificación de sesión	2 bytes
HRN	Número aleatorio de anfitrión	8 bytes

SRN	Número aleatorio de esclavo	8 bytes
SK	Clave de Sesión	20 bytes

TABLA

Valores de Cifrado

Cada esclavo 110 almacena en su memoria 210 un número SN de serie del esclavo único, una clave EK de cifrado de secreto única, una semilla SEMILLA del número aleatorio de esclavo y un identificador SID de sesión en una memoria no volátil, tal como la memoria 210. Estos valores se pueden escribir inicialmente en la memoria 210 como parte del proceso de fabricación para el esclavo 110. El número SN de serie del esclavo es el número de serie único del esclavo 100. La clave EK de cifrado secreta es la clave secreta mantenida en tanto el esclavo 110 como el anfitrión 100 que se usa para obtener la clave SK de sesión. La semilla SEMILLA del número aleatorio de esclavo se inicializa con un número aleatorio real durante el proceso de fabricación y es actualizada por el esclavo 110 después de cada ciclo de alimentación con un valor obtenido de éste. La identificación SID de sesión se inicializa a cero o algún otro valor y es aumentada o disminuida por el esclavo 110 con cada ciclo de alimentación.

El funcionamiento del esclavo 110 se describirá a continuación con respecto a la Fig. 2.

Después de que se haya reiniciado el esclavo 110, lo que puede ocurrir, por ejemplo, cuando el esclavo 100 se conecta inicialmente y es alimentado por el anfitrión 100, el esclavo 110 calcula en 10 una nueva identificación SID de sesión basada en la identificación SID de sesión actual que se mantiene en una memoria 210 no volátil dentro del esclavo 110. El valor de la nueva identificación SID de sesión se puede calcular, por ejemplo, aumentando o disminuyendo el valor de la identificación SID de sesión actual.

Además, después del reinicio, el esclavo 110 determina una nueva semilla SEMILLA del número aleatorio de esclavo en 20. Según la realización de ejemplo, $SEMILLA_0$ representa el número aleatorio real escrito en la memoria 210 para la semilla SEMILLA del número aleatorio de esclavo durante el proceso de fabricación del dispositivo 110 esclavo. La semilla $SEMILLA_1$ del número aleatorio de esclavo es el valor de la semilla SEMILLA del número aleatorio de esclavo después del i-ésimo ciclo de alimentación posterior. El valor i-ésimo de la semilla $SEMILLA_i$ del número aleatorio de esclavo se puede actualizar con el valor $SEMILLA_{i-1}$ de la semilla SEMILLA del número aleatorio de esclavo que sigue a un encendido del dispositivo 110 esclavo. En concreto, $SEMILLA_i$ se puede computar usando un algoritmo seguro, tal como un algoritmo hash seguro (SHA). De esta manera, $SEMILLA_i$ se puede representar como:

$$SEMILLA_i = SHA\text{-}1(SEMILLA_{i-1})$$

Donde “SHA-1” es la función hash segura de 160 bit diseñada por la Agencia de Seguridad Nacional. Se entiende que $SEMILLA_i$ se puede calcular usando un algoritmo diferente, lo que incluye un algoritmo seguro diferente, tal como un SHA diferente.

La semilla $SEMILLA_i$ del número aleatorio de esclavo se usa después para computar en 30 un número R aleatorio (o pseudoaleatorio multi byte, tal como un número de 20 bytes, según la ecuación:

$$R = SHA\text{-}1(SN \,\&\, SEMILLA_i \,\&\, SID)$$

donde “&” representa la concatenación. El número SRN aleatorio de esclavo para la sesión se puede calcular para ser un número predeterminado de los bytes más significativos del número R, tal como los 8 bytes más significativos de R:

$$SRN = R[159:96]$$

se entiende que se pueden utilizar funciones y algoritmos distintos del SHA-1 para generar el SRN, así como otro algoritmo basado en hash.

