(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 628 820**

(51) Int. CI.:

G06F 21/00 (2013.01)
G06F 11/30 (2006.01)
G06F 11/22 (2006.01)
G06F 21/52 (2013.01)
G06F 21/56 (2013.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(86) Fecha de presentación y número de la solicitud internacional: **03.11.2011 PCT/US2011/059244**

(87) Fecha y número de publicación internacional: **10.05.2012 WO12061663**

(96) Fecha de presentación y número de la solicitud europea: **03.11.2011 E 11838845 (3)**

(97) Fecha y número de publicación de la concesión europea: **22.02.2017 EP 2635992**

(54) Título: **Uso de toma de huellas digitales de potencia (PFP) para monitorizar la integridad y potenciar la seguridad de sistemas informáticos**

(30) Prioridad:

03.11.2010 US 409670 P
15.04.2011 US 201161475713 P

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
04.08.2017

(73) Titular/es:

VIRGINIA TECH INTELLECTUAL PROPERTIES, INC. (100.0%)
2200 Kraft Drive, Suite 1050
Blacksburg, VA 24060, US

(72) Inventor/es:

REED, JEFFREY H. y
AGUAYO GONZALEZ, CARLOS R.

(74) Agente/Representante:

LOZANO GANDIA, José

USO DE TOMA DE HUELLAS DIGITALES DE POTENCIA (PFP) PARA MONITORIZAR LA INTEGRIDAD Y POTENCIAR LA SEGURIDAD DE SISTEMAS INFORMÁTICOS

DESCRIPCIÓN

La presente invención se refiere a un método para realizar una estimación de integridad en tiempo real de ejecución de una rutina. Según un segundo aspecto, la invención se refiere a un sistema correspondiente.

En circuitos digitales CMOS, con cada transición de bits existe un drenaje de corriente transitoria que resulta de un breve cortocircuito en las compuertas y la carga y descarga de capacitancia parásita en las salidas del circuito. En un procesador, la intensidad de estas corrientes transitorias, por tanto, la potencia total consumida en un ciclo de reloj específico, se determina mediante el número total de transiciones de bits que tienen lugar en ese ciclo. El número de transiciones de bits se determina mediante la secuencia de instrucciones específica ejecutada, así como sus direcciones y parámetros. La toma de huellas digitales de potencia es una solución de estimación de integridad y detección de intrusos para sistemas cibernéticos críticos basada en tomar medidas refinadas del consumo de potencia de un procesador y compararlas con firmas de confianza (patrones que resultan de la secuencia específica de transiciones de bits durante la ejecución) para la detección de anomalías. El enfoque básico tras la toma de huellas digitales de potencia se conoce a partir de Carlos R Aguayo Gonzales *et al* Proceedings of “Power fingerprinting in SDR & CR integrity”, Military Communications Conference, 2009, Milcom 2009. IEEE Piscataway, NJ, EE.UU., páginas 1 a 7, ISBN 978-1-4244-5238-5. El enfoque consiste en caracterizar la ejecución de software de confianza y extraer sus firmas de potencia y usarlas como referencia para comparar perfiles de prueba para determinar si está ejecutándose el mismo código. Un monitor de toma de huellas digitales de potencia (PFP) consiste en tres elementos principales comunes a todos los sistemas de reconocimiento de patrón, tal como se muestra en la figura 1: detección 110, extracción 120 de características, y detección/clasificación 130. La detección implica medir el drenaje de corriente instantánea de hardware digital, lo que puede lograrse usando una sonda de corriente comercial y a osciloscopio de alto rendimiento. La extracción de características es un aspecto crítico para la PFP e implica la identificación de propiedades estadísticas y temporales del consumo de potencia que identifica de manera única la ejecución de una rutina de software dada. Esta es una tarea difícil que requiere un conocimiento profundo de la arquitectura del procesador y la estructura del software, pero que puede facilitarse construyendo el propio software con determinadas características que potencian las firmas y mejoran el determinismo. De manera ideal, se extrae una firma de cada trayectoria de ejecución en el código. En los casos en los que esto no es viable, solo se caracterizan y monitorizan unas pocas secciones críticas, tal como módulos de kernel del OS y aplicaciones de núcleo.

En el enfoque de toma de huellas digitales de potencia general, se sitúa un sensor 110 en la placa del procesador tan cerca de las clavijas de alimentación como sea posible. El sensor capta el drenaje de corriente instantánea del procesador. El sensor puede ser una sonda de corriente comercial, una resistencia en derivación o un espejo de corriente. La señal del sensor debe digitalizarse a una velocidad mayor que la velocidad de reloj principal del procesador. Si el procesador tiene un bucle de bloqueo de fase interno para aumentar la frecuencia de funcionamiento, entonces pasa a ser la frecuencia de reloj eficaz. Se han obtenido resultados satisfactorios usando 3,5X la frecuencia de reloj eficaz, pero esto no representa un límite inferior. Pueden usarse varios mecanismos para reducir los requisitos de muestreo.

Después de que el drenaje de corriente instantánea se haya digitalizado para dar un perfil de potencia, se aplican diferentes técnicas de procesamiento de señal para extraer características discriminatorias de los perfiles. Después de haberse extraído las características, se hacen pasar a través de un clasificador o detector 130 supervisado que se ha entrenado previamente usando perfiles 140 de software de confianza. En última instancia, este detector toma la decisión de si la ejecución de software corresponde o no al software autorizado. En la figura 1 se presenta una descripción pictográfica del enfoque de toma de huellas digitales de potencia general en la técnica anterior.

La decisión de si las características de un perfil de potencia específico corresponden a la ejecución autorizada se realiza mediante un detector diseñado cuidadosamente, que compara perfiles de potencia de entrada con todas las firmas 140 almacenadas del código autorizado. Cuando los perfiles observados no pueden hacerse coincidir con ninguna de las firmas almacenadas, dentro de una tolerancia razonable, se determina que se ha producido una intrusión. Aunque la diferencia para cada característica puede ser pequeña, la confianza de evaluar una intrusión puede ser muy alta y establecerse arbitrariamente debido al gran número de características.

Sin embargo, las técnicas y procedimientos actuales deben potenciarse y mejorarse para ir al compás de la tecnología y las prácticas que están desarrollándose y usándose por aquellos que buscan superar o derrotar las salvaguardias que se basan en la toma de huellas digitales de potencia para determinar la integridad de ejecución de sistemas informáticos.

Sumario de la invención

Por tanto, un objeto de la presente invención es proporcionar procedimientos para potenciar la estimación de integridad de ejecución de sistema objetivo determinada mediante toma de huellas digitales de potencia (PFP). La

invención resuelve este problema por medio de un método según la reivindicación 1 y un sistema según la reivindicación 7.

La invención puede llevarse a cabo: integrando PFP en la fase de detección de seguridad de defensa en profundidad exhaustiva; desplegando una red de nodos habilitados para PFP; ejecutando dispositivos que no son de confianza con entradas predefinidas que fuerzan una secuencia de estado específica y ejecución de software específica; incorporando información de identificación de módulo en señalización de sincronización; combinando señales de diferentes elementos de placa; usando firmas de malware para potenciar la realización de PFP; mediante caracterización automática y extracción de firmas; proporcionando actualizaciones de firmas seguras; protegiendo contra ataques de canal lateral; realizando estimación de integridad en tiempo real en una plataforma incorporada monitorizando su consumo de potencia dinámico y comparándolo con firmas de código de confianza; precaracterizando el consumo de potencia de la plataforma y concentrándose en secciones de perfil que portan la mayor parte de la información sobre el estado de ejecución interno; mejorando la capacidad de PFP para detectar desviaciones de la ejecución autorizada en plataformas incorporadas comerciales.

Un aspecto de la invención es un método para realizar una estimación de integridad en tiempo real de ejecución de una rutina en una plataforma de procesamiento informático. Esto se logra monitorizando la ejecución de la rutina rastreando el consumo de potencia de un procesador, mediante la toma de muestras del procesador durante la ejecución de la rutina. Se emplea una técnica de caracterización de plataforma que detecta secciones de los perfiles, es decir, aquellas secciones que presentan la mayor dependencia de transiciones de estado en el procesador. Estas secciones se usan para seleccionar características que portan la mayor parte de la información. Esta caracterización de plataforma se aplica a la plataforma y puede usarse para todas las rutinas ejecutadas en la plataforma. La siguiente etapa es obtener, a partir de una caracterización de características seleccionadas de la rutina, tal como se contiene en las secciones identificadas en la caracterización de plataforma, un conjunto de huellas digitales de potencia de confianza de la rutina. Después, se establece un umbral para una tasa de falsas alarmas específica basándose en la distribución de probabilidad de distancia a partir de una firma compuesta por las huellas digitales de confianza. Después, se compara una biblioteca de las huellas digitales de confianza con características extraídas de perfiles a partir de la ejecución de código que no es de confianza, y después se determina una distancia entre las huellas digitales en la biblioteca y las características extraídas a partir de la ejecución del código que no es de confianza. Se notifica una excepción si la distancia supera el umbral.

Se describen diversos procedimientos para mejorar el funcionamiento, eficiencia, utilidad y rendimiento de sistemas de estimación de integridad y detección de intrusión basados en la toma de huellas digitales de potencia (PFP). Los diferentes procedimientos incluyen:

- Incorporar información de identificación de módulo en señalización de sincronización
- Monitorización de PFP mejorada combinando señales de diferentes elementos de placa
- Usar firmas de malware para potenciar el rendimiento de PFP, generalizando basándose en la tecnología de monitorización de baterías existentes.
- Caracterización automática y extracción de firmas
- Actualizaciones de firmas de seguras
- Respuesta a violaciones de integridad y seguridad en capas
- Protección contra ataques de canal lateral

También se describen métodos y aparatos para:

- Red de monitores de PFP distribuida para monitorizar la dinámica y comportamiento de malware
- Aplicación de PFP para análisis de confianza de cadena de suministro
- Gestión de derechos digitales y alquileres de ejecución limitada
- Predicción de falla basándose en PFP

Breve descripción de los dibujos

Los objetos, aspectos y ventajas anteriores y otros se entenderán mejor a partir la siguiente descripción detallada de realizaciones preferidas de la invención haciendo referencia a los dibujos, en los que:

La figura 1 es un diagrama de bloques general de toma de huellas digitales de potencia.

La figura 2 es un diagrama que muestra ubicaciones de sensor ideales para un monitor de PFP.

La figura 3 es un diagrama que muestra ubicaciones de sensor ideales para placas de múltiples procesadores.

La figura 4 es un diagrama que muestra un ejemplo de desencadenamiento con una señal física.

La figura 5 es un diagrama esquemático que muestra la inserción de instrucción estratégica de PFP para la sincronización y desencadenamiento.

La figura 6 es un diagrama esquemático que muestra el acceso indirecto a recursos físicos en el paradigma de controlador de dispositivo Linux.

La figura 7 es un diagrama de flujo que muestra el procedimiento de caracterización de código de confianza.

La figura 8 es un gráfico que muestra un ejemplo de procesamiento previo de perfiles calculando su densidad espectral de potencia.

La figura 9 es un gráfico que muestra un procesamiento previo de muestra de perfiles en el dominio de tiempo.

La figura 10 es un gráfico que muestra la diferencia de PSD a partir de la ejecución de perfiles de prueba frente a una firma almacenada.

La figura 11 es un diagrama de flujo que muestra el procedimiento de diseño de detector.

La figura 12 es un gráfico que muestra una distribución de probabilidad de muestra de ejecución código de confianza usada para el diseño de detector y selección de umbral.

La figura 13 es un diagrama de flujo que muestra el procedimiento de operación de estimación de integridad de PFP.

La figura 14 es un diagrama esquemático que muestra una configuración de realización de muestra para la plataforma Android.

La figura 15 es una representación gráfica de distribuciones de muestra que resultan de la ejecución de la rutina sin manipulación indebida original.

La figura 16 es una representación gráfica de distribuciones de muestra que resultan de la ejecución de la rutina manipulada indebidamente.

La figura 17 es un gráfico de detalle de perfil de muestra que muestra diferentes secciones de los perfiles que contienen diferentes niveles de información discriminatoria.

La figura 18 es una representación esquemática de una caracterización de plataforma que usa una proyección lineal desde el punto de vista más informativo.

La figura 19 es un diagrama esquemático que muestra una configuración de medición de referencia para caracterización y monitorización de consumo de potencia de plataforma usando PFP.

La figura 20 es un gráfico que muestra un perfil de muestra de ejecución de código de nivel inicial para evaluar la capacidad para detectar un cambio de consumo de potencia mínimo.

La figura 21 es una representación gráfica de una firma promedio de la ejecución de código de nivel inicial, en la que cada punto representa una proyección en un espacio euclidiano de n dimensiones.

La figura 22 es un gráfico que muestra una distribución de muestra de distancias euclidianas a partir de la firma promedio extraída de la ejecución del código de nivel inicial.

La figura 23 es un gráfico que muestra una distribución de muestra de distancias euclidianas de la firma de nivel inicial en el espacio transformado obtenido usando PCA.

La figura 24 es un gráfico que muestra los baricentros de perfiles a partir de instrucciones de perfil para LDA.

La figura 25 es un gráfico que muestra distribución de muestra de distancias euclidianas a partir de la firma de nivel inicial en el espacio transformado obtenido usando LDA.

La figura 26 es un diagrama de bloques esquemático de una plataforma objetivo a modo de ejemplo para detectar

desviaciones de la ejecución de software autorizada.

La figura 27 es un diagrama esquemático que muestra diferentes capas en un enfoque de defensa en profundidad de la seguridad cibernética.

