(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 783 878**

(51) Int. CI.:

G06F 21/53 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **21.11.2017 E 17202682 (5)**

(97) Fecha y número de publicación de la concesión europea: **11.03.2020 EP 3324324**

(54) Título: **Procedimiento de protección de un dispositivo electrónico que ejecuta un programa contra ataques por inyección de error y por confusión de tipo**

(30) Prioridad:

21.11.2016 FR 1661310

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
18.09.2020

(73) Titular/es:

IDEMIA IDENTITY & SECURITY FRANCE (100.0%)
2, Place Samuel de Champlain
92400 Courbevoie, FR

(72) Inventor/es:

BAILLY, ALEXIS;
MAGHREBI, HOUSSEM;
SERRE, AHMADOU y
BRUGNON, MARC

(74) Agente/Representante:

ELZABURU, S.L.P

DESCRIPCIÓN

Procedimiento de protección de un dispositivo electrónico que ejecuta un programa contra ataques por inyección de error y por confusión de tipo

Ámbito de la invención

La presente invención concierne a un procedimiento de protección de un dispositivo electrónico que ejecuta un programa contra ataques por inyección de error y por confusión de tipo, puestos en práctica sobre una variable destinada a ser utilizada por el programa.

Estado de la técnica

De modo conocido, un ataque por inyección de error consiste en perturbar el entorno físico de un dispositivo electrónico que ejecuta un programa, de modo que se modifique el valor memorizado por el dispositivo de una variable destinada a ser utilizada por el programa. Tales perturbaciones pueden ser producidas de diferentes maneras: variación de una tensión de alimentación, variación de una frecuencia de reloj del dispositivo, emisión de radiación electromagnética o láser, etc.

Para proteger una variable contra un ataque por inyección de error, existen varios tipos de contramedidas.

Una primera contramedida para proteger una variable contra un ataque por inyección de error consiste simplemente en duplicar el contenido de la pila de ejecución en una pila de salvaguarda. Para verificar la integridad de una variable, se compara el valor de la variable memorizada en la pila de ejecución y el valor de su copia en la pila de salvaguarda. Si los dos valores comparados son diferentes, se detecta un error.

Una segunda contramedida, descrita en el documento EP1960934B1, consiste en calcular un dato de control de integridad Q de la variable y memorizar este dato de control de integridad en una pila de control específica. Para aplicar este principio de protección, se asignan dos pilas independientes en la memoria del dispositivo, como está representado en la figura 1a:

- una pila de ejecución P1 que contiene el valor de cada variable destinada a ser utilizada por el programa, y
- una pila de control P2 que contiene los datos de control de integridad Q de cada variable.

Están previstos igualmente dos registros de índices:

- un primer registro de índices ind1 que contiene datos de direccionamiento adaptados para localizar las variables contenidas en la pila de ejecución P1, y
- un segundo registro de índices ind2 que contiene datos de direccionamiento adaptados para localizar los datos de control de integridad contenidos en la pila de control P2.

Para verificar la integridad de una variable, se lee el valor corriente de esta variable presente en la pila de ejecución (siendo localizado este valor gracias a un dato de direccionamiento asociado a la variable, presente en el primer registro de índices ind1), se pone en práctica un cálculo de integridad sobre la base del valor leído y se compara el resultado de este cálculo con el dato de integridad asociado a la variable que está presente en la pila de control (siendo localizado el dato de integridad gracias a un dato de direccionamiento correspondiente en el segundo registro de índices ind2). Si los dos valores comparados son diferentes, se detecta un error.

Sin embargo, la solución presentada en el documento EP1960934B1 no permite proteger una variable contra un ataque denominado por « confusión de tipo », puesto en práctica por ejemplo para atacar un programa JavaCard ejecutado por una máquina virtual. Un ataque por confusión de tipo consiste en forzar una referencia de un tipo dado hacia un objeto de tipo diferente y después en acceder a un objeto a través de una referencia de tipo inválida con el fin de realizar operaciones no permitidas por la máquina virtual, por ejemplo acceder a la memoria más allá de los límites del objeto.

De esta manera, para proteger variables contra ataques por inyección de error e igualmente por confusión de tipo, se podría considerar utilizar, de acuerdo con la figura 1b:

- tres pilas diferentes: la pila de ejecución P1 para memorizar el valor V de una variable, y una pila de control P2 para memorizar un dato de control de integridad Q representativo del tipo de variable, y una pila de salvaguarda P3 para memorizar una copia del valor V;
- tres registros de índices diferentes: ind1, ind2 e ind3 que contienen datos de direccionamiento A, A2, A3 para localizar los datos contenidos respectivamente en las tres pilas P1, P2 y P3.

En este contexto, un atacante debería modificar el valor V en la pila P1, la copia de este valor en la pila P3 y el dato Q en la pila P2 de modo coherente con el fin de tener éxito en su ataque. Ahora bien, la probabilidad de poner en

marcha tales modificaciones coherentes es pequeña. La coherencia de las informaciones contenidas en las tres pilas P1, P2, P3 ofrece por tanto un alto nivel de protección de la variable.

Sin embargo, dicha solución tendría el inconveniente de ser consumidora de memoria. En particular, con un dispositivo electrónico tal como una tarjeta con chip, el número de registros de índices utilizables simultáneamente puede estar muy limitado (por ejemplo solamente 8 registros de índices) y estos registros de índices pueden resultar ya utilizados para otros fines por dicho dispositivo; la solución considerada en la figura 1b puede resultar ser entonces simplemente no posible.

Exposición de la invención

Un objetivo de la invención es proteger una variable memorizada en una pila de ejecución contra ataques por inyección de error y por confusión de tipo mediante un consumo de memoria suplementaria reducido.

Se propone entonces, según un primer aspecto de la invención, un procedimiento de protección de un dispositivo electrónico que ejecuta un programa contra ataques por inyección de error y por confusión de tipo susceptibles de afectar a una variable destinada a ser utilizada por el programa, comprendiendo el procedimiento las etapas de.

- cálculo de datos de control de integridad de la variable, dependiendo los datos de control de integridad:
    - de un tipo de variable, y
    - de un valor de la variable memorizado en una pila de ejecución y/o de un primer dato de direccionamiento memorizado en un primer registro de índices, estando adaptado el primer dato de direccionamiento para localizar el valor memorizado en la pila de ejecución,
- memorización de los datos de control de integridad de la variable en al menos una pila de control diferente de la pila de ejecución,
- memorización, en un segundo registro de índices, de un único segundo dato de direccionamiento adaptado para localizar los datos de control de integridad en la o en cada pila de control.
- verificación de integridad de la variable puesta en práctica cuando un programa controla la lectura del dato de direccionamiento de la variable y/o una lectura del valor de la variable, comprendiendo la citada verificación una comparación entre los datos de control de integridad memorizados en la o cada pila de control y nuevos datos de control de integridad calculados sobre la base:

  de un tipo de la variable visto por el programa durante el control de lectura, y/o del valor de la variable presente en la pila de ejecución y

  del dato de direccionamiento presente en el primer registro de índices durante el control de lectura.

El hecho de que los datos de control de integridad memorizados dependan no solamente del valor de la variable y/o del primer dato de direccionamiento, sino igualmente del tipo de variable permite proteger la variable contra ataques por inyección de error pero igualmente contra los ataques por confusión de tipo.

Además, el procedimiento propuesto ofrece una economía de consumo de memoria suplementaria para asegurar esta doble protección, siendo esta economía de naturaleza diferente según el número de pilas de control utilizadas:

- En el caso en que los datos de controles de integridad sean memorizados en una sola y misma pila de control, esta economía reside en la unicidad misma de esta pila de control que viene en suplemento de la pila de ejecución.
- En el caso en que los datos de controles de integridad sean memorizados en varias pilas de controles, se obtiene una economía de registros de índices. En efecto, en la medida en que uno solo dato de direccionamiento sirve para localizar los datos de control en las diferentes pilas de controles utilizadas, solo es necesario un registro de índices suplementario.

