(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 683 998**

(51) Int. CI.:

G11C 29/12 (2006.01)
G11C 16/22 (2006.01)
G11C 7/24 (2006.01)
G06F 12/14 (2006.01)
G06F 21/79 (2013.01)
G11C 29/44 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **15.05.2014 E 14168529 (7)**

(97) Fecha y número de publicación de la concesión europea: **16.05.2018 EP 2945092**

(54) Título: **Dispositivo de memoria con modo de prueba segura**

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
01.10.2018

(73) Titular/es:

WINBOND ELECTRONICS CORP. (100.0%)
No. 8 Keya 1st Rd., Daya District, Central Taiwan Science Park,
Taichung City, Taiwan., TW

(72) Inventor/es:

TASHER, NIR;
KALUZHNY, URI;
WEISER, TSACHI y
TEPER, VALERY

(74) Agente/Representante:

CARVAJAL Y URQUIJO, Isabel

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

Dispositivo de memoria con modo de prueba segura

Campo de la invención

La presente invención se relaciona en general con la memoria asegurada, y particularmente a métodos y sistemas para asegurar un dispositivo de memoria en modo de prueba.

Antecedentes de la invención

Algunos dispositivos de memoria seguros cifran los datos almacenados usando una clave secreta. Los dispositivos de memoria seguros pueden ser vulnerables a diversos ataques destinados a acceder o modificar información almacenada sensible, y/o interrumpir el funcionamiento del dispositivo de memoria.

En el documento de los Estados Unidos US 2009/0052263 A1 se proporciona un circuito de escritura para impulsar una línea de entrada/salida global para escribir los mismos datos en las celdas de memoria de acuerdo con una combinación de una primera señal de datos de prueba y una segunda señal de datos de prueba en un modo de prueba independientemente de las señales de datos de entrada.

Resumen de la invención

En contraste, se sugiere un método en un dispositivo de memoria que comprende un controlador de memoria y una memoria que opera en un modo de prueba, que incluye un vector de datos de prueba para ser escrito en la memoria que se recibe por el controlador de memoria. El vector de datos de prueba se escribe en la memoria solo si el vector de datos de prueba pertenece a un conjunto predefinido de vectores de datos de prueba almacenados en el controlador de memoria. Si el vector de datos de prueba no pertenece al conjunto de vectores de datos de prueba, el vector de datos de prueba recibido se convierte en uno de los vectores de datos de prueba del conjunto predefinido de vectores de datos de prueba por el controlador de memoria, y el vector de datos de prueba convertido se escribe por el controlador de memoria a la memoria.

En algunas realizaciones, convertir el vector de datos de prueba recibido en uno de los vectores de datos de prueba del conjunto predefinido de vectores de datos de prueba incluye seleccionar un subconjunto de bits del vector de datos de prueba recibido, y reemplazar los bits restantes del vector de datos de prueba recibidos con repeticiones periódicas del subconjunto seleccionado.

En otras realizaciones, en cualquiera de los vectores de prueba, todos los bits de orden par equivalen a un primer valor de bit y todos los bits de orden impar equivalen a un segundo valor de bit, y convertir el vector incluye seleccionar, en el vector, un bit par representativo y un bit impar representativo, reemplazando los bits de orden par del vector con el bit par representativo, y reemplazando los bits de orden impar del vector con el bit impar representativo.

En una realización, probar la memoria incluye leer una palabra de datos de un vector de prueba previamente escrito, y exponer afuera la información codificada del dispositivo de memoria acerca de errores en la palabra de datos leídos. En otra realización, exponer la información codificada incluye exponer un número total de los errores. En aún otra realización, exponer la información codificada incluye no exponer ubicaciones verdaderas de los errores.

En algunas realizaciones, no exponer las ubicaciones verdaderas de los errores incluye desplazar bits de orden par y de orden impar que indican errores a diferentes ubicaciones de orden par o de orden impar respectivas. En otras realizaciones, exponer la información codificada incluye exponer adicionalmente una ubicación verdadera de un subconjunto de los errores.

Se proporciona adicionalmente, de acuerdo con una realización de la presente invención, un dispositivo de memoria que incluye una memoria y un controlador de memoria. El controlador de memoria está configurado para operar en modo de prueba, para recibir un vector de datos de prueba para escribir en la memoria, para escribir el vector de datos de prueba en la memoria solo si el vector de datos de prueba pertenece a un conjunto predefinido de vectores de datos de prueba almacenados en el controlador de memoria y, si el vector de datos de prueba no pertenece al conjunto predefinido de vectores de datos de prueba, para convertir el vector de datos de prueba recibido en uno de los vectores de datos de prueba del conjunto predefinido de vectores de datos de prueba, y para escribir el vector de datos de prueba convertido en la memoria. Debe entenderse que la presente invención está definida por las reivindicaciones independientes. Otras realizaciones preferidas se definen por las reivindicaciones dependientes. La presente invención se comprenderá más completamente a partir de la siguiente descripción detallada de las realizaciones de la misma, tomada junto con los dibujos en los cuales:

Breve descripción de los dibujos

La Figura 1 es un diagrama de bloques que ilustra esquemáticamente un sistema para probar un dispositivo de memoria, de acuerdo con una realización de la presente invención;

La Figura 2 es un diagrama de flujo que ilustra esquemáticamente un método para escribir patrones de prueba de forma segura, de acuerdo con una realización de la presente invención;

La Figura 3 es un diagrama de flujo que ilustra esquemáticamente un método para probar un dispositivo de memoria, de acuerdo con una realización de la presente invención; y

La Figura 4 es un diagrama que ilustra esquemáticamente un método para probar un dispositivo de memoria que usa una función unidireccional, de acuerdo con una realización de la presente invención.

Descripción detallada de las realizaciones

Resumen

Las realizaciones de la presente invención que se describen aquí proporcionan métodos y sistemas mejorados para asegurar el dispositivo de memoria contra ataques que intentan explotar el modo de prueba del dispositivo de memoria. En el modo de prueba, los patrones de prueba típicamente se almacenan y se leen sin cifrar, con el fin de facilitar la interpretación de los resultados de la prueba. Aunque normalmente solo los probadores autorizados tienen permiso para probar un dispositivo de memoria, las partes no autorizadas pueden intentar acceder o modificar los datos almacenados, interrumpir el funcionamiento del dispositivo de memoria o atacar el dispositivo de memoria explotando las vulnerabilidades del modo de prueba del dispositivo. En la descripción que sigue, se asume que el dispositivo de memoria se comunica con un servidor externo (por ejemplo, el probador) a través de un enlace no seguro.

En algunas realizaciones, el modo de prueba especifica un pequeño conjunto de patrones de prueba predefinidos (también denominados aquí como vectores de prueba). Cuando se opera en modo de prueba, el dispositivo de memoria permite escribir en la matriz de memoria solamente palabras de datos que pertenecen al conjunto de patrones válidos predefinidos. Además, cuando se recibe una palabra de datos que no pertenece al conjunto de patrones de prueba, el dispositivo de memoria convierte la palabra de datos en uno de los patrones de prueba predefinidos antes del almacenamiento. Al permitir escribir en el dispositivo de memoria bajo patrones de prueba limitados en lugar de datos significativos, la seguridad del dispositivo mejora significativamente.