El anfitrión 100 computa el número HRN aleatorio de anfitrión usando una computación similar a la descrita anteriormente para generar el número SRN aleatorio de esclavo, o cualquier otro algoritmo generador de números aleatorios o pseudoaleatorios.

El anfitrión 100 y el esclavo 110 se comunican usando comandos y datos que son cifrados. En una realización de ejemplo, el anfitrión 100 y el esclavo 110 cifran los comandos y los datos a comunicar entre sí usando un cifrado de flujo. Por ejemplo, el anfitrión 100 y el esclavo 110 pueden utilizar el cifrado de flujo RC4 debido a su bajo coste computacional. Se entiende, sin embargo, que cualquier esquema de cifrado y/o cifrado de flujo puede ser utilizado por el anfitrión 100 y el esclavo 110 para comunicar la información entre ellos. En términos generales, una sesión de cifrado es establecida mediante el intercambio de valores entre el anfitrión 100 y el esclavo 110, a partir de lo cual el anfitrión 100 y el esclavo 110 calculan de manera independiente una clave SK de sesión basada en los valores

intercambiados y un valor secreto conocido por cada uno. La clave SK de sesiones se usa después para inicializar el cifrado, que como se discutió en la realización de ejemplo es un cifrado de flujo.

Para establecer una sesión de cifrado en 40 para comunicar la información cifrada entre el anfitrión 100 y el esclavo 110, el anfitrión 100 envía al esclavo 110 el valor HRN aleatorio de anfitrión. En respuesta, el esclavo 110 envía al anfitrión 100 el número SRN aleatorio de esclavo y la identificación SID de sesión en respuesta. Después, tanto el anfitrión 100 como el esclavo 110 calculan en 50 la clave SK de sesión como sigue:

$$SK = HMAC(EK,HRN \& SRN \& SID)$$

donde HMAC es el código de autenticación de mensaje basado en hash. Como se mencionó anteriormente, la clave EK de cifrado secreta es conocida por tanto el anfitrión 100 como el esclavo 110, pero no se transmite a través del bus 120. La clave SK de sesión puede ser, por ejemplo, de 20 bytes de longitud y no se comunica a través del bus 120.

Se entiende que otras funciones de cifrado, tales como otra función basada en hash, se pueden utilizar para generar la clave SK de sesión. Se entiende además que cualquier esquema de cifrado se podría usar, y la realización de ejemplo usa el cifrado de flujo RC4 por su bajo coste computacional.

Según una realización de ejemplo, los bytes más significativos de la clave SK de sesión, tales como SK[159:32] (16 bytes), se pueden usar para inicializar el cifrado de flujo en 60 en el comienzo de la sesión de cifrado. Después de la inicialización, el cifrado produce una secuencia de bytes K_0 K_1 K_2 K_3 .... Tanto el anfitrión 100 como el esclavo 110 computan la misma secuencia de bytes K ya que cada uno inicializa el flujo de cifrado con la misma clave SK de sesión. El anfitrión 100 entonces es capaz de cifrar en 60 un paquete de comando para su transmisión al esclavo 110 realizando una operación OR exclusiva ("XOR") de los bytes de comando y datos con K_i, donde el valor i es aumentando por cada byte cifrado. Tras la recepción del paquete de comando cifrado, el esclavo 110 descifra después en 60 el paquete recibido realizando la operación XOR de los bytes con los mismo K bytes del cifrado. De manera similar, el esclavo 110 cifra en 60 el paquete de respuesta y transmite el paquete de respuesta cifrado que el anfitrión 100 es capaz de descifrar usando los mismos K bytes usados por el esclavo 110 al cifrar el paquete de respuesta.

Como se mencionó anteriormente, los bytes más significativos de la clave SK de sesión se pueden usar para una sesión de cifrado. Los bytes menos significativos de la clave SK de sesión, en este caso SK[31:0] (4 bytes), se pueden usar para inicializar en 70 el generador de direcciones de esclavo por el esclavo 110 y el anfitrión 100.