El procedimiento según este primer aspecto de la invención puede demás comprender las características siguientes, tomadas solas o en combinación cuando esto sea técnicamente posible.

Los datos de control de integridad calculados pueden depender de, o comprender:

- un primer dato de control de integridad dependiente del tipo de la variable,
- un segundo dato de control de integridad dependiente del valor de la variable y/o del primer dato de direccionamiento.

El primer dato de control de integridad puede ser calculado por aplicación de una primera función predeterminada al valor de la variable y al dato de direccionamiento.

La aplicación de la primera función predeterminada comprende el cálculo de un código de control de error en un dato dependiente del valor de la variable y/o del dato de direccionamiento de la variable.

El código de control de error es un código de redundancia longitudinal.

La aplicación de la primera función predeterminada comprende el cálculo de la separación exclusiva del valor de la variable y del dato de direccionamiento de la variable.

El segundo dato de control de integridad puede ser calculado por codificación del tipo de la variable en un número de bits predeterminado, siendo el número de bits función de un número total de tipos de variables susceptibles de ser memorizadas en la pila de ejecución.

Los datos de control de integridad de variable pueden ser o depender del resultado de la aplicación de una segunda función predeterminada al primer dato de control de integridad y al segundo dato de control de integridad.

La segunda función predeterminada puede ser una concatenación aplicada al primer dato de control de integridad y al menos a una porción del segundo dato de control de integridad.

Como se indicó anteriormente, los datos de control de integridad pueden ser memorizados en una sola y misma pila de control.

Alternativamente,

- el primer dato de control de integridad es memorizado en una primera pila de control,
- el segundo dato de control de integridad es memorizado en una segunda pila de control,
- el único dato de direccionamiento está adaptado para localizar por una parte el primer dato de control de integridad en la primera pila de control y por otra para localizar el segundo dato de control de integridad en la segunda pila de control.

Puede estar previsto además que:

- el primer dato de control de integridad se memorice en una primera dirección desplazada una desviación con respecto a una dirección de referencia de la primera pila de control, siendo contada la desviación en número de palabras direccionables en la primera pila de control,
- el segundo dato de control de integridad se memorice en una segunda dirección desplazada dicha desviación con respecto a una dirección de referencia de la segunda pila de control,
- el único dato de direccionamiento comprende, es, o depende de la citada desviación.

El segundo dato de control de integridad puede ser o puede comprender una copia del valor de la variable.

El primer dato de control de integridad puede tener un tamaño inferior o igual al tamaño de una palabra direccionable en la segunda pila de control.

Una palabra direccionable en la primera pila de control puede tener un tamaño inferior a una palabra direccionable en la segunda pila de control.

Se propone igualmente, según un aspecto de la invención, un producto programa de ordenador que comprende instrucciones de código de programa para la ejecución de las etapas del procedimiento según el primer aspecto de la invención, cuando este procedimiento es ejecutado por al menos un procesador.

Se propone igualmente, según un tercer aspecto de la invención, un dispositivo electrónico, tal como una tarjeta con chip, que comprende:

- al menos un procesador configurado para ejecutar un programa,
- al menos una memoria adaptada para contener una pila de ejecución y al menos una pila de control,

en el cual el procesador está configurado para:

- calcular datos de control de integridad de una variable memorizada en la pila de ejecución y destinada a ser utilizada por el programa, dependiendo los datos de control de integridad:
    - del tipo de la variable, y

- del valor de la variable y/o de un primer dato de direccionamiento memorizado en un primer registro de índices, estando adaptado el primer dato de direccionamiento para localizar la variable en la primera pila de ejecución,

- controlar la memorización, en la o cada pila de control, de los datos de control de integridad de la variable,
- controlar la memorización, en un segundo registro de índices, de un único segundo dato de direccionamiento adaptado para localizar los datos de control de integridad en la o cada pila de control,
- controlar la verificación de integridad de la variable puesta en práctica cuando un programa controla una lectura del dato de direccionamiento de la variable y/o una lectura del valor de la variable, comprendiendo la citada verificación una comparación entre los datos de control de integridad memorizados en la o cada pila de control y nuevos datos de control de integridad calculados sobre la base:

  de un tipo de la variable visto por el programa durante el control de lectura, y

  del valor de la variable presente en la pila de ejecución y/o del dato de direccionamiento presente en el primer registro de índices durante el control de lectura.

Descripción de las figuras

Otras características, objetivos y ventajas de la invención se pondrán de manifiesto en la descripción que sigue, la cual es puramente ilustrativa y no limitativa, y que debe ser leída en relación con los dibujos anejos, en los cuales:

- La figura 1a, ya analizada, representa esquemáticamente el contenido de una memoria en el transcurso de la puesta en práctica de un procedimiento conocido del estado de la técnica, que protege una variable contra ataques por inyección de error.
- La figura 1b, ya analizada, representa esquemáticamente el contenido de una memoria en el transcurso de la puesta en práctica de un procedimiento susceptible de proteger una variable contra ataques por inyección de error y por confusión de tipo,
- La figura 2 representa esquemáticamente un dispositivo electrónico según un modo de realización de la invención,
- La figura 3 representa esquemáticamente el contenido de una memoria en el transcurso de la puesta en práctica de un procedimiento de protección de una variable contra ataques por inyección de error y por confusión de tipo, según un primer modo de realización de la invención.
- La figura 4 es un organigrama de etapas del procedimiento según el primer modo de realización de la invención,
- La figura 5 representa esquemáticamente el contenido de una memoria en el transcurso de la puesta en práctica de un procedimiento de protección de una variable contra ataques por inyección de error y por confusión de tipo, según un segundo modo de realización de la invención,
- La figura 6 es un organigrama de etapas del procedimiento según el primer modo de realización de la invención.

En el conjunto de las figuras, los elementos similares llevan referencias idénticas.

Descripción detallada de la invención

Dispositivo electrónico de protección contra ataques por inyección de error y por confusión de tipo

En referencia a la figura 2, un dispositivo electrónico 1 comprende al menos un procesador 2 y al menos una memoria 4.

La memoria 4 comprende al menos una memoria volátil 6, por ejemplo de tipo RAM. La memoria volátil 6 tiene la función de memorizar temporalmente datos, por ejemplo datos calculados por el procesador 2. El contenido de la memoria volátil 6 queda borrado con un apagado del dispositivo electrónico 1.

La memoria 4 comprende además una memoria no volátil 8, por ejemplo de tipo de disco duro, SSD, flash, EEPROM, etc. La memoria no volátil 8 tiene la función de memorizar datos de manera persistente, en el sentido de que un apagado del dispositivo electrónico 1 no borra el contenido de la memoria no volátil.

El procesador 2 está adaptado para ejecutar instrucciones de código de programa que pertenecen a un juego de instrucciones predeterminado.

Por extensión, el procesador 2 está adaptado para ejecutar un programa que se presente en forma de un binario compilado que comprenda instrucciones de códigos que pertenezcan a este juego de instrucciones predeterminado. Se tomará a continuación el ejemplo de un programa de tipo máquina virtual.

La máquina virtual está configurada para interpretar un programa objetivo, presentándose el programa objetivo en forma de un binario que comprende instrucciones de código en un formato diferente del juego de instrucciones antes citado, cuando la máquina virtual es ejecutada por el procesador 2.

Por ejemplo, una máquina virtual JavaCard está configurada para interpretar un « bytecode » resultante de un código fuente en el lenguaje de programación JavaCard, que es un lenguaje de programación orientado a objetos.