En algunas realizaciones, el dispositivo de memoria lee a partir de la memoria un patrón de prueba previamente escrito, el cual posiblemente contiene uno o más errores. El dispositivo de memoria convierte el patrón de lectura en uno de los patrones válidos y compara el resultado con el patrón de lectura, con el fin de generar una palabra de bits erróneos. Luego, el dispositivo de memoria manipula la palabra de los bits erróneos para camuflar las ubicaciones de error exactas, pero por otro lado proporciona al probador información significativa para probar y depurar. Esta manipulación puede involucrar, por ejemplo, el desplazamiento de los bits erróneos a otras ubicaciones de bits. En otras palabras, el dispositivo de memoria rastrea el número de errores, pero oculta las ubicaciones verdaderas de los errores. En una realización, el dispositivo de memoria codifica adicionalmente el número de errores y la ubicación del bit erróneo más significativo.

En algunas realizaciones, el servidor escribe datos de prueba en un área de memoria dada (no necesariamente un área contigua). El servidor proporciona adicionalmente al dispositivo de memoria un resultado esperado de una función unidireccional aplicada a los datos almacenados en esta área de memoria. La función unidireccional se define de tal manera que no es posible recuperar ninguna de las palabras de datos en el área de memoria del resultado. Para probar la integridad del área de memoria, el dispositivo de memoria vuelve a calcular el resultado de la función unidireccional y compara el resultado recalculado con el resultado esperado proporcionado por el servidor.

Cualquier diferencia entre los resultados esperados y recalculados indica que los datos leídos del área de memoria contienen uno o más errores. El dispositivo de memoria produce solo el resultado binario de la comparación, exponiendo así solo la información esencial. Este esquema de prueba permite probar la integridad de los datos escritos, pero hace que sea imposible adivinar los datos en sí.

En algunas realizaciones, el dispositivo de memoria proporciona protección contra un ataque que intenta adivinar el contenido de la memoria enviando múltiples versiones diferentes del resultado de la función unidireccional. El dispositivo de memoria cuenta el número de veces que el dispositivo recibe dicho resultado esperado, y si este número excede un umbral predefinido, el dispositivo desactiva el modo de prueba o toma otras medidas de protección adecuadas.

En una realización, el dispositivo de memoria desactiva la prueba de la memoria al detectar que una o más claves secretas están instaladas. Las claves secretas se pueden usar, por ejemplo, para cifrado y/o autenticación. Por otra parte, cuando las claves secretas no están instaladas, el dispositivo no puede llevar a cabo operaciones de cifrado y/o autenticación. Por lo tanto, cuando se instala una clave secreta, el dispositivo de memoria deshabilita la escritura, lectura o ambos, datos sin cifrar hacia/a partir de la memoria, por ejemplo, con el propósito de prueba. En esta realización, para permitir la prueba, el dispositivo de memoria primero borra toda la memoria, que incluye cualquier clave secreta instalada. Alternativamente, el dispositivo de memoria elimina solo las claves secretas y cualquier otra información secreta de la memoria.

Las técnicas divulgadas permiten probar de forma segura un dispositivo de memoria de diversas maneras, tal como limitando los datos que pueden escribirse en un pequeño conjunto de patrones de prueba, exponiendo solo información codificada y limitada sobre los resultados de las pruebas, y habilitando la prueba de memoria solo después del borrado de cualquier clave secreta y/o datos secretos.

Descripción del sistema

La Figura 1 es un diagrama de bloques que ilustra esquemáticamente un sistema 20 para probar un dispositivo 24 de memoria, de acuerdo con una realización de la presente invención. El sistema 20 comprende un servidor 28, el cual escribe datos y/o comandos en el dispositivo 24 de memoria, y lee datos y/o resultados de prueba a partir del dispositivo de memoria. El dispositivo 24 de memoria comprende un controlador 32 de memoria, que se comunica con el servidor 28 usando una interfaz 34 de memoria sobre un enlace 38 no seguro.

El dispositivo 24 de memoria comprende además una memoria 40 no volátil que almacena los datos recibidos del controlador 32 de memoria para su almacenamiento, y recupera los datos almacenados para el controlador 32 de memoria tras la solicitud. En la realización de la Figura 1, la memoria 40 comprende una memoria flash. En realizaciones alternativas, se puede usar cualquier otra memoria adecuada no volátil de cualquier tecnología adecuada, tal como, por ejemplo, unidad de estado sólido (SDD) con base en, EEPROM, ROM programable de una sola vez (OTP), RAM resistiva (RRAM), almacenamiento magnético como una unidad de disco duro (HDD), almacenamiento óptico y similares. En la descripción que sigue, se asume que la memoria 40 comprende una memoria flash, como un ejemplo representativo de una memoria no volátil, pero las técnicas descritas son aplicables a cualquier otra memoria adecuada.

Una unidad 44 de configuración de modo configura el dispositivo 24 de memoria para que funcione en uno de los dos modos de funcionamiento, denominados aquí modos operativos y de prueba. Aunque el modo de prueba está restringido típicamente a usuarios autorizados, los atacantes no autorizados pueden forzar al dispositivo al modo de prueba de funcionamiento.

Cuando está en modo operativo, el controlador 32 de memoria puede aplicar operaciones criptográficas a los datos que se almacenarán en la memoria 40 flash, y a los datos recuperados de la memoria flash, utilizando un motor 52 criptográfico. Cuando está en modo operativo, un selector 48 se interconecta entre el motor 52 y la memoria 40 flash. En algunas realizaciones, el motor 52 cifra los datos antes de almacenarlos, y descifra los datos leídos de la memoria flash.

Alternativamente o adicionalmente, el motor 52 calcula una indicación criptográfica de los datos, y almacena la indicación junto con los datos en la memoria 40 flash. Cuando se recuperan los datos, el motor 52 puede validar la integridad de los datos calculando la indicación de los datos leídos y comparados con la indicación almacenada. Una unidad 56 de clave secreta contiene una o más claves secretas para ser utilizadas por el motor 52 criptográfico.

Cuando se opera en el modo de prueba, el dispositivo 32 de memoria configura el selector 48 para interconectar una unidad 60 de lectura/escritura (R/W) de prueba a la memoria 40 flash en lugar del motor 52 criptográfico. Por lo tanto, en el modo de prueba, la información que puede ser intercambiada entre la memoria flash y el servidor es controlada por la unidad 60 de prueba R/W.