La figura 28 es un diagrama esquemático que muestra el alcance de monitorización de PFP dentro de una solución de seguridad de defensa en profundidad en capas.

La figura 29 es un diagrama esquemático que muestra nodo de señuelo de PFA para monitorizar y recopilar información.

La figura 30 es un diagrama esquemático que muestra una red de señuelos de PFP.

La figura 31 es un diagrama de flujo de un análisis de confianza de cadena de suministro usando PFP.

La figura 32 es un diagrama esquemático que muestra posibles fuentes de firmas de referencia para el análisis de confianza de cadena de suministro usando PFA.

La figura 33 es un diagrama que muestra el uso de un registro de IO para proporcionar señalización de sincronización e identificación a un monitor de PFP.

La figura 34 es un diagrama que muestra la incorporación de señalización de sincronización e identificación de PFP en perfiles de potencia.

La figura 35 es un diagrama que muestra una configuración de muestra para combinar múltiples señales para la estimación de integridad de PFP.

La figura 36 es un diagrama de flujo de un procedimiento para extraer características que excluyen falla de dispositivo para PFP.

La figura 37 es un diagrama esquemático que muestra relaciones entre los diferentes elementos de sistema que interactúan para la caracterización automática y extracción de características.

La figura 38 es un diagrama de una estructura para impedir que ataques de canal lateral aprovechen un monitor de PFP incorporado.

La figura 39 es un diagrama de una estructura para impedir que ataques de canal lateral aprovechen un monitor de PFP externo.

Descripción detallada de una realización preferida de la invención

La toma de huellas digitales de potencia (PFP) es una técnica que permite que un monitor externo estime la integridad de ejecución de un sistema cibernético. La PFP se basa en la información de estado de ejecución portada por el consumo de potencia dinámico de un procesador. Mediante el uso de esta información, junto con firmas precaracterizadas a partir de referencias de confianza, la PFP puede determinar la integridad de ejecución en sistemas objetivo. Para la aplicación práctica de PFP, se necesita implementar un aparato específico y seguir procedimientos específicos para proporcionar una solución de monitorización eficaz. En el presente documento, se describen diversos procedimientos para mejorar el funcionamiento, eficiencia, utilidad y rendimiento de una solución de monitorización de PFP.

Aplicación de PFP para detectar modificaciones de software en teléfonos inteligentes y otros dispositivos incorporados.

La seguridad cibernética se ha vuelto un elemento crítico para la seguridad nacional. Los microprocesadores están muy extendidos en casi todos los aspectos de la vida moderna. Los avances tecnológicos en las áreas de tecnología de la información avanzan a un ritmo más rápido que las soluciones de seguridad necesarias para protegerlos. La amenaza de ataques cibernéticos permanece constante con consecuencias posiblemente devastadoras para la infraestructura crítica y seguridad nacional. La infraestructura cibernética se ha vuelto tan importante que, ahora, el ciberespacio se considera un nuevo campo de guerra y un elemento crítico para la seguridad nacional que necesita protegerse todo tipo de amenazas, incluyendo los adversarios subvencionados por el estado.

Se describe una técnica para realizar una estimación de integridad en tiempo real en teléfonos inteligentes y otras plataformas incorporadas monitorizando su consumo de potencia dinámico y comparándolo con firmas a partir de un código de confianza. El método y tecnología descritos se construyen a partir del concepto general de toma de huellas digitales de potencia y proporcionan mejoras para la aplicación general en dispositivos comerciales complejos. Se presentan ejemplos de realizaciones preferidas de las técnicas generales que van a usarse como

referencias y ejemplos. Sin embargo, las técnicas son generales y pueden adaptarse a cualquier plataforma cibernética.

Como parte del enfoque, también se describe una metodología para precaracterizar la manera en la que una plataforma y procesador específicos consumen potencia para mejorar el rendimiento del enfoque concentrando esfuerzos de clasificación en las secciones de los perfiles que portan la mayor parte de la información sobre el estado de ejecución interno del procesador e ignoran características extremadamente ruidosas o redundantes que pueden perjudicar rendimiento.

El objetivo es potenciar el enfoque de toma de huellas digitales de potencia (PFP) general para definir una técnica fiable para detectar modificaciones de software no autorizadas en teléfonos inteligentes, sistemas incorporados y sistemas de información generales. En la figura 1 se representa el enfoque de la técnica anterior general.

El método de PFP general comienza recopilando medidas refinadas del consumo de potencia durante la ejecución de código de confianza. Se necesita que el sensor 110 recopile una representación de medida directa o indirecta del consumo de potencia dinámico o drenaje de corriente instantánea del procesador. El sensor 110 puede implementarse por medio de una sonda de corriente comercial, un sensor de efecto Hall, sensor de campo magnético compuesto, piezoeléctrico/magnetostrictivo, bobina de Rogowski, un espejo de corriente de alto ancho de banda, o una resistencia en derivación de precisión de baja resistencia simple. Obsérvese que los sensores deben cumplir los requisitos establecidos por las técnicas de extracción de características específicas seleccionadas.

La ubicación física del sensor es un elemento crítico para el éxito de este enfoque. La ubicación 210 ideal se muestra en la figura 2 en la señal V_{DD} del procesador 205. Si esta ubicación no es viable, o introduce ruido de fuente de alimentación excesivo, entonces también se muestra la segunda mejor ubicación 220. Si el sensor 220 se sitúa en la segunda ubicación los perfiles de cobre con su capacitancia e inductancia parásitas junto con los condensadores 215 de desacoplamiento crean un filtro pasobajo (LP) RLC que afecta a los perfiles de corriente. Para la PFP resulta beneficioso precaracterizar este efecto de hardware identificando la función de transferencia, H, del filtro LP usando un analizador de redes comercial u otra técnica de identificación de sistema. El efecto del filtro LP inherente puede minimizarse haciendo pasar los perfiles a través de otro filtro con la función de transferencia inversa, H_{inv}. Se recomienda implementar el filtro inverso digitalmente. Puesto que la inversión directa de H puede conducir a un filtro inestable, es necesario seleccionar la aproximación estable más cercana de H_{inv}.

En la figura 2, puede proporcionarse $V_{DD\_núcleo}$ 225 mediante diferentes fuentes. Para procesadores simples, viene directamente de los reguladores de tensión. Para plataformas más sofisticadas, puede venir de un sistema de gestión de potencia y periféricos, que es un circuito complejo que proporciona una gran red de servicios que incluye la entrega de diferentes niveles de tensión requeridos, restablecer e interrumpir la gestión, y otras gestiones de periféricos. Los gestores de potencia son sistemas complejos que combinan diferentes señales y añaden interferencia desde la perspectiva de PFP y tienden a ocultar las firmas de potencia. Para un sistema con un circuito de gestión de potencia, se recomienda diseñar la placa de sistema con las provisiones necesarias para colocar el sensor de corriente después del sistema de gestión de potencia para evitar la interferencia adicional y facilitar la extracción de firmas. En la situación de mejor caso, el sensor de potencia se incluirá en el sistema de gestión de potencia como otro servicio proporcionado, facilitando la integración de PFP.

En el caso de múltiples procesadores en la placa, puede repetirse el mismo principio para cada uno de los procesadores, tal como se muestra en la figura 3, en la que el n-ésimo procesador 206 se monitoriza preferiblemente en 211 o en una segunda mejor ubicación 221 después del condensador 216 de desacoplamiento. En este caso, el detector debe diseñarse para combinar y considerar perfiles de ambos sensores. Para procesadores de múltiples núcleos en el mismo paquete, se aplican los mismos principios que en el ejemplo de múltiples procesadores, pero la ubicación y viabilidad dependerán de la arquitectura del procesador, el número de núcleos alimentados por cada carril, y requisitos de desacoplamiento.

Con el sensor en su sitio, la siguiente etapa es caracterizar código de confianza. Este procedimiento se logra ejecutando repetidamente el código de confianza objetivo en un entorno controlado (incluyendo aislamiento del software objetivo, establecimiento de entradas usadas durante la ejecución, e inserción de marcadores específicos que ayudan a sincronizar perfiles). Los marcadores pueden ser de diferente naturaleza y ayudar con el desencadenamiento y la sincronización. Los posibles marcadores incluyen señales físicas (como cambiar el nivel de tensión de una clavija) o una secuencia específica de instrucciones que produce una secuencia de consumo de potencia conocida. Un ejemplo de una señal 410 desencadenante física se muestra en la figura 4. El concepto de inserción de instrucciones para el desencadenamiento se representa en la figura 5. En este caso las instrucciones 515 de montaje adicionales se eligen para producir un patrón 510 conocido en los perfiles, habitualmente una fuerte variación en el drenaje de corriente durante un periodo de tiempo corto para ayudar a indicar cuándo se ejecuta un código 510 específico.

Cuando la aplicación 610 objetivo está ejecutándose en el espacio de usuario en una plataforma que implementa el paradigma de controlador de dispositivo Linux, o en cualquier otro sistema operativo con acceso indirecto a señales físicas, tal como se describe en la figura 6, es necesario compensar las incertidumbres inherentes en la ejecución y

el sincronismo provocadas por el acceso indirecto. En este caso, la instrucción 515 de desencadenante se ejecutará en el espacio 610 de usuario que no tiene acceso directo a la memoria 640 física, y solo puede acceder a los registros 632 necesarios para crear la señal 650 física por medio de un controlador 631 de dispositivo ubicado en el espacio 620 de kernel. Las incertidumbres en la ejecución y el sincronismo existen debido a que el acceso a archivos requiere que los procedimientos esperen (ejecución en bloques) a la señalización de sincronización apropiada durante la cual el kernel 620 programa otro procedimiento para que se ejecute.

Aunque no se requiere que los marcadores 630 permanezcan en el código final, el procedimiento de estimación de tiempo de ejecución se ve facilitado si permanecen en su sitio. En el caso en el que los marcadores se dejan en la versión desplegada, es necesario garantizar que las instalaciones o servicios usados para los marcadores permanecerán aún en la plataforma desplegada (por ejemplo si se supone que el marcador enciende un LED 650, ese LED 650 debe existir en la plataforma desplegada).

Es importante observar que durante la caracterización es necesario usar el código exacto que va a desplegarse. Esto incluye usar exactamente las mismas herramientas para construir el software, con el mismo nivel de optimización, etc.

Para un mejor rendimiento, la caracterización debe ser un procedimiento independiente e iterativo, durante el que la estructura de código de fuentes junto con los marcadores respectivos se desarrollan conjuntamente para producir las firmas más fuertes con la menor varianza entre las diferentes instancias de ejecución.

Puede necesitarse recopilar varios perfiles de la ejecución del código de confianza para calcular su promedio y reducir el impacto de ruido aleatorio inherente a cualquier sistema físico. El procedimiento de caracterización se representa en la figura 7. Después de insertar 710 marcadores en el código, se ejecuta el software de confianza y se captan 720 los perfiles de potencia resultantes. Esto se hace para todas las trayectorias 730 de ejecución significativas, usando una entrada 735 predefinida si es necesario. Las variaciones debidas a parámetros aleatorios se eliminan usando PCA (análisis de componentes principales) 740. Se extraen características 750 discriminatorias y se realizan el análisis estadístico, el cálculo del promedio y la agrupación 760 para generar un conjunto de firmas 770 autorizadas.

Las firmas pueden extraerse de dominios de señal diferentes y ser multidimensionales. Además, pueden usarse múltiples firmas para identificar un único fragmento de código.

Procesamiento de perfil y extracción de características

El procedimiento de preparación de perfiles de prueba que van a compararse con la firma almacenada se conoce como procesamiento previo y extracción de características. El procesamiento previo de perfil implica tareas generales para condicionar los perfiles para extraer las características discriminatorias seleccionadas, por ejemplo convirtiendo los perfiles en el dominio apropiado o alineando los perfiles con referencia a un marcador específico. Un ejemplo de procesamiento previo de perfil se muestra en la figura 8, en la que perfiles de dominio de tiempo de la ejecución de software de prueba en una placa BeagleBoard con un procesador OMAP3 se convierten en primer lugar al dominio de frecuencia calculando su densidad espectral de potencia.

Otro ejemplo de procesamiento previo básico es alinear los perfiles de dominio de tiempo, tal como se muestra mediante la alineación de la ejecución de base y perfiles alternos (-1 transición de bits) en la figura 9, antes de que se pase a un detector de correlación. En este ejemplo, cada perfil de N muestras se considera como un punto en un espacio euclidiano multidimensional.

La extracción de características es el procedimiento de calcular el dato estadístico de prueba final (a partir de nuevos perfiles) que se pasa a los detectores y se usa para determinar la integridad. Este procedimiento es único para cada característica seleccionada. Por ejemplo, en análisis de correlación de dominio de tiempo básico, el procesamiento previo puede incluir una sincronización gruesa y compensación para patrones de consumo de potencia de plataformas específicas, mientras que la extracción de características implica la comparación con la firma almacenada calculando el factor de correlación o la distancia euclidiana. Un ejemplo de extracción de características se muestra en la figura 10, que muestra el error de PSD en dB de perfiles de prueba que corresponden a la ejecución del código de confianza y el código manipulado indebidamente en el procesador OMAP3 de la placa BeagleBoard siguiendo el ejemplo de PSD en la figura 8. El uso de este vector diferencia, el dato estadístico de prueba final o característica discriminatoria que se pasa al detector puede representarse mediante el error cuadrático medio o cualquier otra medida de error o distancia.