El esclavo 110 puede usar una dirección de 10 bits en el bus 120. Según una realización de ejemplo en la que el anfitrión 100 es un dispositivo de impresión y cada dispositivo 110 esclavo es un cartucho de tóner/tinta diferente, los cuatro bits más significativos de la dirección de esclavo pueden ser fijos y asignados a un valor que corresponde con un tipo de tinta o tóner – cian, magenta, amarillo o negro, por ejemplo. Los seis bits menos significativos de la dirección de esclavo de 10 bits pueden ser establecidos entonces mediante un generador de números pseudoaleatorios (PRNG) dentro del esclavo 110 y el anfitrión 100. Después de que el esclavo 110 es reiniciado, los seis bits menos significativos de su dirección de esclavo, esto es, la dirección I^2C de esclavo, en el bus 120 son 0. Cuando el anfitrión 100 da instrucciones al esclavo 110 para cambiar su dirección de esclavo en 70, los seis bits menos significativos de la dirección de esclavo se establecen a partir de los bits predeterminados en el siguiente valor del PRNG.

De acuerdo con una realización de ejemplo, el PRNG puede ser un generador de congruencia lineal (LCG) y puede generar el número X_0 pseudoaleatorio como sigue:

$$X_n = 2891336453\ X_{n-1} * 1523469037 \bmod 2^{32}$$

donde X_{n-1} representa el valor actual del número X_n. Se entiende que se pueden utilizar otros LCG y/o PRNG para generar el número pseudoaleatorio X_n.

Según una realización de ejemplo, el LCG se inicializa con un número predeterminado de bytes de la clave SK de sesión, tales como los cuatro bytes menos significativos, SK[31:0], de manera tal que:

$$X_0 = SK[31:0]$$

Después el anfitrión 110 lee la respuesta al comando de establecimiento de dirección, se calcula el siguiente valor del LCG (X_n) y se establece la dirección de esclavo (I^2C) en 80 para ser un subconjunto predeterminado de bits de X_n. En una realización de ejemplo,

$$\text{Dirección de Esclavo}[5:0] = X_n[29:24]$$

El anfitrión 100 envía los comandos para cambiar las direcciones al esclavo 110 de una manera periódica, después

de lo que el anfitrión 100 y el esclavo 110 computan cada uno la nueva dirección X_n para el esclavo 110.

Después de esto, el esclavo 110 no responderá a las solicitudes de sondeo de dirección hasta después de que haya recibido una solicitud de estado desde el anfitrión 100 usando la nueva dirección X_n.

El anfitrión 100 y el esclavo 110 se comunican usando los paquetes de comando y respuesta a través del bus 120. Los paquetes contienen un valor de verificación de redundancia cíclica (CRC) para comprobar en busca de errores de datos en los paquetes. Si la verificación CRC falla en el esclavo 110, entonces el esclavo 110 devuelve una respuesta CRC al anfitrión 100. Si la verificación CRC falla en el anfitrión 100, entonces el anfitrión 100 retransmite el paquete de comando anterior sin avanzar el cifrado de flujo. En cualquier caso, el anfitrión 100 retransmite el paquete de comando de nuevo sin cambiar su contenido. Este enfoque mantiene el anfitrión 100 y el esclavo 110 sincronizados en el flujo de cifrado y también evita que se usen los mismos bytes de cifrado para cifrar diferentes datos.

El sistema de comunicación anfitrión-esclavo descrito anteriormente usa un esquema de comunicación cifrado basado en paquetes. Se proporciona un medio para la detección y corrección de errores utilizando verificación CRC y retransmisión de paquetes. El anfitrión 100 y el esclavo 110 intercambian valores de modo que cada uno computa una clave SK de sesión a partir de una clave secreta conocida por tanto el anfitrión 100 como el esclavo 110 pero no la intercambian a través del bus 120. La clave SK de sesión se usa después para inicializar tanto el cifrado de flujo como la función de dirección de bus. Con respecto a lo anterior, el anfitrión 100 y el esclavo 110 cifran y descifran cada uno sus comunicaciones realizando la operación XOR de los datos transmitidos/recibidos con los bytes del cifrado de flujo. El anfitrión 100 cambia de manera periódica y/u ocasional las direcciones de esclavo a través del bus 120.