En algunas realizaciones, cuando se escriben datos en la memoria flash en el modo de prueba, la unidad 60 R/W puede escribir en la memoria flash solamente palabras de datos que pertenecen a un pequeño conjunto de patrones 64 de prueba predefinidos. Además, unidad 60 R/W convierte cualquier dato recibido de la interfaz 34 que no concuerde con ninguno de los patrones 64 en uno de los patrones de prueba permitidos. Como resultado, un usuario no autorizado no puede escribir información significativa en la memoria flash y, por lo tanto, no puede modificar la información sensible que está almacenada en la memoria flash.

Cuando el dispositivo 24 de memoria está en modo de prueba, y lee datos de la memoria 40 flash, la unidad 60 R/W comprueba si hay errores en los datos recuperados. La unidad 60 R/W codifica la información con respecto a los errores, de modo que solo se expone información esencial sobre los errores y entrega la información codificada al

servidor 28 a través de la interfaz 34, exponiendo así solo información mínima y esencial acerca el enlace 38 no seguro.

En algunas realizaciones, la prueba del dispositivo 24 de memoria se basa en el cálculo de un resultado de una función unidireccional sobre algunos datos que se almacenan en la memoria flash. El término "función unidireccional" se refiere a una función que cumple dos criterios: (i) Un cambio en los datos causa un cambio en el resultado de la función con una probabilidad muy alta, y (ii) los datos no se pueden recuperar del resultado.

La implementación de la función unidireccional se realiza típicamente dentro de la unidad 60 R/W, pero alternativamente puede ser parte del motor 52 criptográfico, o dividirse entre los dos. Cuando el servidor almacena datos en el dispositivo de memoria, la unidad 60 R/W almacena junto con los datos y devuelve al servidor el resultado de la función unidireccional aplicada a esos datos. Alternativamente, el servidor comprende medios para calcular el resultado de la función unidireccional respectivo y almacenar el resultado en una memoria local del servidor.

Cuando se realiza la prueba, el servidor proporciona a la unidad 60 R/W el resultado esperado de la función unidireccional y, al leer los datos, la unidad 60 R/W vuelve a calcular la función unidireccional sobre los datos leídos y compara al resultado esperado. Si los dos resultados coinciden, se asume que los datos se leen correctamente, con una alta probabilidad. La unidad 60 R/W puede enviar al servidor 28 solo el resultado de comparación, o el resultado recalculado de la función unidireccional.

En algunas realizaciones, la unidad 60 R/W rastrea adicionalmente el número de veces que la unidad R/W recibe un resultado esperado de la función unidireccional, para ser comparado con el resultado recalculado. Cuando el número de veces supera un umbral predefinido, esto puede indicar un ataque de fuerza bruta, y el dispositivo puede protegerse de dicho ataque desactivando el modo de prueba del dispositivo de memoria o tomando cualquier otra medida de protección adecuada.

En algunas realizaciones, la unidad 60 R/W comprende medios para detectar si la clave 56 secreta está instalada, o no, y puede desactivar la prueba de la memoria hasta que se borre la clave secreta y/u otra información sensible, como se explicó, en una realización, a continuación.

La configuración del dispositivo 24 de memoria en la Figura 1 es una configuración de ejemplo, que se elige puramente en aras de la claridad conceptual. En realizaciones alternativas, también se puede usar cualquier otra configuración adecuada de un dispositivo de memoria. Los diferentes elementos del dispositivo 24, tales como la unidad 60 R/W y el motor 52 criptográfico, pueden implementarse usando cualquier hardware adecuado, tal como en un Circuito Integrado Específico de Aplicación (ASIC) o una Disposición de Compuerta Programable en Campo (FPGA). En algunas realizaciones, algunos elementos del dispositivo 24 pueden implementarse usando software, o usando una combinación de elementos de hardware y software.

En algunas realizaciones, ciertos elementos del sistema 20, tales como el servidor 28 o elementos del controlador 32 de memoria, pueden comprender un procesador de propósito general, el cual está programado en un software para llevar a cabo las funciones descritas aquí.

El software puede descargarse al procesador en forma electrónica, a través de una red, por ejemplo, o alternativamente, puede proporcionarse y/o almacenarse en medios tangibles no transitorios, como una memoria magnética, óptica o electrónica.

Pruebas seguras de la memoria

La Figura 2 es un diagrama de flujo que ilustra esquemáticamente un método para escribir patrones de prueba de forma segura, de acuerdo con una realización de la presente invención. En el método de la Figura 2, se asume que el dispositivo de memoria funciona en modo de prueba. Se asume además que la memoria 40 flash almacena elementos de datos de 32 bits, y que los patrones 64 de prueba comprenden cuatro patrones de 32 bits, cuyos valores se representan en la Tabla 1 a continuación. En cualquiera de los patrones de prueba en la Tabla 1, todos los bits en las posiciones pares de bits tienen un primer valor de bit, y todos los bits en las posiciones de bit impares tienen un segundo valor de bit.

Tabla 1: patrones de prueba válidos

ID Patrón	Valor Binario (32 Bits)	Valor Hexadecimal
0	00000000000000000000000000000000	0x00000000
1	01010101010101010101010101010101	0x55555555
2	10101010101010101010101010101010	0xAAAAAAAA
3	11111111111111111111111111111111	0xFFFFFFFF

El método comienza en una etapa 100 de recepción, recibiendo a partir de la unidad 60 R/W del servidor 28 una palabra de datos a escribir. En el modo de prueba, la palabra de datos debería pertenecer típicamente a patrones 64 de prueba (por ejemplo, la palabra de datos es igual a uno de los patrones en la Tabla 1 anterior). En realizaciones alternativas, en lugar de recibir las palabras de datos directamente, la unidad R/W recibe del servidor un comando que incluye la palabra de datos a escribir.

En una etapa 104 de comprobación, la unidad 60 R/W verifica si la palabra de datos recibida concuerda con cualquiera de los patrones 64 de prueba. Si la palabra de datos difiere de todos los patrones de prueba válidos, la unidad R/W convierte la palabra de datos en una patrón de prueba válido en una etapa 108 de conversión. De lo contrario, la unidad R/W deja la palabra de datos sin cambios. Se tiene en cuenta que en ambos casos, la unidad R/W genera un patrón de prueba válido. En algunas realizaciones, cuando la palabra de datos recibida no coincide con ninguno de los patrones de prueba, el dispositivo de memoria ignora la palabra de datos o el comando que incluye esta palabra de datos.

La unidad 60 R/W puede usar cualquier método adecuado para convertir la palabra de datos recibida en un patrón válido. En algunas realizaciones, para llevar a cabo la conversión, la unidad R/W selecciona primero un bit par representativo y un bit impar representativo entre los 32 bits de la palabra de datos recibida. La unidad R/W puede seleccionar cualquier bit representativo par e impar adecuado, como, por ejemplo, los dos bits más significativos (MS), o los dos bits menos significativos (LS) de la palabra de datos recibida. Luego, la unidad R/W convierte la palabra de datos recibida en un patrón de prueba válido, reemplazando los bits de orden par con el bit representativo par, y los bits de orden impar con el bit representativo impar. En términos generales, la unidad R/W puede convertir la palabra de datos recibida en un patrón válido seleccionando cualquier subconjunto adecuado de bits de la palabra de datos, y reemplazando los bits restantes de la palabra de datos con repeticiones periódicas del subconjunto seleccionado.