Diseño de detector

Después de que se hayan extraído las firmas y se hayan seleccionado las características discriminatorias, la siguiente etapa en el procedimiento de PFP es diseñar detectores óptimos para realizar la estimación de integridad final. Estos detectores tomarán la decisión final de si un perfil de prueba debe considerarse una intrusión durante la operación de monitorización. El procedimiento de diseño de detector y la operación de monitorización normal son

muy similares. En el diseño de detector, se captan perfiles de prueba de la ejecución de software de confianza y se procesan para extraer las características discriminatorias seleccionadas y se comparan con las firmas almacenadas. Se recopilan y se procesan varios perfiles y se usan sus distribuciones de muestra estadísticas para identificar un umbral que produce los objetivos de rendimiento esperados. El procedimiento de diseño de detector se muestra en la figura 11. La entrada 1110 aleatoria o predefinida se proporciona al software 1120 de confianza y se captan perfiles de prueba nuevos a partir de su ejecución. Los resultados se alinean y sincronizan 1130, y los perfiles se someten a procesamiento previo y se condicionan 1140. Usando firmas 770 autorizadas para la comparación, se extraen las características discriminatorias seleccionadas y se genera 1150 una medida de distancia. Después, se realiza 1160 análisis estadístico y ajuste de distribución con las medidas resultantes. Finalmente, se aplica 1170 el criterio de Neyman-Pearson para determinar un umbral que cumpla los objetivos de rendimiento esperados.

Un enfoque común para crear detectores óptimos implica la aplicación del criterio de Neyman-Pearson para maximizar la probabilidad de detección para una probabilidad de falsa alarma dada. Como breve recordatorio de este criterio, que surge a partir de la teoría de prueba de hipótesis básica, se establece una probabilidad de falsa alarma objetivo basándose en la tolerancia y el coste estimado de cometer un error en la decisión final. Usando una estimación de la distribución de probabilidad de las características discriminatorias del código de confianza, se calcula un umbral de distancia que produce la probabilidad de falsa alarma esperada al tiempo que maximiza la probabilidad de detección correcta. Un ejemplo de este procedimiento se muestra en la figura 12, en la que se calcula un umbral 1220 de distancia para una distribución 1210 de probabilidad que produce una probabilidad 1230 de falsa alarma esperada.

Sin embargo, es importante observar que existen diferentes técnicas que pueden producir resultados mejorados dependiendo de la naturaleza de las características discriminatorias seleccionadas. Otras técnicas para el diseño de detector y entrenamiento de máquinas incluyen: redes neuronales, máquinas de vector de soporte y modelos de Markov ocultos.

Operación de monitorización

Después de que se hayan extraído las firmas de la ejecución de código de confianza, se hayan seleccionando características discriminatorias, y se hayan diseñado detectores óptimos, el monitor de PFP está listo para estimar la integridad de software de prueba. Tal como se mencionó anteriormente, el procedimiento de estimación de integridad normal es muy similar al procedimiento de diseño de detector. Durante el funcionamiento normal, el monitor también extrae las características discriminatorias seleccionadas a partir de perfiles de potencia tras el procesamiento previo necesario, pero en vez de recopilar los datos estadísticos de varios perfiles tal como se realizó para el diseño de detector, se hacen pasar a través del detector apropiado para compararlos con los umbrales respectivos y determinar el estado de integridad de la ejecución de código de prueba. El detector compara los perfiles de prueba con todas las firmas conocidas y, si ningún dato estadístico de prueba individual es suficiente para determinar que se ha ejecutado el código autorizado, entonces se notifica una intrusión. Este procedimiento se representa en el diagrama mostrado en la figura 13. El software objetivo se ejecuta 13310 durante el funcionamiento normal o usando una entrada predefinida para captar perfiles 1320 de prueba, que después se alinean y sincronizan 1330, y después se someten a procesamiento previo y se condicionan 1340. Después, el detector compara 1350 las características extraídas con las firmas 1370 conocidas para determinar una distancia, usando el umbral 1220 predefinido para tomar una decisión 1360 de estimación de integridad.

Resultados de muestra

Con el fin de mostrar a modo de ejemplo el procedimiento de PFP en teléfonos inteligentes y otras plataformas incorporadas, se describe una implementación de referencia de esta técnica usando una placa BeagleBoard revisión C4 con el procesador ARM (OMAN3 a 720 MHz) que se ejecuta en la plataforma Android. La placa 1410 BeagleBoard se modifica ligeramente cortando las pistas 1420 principales que proporcionan potencia al carril de potencia de núcleo para conectar una sonda 1430 de corriente. Se implementa el sistema de captación usando un osciloscopio 1440 en tiempo real comercial y una sonda 1430 de corriente. El osciloscopio se configura a una velocidad de muestreo de 2,5 GSps y se recopilan un total de 30 mil muestras en cada pista iniciado mediante el desencadenante 1450. La configuración se describe en la figura 14.

Se desarrolló una aplicación de prueba básica para demostrar el procedimiento y mostrar viabilidad. Esta aplicación básica consiste en un contador simple que visualiza un número entero creciente en la pantalla del dispositivo. El funcionamiento de la aplicación se describe en la lista 1 y consiste en una estructura de aplicación de Java para Android típica con una rutina de inicialización que prepara la pantalla para mostrar un cuadro de texto y establece una variable de número entero usada como contador. También existe una rutina denominada “DisplayCounter” que se encarga de aumentar el valor del contador y visualizarlo en la pantalla. Esta rutina se configura como tarea recurrente que se solicita cada segundo.

LISTA 1. Pseudocódigo de aplicación de prueba de Android

```
Inicializar
```

```
Display-Counter ()
{
    contador = IncrementValue(contador);
    Visualizar datos
}
Suspender durante un segundo
DisplayCounter al reactivarse
```

La rutina IncrementValue crítica se implementó en código C nativo y se incluyó como biblioteca externa mediante el conjunto de herramientas NDK de Android, en vez de la implementación Java tradicional. Antes de la sección crítica, se establece un desencadenante 1450 físico para indicar 1460 a los sistemas de captación que empiecen a recopilar perfiles de potencia.

Solo se caracteriza y monitoriza la rutina IncrementValue crítica mostrada en la lista 2.

LISTA 2. Pseudocódigo de rutina nativa monitorizada en C

```
/*Rutina nativa crítica*/
int incrementValue(int Val)
    /*LED desencadenante usr1*/
    Abrir archivo de control de controlador de dispositivos
    Escribir 1 en el archivo
    /*Incrementar Val*/
    Val++;
    /*Procesamiento adicional general*/
    i = 1000;
    mientras (i) i - -;
    /*Restablecer desencadenante LED usr1 */
    Escribir 0 en el archivo
    Cerrar archivo de control de controlador
    devolver Val;
```

La extracción de firmas se realiza en el dominio de frecuencia simplemente calculando el promedio del PSD de varios perfiles a partir de la ejecución de código de confianza. Se ignora la información de fase del perfil. Se calcula el promedio del PSD de doscientos perfiles junto con la producción de la firma.

Las características discriminatorias también se extraen en el dominio de frecuencia mediante un error cuadrático medio entre la firma y el PSD de perfiles de prueba (en dB). Se calcula el promedio del PSD de los últimos tres perfiles de prueba juntos antes de calcular el MSE. Solo se usan los primeros 200 MHz del PSD en el cálculo de MSE.

Este procedimiento para la extracción de firmas produce una característica discriminatoria monodimensional.

El diseño de detector se realizó usando el criterio de Neyman-Pearson descrito anteriormente usando una probabilidad de falsa alarma objetivo, P_{F4}, del 1%. Los datos estadísticos de muestra del perfil se extraen de una muestra de 200 perfiles de la ejecución del código de confianza.

La distribución de muestra se ajustó a una distribución de Rayleigh con una media y una varianza iguales a la media y varianza de la distribución de muestra de entrenamiento. Usando esta distribución, se calcula la distribución de probabilidad inversa para encontrar el umbral que produce la P_{FA} objetivo del 1% objetivo.

Para someter a prueba la capacidad de detectar desviaciones de ejecución del código de confianza, se somete a prueba el monitor diseñado previamente usando una versión ligeramente manipulada de manera indebida de la aplicación. La aplicación manipulada indebidamente, mostrada en la lista 3, está diseñada para emular un ataque encubierto en el que la intrusión permanece inactiva hasta que se cumple una condición específica. La intrusión consiste en una modificación muy simple en la que se escribe un archivo sólo cuando el valor de un contador alcanza un valor específico (la condición).

LISTA 3. Pseudocódigo de rutina nativa crítica manipulada indebidamente

```
/*Rutina nativa crítica*/
Int. incrementValue(int Val)
{
    /*LED desencadenante usr1*/
    Abrir archivo de control de controlador de dispositivo
    Escribir 1 en el archivo
    /* Manipulación indebida */
    si (Val == 1) {
    //abrir archivo temporal
    //escribir Val en el archivo
    //cerrar archivo
    }
    /*Incrementar Val*/
    Val++;
    /*Procesamiento adicional general*/
    i = 1000;
    mientras (i) i--;
    /*Restablecer desencadenante LED usr1*/
    Escribir 0 en el archivo
    Cerrar archivo de control de controlador
    devolver Val;
}
```

Es importante observar que la escritura de archivo que tiene lugar en la manipulación indebida solo se produce una vez durante la ejecución (es decir cuando el contador es 1). El resto del tiempo, cuando se solicita la rutina, la condición no se cumple y no se escribe el archivo adicional. Por tanto, durante la mayor parte del tiempo que se solicita la rutina, la única modificación desde un punto de vista lógico es una evaluación adicional de una condición dada.

Resultados del funcionamiento

Los resultados de ejecutar el monitor cuando se ejecuta la versión sin manipulación indebida original de la rutina se muestran en la figura 15.

Se puede observar que para la duración de la prueba, solo unas pocas instancias pasaron el umbral 1510, lo cual concuerda con la probabilidad de falsa alarma diseñada.

Los resultados de ejecutar el monitor con la versión manipulada indebidamente de la aplicación se muestran en la figura 16. Obsérvese que ninguna instancia se clasifica erróneamente como ejecución autorizada y cada ejecución única de la aplicación manipulada indebidamente se marcará como intrusión por encima del umbral 1610. Además, es importante observar que debido a la ejecución condicionada de la manipulación indebida, solo una vez durante las instancias de ejecución usadas en estos resultados se escribió realmente el archivo. El resto del tiempo, solo se comprobó una condición, y cuando no se cumplió, se reanudó la ejecución normal.

Caracterización de plataforma y evaluación de sensibilidad mínima

Las medidas refinadas del consumo de potencia pueden conducir a información redundante que añade muy poca información discriminatoria, pero que pueden añadir un ruido significativo e incertidumbre a las firmas. En el dominio de tiempo esto tiene el aspecto de la figura 17. En este caso, se desea centrar la atención en las secciones de los perfiles (dimensiones) que tienen la mayor varianza 1710 entre las dos ejecuciones, al contrario que las otras secciones, por ejemplo 1720, que muestran poca varianza entre las dos ejecuciones. Por otra parte, cuando se caracteriza una rutina de software específica que toma parámetros aleatorios, el efecto de estos parámetros aleatorios es introducir ruido en las firmas, lo que acaba reduciendo el rendimiento y aumentando la probabilidad de falsa alarma. En este caso, se desea centrar la atención en las dimensiones (por ejemplo 1720) que permanecen constantes durante la ejecución del software objetivo, al tiempo que se ignoran aquellas que añaden ruido. En este caso, se desean ignorar las dimensiones que presentan gran varianza (por ejemplo 1710).

Para mejorar el rendimiento de PFP, es necesario reducir el número de características analizadas concentrándose solo en aquellas que portan la mayor parte de la información. Esto se logra precaracterizando las características que portan la mayor parte de la información para una plataforma como parte del entrenamiento y después se elimina información redundante durante el procesamiento previo antes de que los perfiles pasen a los detectores.

Antecedentes técnicos

En sistemas de reconocimiento de patrón tradicionales, el procedimiento de selección de un subconjunto de características que maximiza un criterio específico (en el caso de PFP se desea maximizar la información discriminatoria PFP), se conoce como selección de características óptimas. En sistemas de agrupación, esto se logra normalmente protegiendo los perfiles, x, hasta un espacio transformado con menos dimensiones que los más útiles (o perspectiva de información) por medio de una transformación lineal.

Esta transformación se describe como

$$y = Wx$$

en la que W es una matriz de transformación lineal diseñada cuidadosamente que cuando se aplica a perfiles de prueba, produce un perfil transformado con una dimensionalidad menor que maximiza unos criterios particulares. Existen diferentes criterios para identificar la transformación óptima. Dado que se intenta optimizar la selección de características en cuanto a la información discriminatoria, resulta natural seguir un enfoque teórico de información. Esta optimización se ha realizado anteriormente y puede encontrarse en varias fuentes en la bibliografía de reconocimiento de patrón, por ejemplo véase J. T. Tou y R. C. Gonzalez. “Pattern Recognition Principles”. Addison-Wesley Publishing Company, 1974.

Análisis de componentes principales

Un enfoque bien conocido para determinar la W apropiada que optimiza la entropía (o información) en los perfiles se conoce como análisis de componentes principales (PCA). Se supone que las matrices de covarianza de las diferentes clases, C_i, se distribuyen normalmente y son idénticas $C_i = C$. Por tanto, los vectores propios pueden considerarse como portadores de información para los perfiles en consideración. Algunos de estos vectores portan más información discriminatoria en el sentido de clasificación que otros, que pueden eliminarse con seguridad sin mucha penalización de rendimiento. No debe sorprender que los vectores de característica óptima estén vinculados a esos vectores propios y se usen para crear la matriz de transformación W agregando vectores propios en orden descendente según el valor propio correspondiente. Dado que en PFP solo se necesita un único punto por ciclo de reloj, la matriz de transformación W viene dada por el vector propio de la matriz de covarianza asociada al mayor valor propio.