Se supone en lo que sigue que la máquina virtual está memorizada en la memoria no volátil 8, lo mismo que un programa objetivo interpretable por la máquina virtual.

El dispositivo electrónico 1 es por ejemplo una tarjeta con chip, tal como una tarjeta SIM.

Procedimiento de protección contra ataques por inyección de error y por confusión de tipo (primer modo de realización)

En referencia a las figuras 3 y 4, un procedimiento de protección según un primer modo de realización comprende las etapas siguientes.

El procesador 2 inicia la ejecución de la máquina virtual.

La máquina virtual asigna en la memoria virtual 6 una pila de ejecución P1 asociada a un proceso del programa objetivo que haya que ejecutar. Esta pila de ejecución P1 está destinada a contener variables de las cuales tiene necesidad el proceso del programa objetivo en el transcurso de su ejecución por la máquina virtual.

La pila de ejecución P1 está caracterizada por: una dirección de pila AP1, un tamaño, y un tamaño m de palabra direccionable en la pila P1.

En el presente texto, se considera que una palabra direccionable (« word adressable ») en una pila es una unidad más pequeña que dispone de una dirección de memoria propia en la pila. Por consiguiente, el tamaño de una palabra direccionable es igual a la desviación entre dos direcciones consecutivas en la pila.

La dirección de pila AP1 es una dirección de inicio de la pila P1, como está representado en la figura 3, o bien una dirección de final de la pila P1.

El tamaño de la pila P1 es igual al número de palabras direccionables memorizables en la pila de ejecución P1 multiplicado por el tamaño de una de estas palabras direccionables. Por ejemplo, si el tamaño m de una palabra direccionable en la pila P1 es de 32 bits y el número de palabras direccionables simultáneamente memorizables en la pila P1 es 1024, el tamaño de la pila de ejecución P1 es de 32 x 1024 = 32768 bits.

La máquina virtual asigna además en la memoria virtual 6 un primer registro de índices ind1 asociado a la pila de ejecución P1. El primer registro de índices ind1 es distinto de la pila de ejecución P1.

La máquina virtual asigna además en la memoria virtual 6 una pila de control P2, distinta de la pila de ejecución P1 y del primer registro de índices ind1. Como se verá en lo que sigue, esta pila de control P2 sirve para detectar la presencia de ataques por inyección de error y la presencia de ataques por confusión de tipo.

La máquina virtual asigna además en la memoria virtual 6 un segundo registro de índices ind2 asociado a la pila de control P2. Este segundo registro de índices ind2 es distinto de la pila de ejecución P1, de la pila de control P2 y del primer registro de índices ind1.

Como la pila de ejecución P1, la pila de control P2 está caracterizada por: una dirección de pila AP2 (por ejemplo una dirección de inicio o de final de pila), un tamaño, y un tamaño n de palabra direccionable en la pila de control P2.

El tamaño de palabra direccionable n en la pila de control P2 es preferentemente inferior al tamaño de palabra direccionable m en la pila de ejecución P1, con el fin de limitar el consumo de memoria suplementaria inducido por la asignación de la pila de control P2, además de la pila de ejecución P1.

En el transcurso de la ejecución del proceso del programa objetivo por la máquina virtual, la máquina virtual controla la memorización de una variable Z en la memoria volátil 6.

La variable Z tiene un valor V, un dato de direccionamiento A y un tipo T.

La máquina virtual memoriza el dato de direccionamiento A en el primer registro de índices ind1, estando adaptado el dato de direccionamiento A para localizar el valor V de la variable Z en la pila de ejecución P1 (etapa 102).

La máquina virtual memoriza el valor V de la variable Z en la pila de ejecución P1, en el emplazamiento de memoria indicado por el dato de direccionamiento A (etapa 104).

Por ejemplo, el dato de direccionamiento A es una desviación (o dirección relativa u « offset » en inglés) entre la dirección de la pila de ejecución P1 y la dirección de la variable Z, siendo contada esta desviación en número de palabras direccionables en la pila de ejecución P1. Así, basta añadir (o suprimir) a la dirección de la pila de ejecución P1 esta desviación multiplicada por el tamaño m de palabra direccionable en la pila de ejecución P1 para obtener la dirección de la variable Z, y así acceder al valor V de esta variable Z.

Alternativamente, el dato de direccionamiento A es directamente la dirección del valor V en la pila de ejecución P1.

De modo en sí conocido, el tipo de una variable define los valores que puede tomar la variable así como los operadores que se la pueden aplicar cuando la misma es utilizada por la máquina virtual. A modo de ejemplo, el tipo « char » está codificado en 8 bits y por tanto una variable de este tipo puede tomar 256 valores diferentes. El número total de tipos de variables susceptibles de ser memorizadas en la pila de ejecución P1 por la máquina virtual está predeterminado.

La máquina virtual determina el tipo T de la variable Z de la cual el valor A y la zona de direccionamiento están memorizados. Esta determinación es puesta en práctica por la máquina virtual por interpretación del « bytecode »,

La máquina virtual calcula datos de control de integridad de la variable Z (etapa 106).

Estos datos de control de integridad son memorizados en la pila de control P2 (etapa 108).

La máquina virtual memoriza por otra parte un segundo dato de direccionamiento A2 en el segundo registro de índices ind2, estando adaptado el segundo dato de direccionamiento A2 para localizar los datos de control de integridad en la pila de control P2 (etapa 110).

El segundo dato de direccionamiento A2 es por ejemplo del mismo formato que el dato de direccionamiento A.

En variante, el segundo dato de direccionamiento A2 puede resultar de la aplicación de una función predeterminada h al dato de direccionamiento A, de acuerdo con la fórmula siguiente:

$$A2 = h(A)$$

La función h puede ser la función identidad o bien otra función, por ejemplo una permutación.

Los datos de control de integridad dependen:

- por una parte del tipo T de la variable, y
- por otra del valor V de la variable Z y/o del dato de direccionamiento A de la variable Z adaptado para localizar el valor V en la pila de ejecución P1.

En lo que sigue, se describe un modo de realización particular en el cual los datos de control de integridad dependen del tipo T de la variable Z, del valor V de la variable Z, y también del dato de direccionamiento A, adaptado para localizar el valor V de la variable Z en la pila de ejecución P1.

El cálculo 106 de los datos de control de integridad comprende las etapas siguientes.

La máquina virtual calcula un dato X, representativo del tipo de variable Z.

El dato X es una codificación del tipo de la variable Z en un número de bits predeterminado n1, siendo este número de bits función del número total k de tipos de variables susceptibles de ser memorizadas en la pila de ejecución P1.

Preferentemente, se elige n1 como sigue:

$$n1 = |\log_2 k|$$

donde $|x|$ desgina la parte entera por exceso del número real x (denominado igualmente « techo » de x o « ceiling » en inglés). Dicho de otro modo, se tiene:

$$|x| - 1 < x \leq |x|$$

Por ejemplo, si el número total de tipos vale 4, entonces el dato X es codificado en 2 bits: los valores binarios 00, 01, 10, 11 son los códigos respectivos para los cuatro tipos.

La máquina virtual calcula además un dato de control de integridad Y por aplicación de una función predeterminada f al valor V de la variable Z y al dato de direccionamiento A de la variable Z. Se tiene por tanto:

$$Y = f(V,A)$$

Los datos de control de integridad memorizados en la pila de control P2 son o dependen de los datos X e Y.

Por ejemplo, la aplicación de la función f comprende el cálculo de un código de control de error en un dato intermedio dependiente del valor V de la variable Z y/o del dato de direccionamiento A de la variable Z, tal como un código de redundancia longitudinal (« LRC » en inglés). El dato intermedio puede ser la disyunción exclusiva del valor V y de la dirección A (operador « XOR »).