En una realización, en la etapa 108, la unidad 60 R/W convierte la palabra de datos en un patrón válido duplicando el valor binario de los dos bits más significativos (MS), quince veces. Por ejemplo, la unidad R/W convierte la palabra de datos de 32 bits 00XX...XXXX en la cual X denota un valor binario que puede ser '0' o '1', en el patrón 0x00000000, y la palabra 01XX...XXXX al patrón 0x55555555 (binario "0101...01010101").

En una etapa 112 de almacenamiento, la unidad 60 R/W escribe la salida de la etapa 108, la cual como se explicó anteriormente es igual a un patrón de prueba válido, a la memoria 40 flash, y el método termina entonces.

El esquema de escritura descrito en la Figura 2 impide que un usuario no autorizado escriba información significativa en la memoria en un intento de leer o manipular información segura o de causar otro daño al dispositivo.

La Figura 3 es un diagrama de flujo que ilustra esquemáticamente un método para probar un dispositivo de memoria, de acuerdo con una realización de la presente invención. El método puede llevarse a cabo, por ejemplo, después de escribir la memoria usando el método de la Figura 2 anterior, o independientemente del método de la Figura 2. La descripción de la Figura 3 se acompaña con un ejemplo numérico en el cual el patrón 0x55555555 (binario "0101...01010101") se escribió previamente en la memoria flash. Los resultados numéricos respectivos adicionales que corresponden a diversas etapas en el diagrama de flujo se representan a continuación en la Tabla 2.

El método comienza por la unidad 60 R/W que lee una palabra que está almacenada en la memoria 40 flash, en una etapa 200 de lectura. En algunas realizaciones, la unidad R/W ejecuta la etapa 200 en respuesta a la recepción de un comando respectivo (no se muestra) del servidor, para leer una o más ubicaciones en la memoria. Cualquiera de los 32 bits de la palabra leída en la etapa 200 puede ser correcto o erróneo. En el ejemplo de la Tabla 2, se asume que la palabra de lectura contiene 21 errores, y en lugar de leer el patrón 0x55555555 previamente escrito, la unidad 60 R/W lee R_WORD=0x93209A6A.

En una etapa 204 de conversión de lectura, la unidad 60 R/W convierte R WORD en uno de los patrones 64 de prueba. En una realización, la conversión es similar a las operaciones de verificación y conversión condicionadas descritas en las etapas 104 y 108 respectivas anteriormente. En el ejemplo de la Tabla 2, los dos bits MS de R_WORD son iguales a "10" y, por lo tanto, R_WORD se convierte en el patrón R_PATTERN=0xAAAAAAAA.

En una etapa 208 de extracción de errores, la unidad R/W compara entre R_WORD y R_PATTERN, utilizando una operación XOR en modo bit a 32 bits, la cual da como resultado una palabra de errores de 32 bits, denominada R_ERRORS. Se tiene en cuenta que cuando R_WORD no contiene errores, R_ERRORS es igual a cero.

En una etapa 212 de extracción de bit de error MS, la unidad R/W extrae una palabra de 32 bits, denominada MS_ERROR_BIT, y que contiene un solo bit '1' en la posición del bit erróneo MS de R_ERRORS. Por ejemplo, la unidad 60 R/W puede borrar la palabra MS_ERROR_BIT completa, y luego definir un bit '1' en la posición del bit que no sea cero más a la izquierda encontrado en R_ERRORS. En el ejemplo de la Tabla 2, MS_ERROR_BIT=0x20000000. En algunas realizaciones, la unidad 60 R/W no proporciona información explícita sobre el bit de error MS, y puede omitirse la etapa 212.

Además en la etapa 212, la unidad 60 R/W extrae una palabra de 32 bits que contiene los bits de error de menos significativos (LS) (si los hay), y que se denota LS_ERROR_BITS. LS_ERROR_BITS es igual a R_ERRORS excluyendo el bit de error MS (por ejemplo, calculado mediante una operación XOR en modo bit entre R_ERRORS y MS_ERROR_BIT). En el ejemplo de la Tabla 2, LS_ERROR_BITS=0x198A30C0;

En una etapa 216 de codificación de error, la unidad 60 R/W codifica los errores en LS_ERROR_BITS en una palabra de 32 bits, a partir de la cual puede concluirse el número total de errores, pero las ubicaciones verdaderas de los bits erróneos no están expuestas. La unidad 60 R/W procesa por separado los bits en las ubicaciones pares e impares de LS_ERROR_BITS como se describe aquí. En la descripción que sigue, los bits "1" par e impar se refieren a los "1" bits cuya posición de bit o ubicación en la palabra de 32 bits es par o impar, respectivamente.

Para procesar los bits "1" pares en la etapa 216, la unidad R/W mueve los bits "1" pares en LS_ERROR_BITS a las ubicaciones pares más a la derecha que están disponibles (es decir, están despejadas). El resultado de esta operación se denota EVN_SHIFT_ERRORS y tiene la forma 000...010101, en la cual el número de unidades es igual al número '1' bits par en LS_ERROR_BITS. En el ejemplo de la Tabla 2, LS_ERROR_BITS contiene cuatro '1' bits pares, y por lo tanto EVN_SHIFT_ERRORS=0x00000055.

De manera similar, para procesar los bits "1" impares, la unidad R/W mueve los bits "1" impares en LS_ERROR_BITS a las ubicaciones impares disponibles más a la derecha. El resultado de esta operación se denota ODD_SHIFT_ERRORS, y tiene la forma 000...101010, en la cual el número de unidades es igual al número de bits impares '1' en LS_ERROR_BITS. En el ejemplo de la Tabla 2, hay seis bits "1" impares en LS_ERROR_BITS y, por lo tanto, ODD_SHIFT_ERRORS=0x00000AAA.

Teniendo en cuenta que las palabras EVN_SHIFT_ERRORS y ODD_SHIFT_ERRORS transmiten información sobre el número de errores, pero a partir de los cuales no se pueden recuperar las ubicaciones verdaderas de los errores.

En una etapa 220 de codificación de error, la unidad 60 R/W combina el bit de error MS MS_ERROR_BIT de la etapa 212, los errores EVN_SHIFT_ERRORS pares e impares y ODD_SHIFT_ERRORS de la etapa 216, y el patrón R_PATTERN de la etapa 204, para producir ENCODED_ERRORS. En algunas realizaciones, la operación de combinación en la etapa 220 comprende una operación XOR en modo bit entre R_PATTERN, MS_BIT_ERROR, EVEN_SHIFTED_ERRORS, y ODD_SHIFTED_ERRORS. En el ejemplo de la Tabla 2 ENCODED_ERRORS=0x8AAAA055. El dispositivo de memoria entrega entonces ENCODED_ERRORS de la etapa 220 al servidor 28.