Las ventajas sobre los sistemas existentes incluyen la detección y corrección de errores, las comunicaciones cifradas, y el método de cambio de dirección seguro que estará siempre sincronizado entre el anfitrión 100 y el esclavo 110. La detección y corrección de errores aumenta la fiabilidad en ambientes ruidosos. El cifrado de datos evita que un atacante analice el tráfico de bus para aprender el significado de los comandos y los datos compartidos entre el anfitrión 100 y el esclavo 110. Cuando se implementa en un sistema en el que el anfitrión 100 es una impresora y el esclavo 110 está asociado con un cartucho de tóner o de tinta consumible, el método de cambio de direcciones anteriormente descrito permite que un esclavo 110 se mueva de impresora a impresora sin problemas mientras se mantiene una comunicación segura con la impresora conectada.

Aunque lo anterior describe realizaciones de ejemplo, son posibles muchas variaciones dentro del alcance de la presente descripción. Por ejemplo, como se discutió anteriormente se usa un cifrado de flujo para cifrar los datos debido a su simplicidad. De manera alternativa, un cifrado de bloques, tal como el Estándar de Cifrado Avanzado (AES), ofrecería una seguridad relativamente mayor pero con un alto coste computacional. En dicha realización alternativa, alguna o todas las claves SK de sesión determinadas se usarían para realizar el cifrado y descifrado de la información a transmitir y de la información recibida, respectivamente, de acuerdo con el cifrado de bloques concreto utilizado. El protocolo corrige los errores mediante la retransmisión de paquetes. Además, se podría usar un esquema de corrección de errores precoz en el que los bits de corrección de errores se incluyen en el paquete transmitido. Aún más, se podría usar un bus direccionado diferente, tal como el Bus en Serie Universal (USB), para el bus 120 en lugar de un bus que utiliza el protocolo I^2C.

La descripción anterior de una o más realizaciones de ejemplo se ha presentado con propósitos de ilustración. No está destinado a ser exhaustivo o limitar la aplicación a las formas precisas descritas, y obviamente son posibles muchas modificaciones y variaciones a la luz de la enseñanza anterior. Se entiende que la invención se puede poner en práctica de maneras distintas a las específicamente establecidas en la presente memoria sin salir del alcance de la invención. Se pretende que el alcance de la aplicación sea definido por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Una circuitería (230) de dispositivo esclavo, incluida en un dispositivo (110) esclavo, comprendiendo la circuitería de dispositivo esclavo:

un procesador (200) y una memoria (210) acoplada al mismo que tiene almacenadas dentro instrucciones de código de programa que, cuando son ejecutadas por el procesador (200), provocan al procesador (200) a:

después de que la circuitería (230) del dispositivo esclavo se reinicie, determinar el valor de semilla (20) basado en un valor de semilla del dispositivo esclavo anterior al reinicio del dispositivo esclavo;

recibir un número de anfitrión desde un anfitrión (100) que es sustancialmente aleatorio;

generar una clave de sesión (50) basada en el valor de semilla determinado y el número de anfitrión, siendo la clave de sesión sustancialmente aleatoria; y

realizar las operaciones cifrado y descifrado (60) en base a la clave de sesión generada en los datos a transmitir y los datos recibidos por la circuitería (230) de esclavo, respectivamente, y determinar un valor de dirección (80) en base a la clave de sesión generada para comunicarse con el anfitrión (100),

en donde sustancialmente aleatorio comprende ser uno de entre un número aleatorio real y un número pseudoaleatorio, y en donde una primera parte de la clave de sesión es usada por el procesador (200) que realiza las operaciones de cifrado y descifrado (60) en los datos a transmitir y a recibir, respectivamente, y una segunda parte de la clave de sesión se usa para generar el valor de dirección.

2. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde el procesador (200) calcula un valor del identificador de sesión (10) después de que la circuitería (230) del dispositivo esclavo se reinicie, estando basado el valor del identificador de sesión en un valor de identificador de sesión anterior a que la circuitería (230) del dispositivo esclavo se reiniciara, estando basada la clave de sesión en el valor del identificador de sesión calculado.

3. La circuitería (230) del dispositivo esclavo de la reivindicación 2, en donde el valor del identificador de sesión calculado es el valor del identificador de sesión anterior a que la circuitería (230) del dispositivo esclavo se reiniciara que es aumentado o disminuido.

4. La circuitería (230) del dispositivo esclavo de la reivindicación 2, en donde el procesador (200) determina un número sustancialmente aleatorio (30) basado sólo en el valor de semilla determinado, el valor del identificador de sesión calculado y el número de serie del dispositivo esclavo, y en donde la clave de sesión está basada en el número sustancialmente aleatorio.

5. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde el valor de semilla determinado se determina usando un algoritmo hash seguro.

6. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde el valor de semilla determinado es sustancialmente aleatorio.

7. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde la clave de sesión está basada en una clave de cifrado secreta del dispositivo esclavo.

8. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde las instrucciones que provocan que el procesador determine la clave de sesión (50) usan un código de autenticación de mensaje basado en hash (HMAC).

9. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde las operaciones de cifrado y descifrado forman parte de un cifrado de flujo para comunicarse con el anfitrión.

10. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde el valor de semilla determinado está basado en un valor de semilla del dispositivo esclavo anterior a que el dispositivo esclavo se reiniciara.

11. La circuitería (230) del dispositivo esclavo de la reivindicación 1, en donde un envase de tóner incluye la circuitería de dispositivo esclavo.

12. La circuitería (230) de dispositivo esclavo de la reivindicación 1, en donde el procesador (200) funciona como un generador de números pseudoaleatorios que se inicializa con una segunda parte de la clave de sesión, y una parte de la salida del generador de números pseudoaleatorios forma una parte del valor de dirección del dispositivo (110) esclavo.

13. La circuitería (230) del dispositivo esclavo de la reivindicación 2, en donde el procesador determina un número sustancialmente aleatorio basado en un algoritmo SHA-1 del valor de semilla determinado, del valor del identificador de sesión calculado y del número de serie del dispositivo (110) esclavo. identifica un número aleatorio de esclavo (30) a partir de una parte del número sustancialmente aleatorio, y determina la clave de sesión (50) basado en un

código de autenticación de mensajes basado en hash (HMAC) del número de anfitrión recibido del anfitrión (100), el número aleatorio de esclavo, una clave de cifrado secreta del dispositivo (110) esclavo y el valor del identificador de semilla, siendo la clave de sesión un número sustancialmente aleatorio.