De esta manera, en un modo de realización, se tiene:

$$Y = LRC(V \text{ XOR } A)$$

Los datos de control de integridad pueden estar formados por un dato resultante de la aplicación de una función g predeterminada a los datos X e Y. En otras palabras, los datos de control de integridad están formados por el dato:

$$g(X,Y)$$

En un modo de realización particular, g es una concatenación aplicada al dato X (cuyo número de bits es n1) y al menos a una porción del dato Y, siendo la citada porción de longitud n2 en número de bits. Se tiene:

$$g(X,Y) = X||Y$$

donde || designa el operador de concatenación.

Por ejemplo, la porción de Y está formada por los n2 bits de mayor peso del dato Y.

Preferentemente, se eligen n1 y n2 tal que n = n1 + n2. Si n1 se determina en función del número total de tipos de variables posibles, y si no es posible elegir n, se ajusta n2 de modo que se verifique esta igualdad. De esta manera, los datos de control de integridad ocupan un espacio de tamaño mínimo en la pila de control P2, al tiempo que son portadores de una cantidad significativa de información.

Las etapas que preceden son puestas en práctica cada vez que la máquina virtual actualiza el valor de la variable Z en la pila de ejecución P1.

Las etapas que preceden son puestas en práctica además cada vez que debe ser memorizada una nueva variable en la memoria virtual 6 (por tanto especialmente cuando un nuevo valor de variable es memorizado en un emplazamiento libre de la pila de ejecución P1).

Posteriormente, la máquina virtual interpreta una instrucción de código del programa objetivo que controla una lectura del dato de direccionamiento A de la variable Z y/o una lectura del valor V de la variable Z.

La máquina virtual pone en práctica una verificación de integridad de la variable a partir de los datos de control de integridad X, y memorizados en la pila de control P2 (etapa 112).

La verificación de integridad comprende subetapas de:

- lectura del dato de direccionamiento de la variable Z presente en el primer registro de índices ind1 durante el control en lectura,
- localización del valor de la variable Z presente en la pila de ejecución P1 sobre la base del dato de direccionamiento leído,
- lectura, en la pila de ejecución P1, del valor de la variable Z así localizado,
- determinación, a partir de la citada instrucción de código del programa, del tipo de la variable Z visto por el programa.

La verificación de integridad comprende además un cálculo de nuevos datos de control de integridad sobre la base de los datos leídos, por medio de las mismas funciones que las puestas en práctica para calcular los datos de control de integridad X, Y previamente memorizados en la pila de control P2.

La verificación de integridad comprende además etapas de:

- lectura, en el segundo registro de índices ind2, del dato de direccionamiento relativo a los datos de control de integridad de la variable, y
- localización del dato de integridad presente en la pila de control P2 sobre la base del dato de direccionamiento leído A2.

La verificación de integridad comprende además una comparación entre los nuevos datos de control de integridad calculados y los datos de control de integridad X, Y leídos desde la pila de control P2.

En un caso de funcionamiento normal, los valores A, V, T no han cambiado desde la memorización 102, 104 de la variable. Por consiguiente, los nuevos datos de control de integridad y los datos de control de integridad leídos desde la pila de control P2 son iguales. En este caso, la máquina virtual continúa ejecutando el programa objetivo.

Se consideran ahora los diferentes casos de ataque del dispositivo electrónico.

En el caso en que se haya puesto en práctica un ataque por inyección de error sobre el valor V de la variable Z, el valor V memorizado en la pila de ejecución P1 se ha modificado en un valor V'.

En el caso de ataque en que se haya puesto en práctica una inyección de error en la dirección A de la variable, el valor A memorizado en el primer registro de índices ind1 se ha modificado en un valor A'.

En un tercer caso de que haya sido puesto en práctica un ataque por confusión de tipo de la variable Z, la máquina virtual cree por error que el tipo de variable memorizado no es T sino otro tipo T'.

En uno cualquiera de estos tres casos (o una combinación de los mismos), los nuevos datos de control de integridad calculados en el transcurso de la etapa de verificación de integridad son diferentes de los datos de control de integridad leídos desde la pila de control P2. La máquina virtual puede entonces generar una señal de error, detener la ejecución del proceso del programa concernido por la variable atacada, detener todos los procesos del programa, detenerse completamente, o iniciar un apagado o un nuevo arranque del dispositivo electrónico.

El procedimiento propuesto permite así proteger el dispositivo electrónico 1 contra ataques por inyección de error y por confusión de tipo, sin por ello requerir una cantidad importante de memoria suplementaria. En efecto, se utiliza una única pila de control P2 para memorizar los datos de control de integridad X, Y que permiten detectar tanto un ataque por inyección de error como un ataque por confusión de tipo.

Procedimiento de protección contra ataques por inyección de error y por confusión de tipo (segundo modo de realización)

En referencia a la figura 5, un procedimiento según un segundo modo de realización comprende las etapas siguientes para proteger una variable Z contra ataques por inyección de error y por confusión de tipo.

La variable Z tiene un valor memorizado en una pila de ejecución P1 idéntico al del primer modo de realización, y un dato de direccionamiento A memorizado en un primer registro de índices ind1 idéntico al del primer modo de realización (etapas 202, 204).

La máquina virtual asigna en la memoria virtual 6 una pila de control P2 similar a la del primer modo de realización, y una pila de control P3 diferente de las pilas de ejecución P1 y de control P2.

La máquina virtual asigna además en la memoria virtual 6 un registro de índices ind2.

La pila de control P3 es del mismo tamaño que la pila de ejecución P1; el tamaño m de palabra direccionable en la pila P3 es igual al tamaño de palabra direccionable en la pila de ejecución P1.

La máquina virtual memoriza en la pila de control P3 una copia del valor V de la variable (etapa 206).

En variante, la máquina virtual memoriza en la pila de control P3 un dato de control de integridad dependiente del valor V de la variable Z y/o del dato de direccionamiento A de la variable Z adaptado para localizar el valor V en la pila de ejecución P1 (por ejemplo el dato Y anteriormente descrito en el marco del primer modo de realización).

Por otra parte, la máquina virtual calcula y memoriza en la pila de control P2 un dato de control de integridad del tipo T de la variable Z (por ejemplo el dato X anteriormente descrito en el marco del primer modo de realización) (etapas 208, 210).

De esta manera, a diferencia del primer modo de realización, que solamente utiliza una sola y misma pila de control para memorizar los datos de control de integridad, los datos de control están repartidos en varias pilas de controles diferentes P2 y P3.

La máquina virtual memoriza por otra parte en el registro de índices ind2 un único dato de direccionamiento A2 adaptado para localizar por una parte la copia del valor V en la pila de control P3 y por otra para localizar en la pila de control P2 el dato de control de integridad del tipo T de la variable Z (etapa 212).

En otras palabras, las dos pilas suplementarias P2 y P3 asignadas comparten el mismo registro de índices ind2, lo que permite economizar la cantidad de memoria consumida para proteger la variable Z.

Posteriormente, la máquina virtual interpreta una instrucción de código del programa objetivo que controla una lectura del dato de direccionamiento A de la variable Z y/o una lectura del valor V de la variable Z.

La integridad de la variable Z es verificada (etapa 214) por la máquina virtual.

Para verificar la integridad de la variable Z, se ponen en práctica sensiblemente las mismas etapas que las descritas en el marco del primer modo de realización.

La verificación de integridad comprende así etapas de:

- lectura, en el primer registro de índices ind1, del dato de direccionamiento A de la variable Z,
- localización del valor V de la variable presente en la pila de ejecución P1 sobre la base del dato de direccionamiento A leído,
- lectura, en la pila de ejecución P1, del valor V de la variable localizada,
- determinación del tipo T de la variable visto por el programa.