En la realización descrita anteriormente, la unidad 60 R/W aplica XOR en modo bit con R_PATTERN dos veces: en primer lugar en la etapa 208 al extraer los errores, y en segundo lugar en la etapa 220 cuando codifica los errores. Además, entre la aplicación de las dos operaciones XOR, los bits '1' pares e impares se reposicionan en otras ubicaciones pares e impares respectivas. Teniendo en cuenta que si dos bits de MS en R_WORD contienen errores, el valor de R_PATTERN difiere del valor de patrón verdadero que se almacenó. En este caso, los roles de los bits erróneos y correctos en R_ERRORS se conmutan (es decir, ceros indican los bits erróneos, y unos indican los bits correctos), pero la operación XOR en la etapa 220 cambia estos roles nuevamente en orden.

En la realización descrita anteriormente, la unidad R/W retiene la ubicación del bit de error MS, y desplaza los errores del bit LS a las ubicaciones disponibles más a la derecha. Este esquema retiene así la posición del bit de error MS, pero para los bits de error LS, la unidad R/W genera el número de errores sin exponer sus ubicaciones verdaderas.

Las siguientes etapas descritas son ejecutadas por el servidor 28 después de recibir, a partir del dispositivo de memoria, el resultado ENCODED_ERRORS de la etapa 220 anterior. En una etapa 224 de comparación, el servidor compara entre ENCODED_ERRORS y el patrón de prueba esperado (W_PATTERN). El servidor puede realizar la comparación utilizando una operación XOR en modo bit entre ENCODED_ERRORS y W_PATTERN. El resultado de la comparación se denomina EST_ERRORS, y en el ejemplo de la Tabla 2, EST_ERRORS=0xDFFFF500.

En una etapa 228 de decodificación de errores, el servidor extrae de la información de EST_ERRORS acerca de los respectivos errores de lectura. En la presente realización, el servidor extrae la ubicación exacta del error MS y el número total de errores. En el ejemplo de la Tabla 2, el error de MS se encuentra en la posición más a la izquierda, y hay 21 errores en total.

Tabla 2: Un ejemplo numérico para aclarar el método de la Figura 3

Palabra de 32 bits	Valor binario	Valor hexadecimal
W_PATTERN	01010101010101010101010101010101	0x55555555
R_WORD	10010011001000001001101001101010	0x93209A6A
R_PATTERN	10101010101010101010101010101010	0xAAAAAAAA
R_ERRORS	00111001100010100011000011000000	0x398A30C0

MS_ERROR_BIT	00100000000000000000000000000000	0x20000000
LS_ERROR_BITS	00011001100010100011000011000000	0x198A30C0
EVN_SHIFT_ERRORS	00000000000000000000000001010101	0x00000055
ODD_SHIFT_ERRORS	00000000000000000000101010101010	0x00000AAA
ENCODED_ERRORS	10001010101010101010000001010101	0x8AAAA55
Patrón esperado	01010101010101010101010101010101	0x55555555
EST_ERRORS	11011111111111111111010100000000	0xDFFFF500
Total de errores	21	
MS Error bit	10000000000000000000000000000000	

La configuración de prueba descrita en la Figura 3 es de ejemplo, y también se pueden usar otras configuraciones de prueba. Por ejemplo, en algunas realizaciones, el servidor envía el patrón esperado al dispositivo de memoria, que compara el patrón con el valor almacenado, y retorna a la información codificada del servidor acerca de errores de lectura, si los hay.

En las realizaciones divulgadas, la información codificada con respecto a los errores de lectura incluye la ubicación exacta del error de MS, y el número total de errores. En realizaciones alternativas, la información incluye solo la ubicación del error de MS, o el número total de errores. Además, alternativamente, la información de errores puede incluir la ubicación de un error que no sea el de MS (por ejemplo, la ubicación del error LS) y/o las ubicaciones de múltiples errores.

La realización divulgada se implementa usando cuatro patrones de prueba como se representa en la Tabla 1 anterior. En realizaciones alternativas, sin embargo, se puede usar cualquier otro conjunto adecuado de patrones válidos. Por ejemplo, la cantidad de patrones puede ser distinta de cuatro. Como otro ejemplo, los valores de los patrones pueden ser diferentes de los patrones en la Tabla 1 anterior.

En el método de la Figura 3, los bits erróneos se desplazan a las posiciones disponibles más a la derecha par o impar. Alternativa o adicionalmente, los bits erróneos se pueden desplazar a posiciones pares o impares disponibles que no sean las posiciones más a la derecha. Por ejemplo, si se produce un solo error en la posición del bit LS, este bit de error puede reubicarse en cualquier otra posición de bit impar.

Aunque el método de la Figura 3 describe la prueba de una memoria cuyo tamaño de datos de palabra es de 32 bits, el método es aplicable a cualquier otro tamaño de palabra adecuado.

Prueba asegurada con base en la función unidireccional

La Figura 4 es un diagrama que ilustra esquemáticamente un método para probar un dispositivo de memoria que usa una función unidireccional, de acuerdo con una realización de la presente invención. El método de la Figura 4 comprende una fase de escritura y una fase de lectura. En la fase de escritura, el servidor escribe datos de prueba en un área 260 de memoria en el dispositivo de memoria. Se asume que el servidor posee un resultado esperado de aplicar una función unidireccional sobre los datos de prueba, y que la unidad 60 R/W comprende medios para calcular la misma función unidireccional sobre los datos escritos.

El servidor y la unidad 60 R/W pueden usar cualquier función $F(\cdot)$ unidireccional adecuada que tenga la propiedad de que, dada una entrada A, B=F(A), sea fácil de calcular, pero dado el resultado B, la reproducción de A no es computacionalmente inviable. Además, dadas dos entradas diferentes A y A', la probabilidad de que F(A)=F(A') es muy baja. Las funciones unidireccionales de ejemplo incluyen las funciones SHA-1 y SHA-2 no significativas criptográficas.

En la fase de lectura, el dispositivo de memoria comprueba si se ha producido algún cambio en los datos almacenados en el área 268 de memoria MEM_A, denominada MEM_A', es la misma área de memoria que MEM_A (es decir, comprende las mismas direcciones de memoria), pero el contenido en el área 268 puede ser diferente del contenido original escrito en el área 260 debido a errores.

En algunas realizaciones, para probar la integridad de los datos en MEM_A, la unidad R/W calcula F (MEM_A') y compara el resultado con el resultado esperado F(MEM_A), el cual es proporcionado por el servidor. El servidor puede iniciar la fase de lectura inmediatamente después de escribir los datos de prueba, o en cualquier otra ocasión adecuada. Si F(MEM_A') es igual a F(MEM_A), la integridad de los datos se verifica con alta probabilidad. La unidad 60 R/W entrega al servidor 28 solo el resultado binario de aprobación/falla de la prueba de validación de integridad. Este esquema de prueba comprueba si los datos de prueba se escribieron correctamente al comparar no los datos en sí, sino una indicación de los datos, a la vez que se expone solo un resultado de aprobación/falla binario. Además, el uso de una función unidireccional criptográficamente fuerte hace que no sea factible que un atacante adivine el contenido almacenado.