La transformación lineal puede interpretarse como una proyección del perfil de prueba en un espacio transformado de dimensionalidad menor desde la perspectiva más informativa. El PCA puede aplicarse de diferentes maneras,

dependiendo del objetivo específico. Desde una perspectiva de agrupamiento, se prefiere construir W usando los vectores propios asociados a los valores propios más pequeños, ya que esto producirá un agrupamiento más ajustado en el espacio transformado. Por otra parte, también es posible usar los vectores propios asociados a los mayores valores propios cuando se usan perfiles de ejecuciones diferentes. Cuando se aplica de esta manera, el PCA seleccionará las características que presentan la mayor varianza entre clases. Suponiendo que las matrices de covarianza son idénticas, estos vectores propios representarán las características que contienen la máxima información discriminatoria entre los perfiles específicos usados para PCA.

Análisis discriminante lineal (LDA)

El PCA selecciona un subconjunto de características en orden ascendente o descendente en cuanto a la varianza para optimizar la entropía de perfil. Sin embargo, no considera que las diferencias específicas entre clases para seleccionar un conjunto óptimo de características que maximiza la distancia entre distribuciones de análisis discriminante lineal (LDA) y maximiza la divergencia entre distribuciones, que es una medida de distancia entre distribuciones de probabilidad. La divergencia está estrechamente relacionada con el concepto de entropía relativa en teoría de la información.

Usando información específica de diferentes clases y divergencia como criterio de optimización, el LDA identifica la matriz de transformación óptima para proyectar los perfiles a partir de la perspectiva única que produce la separación máxima entre ellos. Esto se debe a que el vector de transformación W es normal al hiperplano discriminante óptimo entre ambas distribuciones.

Siguiendo la suposición de que los perfiles se distribuyen de manera normal, puede mostrarse [TOU] que la matriz de transformación que produce un extremo de divergencia viene dada por el único vector propio de $C^{-1}\delta\delta^T$ asociado con un valor propio distinto de cero. Este vector viene dado por

$$W_0 = C^{-1}(\mu_1 - \mu_0)$$

donde W_0 proporciona la proyección óptima para separar ambas clases mientras que μ_0 y μ_1 son los baricentros respectivos para las dos clases de entrenamiento. El LDA puede extenderse a M clases discriminantes. En este caso, habrá M-1 vectores propios asociados a valores propios distintos de cero.

Caracterización de consumo de potencia de plataforma

Tal como se mencionó anteriormente, no todas las muestras en un perfil de prueba son igual de importantes para determinar si ha tenido lugar o no una desviación de ejecución. Debido a la gran razón de sobremuestreo y a la naturaleza de los perfiles de potencia, existen algunas secciones de los perfiles que portan más información discriminatoria que otras. Para la PFP, el objetivo es identificar una transformación lineal que reduzca la dimensionalidad de los perfiles eliminando la redundancia al tiempo que se enfatizan las dimensiones que portan la mayor parte de la información.

La idea es transformar características discriminatorias para reducir dimensiones usando una proyección lineal de los perfiles usando una matriz de transformación óptima. En el dominio de tiempo, las secciones de perfil que se corresponden con un ciclo 1810 de reloj completo se reducen a un solo punto 1820 en el espacio transformado, tal como se representa en la figura 18. También tienen que diseñarse clasificadores para funcionar en el espacio transformado, reduciendo el número de dimensiones que necesita para considerarse durante la operación de monitorización normal.

Se realiza caracterización en condiciones controladas en el laboratorio y solo se requiere una vez por plataforma. Tal como se describió en las secciones anteriores, existen dos enfoques generales para identificar la matriz de transformación óptima: PCA y LDA.

Caracterización de plataforma usando PCA

Para crear una matriz de transformación usando PCA, es necesario observar el consumo de potencia del procesador durante ciclos de reloj aleatorios. Los perfiles se alinean para cada ciclo de reloj para mostrar claramente las secciones de los perfiles que se ven más afectadas por el comportamiento dinámico de la ejecución de procesador. Una vez que se alinean los perfiles, se usa el PCA para identificar el vector de transformación que representa la mayor parte de la varianza en los perfiles.

La realización de la caracterización de plataforma usando PCA es relativamente fácil implementar y muy adecuada para plataformas complejas en las que el control de contenido en la canalización resulta demasiado difícil.

Caracterización de plataforma usando LDA

Realizar la caracterización de consumo de potencia de plataforma usando LDA requiere el desarrollo de dos rutinas cuidadosamente adaptadas. Estas rutinas deben ejecutar las instrucciones específicas con direcciones y parámetros específicos en la secuencia correcta para crear dos conjuntos de perfiles que muestran diferencia predeterminadas durante un ciclo de reloj específico. Perfiles de entrenamiento a partir de la ejecución de ambas rutinas proporcionan las dos clases para las que LDA encontrará el hiperplano discriminatorio óptimo, que a su vez pasará a ser el vector de transformación óptimo.

El objetivo de la rutina de caracterización especial es ejecutar una secuencia de instrucciones cuidadosamente elaborada para cargar de manera apropiada la canalización de tal manera que en un ciclo de reloj específico hay un cambio conocido durante cada etapa de ejecución (captación, fijación, ejecución, etc.). Los cambios deben ser relativamente pequeños, preferiblemente debidos a cambiar unos pocos bits en los registros respectivos. La rutina de caracterización no es única, sino que es específica de plataforma ya que depende de la arquitectura, conjunto de instrucciones, etc. de la plataforma que está caracterizándose. Diferentes procesadores requerirán probablemente una secuencia diferente.

Una vez captados y sincronizados los perfiles a partir de la ejecución de ambas secuencias, se usa LDA para encontrar el vector de transformación óptimo W. Se espera que la caracterización de plataforma usando LDA proporcione el mejor rendimiento, dada la disponibilidad de dos clases conocidas, pero sus implementaciones son más complejas que PCA.

Resultados de implementación de referencia de caracterización de consumo de potencia de plataforma

Para esta implementación de referencia, se usa una placa base con un microcontrolador PIC18LF4620 de 8 bits de Microchip Technology Inc., similar a los usados en el kit de demostración PICDEM Z, previsto como plataforma de evaluación y desarrollo para IEEE 802.15.4. Este es un microcontrolador incorporado popular sin unidad de gestión de memoria.

La placa base de procesador está ligeramente modificada con el fin de potenciar las características de consumo de potencia. De la placa se retiran un total de seis condensadores de desacoplo que representan en total 6 microF acumulados. La función de estos condensadores es mitigar el estrés impuesto sobre las fuentes de alimentación por los fuertes picos de corriente provocados por procesadores digitales. Es importante observar que no se necesitará eliminar condensadores de desacoplo si el sensor de corriente se coloca más cerca de las clavijas de alimentación de procesador, o si se cancela el efecto del filtro resultante LP usando procesamiento de señales.

La recopilación de perfiles se realiza usando un osciloscopio 1910 en tiempo real Tektronix TDS 649C y una sonda 1920 de corriente Tektronix TC-6. La sonda está conectada justo más allá de los reguladores de tensión en la placa base. El osciloscopio está configurado para 500 MS/s y 10 mV. El desencadenante se acciona mediante un LED1 1930, y está configurado para nivel de 40 mV de flanco de bajada, y no se mantienen muestras previas al desencadenante. Se recopilan un total de L = 30.000 muestras después de cada acontecimiento desencadenante. La configuración de medición se representa en la figura 19. Se captan perfiles y se transfieren a un ordenador central usando GPIB para su análisis posterior.

Se desarrolla una rutina de muestra para este experimento con un propósito doble 1) proporcionar rutinas de entrenamiento para realizar la caracterización de plataforma y 2) proporcionar un cambio de referencia para medir el rendimiento del enfoque. Se comienza por describir el uso de evaluación de la rutina y proporcionar un rendimiento de nivel inicial para comparación. La rutina de prueba se muestra en la lista 4 y se ejecuta en un bucle infinito. En esta rutina, el contenido del registro W se alterna de 00 a 0f usando diferentes instrucciones. Obsérvese que la lógica real en la rutina no tiene ningún impacto sobre el rendimiento de toma de huellas digitales de potencia. Esta rutina se eligió porque resulta fácil controlar el número de transiciones de bits que se producen. Sin embargo, los resultados no dependen del software específico que está ejecutándose. Por tanto, esta rutina proporciona un ejemplo representativo.

LISTA 4.

```
BYTE i; //addr 00

BYTE j; //adder 01

BYTE k; //add 10

BYTE 1; //addr 11

// Inicializar el sistema

BoardInit();
```

```
// Inicializar variables de datos
_asm
movlw 0x07
movwf i, 0 //addr 0x00
movlw 0x0f
movwf j, 0 //addr 0x01
movlw 0x0f //establecer para cambio mínimo
movwf k, 0 //add 0x10
movlw 0x1f
movwf 1, 0 //add 0x11
movlw 0x00
_endasm
//bucle infinito de código objetivo
mientras (1) {
    TMR0H - 0x00; //Reiniciar T1M0
    TMR0L - 0x00;
    LED_2 = 1; //Desencadenante
    LED_2 = 0;
    _asm
        nop
        iorwf j, 0, 0 //w = 0f
        andlw 0x00 //w = 00
        movf j, 0, 0 //w = 0f
        andlw 0x00 //w = 00
        movf k, 0, 0 //w = 0f posibilidad en k (un bit)
        movlw 0x00 //w = 00
        xorwf j, 0, 0 //w = 0f
        movlw 0x00 //w = 00
        iorwf j, 0, 0 //w = 0f
        xorlw 0x00 //w = 00
        nop
        ... x 10
        nop
```

```
        _endasm
}
```

La rutina, tal como se muestra la lista 4, representa la ejecución de base. Previamente al código objetivo, se crea un desencadenante usando un LED en la placa. El desencadenante se usa para sincronizar la captación de perfiles con el osciloscopio. La instrucción “NOP” entre el desencadenante y el código objetivo se incluye como tampón para aislar los perfiles objetivo de cualquier efecto residual del desencadenante. Una vez dentro del bucle principal, el registro W se alterna de 00 a 0f creando cuatro transiciones de bits en ese registro en cada instrucción. El código alternativo, o modificado, tiene una transición menos de bits. En la línea 15, se cambia el contenido de la variable j de 0f a 07. De esta manera, cuando está ejecutándose el código objetivo, en la línea 35, se carga el parámetro k en el registro W que pasa de 00 a 07, realizándose una transición de solo tres bits en el registro para esa instrucción. Obsérvese que solo hay una diferencia de un bit entre este código modificado y la ejecución de base que carga el registro W con 0f y que todo lo demás en la ejecución se mantiene igual, incluyendo instrucciones, parámetros y direcciones. Obsérvese que este cambio de un bit afecta realmente a dos ciclos de reloj, ya que hay una transición menos que entra en esa instrucción y una menos que sale de la misma. Siguiendo al código objetivo hay una cadena de instrucciones “NOP” antes de repetirse el bucle.

En la figura 20 se muestra un detalle de un perfil típico. En esta figura se capta un ciclo de ejecución completo del código objetivo. Los efectos del desencadenante sobre los perfiles de potencia son claramente visibles como dos escalones 2010 y 2020 cuadrados. También pueden apreciarse ciclos de instrucciones individuales. Pueden identificarse como conjuntos de cuatro picos que se repiten cada 125 muestras. Usando información de sincronismo a partir de la documentación del procesador, puede determinarse la sección del perfil que corresponde a la ejecución del código objetivo. En la figura 20, esta sección está destacada como una línea 2030 continua que abarca diez ciclos de instrucciones. Esto concuerda con el código real, que consiste en diez instrucciones de conjunto, tardando cada una un ciclo de bus en ejecutarse.

Se captan varios perfiles de cada una de las ejecuciones de base y alternativa y se calcula el promedio de los perfiles de cada ejecución juntos para proporcionar una imagen limpia de ambas ejecuciones que muestra el efecto total de una transición menos de bits. Los perfiles promedio se muestran en la figura 21. En esta imagen, se muestran los diez ciclos de reloj correspondientes a la ejecución del código de base y parece que los perfiles de cada ejecución están alineados. Sin embargo, alrededor del índice 650 de muestra puede observarse una pequeña diferencia entre los dos perfiles. La diferencia (en 2110) es más apreciable en la parte superior de la figura 21, que proporciona un aumento. Junto con la proximidad de los baricentros de ambas situaciones, también resulta evidente que los perfiles están ampliamente correlacionados debido a sobremuestreo y también que solo determinadas secciones de los perfiles portan información discriminatoria útil.

Con fines de comparación, se proporcionan los resultados de un enfoque de clasificación simplista en el dominio de tiempo sin precaracterización de plataforma. Se usa un clasificador de distancia mínima básico. En este enfoque, cada perfil captado de longitud L = 1250 (la longitud del código objetivo) representa un punto en un espacio euclidiano de L dimensiones. La distancia euclidiana se toma a partir del baricentro de la ejecución de base para cada perfil de prueba entrante. Con fines de clasificación, el baricentro de base y los perfiles de prueba representan un único punto, o vector, en un espacio euclidiano multidimensional con 1250 dimensiones. Los perfiles de prueba son diferentes de los de entrenamiento usados para obtener el baricentro de base. Esto se hace para evitar un sesgo en las evaluaciones de un clasificador de distancia mínima para discriminar con exactitud entre diferentes situaciones.