La verificación de integridad comprende después un cálculo de un nuevo dato de control de integridad del tipo, por medio de las mismas funciones que las puestas en práctica para calcular el dato de control de integridad del tipo de la variable Z, dato que ha sido previamente memorizado en la pila de control P2.

Si un dato de integridad que se presenta en otra forma que una copia del valor V ha sido memorizado en la pila de control P3 (por ejemplo el dato Y), la verificación de integridad comprende además un cálculo de un nuevo dato de control de integridad sobre la base de los datos leídos, por medio de la misma función que la puesta en práctica para calcular el dato de control de integridad previamente memorizado en la pila de control P3.

La verificación de integridad comprende además etapas de:

- lectura, en el segundo registro de índices ind2, del dato de direccionamiento A2 del dato de integridad asociado a la variable, y,
- localización del dato de integridad X presente en la pila de control P2 sobre la base del dato de direccionamiento A2 leído
- localización del dato de integridad presente en la pila de control P3 sobre la base del mismo dato de direccionamiento A2 leído (copia del valor de la variable Z, dato Y u otro).

La verificación de integridad comprende además:

- una comparación entre el nuevo dato de control de integridad calculado y el dato de control de integridad X leído desde la pila de control P3, y
- una comparación entre el valor de la variable Z presente en la pila de ejecución P1 y el valor copiado leído desde la pila de control P3, o una comparación entre el nuevo dato de control de integridad calculado Y y el dato de control Y leídos desde la pila de control P3, según la naturaleza del dato memorizado en la pila P3 (copia de V o dato Y).

En caso de doble igualdad, la máquina virtual continúa ejecutando el programa objetivo.

Si no, se considera que la variable Z ha sufrido un ataque (por inyección de error o confusión de tipo). En este caso, la máquina virtual puede poner en práctica entonces los mismos tratamientos que los descritos en el marco del primer modo de realización (generar una señal de error, detener la ejecución del proceso del programa concernido por la variable atacada, etc.).

Se puede observar que el hecho de utilizar una copia del valor V como dato de integridad permite acelerar la etapa de verificación de integridad. En efecto, no es necesario proceder a un cálculo de datos de integridad para verificar que el valor V no ha cambiado: basta comparar los valores presentes en las pilas P1 y P3.

El dato de direccionamiento A2 memorizado en el registro de índices ind2 es por ejemplo un número de palabras direccionables en una o la otra de las pilas P2 y P3, siendo el citado número de palabras direccionables igual a:

- una desviación (o dirección relativa u « ofsett » en inglés) entre una dirección de la copia en la primera pila de control y una dirección AP2 de la pila de control P2,
- una desviación entre una dirección de los datos de control de integridad en la primera pila de control y una dirección AP3 de la pila de control P3.

Las direcciones AP2, AP3 pueden ser direcciones de inicio de pila (como está representado en la figura 5) o direcciones de final de pila.

Como se indicó anteriormente, basta añadir (o disminuir) a la dirección AP2 de la pila P2 esta desviación multiplicada por el tamaño n de palabra direccionable en la pila P2 para obtener la dirección del dato X en la pila de control P2.

Similarmente, basta añadir (o disminuir) a la dirección AP3 de la pila de control P3 esta misma desviación multiplicada por el tamaño m de palabra direccionable en la pila de control P3 para obtener la dirección de la copia del valor de la variable Z memorizada en la pila de control P3.

En variante, el dato de direccionamiento A2 puede resultar de la aplicación de una función predeterminada h al dato de direccionamiento A (por ejemplo cuando este dato de direccionamiento es un « offset ») como se ha representado anteriormente para el primer modo de realización.

El dato de integridad del tipo es por ejemplo el dato X descrito en el marco del primer modo de realización, de longitud n1. Por ejemplo, se elige n1 = n.

Preferentemente, una palabra direccionable en la pila de control P2 tiene un tamaño n inferior al tamaño m de una palabra direccionable en la pila de control P3. Esto permite reducir más la memoria suplementaria consumida para proteger la variable. En efecto, el tamaño de una variable Z puede variar en función de su tipo (es decir que la longitud en número de bits del valor V varía en función del tipo T de la variable Z) e incluso relativamente mucho con respecto a la longitud n1.

REIVINDICACIONES

1. Procedimiento de protección de un dispositivo electrónico (1) que ejecuta un programa contra ataques por inyección de error y por confusión de tipo susceptibles de afectar a una variable (Z) destinada a ser utilizada por el programa, estando caracterizado el procedimiento por que el mismo comprende las etapas de:

- cálculo de datos de control de integridad (X, Y) de la variable (Z), dependiendo los datos de control de integridad:
    - de un tipo (T) de la variable (Z), y
    - de un valor (V) de la variable (Z) memorizado en una pila de ejecución (P1) y/o de un primer dato de direccionamiento (A) memorizado en un primer registro de índices (ind1), estando adaptado el primer dato de direccionamiento (A) para localizar el valor (V) memorizado en la pila de ejecución (P1),
- memorización de los datos de control (X, Y) de integridad de la variable (Z) en al menos una pila de control (P2, P3) diferente de la pila de ejecución (P1),
- memorización, en un segundo registro de índices (ind2), de un único segundo dato de direccionamiento (A2) adaptado para localizar los datos de control de integridad (X, Y) en la o en cada pila de control (P2, P3)
- verificación de integridad de la variable (Z) puesta en práctica cuando un programa controla la lectura del dato de direccionamiento (A) de la variable (Z) y/o una lectura del valor (V) de la variable (Z), comprendiendo la citada verificación una comparación entre los datos de control de integridad memorizados en la o cada pila de control (P2, P3) y nuevos datos de control de integridad calculados sobre la base:
    - de un tipo (T, T') de la variable (Z) visto por el programa durante el control de la lectura, y
    - del valor (V, V') de la variable (Z) presente en la pila de ejecución (P1) y/o del dato de direccionamiento (A, A') presente en el primer registro de índices (ind1) durante el control de lectura.

2. Procedimiento según la reivindicación precedente, en el cual los datos de control de integridad calculados dependen de o comprenden:

- un primer dato de control de integridad (X) dependiente del tipo de la variable,
- un segundo dato de control de integridad (V, Y) dependiente del valor de la variable y/o del primer dato de direccionamiento (A).

3. Procedimiento según la reivindicación precedente, en el cual el primer dato de control de integridad (Y) es calculado por aplicación de una primera función predeterminada al valor (V) de la variable (Z) y al dato de direccionamiento (A).

4. Procedimiento según la reivindicación precedente, en el cual la aplicación de la primera función predeterminada comprende el cálculo de un código de control de error en un dato dependiente del valor (V) de la variable (Z) y/o del dato de direccionamiento (A) de la variable (Z).

5. Procedimiento según la reivindicación precedente, en el cual el código de control de error es un código de redundancia longitudinal.

6. Procedimiento según una de las reivindicaciones 3 a 5, en el cual la aplicación de la primera función predeterminada comprende el cálculo de la disyunción exclusiva del valor (V) de la variable (Z) y del dato de direccionamiento (A) de la variable (Z).

7. Procedimiento según una de las reivindicaciones 2 a 6, en el cual el segundo dato de control de integridad (X) es calculado por codificación del tipo de variable (Z) en un número de bits predeterminado, siendo el número de bits función de un número total de tipos de variables susceptible de ser memorizadas en la pila de ejecución (P1).

8. Procedimiento según una la reivindicaciones 2 a 7, en el cual los datos de control de integridad de variable (Z) son o dependen del resultado de la aplicación de una segunda función predeterminada al primer dato de control de integridad y al segundo dato de control de integridad.