Un atacante no autorizado puede enviar diversas versiones diferentes del resultado esperado de la función unidireccional, en un intento por descubrir el correcto. En una realización, la unidad 60 R/W realiza un seguimiento del número de dichos intentos, y si este número excede un umbral predefinido, la unidad R/W toma medidas de protección adecuadas, tales como deshabilitar el modo de prueba del dispositivo.

Prueba segura mediante el borrado de claves secretas

En algunas realizaciones, el dispositivo 24 de memoria permite realizar pruebas de memoria solo cuando no se instala una clave secreta. En estas realizaciones, si el dispositivo de memoria está configurado para operar en modo de prueba (por ejemplo, usando la unidad 44 de configuración de modo), pero detecta que se ha instalado una clave secreta, el dispositivo de memoria deshabilita la prueba hasta que se borra la memoria completa, que incluye la clave secreta. Alternativamente, es suficiente que el dispositivo de memoria permita la prueba cuando solo se borren la clave secreta y posiblemente las áreas de memoria limitadas que almacenan datos confidenciales.

Las realizaciones del sistema 20 que se divulgan anteriormente son de ejemplo, y se eligen exclusivamente por razones de claridad conceptual. En sistemas de prueba alternativos, también se puede usar cualquier otra realización adecuada. Por ejemplo, aunque las diversas realizaciones anteriores se describen por separado, otros sistemas de prueba se pueden configurar para ejecutar dos o más de dichas realizaciones simultáneamente. Aunque las realizaciones descritas aquí abordan principalmente pruebas seguras de memorias no volátiles, los métodos y sistemas descritos aquí también se pueden usar en otras aplicaciones, tales como en pruebas seguras de sistemas de almacenamiento de cualquier tamaño y tecnología.

Se apreciará que las realizaciones descritas anteriormente se citan a modo de ejemplo, y que la presente invención no se limita a lo que se ha mostrado y descrito particularmente anteriormente. Más bien, el alcance de la presente invención incluye tanto combinaciones como subcombinaciones de las diversas características descritas anteriormente, así como variaciones y modificaciones de las mismas que se les ocurrirían a los expertos en la técnica al leer la descripción anterior y que no se describen en la técnica anterior.

REIVINDICACIONES

1. Un método que comprende:

en un dispositivo (24) de memoria que comprende una memoria (40) y un controlador (32) de memoria que opera en un modo de prueba, el controlador (32) de memoria recibe un vector de datos de prueba para ser escrito en la memoria (40); escribir el vector de datos de prueba en la memoria (40) solo si el vector de datos de prueba pertenece a un conjunto predefinido de vectores de datos de prueba almacenados en el controlador (32) de memoria; y

si el vector de datos de prueba no pertenece al conjunto predefinido de vectores de datos de prueba, convirtiendo por el controlador (32) de memoria el vector de datos de prueba recibido a uno de los vectores de datos de prueba del conjunto predefinido de vectores de datos de prueba, y escribiendo por el controlador (32) de memoria el vector de datos de prueba convertido a la memoria (40).

2. El método de acuerdo con la reivindicación 1, en donde convertir el vector de datos de prueba recibido en uno de los vectores de datos de prueba del conjunto predefinido de vectores de datos de prueba comprende seleccionar un subconjunto de bits del vector de datos de prueba recibido y reemplazar los bits restantes del vector de datos de prueba recibido con repeticiones periódicas del subconjunto seleccionado.

3. El método de acuerdo con la reivindicación 1, y que comprende probar la memoria (40) leyendo una palabra de datos de un vector de prueba previamente escrito, y exponer fuera del dispositivo (24) de memoria información codificada sobre errores en la palabra de datos leídos.

4. El método de acuerdo con la reivindicación 3, en donde exponer la información codificada comprende exponer un número total de errores, en donde exponer la información codificada comprende no exponer ubicaciones verdaderas de los errores, y en donde no exponer las ubicaciones verdaderas de los errores comprende desplazar bits de orden par y de orden impar que indican errores a diferentes posiciones respectivas de orden par o de orden impar .

5. El método de acuerdo con la reivindicación 3, en donde exponer la información codificada comprende exponer adicionalmente una ubicación verdadera de un subconjunto de errores.

6. Un dispositivo (24) de memoria, que comprende:

una memoria (40); y

un controlador (32) de memoria, el cual está configurado para operar en un modo de prueba, para recibir un vector de datos de prueba para escribir en la memoria (40), para escribir el vector de datos de prueba en la memoria (40) solo si el vector de datos de prueba pertenece a un conjunto predefinido de vectores de datos de prueba almacenados en el controlador (32) de memoria y, si el vector de datos de prueba no pertenece al conjunto predefinido de vectores de datos de prueba, para convertir el vector de datos de prueba recibido a uno de los vectores de datos de prueba del conjunto predefinido de vectores de datos de prueba, y escribir el vector de datos de prueba convertido en la memoria (40).

7. El dispositivo (24) de memoria de acuerdo con la reivindicación 6, en donde el controlador (32) de memoria está configurado para convertir el vector de datos de prueba recibido en uno de los vectores de datos de prueba seleccionando un subconjunto de bits del vector de datos de prueba recibido, y reemplazando los bits restantes del vector de datos de prueba recibidos con repeticiones periódicas del subconjunto seleccionado.

8. El dispositivo (24) de memoria de acuerdo con la reivindicación 6, en donde el controlador (32) de memoria está configurado para leer una palabra de datos de un vector de prueba previamente escrito, y para exponer fuera del dispositivo (24) de memoria información codificada sobre errores en la palabra de datos leídos.

9. El dispositivo (24) de memoria de acuerdo con la reivindicación 8, en donde el controlador (32) de memoria está configurado para exponer un número total de errores, en donde el controlador (32) de memoria está configurado para no exponer ubicaciones verdaderas de los errores y en donde el controlador (32) de memoria está configurado para no exponer las ubicaciones verdaderas de los errores desplazando los bits de orden par y orden impar que indican errores a diferentes ubicaciones respectivas de orden par o de orden impar.

10. El dispositivo (24) de memoria de acuerdo con la reivindicación 8, en donde el controlador (32) de memoria está configurado para exponer adicionalmente una ubicación verdadera de un subconjunto de errores.