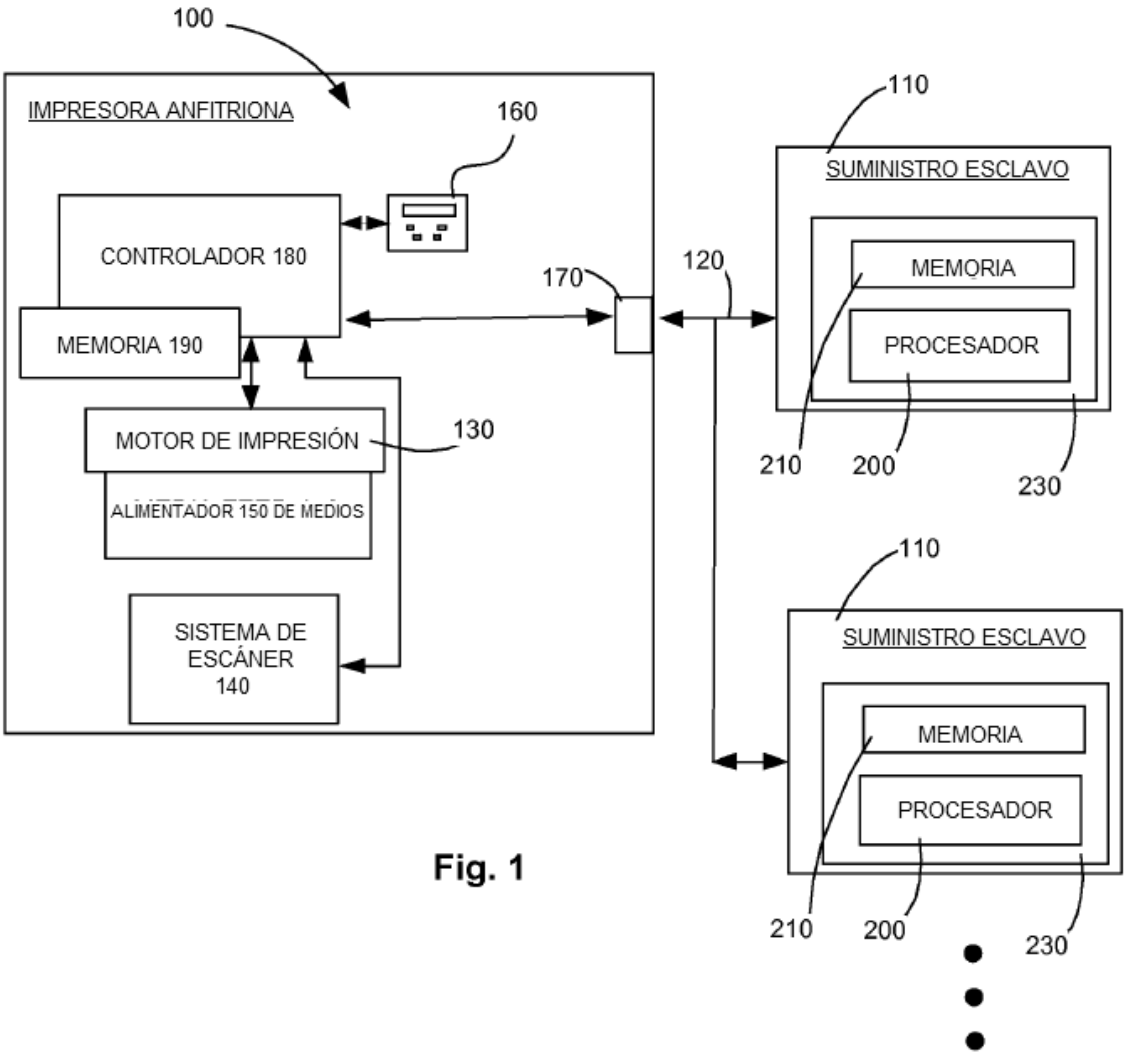
100
IMPRESORA ANFITRIONA
160
CONTROLADOR 180
MEMORIA 190
170
120
MOTOR DE IMPRESIÓN
130
ALIMENTADOR 150 DE MEDIOS
SISTEMA DE ESCÁNER 140
110
SUMINISTRO ESCLAVO
MEMORIA
PROCESADOR
210
200
230
110
SUMINISTRO ESCLAVO
MEMORIA
PROCESADOR
210
200
230