9. Procedimiento según la reivindicación precedente, en el cual la segunda función predeterminada es una concatenación aplicada al primer dato de control de integridad y al menos a una porción del segundo dato de control de integridad.

10. Procedimiento según una de las reivindicaciones precedentes, en el cual los datos de control de integridad (X, Y) son memorizados en una sola y misma pila de control (P2).

11. Procedimiento según una de las reivindicaciones 2 a 9, en el cual

- el primer dato de control de integridad (X) es memorizado en una primera pila de control (P2),
- el segundo dato de control de integridad (V, Y) es memorizado en una secunda pila de control (P3) diferente de la primera pila de control (P2),
- el único dato de direccionamiento (A2) está adaptado para localizar por una parte el primer dato de control de integridad en la primera pila de control (P2) y por otra para localizar el segundo dato de control de integridad en la segunda pila de control (P3).

12. Procedimiento según la reivindicación precedente, en el cual:

- el primer dato de control de integridad (X) es memorizado en una primera dirección desplazada una desviación con respecto a una dirección de referencia de la primera pila de control, contándose esta desviación en número de palabras direccionables en la primera pila de control,
- el segundo dato de control de integridad (V, Y) es memorizado en una secunda dirección desplazada la citada desviación con respecto a una dirección de referencia de la segunda pila de control,
- el único segundo dato de direccionamiento (A2) comprende, es, o depende de la citada desviación.

13. Procedimiento según una de las reivindicaciones 11 a 12, en el cual el segundo dato de control de integridad es o comprende una copia del valor (V) de la variable (Z).

14. Procedimiento según una de las reivindicaciones 11 a 13, en el cual el primer dato de control de integridad (X) tiene un tamaño inferior o igual al tamaño de una palabra direccionable en la primera pila de control (P2).

15. Procedimiento según una de las reivindicaciones 11 a 14, en el cual una palabra direccionable en la pila de control (P2) tiene un tamaño inferior a una palabra direccionable en la segunda pila de control (P3).

16. Producto programa de ordenador que comprende instrucciones de código de programa para la ejecución de las etapas del procedimiento según una de las reivindicaciones precedentes, cuando este procedimiento es ejecutado por al menos un procesador (2).

17. Dispositivo electrónico, tal como una tarjeta con chip, que comprende:

- al menos un procesador (2) configurado para ejecutar un programa,
- al menos una memoria (4) adaptada para contener una pila de ejecución (P1) y al menos una pila de control (P2),

estando caracterizado el dispositivo por que el procesador (2) está configurado para:

- calcular datos de control de integridad (X, Y) de una variable (Z) memorizada en la pila de ejecución (P1) y destinada a ser utilizada por el programa, dependiendo los datos de control de integridad:
    - del tipo de la variable (Z), y
    - del valor (V) de la variable (Z) y/o de un primer dato de direccionamiento (A) memorizado en un primer registro de índices (ind1), estando adaptado el primer dato de direccionamiento (A) para localizar la variable (Z) en la pila de ejecución (P1),
- controlar la memorización, en la o cada pila de control (P2), de los datos de control de integridad (X, Y) de la variable (Z),
- controlar la memorización, en un segundo registro de índices (ind2), de un único segundo dato de direccionamiento (A2) adaptado para localizar los datos de control de integridad (X, Y) en la o cada pila de control (P2, P3),
- controlar la verificación de integridad de la variable (Z) puesta en práctica cuando un programa controla una lectura del dato de direccionamiento (A) de la variable (Z) y/o una lectura del valor (V) de la variable (Z), comprendiendo la citada verificación una comparación entre los datos de control de integridad memorizados en la o cada pila de control (P2, P3) y nuevos datos de control de integridad calculados sobre la base:
    - de un tipo (T, T') de la variable (Z) visto por el programa durante el control de lectura, y
    - del valor (V, V') de la variable (Z) presente en la pila de ejecución (P1) y/o del dato de direccionamiento (A, A') presente en el primer registro de índices (ind1) durante el control de lectura.