Los perfiles de prueba de la ejecución de ambas rutinas tienen las distribuciones de distancia euclidiana mostradas en la figura 22. En este ejemplo simplista, el rendimiento de toma de huellas digitales de potencia no es alentador, ya que casi no hay diferencia entre las distribuciones, que se solapan sustancialmente. Este escaso rendimiento se esperaba, teniendo en cuenta las pequeñas diferencias en el consumo de potencia entre las situaciones de base y alternativa.

Los primeros resultados para la caracterización de plataforma se obtienen a partir de la aplicación de PCA. Para este procedimiento se usan todos los ciclos de reloj correspondientes a la ejecución del presente código objetivo en la rutina mostrada en la lista 4. El perfil correspondiente a la ejecución completa del perfil se divide en secciones diferentes correspondientes a una única ejecución de ciclo de reloj. Entones se alinean las subsecciones y se usa PCA para encontrar el vector de transformación W correspondiente al vector propio que representa la mayor varianza. En este caso, tal como se explicó anteriormente, se toma el perfil con sobremuestreo para un ciclo de reloj y se reduce a un único punto.

Tras realizar la caracterización de plataforma usando PCA, se procesan de nuevo los perfiles de prueba de la rutina de evaluación para demostrar las mejoras del rendimiento de precaracterización de plataforma. Las distribuciones de distancia mínima desde los perfiles de prueba transformados hasta la firma en el nuevo espacio transformado de PCA se muestran en la figura 23.

Se observa una clara separación entre la mayor parte de las distribuciones, lo que representa una clara mejora con respecto al rendimiento de la clasificación simplista mostrado en la figura 22.

Resultados con caracterización de plataforma usando LDA

Con el fin de obtener los perfiles de entrenamiento necesarios para aplicar LDA, se ejecuta la rutina de base y una versión ligeramente modificada. Se obtienen los perfiles de caracterización de plataforma especiales comparando dos conjuntos de perfiles: a partir de la ejecución de base, que es una vez más el código en la lista 4 y una versión ligeramente modificada del mismo mostrada en la lista 5. Los cambios en la ejecución se seleccionan cuidadosamente para provocar una transición menos de bits en cada etapa de ejecución en comparación con la ejecución de base. En esta versión modificada, la instrucción en la línea 36 se cambia de xorwf con opcode 0001 10da a iorwf con opcode 0001 00da (los argumentos opcionales d y a, controlan el bit de destino y de acceso de RAM, respectivamente, y se mantienen con el mismo valor en ambos casos). Durante la ejecución, la diferencia en opcodes provocará una transición menos de bits cuando se fija la palabra de instrucción. El parámetro en la instrucción cambia de j, ubicado en la dirección 0x01, a i, ubicado en la dirección 0x00 en RAM de acceso. Una vez más, el cambio creará una transición menos de bits cuando se ejecute. Además, obsérvese que el contenido de j e i también difiere en un bit. Esto también se traducirá en una transición menos de bits cuando se analice sintácticamente el parámetro, cuando se ejecute la instrucción y cuando se escriban los resultados.

LISTA 5. Rutina modificada para la caracterización de plataforma

```
...

35 movlw 0x00 //w = 00

36 iorwf i, 0, 0 //w = 07

37 movlw 0x00 //w = 00

...
```

Para la caracterización de plataforma solo se usan perfiles correspondientes a la ejecución de la línea 36 en la lista 5. El promedio de estos perfiles (para cada ejecución, la ejecución de base y aquella con una transición menos de bits) se muestra en la figura 24.

Usando estos perfiles, se realiza LDA para identificar el hiperplano discriminatorio óptimo y la transformación lineal que proyectan los perfiles desde la perspectiva más informativa. Los perfiles de prueba procedentes de la rutina de evaluación se procesan de nuevo para demostrar las mejoras de rendimiento de la precaracterización de plataforma. Las distribuciones de distancia mínima desde los perfiles de prueba transformados hasta la firma en el nuevo espacio transformado de LDA se muestran en la figura 25.

Detección de desviaciones de la ejecución de software autorizada en plataformas de radio controladas por software y otros sistemas incorporados

Puede monitorizarse el consumo de potencia dinámico de un procesador para determinar si corresponde a la ejecución esperada o se ha producido una desviación.

Descripción de plataforma

La plataforma objetivo a modo de ejemplo para ilustrar este uso de toma de huellas digitales de potencia (PFP) es una radio controlada por software, en la que la configuración específica del comportamiento de radio se controla mediante software. En la figura 26 se muestra un diagrama de bloques genérico de la plataforma prevista.

En esta plataforma, el comportamiento y la configuración del transceptor 2610 de RF se controlan mediante el procesador 2620. La aplicación 2626 representa la capa más alta e implementa la funcionalidad prevista para el procesador. Con el fin de interaccionar eficazmente con el transceptor 2610 de RF, hay un conjunto de interfaces 2624 de programa de aplicación (API) que retira la complejidad de las interacciones con el hardware a la aplicación principal. Estas API, junto con los controladores requeridos y la implementación 2622 de pila de protocolo, proporcionan un paquete de soporte de placa para el transceptor específico. La pila 2622 de protocolo dispone los datos que van a transmitirse en el formato previamente dispuesto, añadiendo cabeceras requeridas y preparando la carga de modo que el receptor previsto puede extraer la información. También se encarga de extraer la información recibida de dispositivos remotos y presentarla a la capa 2626 de aplicación. El módulo 2612 criptográfico puede implementarse en el transceptor de RF o en software como parte de la pila de protocolo. La figura 26 lo muestra como parte del transceptor. La ubicación del módulo 2612 criptográfico no presenta ninguna diferencia práctica en el enfoque. Las capas de MAC 2614 y de PHY 2616 del transceptor 2610 de RF se encargan del acceso a medios y la transmisión y recepción físicas de la información.

El enfoque descrito caracteriza la ejecución del software 2626 de aplicación, en particular la ejecución de las solicitudes 2624 de API que tienen un impacto sobre el comportamiento del módulo 2612 criptográfico. En este enfoque, el código específico ejecutado como resultado de una solicitud de API se usa para determinar si se usó cifrado y el tipo de cifrado usado. Por ejemplo, si la aplicación exige una clase específica de transmisión cifrada, este enfoque evalúa la ejecución del código que invoca el cifrado. En caso de manipulación indebida maliciosa o accidental, este enfoque proporciona un indicador fiable de la modificación.

Extracción de firmas

Los perfiles de longitud L captados durante la i-ésima ejecución de código autorizado a se representan mediante

$$r_a^{(i)}[n], \quad n = 0, \ldots, L-1$$

Con el fin de evitar la posible interferencia a baja frecuencia de otros componentes de placa, se introduce un filtro de paso alto, no multiplicado, básico, calculando la diferencia entre muestras de perfil

$$d_a^{(i)}[n] = r_a^{(i)}[n] - r_a^{(i)}[n-1]$$

Se usan varios perfiles captados de la ejecución del código autorizado para crear una firma, la huella digital objetivo. Se calcula el promedio de N perfiles para formar la firma objetivo y reducir los efectos de ruido aleatorio en los perfiles individuales

$$s_a[n] = \frac{1}{N}\sum_{i=0}^{N-1} d_a^{(i)}[n], \quad n = 0, \ldots, L-1$$

Extracción de características

El procedimiento de extraer características discriminatorias consiste en una simple correlación de dominio de tiempo frente a la firma objetivo. Sin embargo, la correlación se realiza con j > 0 secciones parciales de la firma y el perfil, cada sección tiene una longitud w = suelo {L}. Esta correlación parcial se realiza para evitar propagar posibles diferencias en los perfiles de potencia a través de una correlación de perfil completo.

La correlación cruzada para diferentes retardos de muestras, 0 ≤ k ≤ w, de sección j de los perfiles viene dada por:

$$\rho_{s_a d_b^{(i)}}(j,k) = \frac{1}{(w-1)\sigma_s \sigma_d} \sum_{n=(j-1)w}^{jw} s_a[n] d_b^{(i)}[k+n] - w\bar{s}\bar{d}$$

donde $\bar{s}$ y σ_s son la media y la desviación estándar de la muestra de la sección correspondiente en s_a y $\bar{d}$ y σ_d son la media y la desviación estándar de la muestra del seccionamiento correspondiente $d_b^{(i)}$.

Con el fin de compensar cualquier desviación de reloj, se mantienen los valores de correlación máxima para diferentes retardos. Esta acción reduce la dimensionalidad de los perfiles a sólo una secuencia de j valores de correlación de picos para cada perfil:

$$\hat{\rho}_{s_a d_b^{(i)}}(j) = \max_k \left\{ \rho_{s_a d_b^{(i)}}(j,k) \right\}$$

En condiciones ideales y con b=a, $\hat{\rho}_{s_a d_b (I)}(j) = 1$ para cada sección j. Cualquier desviación de las características de consumo de potencia se reflejará mediante un factor de correlación reducido.

La característica discriminatoria real o dato estadístico de prueba usada en este trabajo para evaluar perfiles es el valor de correlación de picos mínimo para ese perfil específico

$$x_b^{(i)} = \min_j \{ \hat{\rho}_{s_a d_b^{(i)}}(j) \}$$

$$X_b = x_b^{(i)}; i = 0,\ldots,N-1$$

La variable aleatoria $x_b^{(i)}$ indica la desviación máxima con respecto a la firma de la instancia i del código b. Usando X_b pueden diseñarse detectores apropiados usando diferentes criterios dependiendo de la información estadística que puede recopilarse a partir del sistema a priori.

Respuesta a violaciones de integridad y seguridad en capas

PFP es un enfoque muy eficaz para detectar desviaciones de la ejecución en sistemas cibernéticos. Sin embargo, con el fin de tener una solución completa, es necesario tener una política estructurada para gestionar violaciones de integridad cuando el monitor de PFP detecta una desviación de la ejecución prevista.

Hay tres fases claramente definidas en la seguridad informática:

- Prevención. Incluye mecanismos activos para disuadir, desalentar y prevenir que atacantes lleven a cabo ataques para alterar el sistema, divulgar información, etc.
- Detección. Dado que una prevención absoluta perfecta no es viable, es necesario realizar una monitorización constante de la integridad del sistema
- Respuesta. El conjunto de políticas implementadas para reaccionar frente a ataques satisfactorios.

Ahora se describirá La arquitectura para integrar PFP en un enfoque de seguridad de defensa en profundidad exhaustivo. En este enfoque la PFP proporciona una solución robusta para la fase de “detección” para complementar a varias técnicas diferentes para prevenir y disuadir posibles ataques. La reacción apropiada a diferentes ataques satisfactorios se define en la fase de “respuesta” y se describe según la política de seguridad descrita a continuación.

Aunque lograr una seguridad de sistema requiere un procedimiento y no solo mecanismos o tecnologías aislados, la descripción se centrará en las zonas en las que la PFP puede complementar a los mecanismos de seguridad tradicionales para proporcionar una monitorización continua o intermitente para la estimación de la integridad y la detección de intrusiones. Antes de describir el papel de la PFP, es importante mencionar que el procedimiento de seguridad implica varias etapas, incluyendo:

- Diseño. Seguir enfoques de diseño razonables y diseñar los sistemas para facilitar la implementación de la seguridad, reducir vulnerabilidades, implementar control de acceso, etc. Un ejemplo típico es diseñar el sistema de tal manera que las funciones de seguridad se aíslan del resto de la funcionalidad y en el que se implementan de manera inherente características de control de acceso.
- Desarrollo. Seguir las mejores prácticas de desarrollo para producir productos sostenibles con vulnerabilidades reducidas.
- Despliegue. Asegurarse de que solo se despliegan módulos autorizados. Esto requiere fuertes enfoques de autenticación y sin rechazo.
- Funcionamiento. Mantener un entorno seguro implementando un fuerte control de acceso y otras políticas de seguridad.
- Monitorización. Estimar constantemente la integridad del sistema. PFP, antivirus y sistemas de detección de intrusión en red.
- Respuesta. Definir las políticas y los procedimientos que deben seguirse cuando un ataque es satisfactorio. Deben desarrollarse políticas teniendo en cuenta la criticidad de los sistemas y deben implementarse de manera estricta.

Esta sección describe una arquitectura para integrar un monitor de PFP en una solución de seguridad exhaustiva que incluye mecanismos de seguridad complementarios en los que las vulnerabilidades de una capa quedan cubiertas por la siguiente. Los enfoques y las tecnologías incluidos en las diferentes capas incluyen: cifrado de datos en reposo, fuerte autenticación, control de acceso, resistencia a manipulaciones indebidas, cortafuegos, entornos de pruebas, virtualización y protección física. La arquitectura también proporciona un mecanismo para definir e implementar políticas de seguridad para reaccionar frente a violaciones de integridad detectadas mediante PFP.

La arquitectura define una solución de seguridad en capas en la que un monitor de PFP proporciona una última línea de defensa detectando cuando un intruso consigue penetrar a través de todos los demás mecanismos de defensa. La figura 27 muestra las diferentes capas 2700 en un enfoque de defensa en profundidad. Se pretende que las diferentes capas ralenticen a un adversario y hagan que sea progresivamente más difícil penetrar en una capa de

defensa sin que se detecte. En las capas exteriores están los mecanismos de defensa externos, tales como cortafuegos de Internet, protección física de los equipos y contraseñas y políticas de seguridad (es decir, para prevenir ataques de ingeniería social). Las capas interiores corresponden a diferentes defensas que residen dentro del ordenador 2750 central. Comienzan con el control 2740 de acceso y cifrado de datos en reposo. Continúan con diferentes mecanismos de seguridad previstos para proteger las aplicaciones 2760 y el sistema 2770 operativo. En el núcleo 2780 hay controles para las operaciones de seguridad y kernel más básicas.