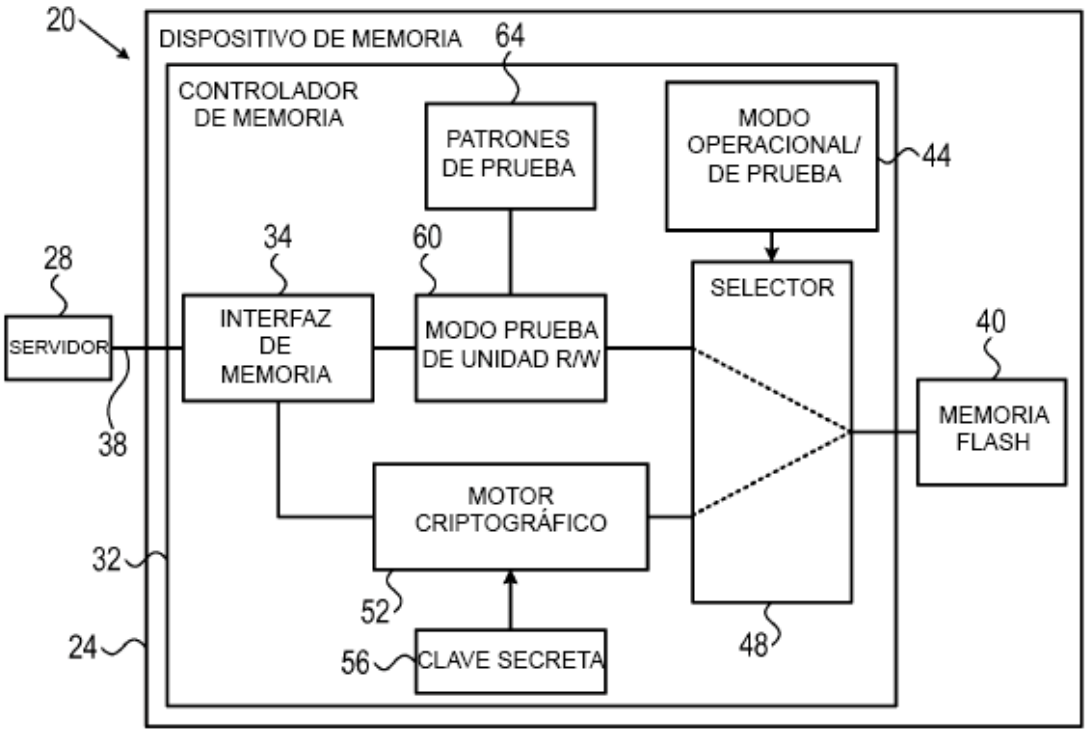
20
DISPOSITIVO DE MEMORIA
64
CONTROLADOR DE MEMORIA
PATRONES DE PRUEBA
MODO OPERACIONAL/ DE PRUEBA
44
34
60
28
SELECTOR
INTERFAZ DE MEMORIA
MODO PRUEBA DE UNIDAD R/W
40
SERVIDOR
MEMORIA FLASH
38
MOTOR CRIPTOGRÁFICO
32
52
24
56
CLAVE SECRETA
48

FIG. 1

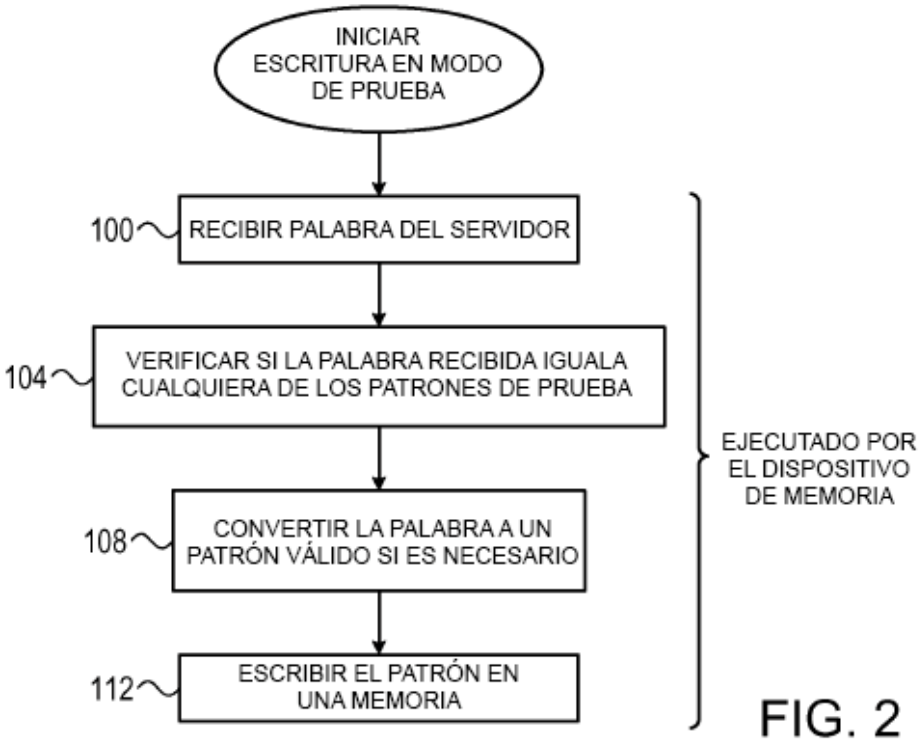
INICIAR ESCRITURA EN MODO DE PRUEBA
100
RECIBIR PALABRA DEL SERVIDOR
104
VERIFICAR SI LA PALABRA RECIBIDA IGUALA CUALQUIERA DE LOS PATRONES DE PRUEBA
108
CONVERTIR LA PALABRA A UN PATRÓN VÁLIDO SI ES NECESARIO
112
ESCRIBIR EL PATRÓN EN UNA MEMORIA
EJECUTADO POR EL DISPOSITIVO DE MEMORIA