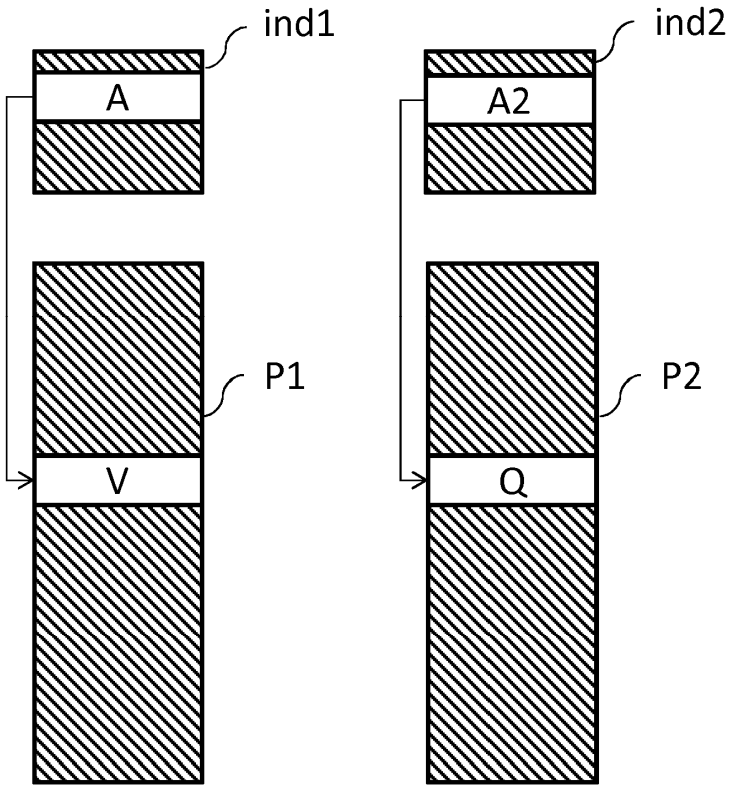

FIG. 1a

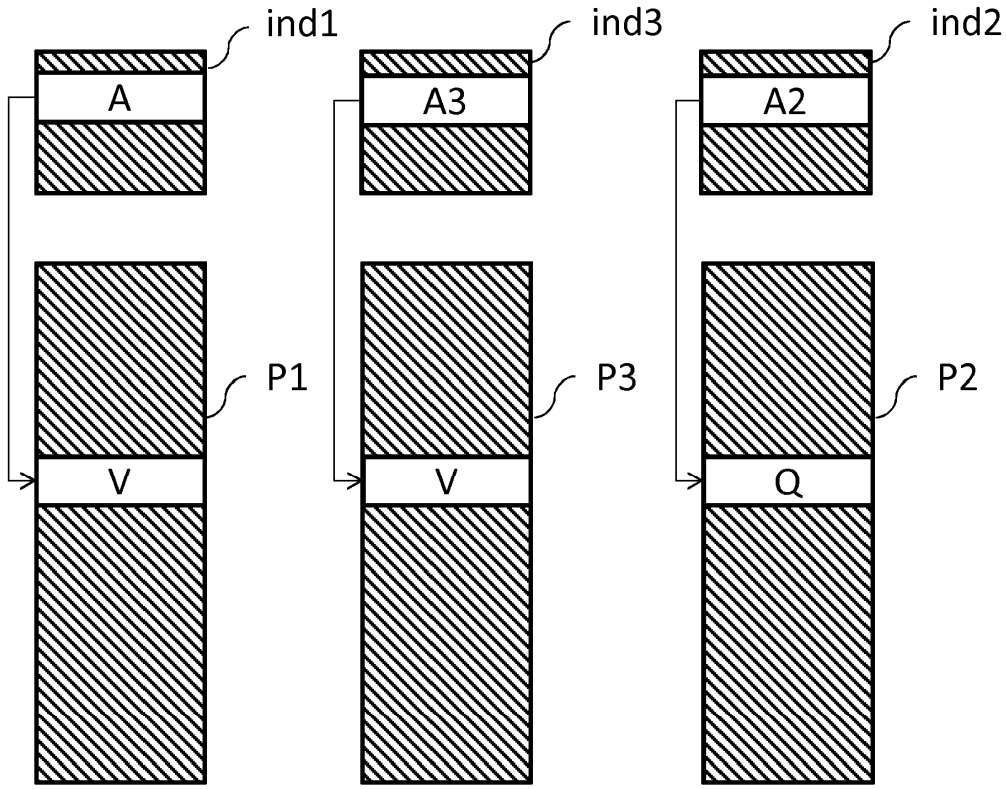

FIG. 1b

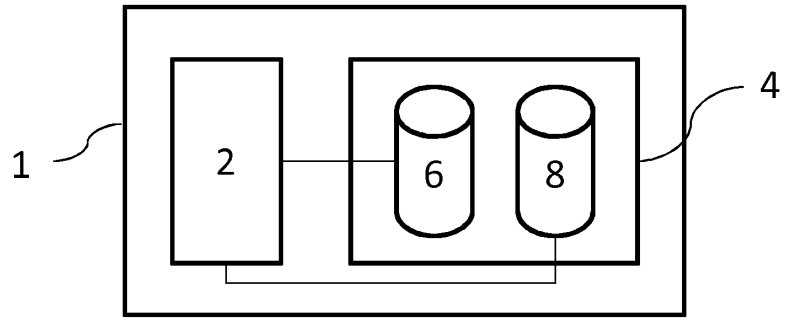

FIG. 2

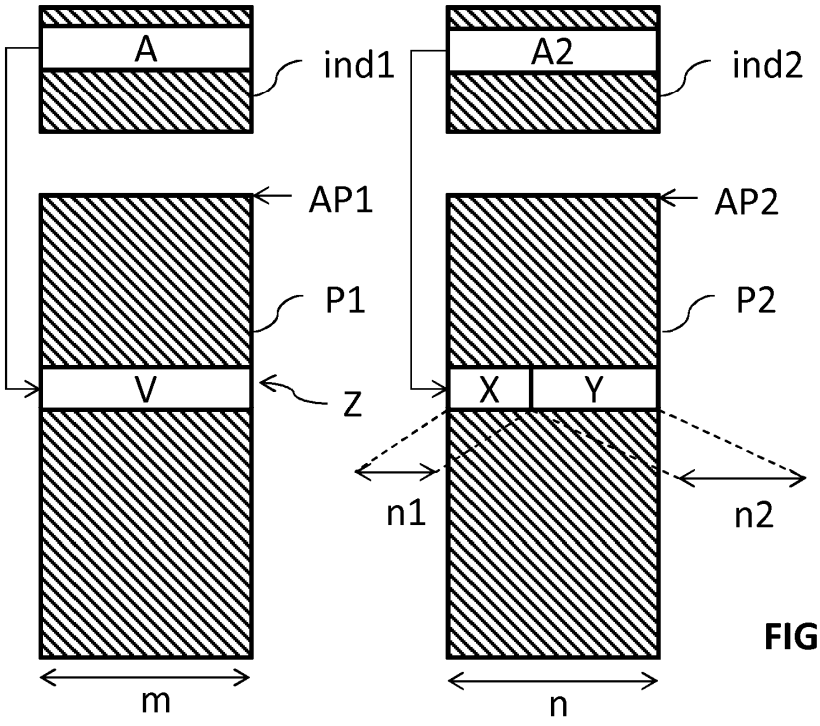

FIG. 3

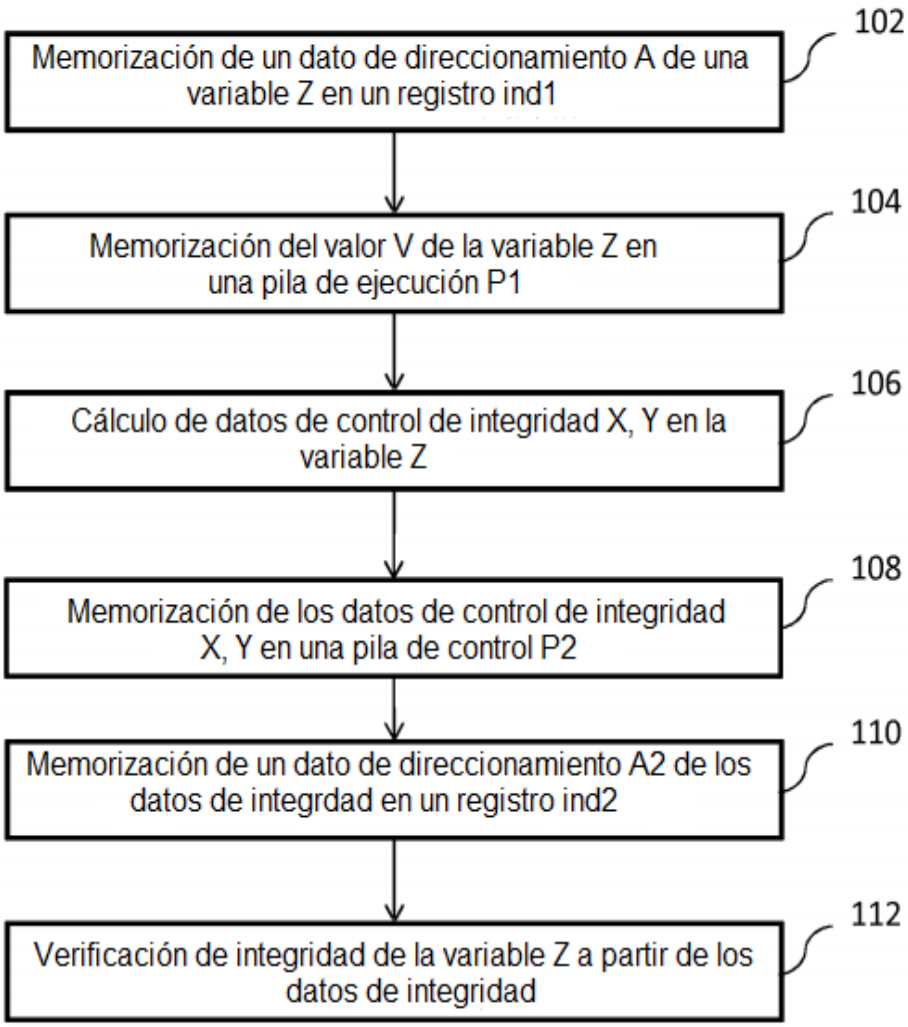
Memorización de un dato de direccionamiento A de una variable Z en un registro ind1
102
Memorización del valor V de la variable Z en una pila de ejecución P1
104
Cálculo de datos de control de integridad X, Y en la variable Z
106
Memorización de los datos de control de integridad X, Y en una pila de control P2
108
Memorización de un dato de direccionamiento A2 de los datos de integrdad en un registro ind2
110
Verificación de integridad de la variable Z a partir de los datos de integridad
112