La PFP puede monitorizar eficazmente la integridad de diferentes capas. A nivel 2780 de núcleo, la PFP puede estimar la integridad de operaciones de seguridad y kernel de las que dependen todos los demás mecanismos. También puede expandirse para monitorizar la integridad de aplicaciones de núcleo en el sistema 2770 operativo, así como la integridad de aplicaciones 2760 a nivel de usuario críticas. Obsérvese que la PFP puede monitorizar la integridad de todos los módulos que residen dentro del alcance del procesador, incluyendo módulos de antivirus y módulos de cifrado, tal como se muestra en la figura 28.

Integrar PFP en un enfoque de defensa en profundidad para la seguridad cibernética permite una gestión más rápida de posibles incidentes antes de que puedan alcanzar sus objetivos y provocar daños.

Las firmas de potencia de la ejecución de otros módulos de seguridad, tales como cifrado y antivirus, se extraen y evalúan at en el tiempo de ejecución. Desde el punto de vista de la PFP, las firmas de un módulo de kernel y un programa de antivirus se extraen de la misma manera y usando las mismas técnicas.

Es importante observar que la PFP puede extenderse a cualquier otro dispositivo para monitorizar su integridad de ejecución. Esto incluye dispositivos que pueden usarse para implementar diferentes capas de seguridad tales como cortafuegos, cierres de seguridad digital, etc.

La última etapa de ciberdefensa con PFP es definir las políticas apropiadas para gestionar diferentes anomalías detectadas mediante las operaciones de monitorización y estimación. Dado que la reacción apropiada frente a intrusiones satisfactorias depende de varios factores únicos para cada plataforma y aplicación, no puede generalizarse la respuesta frente a diferentes intrusiones detectadas por el monitor de PFP. Por tanto, es necesario seguir una arquitectura que acepte e implemente diferentes definiciones de políticas de seguridad que puedan ajustarse a diferentes sistemas al tiempo que se mantienen y se reutilizan las estructuras y los principios de funcionamiento básicos.

Red de monitor de PFP distribuida para monitorizar la dinámica y comportamiento del malware

Esta sección describe el funcionamiento de una amplia red de nodos con capacidades de PFP que se despliegan a lo largo de diferentes regiones geográficas o lógicas para monitorizar la propagación de malware, detectar ataques dirigidos y descubrir las posibles intenciones de adversarios maliciosos. Este enfoque puede aplicarse para descubrir ataques remotos furtivos en zonas geográficas o lógicas específicas.

Una de las principales ventajas de usar PFP para esta aplicación es su sigilo, que impide que adversarios detecten las propias actividades de monitorización, dándoles un falso sentido de subrepción (creer que no se les ha detectado) y engañarles para que continúen con sus actividades, revelando intenciones y capacidades. Esta aplicación de PFP es una potente herramienta para recopilar información.

Funcionamiento

La monitorización con sigilo se logra gracias a la pequeña huella de PFP y a un impacto despreciable sobre la memoria y latencia en el sistema objetivo. La red distribuida de nodos de PFP se implementa usando las siguientes etapas:

1. Habilitar nodos representativos con PFP (equiparlos con un monitor de PFP y extraer firmas de confianza de sus componentes objetivo). El monitor puede estar montado en rack y con recursos, ya que los nodos objetivo solo actúan como señuelo.

2. Desplegar una red de los nodos habilitados con PFP en las zonas geográficas o lógicas objetivo de interés.

3. Monitorizar cada nodo individualmente para detectar violaciones de integridad e intrusiones tal como se representa en la figura 29.

4. Enviar periódicamente los resultados de integridad a una ubicación central para su registro y análisis.

5. En caso de una violación de integridad, el informe debe incluir:

(a) una copia de los perfiles de potencia que experimentaron la violación

(b) la secuencia ordenada de ejecución de módulos no manipulados de manera indebida que se ejecutó antes de la violación

(c) la secuencia ordenada de módulos que se ejecutan tras la violación.

Esta aplicación de PFP se representa en la figura 30. La figura muestra señuelos de PFP en diferentes redes geográficas. Sin embargo, es importante observar que la separación de redes puede ser lógica, como en diferentes secciones de la misma red, o sociopolíticas, como en redes para diferentes agencias gubernamentales o divisiones empresariales.

Los enlaces entre los señuelos y la ubicación de análisis centralizada, representados en la figura 30 como líneas discontinuas, pueden implementarse como una red separada (por ejemplo enlaces inalámbricos dedicados) o realizarse usando redes de área ancha disponibles, tales como le red telefónica conmutada pública (PSTN) o Internet. En cualquier caso, deben colocarse fuertes mecanismos sin rechazo, encargados de proporcionar pruebas (autenticación de alta garantía) del origen y la integridad de los perfiles, para mantener la confiabilidad en el sistema en su conjunto.

Aplicación de PFP al análisis de confianza de cadena de suministro

Subcontratar la producción y fabricación de dispositivos a fabricantes 3220 y plantas de fabricación de semiconductores externos y de confianza abre la puerta a posibles manipulaciones indebidas e infracciones de seguridad. Incluso con proveedores de confianza, existe la posibilidad de que personal externo o disgustado intente alterar el funcionamiento y la funcionalidad de sistemas críticos.

La PFP proporciona un mecanismo 3210 para detectar modificaciones no autorizadas y otras manipulaciones indebidas en software, firmware y hardware introducido por enlaces que no son de confianza en la cadena de suministro a lo largo de todo el ciclo de vida del sistema. La estimación de integridad de nuevos envíos y dispositivos que no son de confianza usando PFP requiere las etapas mostradas en la figura 31. Se usa un generador 3110 de vector de entrada para proporcionar las entradas necesarias para la ejecución del dispositivo objetivo en un entorno 3120 controlado durante el cual se recopilan 3130 perfiles de potencia. Los parámetros de las características individuales del consumo de potencia se compensan 3140 antes de realizar la extracción 3150 de características. Las características resultantes se comparan 3160 con firmas 3170 de referencia almacenadas y a partir de esta comparación se obtiene el resultado de estimación final.

Detectar la integridad de dispositivos digitales usando PFP no es un procedimiento destructivo y solo requiere una colaboración mínima por parte del dispositivo que está evaluándose. Además, las medidas refinadas del consumo de potencia proporcionan visibilidad significativa de la situación de ejecución interna del dispositivo, haciendo que sea extremadamente difícil que una modificación pase desapercibida. Por ejemplo, la PFP puede detectar una manipulación indebida que solo se activa en determinadas condiciones (también conocidas como bombas lógicas y de tiempo) debido a la activación parcial de la funcionalidad adicional o al flujo de ejecución durante comprobaciones de estado. La capacidad de la PFP de detectar funcionalidad adicional o que falta no depende del propósito o las intenciones de las propias inserciones.

Otra ventaja de la PFP es que puede confiarse en una trayectoria de ejecución específica verificada con PFP aunque no se desencadene actividad maliciosa. En otras palabras, si la PFP no detecta una desviación significativa de las firmas, significa que no se ha producido ninguna manipulación indebida o funcionalidad adicional en esa trayectoria de ejecución particular.

Un elemento clave en la realización de análisis de confianza de cadena de suministro con PFP es ejecutar el dispositivo que no es de confianza en un entorno 3120 controlado. Este entorno controlado incluye entradas 3110 predefinidas que fuerzan una secuencia de estado específica y, para dispositivos programables, el software específico que va a ejecutarse. Para algunos sistemas puede ser necesario desarrollar andamiaje de soporte para controlar y aislar la ejecución de componentes específicos. Los vectores de entrada específicos dependen de la funcionalidad del dispositivo o módulo de software y se espera que realicen las trayectorias de ejecución críticas para el funcionamiento del dispositivo. Se necesita usar los mismos vectores de entrada usados para extraer las firmas, para estimar la integridad de los dispositivos que no son de confianza.

Debido a ligeras variaciones de procedimiento durante la fabricación, diferentes dispositivos mostrarán diferentes características de consumo de potencia. Se necesita compensar 3140 estas variaciones en consumo de potencia antes de la extracción 3150 de características para evitar estimaciones erróneas. Esta compensación se realiza por medio de un filtro adaptativo tomas se modifican de manera dinámica para coincidir con la característica específica de los perfiles de consumo de potencia. Este filtro adaptativo permite que el monitor de PFP se concentre en el consumo de potencia resultante de transiciones de bits en el registro de dispositivo durante la ejecución y elimine diferencias en los perfiles debidas a variaciones de fabricación.

El aspecto más crítico para un análisis de confianza de cadena de suministro eficaz usando PFP es la disponibilidad

de firmas 3170 de referencia. Hay diferentes fuentes posibles para tales firmas tal como se representa en la figura 32. La mejor referencia se proporcionará por una implementación 3230 de confianza idéntica (un patrón de referencia). Sin embargo, en muchas ocasiones tal implementación de confianza no está disponible. En estos casos puede extraerse una firma de referencia usando métodos alternativos con diversos grados de error y confiabilidad. Por ejemplo, dos fuentes de referencia alternativas relativamente sencillas incluyen un implementación 3250 anterior del dispositivo (una que se ha sometido a prueba en el tiempo) o una implementación alternativa a partir de un proveedor 3260 diferente. En estos casos, las firmas se extraen a partir de la ejecución de las implementaciones alternativas, reduciendo las posibilidades de dos modificaciones idénticas por proveedores diferentes. Las firmas del primer enfoque pueden no detectar modificaciones no identificadas presentes en la versión anterior. En el último enfoque, un atacante puede crear una modificación idéntica en ambas versiones a partir de los diferentes proveedores para evitar detección.

Usar un modelo 3240 de CAD para obtener las firmas requiere más esfuerzo, pero puede realizarse de manera interna sin basarse en plantas de fabricación de semiconductores externas. Con el fin de extraer las firmas usando un modelo de CAD es necesario simular la ejecución del dispositivo usando vectores de entrada deterministas. Se necesita que el simulador sea preciso en cuanto al consumo de potencia para registrar el nivel de transferencia.

Gestión de derechos digitales y alquileres limitados de ejecución

Otra aplicación novedosa para PFF es la implementación de derechos digitales y la creación de alquiler limitado en cuanto al número de ejecuciones para permitir alquileres basados en el número de ejecuciones.

Este enfoque se implementa extrayendo firmas de la ejecución de software protegido y monitorizando en el tiempo de ejecución las huellas digitales de potencia para implementar la ejecución exclusiva de módulos autorizados. Por ejemplo, puede licenciarse un sistema de software para incluir solo un conjunto de módulos funcionales con un subconjunto de los módulos reservados para un nivel de licencia superior. Las huellas digitales de todos los módulos se extraen antes de la liberación. En el momento de la ejecución un monitor de PFP hace coincidir la ejecución de diferentes módulos con las licencias autorizadas. Cuando se ejecuta un módulo no licenciado, como resultado de una contraseña robada o una infracción en la protección, el monitor de PFP puede informar a la agencia emisora sobre la violación. Además, es posible habilitar un enfoque de alquiler limitado en cuanto al número de ejecuciones de confianza para software protegido. En este caso, el monitor de PFP lleva la cuenta del número de veces que se ha ejecutado el software licenciado e informa a la agencia emisora cuando el alquiler ha caducado.

Puede tomarse un enfoque similar para contenido multimedia licenciado. Usando un monitor de PFP, es posible detectar la reproducción de archivos específicos en reproductores multimedia conocidos usando PFP. En este caso, los datos multimedia protegidos ocupan el lugar de la entrada predeterminada durante la caracterización de PFP. Si se reproducen los mismos datos multimedia en el reproductor específico, las firmas de potencia coincidirán. Por tanto, puede usarse la PFP para detectar la reproducción de datos multimedia licenciados no autorizados.

Predicción de falla basada en PFP

Los componentes de hardware experimentan un proceso de envejecimiento inevitable, que se ve acelerado por el funcionamiento en entornos hostiles o cuando los sistemas funcionan con estrés ambiental continuo. Este envejecimiento se ve reflejado en las características de consumo de potencia de la plataforma. Puede usarse la PFP para monitorizar no solo la correcta ejecución de software sino también la integridad de las plataformas de hardware. Un monitor de PFP puede rastrear de manera continua las características de consumo de potencia del hardware y predecir fallas antes de que se produzcan realmente, dictando cuándo debe sustituirse un sistema o elemento específico.

El rastreo de las características de consumo de potencia en PFP se implementa usando un filtro adaptativo. Es necesario compensar diferencias en el consumo de potencia con respecto a cuándo se extraen las firmas o debidas a condiciones del entorno. El mismo mecanismo de rastreo puede usarse para monitorizar la situación del hardware y comparar las características de consumo de potencia con patrones predeterminados captados en pruebas en laboratorio de los dispositivos. El procedimiento para identificar las características de falla se representa en la figura 36. En este procedimiento, puede lograrse un envejecimiento 3610 acelerado exponiendo el dispositivo objetivo a cambios de temperatura abruptos. El procedimiento de caracterización tiene lugar a intervalos, con una ronda de envejecimiento acelerado seguida por captación 3620 de perfil durante la ejecución de una rutina de prueba. Se recopilan los perfiles para su análisis posterior y se repite el procedimiento hasta que el dispositivo falla. Una vez que el dispositivo falla se examina el conjunto de perfiles para determinar las características específicas que se muestran antes de la falla 3630. Se extraen las características de otros dispositivos similares para proporcionar diversidad estadística y aislar las características 3640 genéricas.