FIG. 2

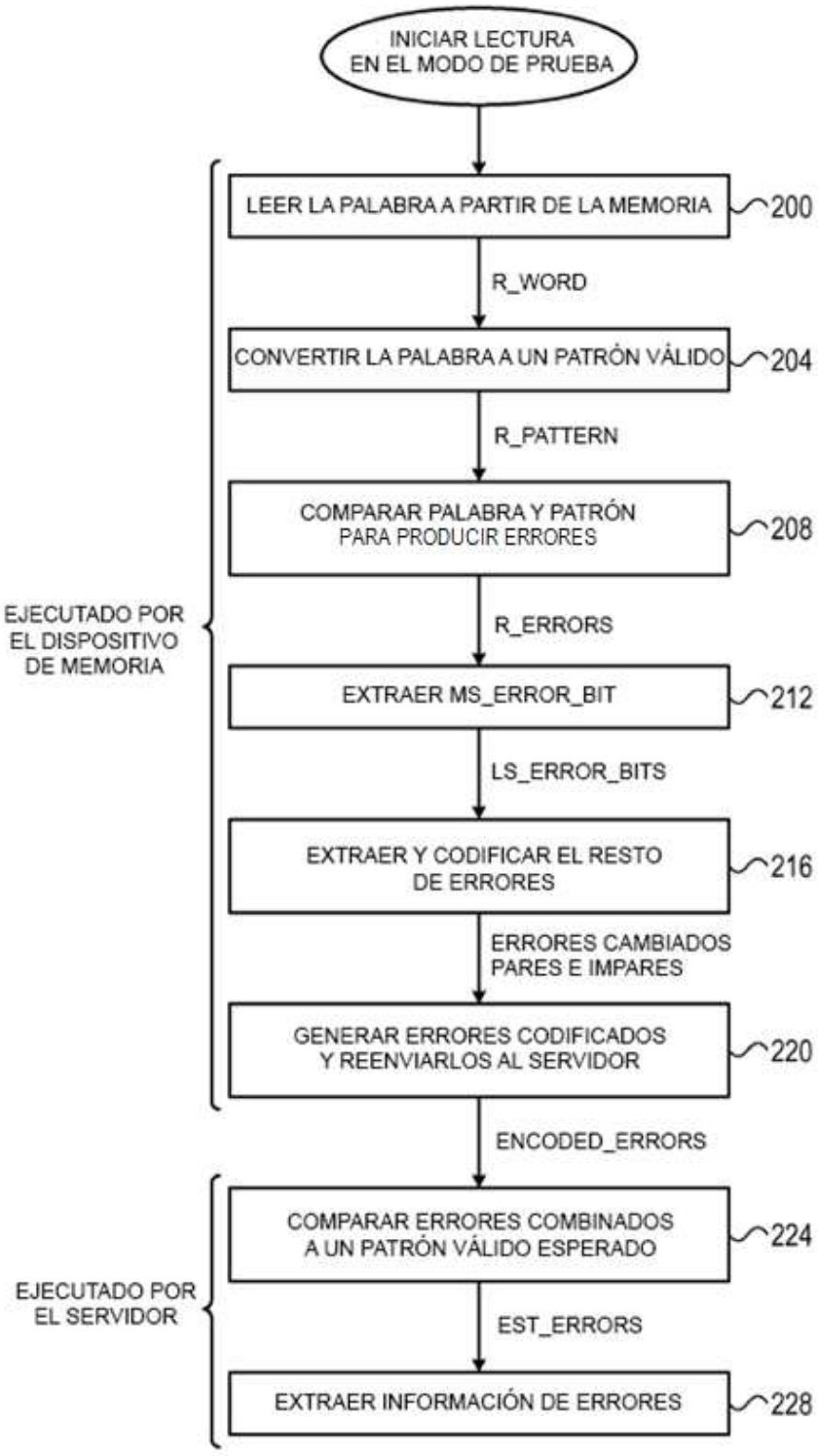
INICIAR LECTURA EN EL MODO DE PRUEBA
LEER LA PALABRA A PARTIR DE LA MEMORIA
200
R_WORD
CONVERTIR LA PALABRA A UN PATRÓN VÁLIDO
204
R_PATTERN
COMPARAR PALABRA Y PATRÓN PARA PRODUCIR ERRORES
208
R_ERRORS
EXTRAER MS_ERROR_BIT
212
LS_ERROR_BITS
EXTRAER Y CODIFICAR EL RESTO DE ERRORES
216
ERRORES CAMBIADOS PARES E IMPARES
GENERAR ERRORES CODIFICADOS Y REENVIARLOS AL SERVIDOR
220
EJECUTADO POR EL DISPOSITIVO DE MEMORIA
ENCODED_ERRORS
COMPARAR ERRORES COMBINADOS A UN PATRÓN VÁLIDO ESPERADO
224
EST_ERRORS
EXTRAER INFORMACIÓN DE ERRORES
228
EJECUTADO POR EL SERVIDOR

FIG. 3

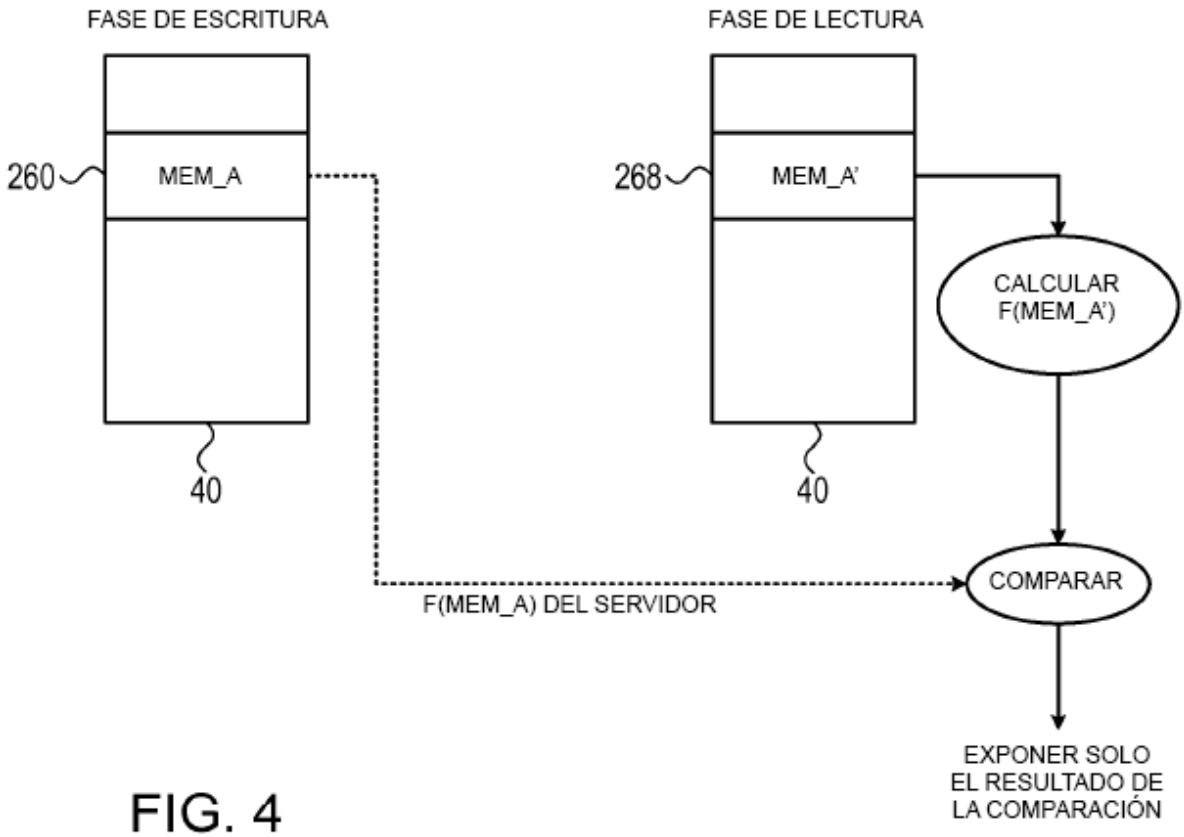
FASE DE ESCRITURA
260
MEM_A
40
FASE DE LECTURA
268
MEM_A'
40
CALCULAR
F(MEM_A')
COMPARAR
F(MEM_A) DEL SERVIDOR
EXPONER SOLO
EL RESULTADO DE
LA COMPARACIÓN

FIG. 4