FIG. 4

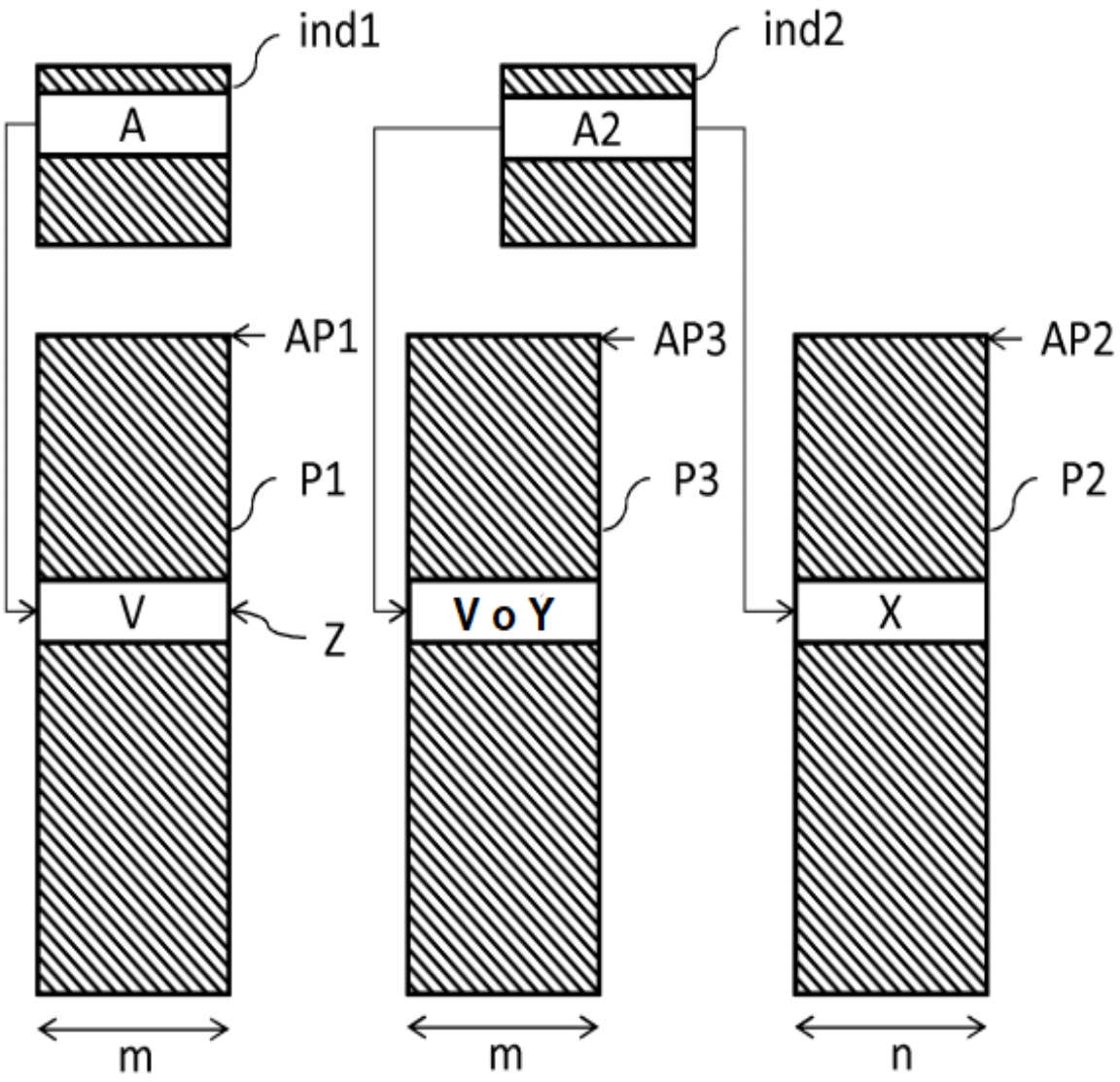

FIG. 5

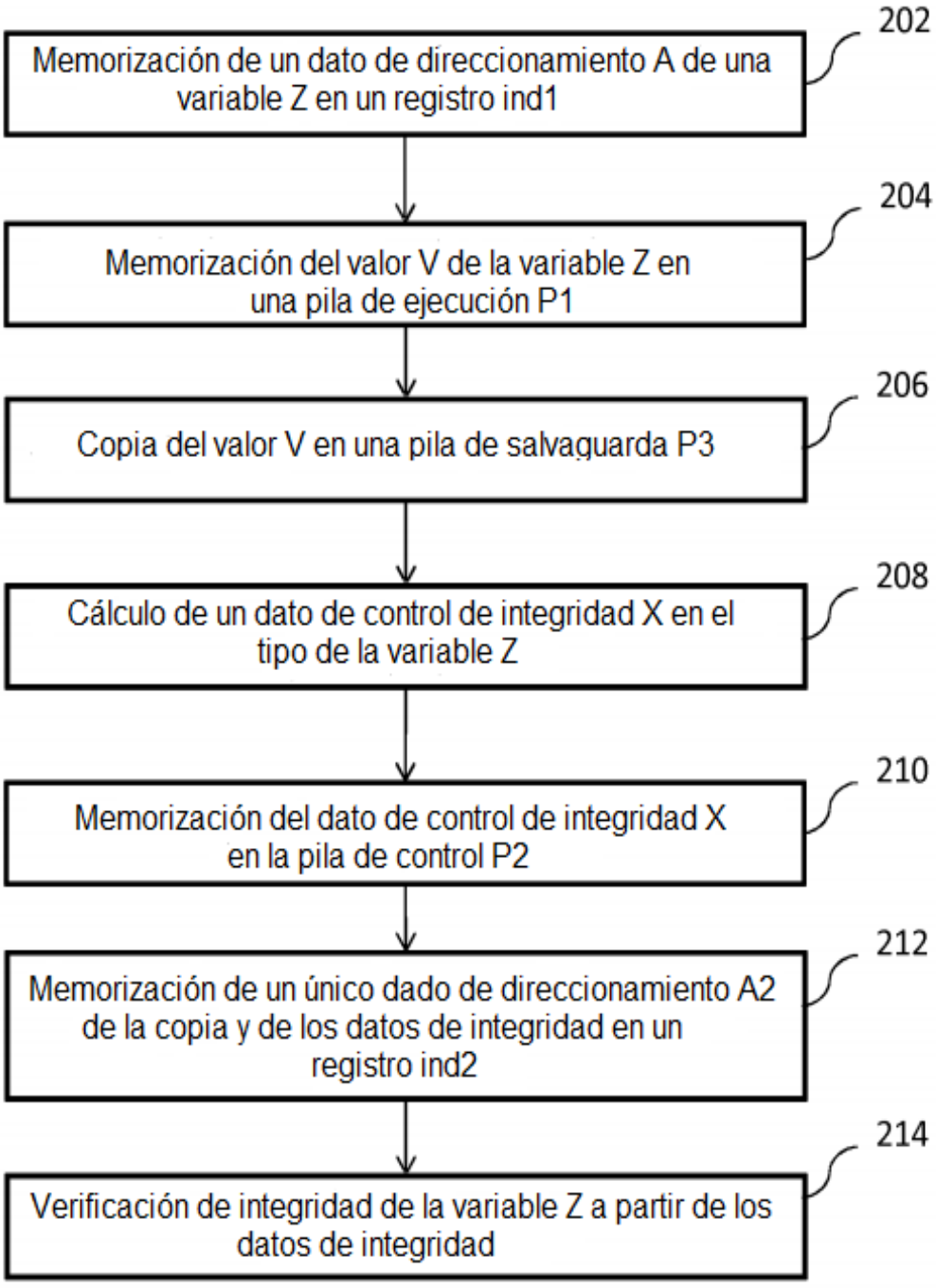
Memorización de un dato de direccionamiento A de una variable Z en un registro ind1
202
Memorización del valor V de la variable Z en una pila de ejecución P1
204
Copia del valor V en una pila de salvaguarda P3
206
Cálculo de un dato de control de integridad X en el tipo de la variable Z
208
Memorización del dato de control de integridad X en la pila de control P2
210
Memorización de un único dado de direccionamiento A2 de la copia y de los datos de integridad en un registro ind2
212
Verificación de integridad de la variable Z a partir de los datos de integridad
214

FIG. 6