Incorporación de información de identificación de módulo en la señalización de sincronización

La PFP requiere una sincronización apropiada con el software que está ejecutándose con el fin de proporcionar una correcta estimación. Hay dos niveles de sincronización en PFP: nivel de ciclo de reloj y nivel de rutina. El primero

puede lograrse fácilmente rastreando los distintos ciclos en el consumo de potencia que se producen a la velocidad de ciclo de reloj o, para plataformas sencillas, analizando con sonda la propia señal de reloj. La segunda sincronización es más difícil de lograr y el procedimiento se facilita incorporando en la propia rutina un desencadenante, o identificador, que informa al monitor 3210 de PFP sobre la ejecución de una rutina específica.

En esta sección se presenta un mecanismo para incorporar una identificación del nodo que está ejecutándose en el mecanismo de desencadenamiento y señalización. Este mecanismo no solo ayuda a informar al monitor de PFP de qué rutina específica está a punto de ejecutarse, sino que también proporciona una señalización de sincronización robusta para una detección de anomalías más precisa y extracción de firmas de comportamiento.

El objetivo final es proporcionar un código de identificación para los diferentes módulos que están caracterizándose que se inserta en los artefactos de sincronización y desencadenamiento para PFP. Hay dos enfoques principales para proporcionar señalización de sincronización e identificación para PFP: 1) crear una señal física adyacente, tal como se muestra en la figura 33, y 2) incorporar una señal en el propio consumo de potencia tal como se muestra en la figura 34. Para el primero, se escribe un código de identificación binario en un registro 3324 de IO físico de un procesador 3320 antes de la ejecución de la rutina 3322. Entonces se transmite 3335 el registro al monitor 3340 de PFP, que capta los perfiles 3315 de potencia del sensor 3310, de manera o bien en paralelo o bien en serie. La longitud del código y registro depende del número de rutinas que se necesita monitorizar. En el sentido más sencillo, puede usarse un registro de un único bit, tal como un LED, para señalizar la ejecución de la rutina objetivo. En el caso de una señalización física separada el desencadenante se codifica como un número binario en el registro de señalización, tal como se muestra en la figura 33.

El segundo enfoque requiere incorporar la señalización de sincronización en el propio consumo de potencia insertando una secuencia de instrucciones 3422 cuidadosamente elaborada que proporciona un patrón de consumo de potencia distintivo. Este enfoque se representa en la figura 34. Las instrucciones en las rutinas de sincronización se eligen de tal manera que las transiciones de bits en sus palabras de código, direcciones y parámetros proporcionan un número específico de transiciones de bits que impulsan en última instancia el consumo de potencia y señalizan al monitor 3340 de PFP que una secuencia específica está a punto de ejecutarse para captar el conjunto correcto de perfiles 3415 procedentes del sensor 3410. Más transiciones de bits dan como resultado un drenaje de corriente mayor. Cuando se desarrolla la secuencia se necesita tener en cuenta la longitud y las características de la canalización. De manera similar al enfoque anterior, la longitud de la secuencia de instrucciones (código) depende del número de rutinas críticas que se necesita identificar. Creando diferentes patrones de consumo de potencia distintos, se elige la propia secuencia para proporcionar diferentes códigos de firmas usados para identificar diferentes módulos.

Es importante observar que la señalización de sincronización es un elemento requerido para una PFP eficaz, ya que permite concentrar los esfuerzos de estimación en las secciones del código que son más importantes. Incorporar un código de identificación en las instalaciones de señalización facilita el procedimiento de estimación, pero no es un requisito necesario. Esto se debe a que usar un único desencadenante permitirá al monitor de PFP captar el conjunto correcto de perfiles y pueden usarse técnicas de clasificación de señales para determinar qué rutina específica se ejecutó o si no puede establecerse ninguna coincidencia fiable (una anomalía).

Monitorización de PFP perfeccionada combinando señales de diferentes elementos de placa

Un monitor de PFP puede usar señales de diferentes elementos del sistema y combinarlas para proporcionar rendimiento y fiabilidad mejorados. Las fuentes de múltiples señales incluyen múltiples procesadores, coprocesadores, periféricos u otros elementos de uso especial introducidos con el único propósito de potenciar la PFP (por ejemplo los registros de IO usados para el desencadenamiento).

Hay diferentes maneras de combinar señales a partir de diferentes fuentes en PFP. Uno de los enfoques principales incluye captar perfiles de potencia a partir de diferentes procesadores u otros circuitos digitales para realizar la estimación de integridad en placas de múltiples procesadores y múltiples núcleos. Otro enfoque es monitorizar otros elementos de los sistemas (consumo de potencia u otros canales laterales y directos) para recopilar información de contexto adicional que va a usarse durante la estimación de integridad. La información de contexto adicional puede usarse para mejorar la sincronización y facilitar la caracterización de comportamiento. La información de contexto puede generarse como producto de funcionamiento de sistema normal o introducirse de manera deliberada en el momento del diseño (por ejemplo los registros de IO usados para el desencadenamiento). En la figura 35 se representa una configuración de muestra de un monitor de PFP que combina múltiples señales.

Pueden captarse señales adicionales a partir de registros de IO de soporte directos, a partir del consumo de potencia de diferentes elementos o a partir de otros canales laterales tales como radiación electromagnética. Combinar señales de diferentes fuentes requiere un detector especialmente diseñado que pueda soportar las diferentes características. Los mecanismos de combinación específicos dependen de la funcionalidad de sistema y la plataforma de soporte. Por ejemplo, en un procesador de múltiples núcleos, pueden explorarse perfiles de potencia de cada núcleo con el fin de encontrar los perfiles correspondientes a una rutina objetivo. Otro ejemplo, en una radio definida por software, la activación del amplificador de potencia (PA) puede detectarse monitorizando el

consumo de potencia y se produce cuando está teniendo lugar una transmisión de radio. La activación del PA puede usarse como mecanismo desencadenante para las rutinas implicadas en la preparación de los datos que van a transmitirse (obsérvese que, en este caso, las rutinas se ejecutan antes de que se produzca el desencadenante).

Uso de firmas de malware para potenciar el rendimiento de PFP

Aunque la aplicación principal de la PFP es la detección de anomalías, hay importantes beneficios en el uso de información disponible a partir de malware conocido para mejorar el rendimiento de la estimación. Cuando se identifica una nueva tendencia de malware, es posible extraer su firma de PFP y añadirla a la biblioteca de firmas conocidas. Estas firmas de malware pueden usarse para perfeccionar el rendimiento de la estimación de integridad de PFP proporcionando detección basada en firma tradicional de malware instalado, de manera similar al software antivirus tradicional. Se necesitará hacer que el monitor conozca la naturaleza individual de cada firma (lista blanca y lista negra) con el fin de evitar evaluaciones incorrectas. También pueden extraerse firmas de malware a partir de patrones de comportamiento en la ejecución. Por ejemplo, determinados tipos de malware, tales como ataques por agotamiento, tienen patrones de ejecución muy distintivos que pueden identificarse fácilmente usando PFP.

El procedimiento de extraer firmas a partir de malware es similar al procedimiento de extraer firmas a partir de software de confianza, en el que los módulos objetivo se ejecutan repetidamente en un entorno controlado y se aplican técnicas de procesamiento de señales diferentes a los perfiles de potencia resultantes para seleccionar las características con las mejores propiedades discriminatorias. Es importante observar que la caracterización de malware se facilita una vez que se ha identificado, aislado y ejecutado el malware en un entorno controlado.

Caracterización automática y extracción de firma

Con el fin de caracterizar eficazmente un nuevo sistema de software, o una nueva versión de un sistema existente, es necesario tener herramientas para caracterizar automáticamente una referencia de confianza y extraer las firmas de PFP que identifican de manera única la ejecución de ese software específico. En un sentido, este procedimiento es similar a las pruebas automáticas porque requiere la ejecución de módulos específicos en condiciones controladas. Sin embargo, a diferencia de las pruebas automáticas, la caracterización de PFP solo se preocupa de "observar" varias instancias de ejecución de diferentes módulos y no intenta evaluar ningún requisito o propiedad.

El propósito de esta sección es describir un enfoque para facilitar la caracterización de sistemas complejos y arquitecturas de software y hacer que sea viable extraer la firma a partir de implementaciones realistas de sistemas cibernéticos de complejidad práctica. Sin este enfoque automático, se tardaría demasiado en caracterizar y extraer las firmas únicas a partir de sistemas complejos (es decir, sistemas comerciales) para usarlas en la toma de huellas digitales de potencia.

El objetivo principal es automatizar el procedimiento de caracterización para los diferentes módulos usando un andamiaje similar a lo que se usa normalmente en las pruebas de software, así como usando una variedad de análisis estadísticos y procesamiento de señales para identificar las mejores características discriminatorias que forman las huellas digitales. El procedimiento comienza cuando se necesita caracterizar una nueva pila de software. Las herramientas necesarias para este procedimiento incluyen: descriptores de módulos críticos, herramientas de procesamiento de señales para la extracción de características, herramientas de diseño de detector, andamiaje para ejecución de módulos (similar al andamiaje de pruebas), generadores de vectores de entrada, generación de informe, y empaquetamiento de firmas. Con el fin de facilitar la comprensión del enfoque, se proporciona una vista de alto nivel del procedimiento que describe los detalles y las interrelaciones entre los diferentes subsistemas. Las relaciones se representan en la figura 37.

- Los descriptores incluyen información requerida sobre los módulos críticos, incluyendo identificadores únicos, dependencias, análisis de entradas (desglose de diferentes clases de entradas), modo de ejecución (relacionado dinámicamente, prioridad, módulo de kernel, etc.).

- La información de los descriptores se usa para implementar los andamiajes para controlar la ejecución aislada de los módulos objetivo. Los andamiajes permiten que el sistema introduzca valores deterministas como entradas para controlar la ejecución de los módulos.

- La información en los descriptores sobre la funcionalidad y los diferentes tipos de entradas se usa para determinar un conjunto adecuado de vectores de entrada.

- Se realiza un análisis de cobertura para identificar las trayectorias de ejecución que se han puesto en práctica, produciendo una medida del nivel de protección para el sistema.

- Una vez cargado el sistema, el operario (que puede ser un sistema automático) ejecuta los diferentes módulos con soporte de los andamiajes y proporcionando los vectores de entrada apropiados. Mientras están ejecutándose los módulos, el monitor de PFP capta medidas del consumo de potencia.

- Entonces se procesan los perfiles de potencia captados por el monitor usando diferentes técnicas de procesamiento de señales para extraer características discriminatorias. Hay un conjunto predefinido de características que van a extraerse para cada componente en diferentes campos y usando diferentes técnicas.

- Tras captarse varios perfiles y analizarse las características respectivas, se realiza un análisis estadístico para diseñar detectores óptimos para distinguir la actividad normal de anomalías basándose en los requisitos específicos para la aplicación.

- Entonces se empaquetan juntos firmas y detectores para desplegarse junto con los monitores que estimarán la integridad de los sistemas objetivo.

Las siguientes secciones incluyen descripciones más detalladas necesarias para implementar satisfactoriamente el enfoque descrito anteriormente.

Descriptores

Los descriptores contienen metainformación sobre los módulos específicos que van a caracterizarse. Se usan para desarrollar artefactos de andamiaje para aislar la ejecución de módulos individuales y proporcionar un entorno controlado para poner en práctica las diferentes trayectorias de ejecución.

Se espera que los descriptores proporcionen en un lenguaje de marcado que leen fácilmente personas y máquinas, tal como el lenguaje de marcado extensible (XML), pero el contenido, lenguaje y estructura dependerán de las herramientas específicas usadas para automatizar el procedimiento de caracterización y pueden ser privados.

La información mínima requerida que se necesita que esté contenida en un descriptor de módulo para la caracterización de PFP incluye:

- Identificadores únicos para describir cada módulo. Los identificadores únicos deben poder leerse por personas y proporcionar la información necesaria para ubicar de manera única el módulo en cuestión. Los elementos en la parte legible por personas incluye empresa, producto, clase, módulo y versión.

- Dependencias. Las dependencias de software y hardware requeridas del módulo.

- Dependencias de estado. Los elementos del estado interno que afectan al comportamiento del módulo y que se necesita controlar para proporcionar una ejecución consistente y determinista.

- Análisis de interfaz. Proporciona un desglose de las diferentes clases de entradas y las clases de entradas requeridas para poner en práctica las diferentes trayectorias de ejecución.

- Modo de ejecución. Describe en qué modo se ejecutará el módulo cuando se despliegue, es decir, estático, para módulos conectados de manera estática; dinámico, para módulos conectados de manera dinámica; kernel o modo protegido, para el modo de funcionamiento que adoptará el procesador cuando ejecute el módulo; y nivel de prioridad.

Generadores de vectores de entrada

La función de los generadores de vectores de entrada es similar a sus equivalentes en las pruebas de software, proporcionar las entradas apropiadas para forzar que el componente entre en una secuencia de estado específica que incluye las diferentes trayectorias de ejecución. Sin embargo, a diferencia de las pruebas, el objetivo para los vectores de entrada de PFP no es encontrar errores de implementación, sino simplemente poner en práctica las diferentes trayectorias de ejecución.