Fig. 1

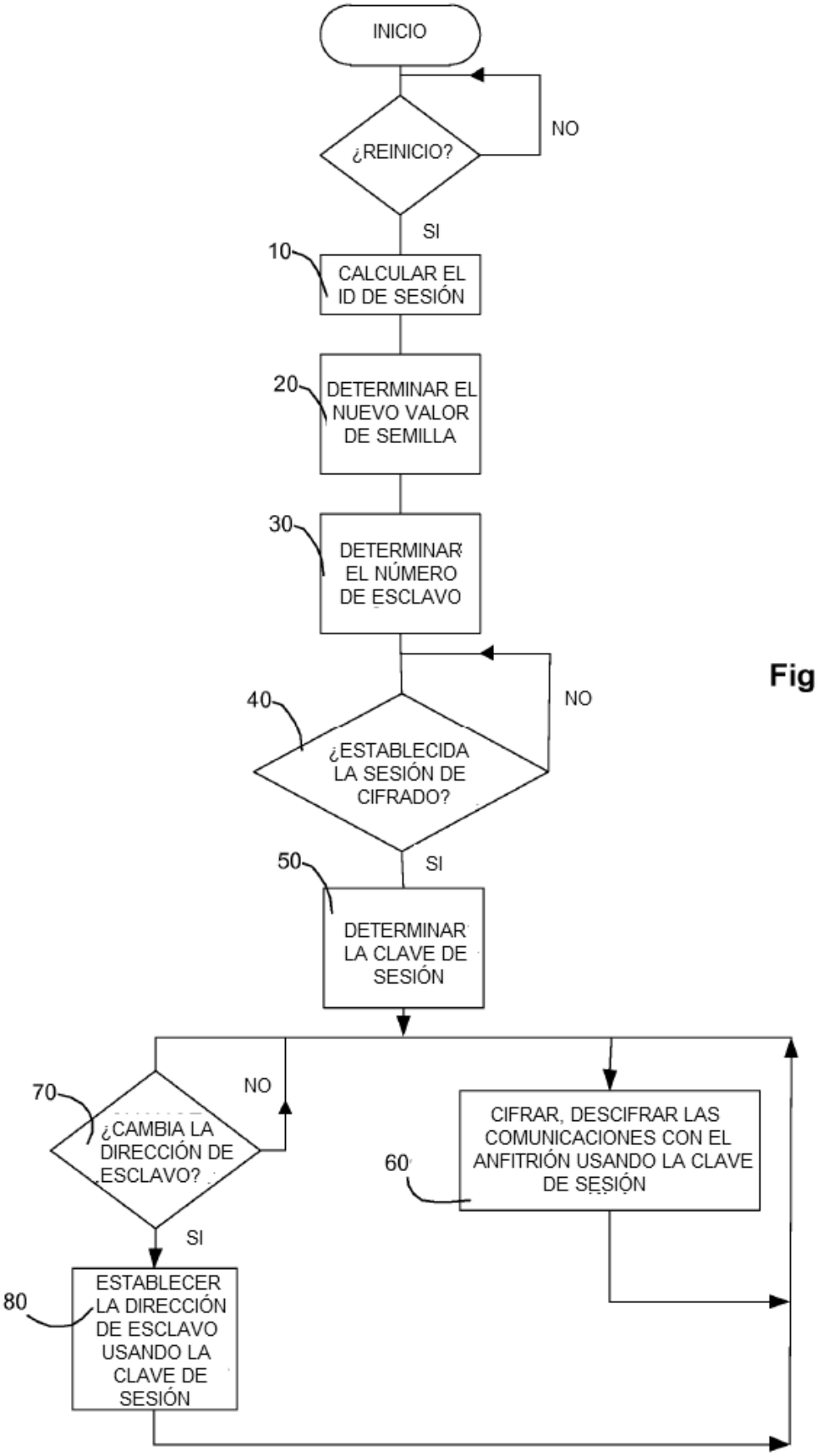
INICIO
¿REINICIO?
NO
SI
10
CALCULAR EL ID DE SESIÓN
20
DETERMINAR EL NUEVO VALOR DE SEMILLA
30
DETERMINAR EL NÚMERO DE ESCLAVO
40
¿ESTABLECIDA LA SESIÓN DE CIFRADO?
NO
SI
50
DETERMINAR LA CLAVE DE SESIÓN
70
¿CAMBIA LA DIRECCIÓN DE ESCLAVO?
NO
SI
80
ESTABLECER LA DIRECCIÓN DE ESCLAVO USANDO LA CLAVE DE SESIÓN
60
CIFRAR, DESCIFRAR LAS COMUNICACIONES CON EL ANFITRIÓN USANDO LA CLAVE DE SESIÓN

Fig. 2