Dependiendo de la naturaleza del sistema objetivo, algunas veces será necesario almacenar los vectores de entrada y distribuirlos junto con firmas para su utilización durante la estimación (es decir, auditoría de integridad). La decisión de si mantener los vectores de entrada depende de la naturaleza de las características seleccionadas y de si pueden eliminarse perfiles debidos a entradas aleatorias.

Los vectores de entrada pueden generarse usando diferentes técnicas, incluyendo enfoques basados en búsquedas (búsqueda aleatoria, ascenso de colinas, algoritmo genético, etc.), exploración parcial, programación lineal y enfoques aleatorios y pseudoaleatorios.

Sin embargo, la identificación real de vectores de prueba eficaces sigue siendo un enfoque ampliamente heurístico que depende de la funcionalidad específica del módulo objetivo y su campo de entrada, así como la información disponible sobre la estructura del módulo. Habrá algunos casos en los que se necesite conocimiento específico de la estructura de ejecución del módulo (qué trayectorias de ejecución existen y las secuencias de estado necesarias para ejecutarlas) para encontrar vectores de entrada significativos dentro de un plazo de tiempo razonable. Además,

algunas veces puede requerirse la entrada directa de un análisis por parte de un experto para proporcionar directrices a las herramientas automáticas con el fin de identificar y generar vectores de prueba eficaces, significativos.

Un elemento clave de la generación de vectores de prueba para PFP es que el objetivo es ejecutar las diferentes trayectorias que se espera que se produzcan una vez desplegado el dispositivo, no encontrar errores. Es un enfoque relativamente peligroso, porque puede alcanzarse un estado de ejecución válido que no se ha caracterizado y, por tanto, se indica como una anomalía. La ventaja es que reduce el espacio de búsqueda a solo unos pocos estados. Para los sistemas más críticos, el espacio de ejecución es relativamente pequeño y los estados de ejecución previstos son un subconjunto.

Informe de cobertura

Usando la información procedente del generador de vectores de entrada es posible generar un informe de cobertura basado en las trayectorias de ejecución atravesadas por los vectores de entrada específicos. Usando información estructural de los módulos objetivo, es posible calcular una medida de cobertura de PFP como porcentaje de las trayectorias existentes en el módulo y las atravesadas usando los vectores de entrada generados. Este informe sólo es una indicación de la cobertura esperada para PFP. Todavía se necesita completar el informe identificando el número de trayectorias de ejecución que proporcionaron realmente firmas de PFP aceptables.

El informe se proporciona al final para proporcionar a los usuarios la información sobre los módulos específicos que pueden monitorizarse usando PFP.

Andamiaje

Con los descriptores y la pila de software, se realiza el procedimiento de andamiaje para aislar la ejecución de los módulos críticos y sus diferentes partes. Esto es similar al procedimiento de andamiaje para pruebas automáticas. El propósito del andamiaje es ejecutar los módulos objetivo en un entorno controlado similar al que se encontrará una vez desplegado el sistema completo con el fin de recopilar los perfiles de potencia durante su ejecución. Dado que se espera que los módulos tengan diferentes trayectorias de ejecución que dependen de las entradas, se necesita que los andamiajes faciliten el uso de diferentes entradas.

Para el caso en el que se necesitan entradas físicas, se necesita que los andamiajes proporcionen las interfaces físicas apropiadas para proporcionar las entradas necesarias.

Esto es un procedimiento parcialmente manual y depende de las características de los módulos objetivo. Afortunadamente, la mayor parte de los elementos necesarios para los andamiajes se solapan en cuanto a la funcionalidad con los andamiajes tradicionales para pruebas automáticas (por ejemplo, pruebas de unidad, integración y sistema), añadiendo solo un poco de trabajo adicional.

Es importante observar que para implementaciones que no son de software, el andamiaje tendrá requisitos similares, aunque la implementación final será diferente. En estos casos, los módulos estarán limitados por las secciones que pueden ponerse en práctica de manera independiente. Para sistemas altamente integrados, esto puede representar un desafío.

Procesamiento de señales y extracción de características

Con los perfiles de potencia correspondientes a la ejecución de los diferentes módulos y sus trayectorias de ejecución individuales captados usando el sensor de potencia/corriente instantánea, se necesita extraer las características discriminatorias que identifican de manera única la ejecución del módulo objetivo. El conjunto exacto de técnicas y análisis de señales necesario para identificar firmas prácticas depende de las características específicas de los módulos objetivo.

Simplemente se describe un marco para la ejecución en paralelo de varias técnicas de extracción de características y procesamiento de señales diferentes para reducir el tiempo global requerido para caracterizar un módulo objetivo.

No hay ningún procedimiento eficaz conocido para determinar las características discriminatorias óptimas para un problema dado. Sin embargo, hay varias técnicas que pueden evaluarse y a partir de las cuales se seleccionan las mejores características discriminatorias. El conjunto de características discriminatorias que se extraen se determina usando una combinación de enfoques heurísticos y experiencia. Entre estas características se incluyen: correlación de dominio de tiempo, distancia euclidiana, análisis cicloestacionario, análisis de frecuencia, etc. El procedimiento para seleccionar las mejores características discriminatorias incluye calcular todas las características diferentes en el conjunto en paralelo y clasificarlas basándose en varianza dentro de la clase. La distancia de Mahalanobis es una medida de muestra para una evaluación de características de este tipo.

Los procedimientos de selección de características y diseño de detector, explicados a continuación, están

estrechamente relacionados, ya que las propiedades estadísticas de los resultados de extracción de características determinan el análisis necesario para determinar un umbral de detección óptimo.

Análisis estadístico y diseño de detector

Se realiza un análisis estadístico con las diferentes características obtenidas a partir de los perfiles de potencia captados durante instancias de ejecución independientes del módulo objetivo. El objetivo del análisis estadístico es seleccionar las características con las mejores calidades discriminatorias y determinar los niveles de umbral, o áreas dentro de las cuales se considerará que un conjunto observado de características se ha generado por el módulo objetivo (un detector).

En PFP, que es un enfoque de detección de anomalías, la probabilidad de falsa alarma (PFA) es una medida de funcionamiento importante que determina el rendimiento del sistema. La PFA se define como la probabilidad de que una instancia de ejecución normal del módulo objetivo presente un fallo fuera del área de aceptación y se clasifique como una anomalía. Se necesita diseñar un detector de PFP para minimizar la PFA al tiempo que se maximiza la probabilidad de identificar correctamente el módulo objetivo. Este es un problema de pruebas de hipótesis clásico y puede aplicarse el criterio de Neyman-Pearson para detectar un umbral. Sin embargo, hay varios otros enfoques que pueden aplicarse.

Dadas suficientes muestras, puede lograrse una PFA arbitraria en PFP. Sin embargo, en sistemas prácticos esto no es factible y debe determinarse un nivel de PFA práctico, finito. La PFA que puede tolerarse depende del módulo específico y la naturaleza de la aplicación en la que se espera que funcione.

De manera ideal, se necesita que firmas de diferentes instancias de la misma ejecución se encuentren dentro de la distancia para sensibilidad mínima calculada durante la caracterización de plataforma. En el caso de que no pueda lograrse esta característica deseada, hay varias maneras de hacer que la PFP proporcione estimaciones precisas. Un enfoque sencillo es calcular el promedio de varios perfiles para desprenderse de algo de ruido.

Empaquetamiento y cifrado de firmas

Una vez que se han caracterizado los módulos objetivo, se empaquetan las firmas resultantes, técnicas de extracción de características, y umbrales para su despliegue junto con los dispositivos. El mecanismo de empaquetamiento y entrega depende de las características del dispositivo y la aplicación. Se necesita almacenar las firmas completas extraídas usando las características seleccionadas y pasarlas a los monitores. Por ejemplo, en el caso de correlación de dominio de tiempo simple, se necesita almacenar el vector completo.

Con el fin de proteger las firmas en reposo o durante el transporte, es necesario cifrarlas para evitar dar a posibles atacantes una referencia exacta de las firmas que está buscando el monitor. Este cifrado puede realizarse usando una variedad de mecanismos para cifrado con clave privada o pública. Sin embargo, es importante observar que aunque un posible atacante adquiera las firmas, todavía será muy difícil hacer coincidir las firmas perfectamente al tiempo que se lleva a cabo un comportamiento malicioso.

Actualización de firmas segura

Cundo se actualiza un sistema desplegado que está monitorizándose usando PFP, también es necesario actualizar las firmas de PFP de una manera fiable y segura con el fin de mantener una estimación de integridad eficaz. Esta es una etapa crítica, ya que la confiabilidad de la estimación depende de la gestión apropiada de firmas. Para que este procedimiento de actualización sea seguro, es necesario verificar la integridad y autenticidad de la firma. En esta sección se describe el mecanismo necesario para proporcionar una actualización de firmas de PFP segura.

Para monitores de PFP ampliamente desplegados, las firmas deben distribuirse junto con otras actualizaciones de software. Para monitores de PFP centralizados, pueden entregarse actualizaciones por separado de las actualizaciones de software. El principal desafío en la actualización de firmas de PFP es la autenticación (es decir, asegurarse de que el remitente es una entidad autorizada y de que la propia firma es correcta y no se ha manipulado de manera indebida o alterado de ninguna manera). El desafío no es tan difícil en el caso de monitores de PFP centralizados, en los que pueden distribuirse firmas usando medios físicos o redes de confianza y en los que pueden realizarse preparaciones previamente a la transferencia de firmas confidenciales.

En el caso de monitores de PFP ampliamente distribuidos, en los que el intercambio de firmas no puede realizarse usando medios físicos o redes de confianza alternativas, se necesita realizar la actualización de firmas junto con la actualización de software real. En este caso, hay varios puntos vulnerables que puede aprovechar un atacante con suficiente conocimiento del sistema de PFP. Por ejemplo, si no se autentica de manera apropiada, el procedimiento de actualización puede alterarse mediante un ataque de intermediario.

Procedimiento y operación de actualización de firmas segura.

Enfoques conocidos para una distribución de contenido segura usados comúnmente en la programación por el aire y distribución de actualizaciones de software pueden adaptarse para actualizaciones de firmas de PFP. La actualización de firmas segura puede considerarse desde dos puntos de vista diferentes: el originador de firma auténtica y el monitor de PFP. Desde el lado de la generación de firma, es necesario proporcionar información de autenticación eficaz junto con la firma y cifrar los fragmentos de firma con un esquema de clave rotatorio.

Otras técnicas que pueden aplicarse para permitir una actualización de firma segura incluyen:

- Aleatorizar tanto el tampón de muestra como los elementos de firma
- Cifrado con clave pública o simétrica
- Cambiar la clave de cifrado que va a usarse para desaleatorizar la firma y perfiles según una secuencia conocida (secuencia de PN) que actualiza su índice tras cada actualización de firma.

Protección contra ataques de canal lateral

La PFP usa los mismos principios para la estimación de integridad que aprovechan los ataques de canal lateral maliciosos. Por tanto, con el fin de impedir que posibles adversarios aprovechen la infraestructura de PFP para realizar ataques de canal lateral, es necesario proteger los perfiles proporcionados por el sensor limitando el acceso a los mismos. Esto es especialmente importante cuando los perfiles de potencia se transmiten usando una conexión inalámbrica. Esta sección describe un mecanismo para proteger el acceso no autorizado a perfiles de potencia, de los que puede hacerse un mal uso en ataques de canal lateral.

Funcionamiento

La protección para el acceso a perfiles se logra cifrando o aleatorizando los perfiles usando una clave compartida entre el sensor de PFP y el monitor de PFP. Desde este punto de vista, hay dos modos básicos de funcionamiento para PFP: monitor incorporado (sensor y digitalizador) y monitor externo.

En funcionamiento incorporado, se cifran o aleatorizan perfiles con una clave privada fuerte (cifrado con clave simétrica). Realizar esta etapa de cifrado es especialmente importante cuando los perfiles de potencia se transmiten de manera inalámbrica para su procesamiento externo. El procedimiento de cifrado se describe en la figura 38. La salida analógica del procesador 3810 se monitoriza por el sensor 3820 y se convierte en un convertidor 3830 de analógico a digital y se alimenta al elemento 3850 de cifrado. La salida analógica del procesador 3810 que se monitoriza por el sensor 3820 y se convierte por el convertidor 3830 de analógico a digital puede ponerse en una memoria 3840 intermedia y después alimentarse al elemento 3850 de cifrado. El elemento 3850 de cifrado puede ocultar la información apropiada de atacantes de canal lateral de varias maneras, incluyendo cifrado en bloques de las muestras de bits o aleatorizándolas (en efecto un elemento de cifrado de transposición en el que la clave es una permutación).

Para monitores externos, la conexión física que da acceso a los perfiles se proporciona mediante un conmutador digital que requiere una contraseña. En este caso, los puntos de contacto para el monitor externo se proporcionan por un chip de gestión de potencia en la plataforma. El chip de gestión de potencia puede ser tan sencillo como un regulador de tensión, pero para la mayoría de los procesadores comerciales usados en teléfonos inteligentes modernos, los chips de gestión de potencia son mucho más complejos. Cuando se conecta el monitor apropiado, el gestor de potencia habilitado para PFP lee la contraseña del monitor externo una vez conectado y después redirige la corriente de fuente de alimentación para que pase a través del sensor externo, lo que permite que el monitor externo capte el drenaje de corriente instantánea o consumo de potencia. La figura 39 muestra una representación gráfica de este procedimiento.

Es importante observar que las soluciones descritas en este caso no pretenden impedir que atacantes lleven a cabo ataques de canal lateral contra los sistemas objetivo. En vez de eso, pretenden impedir que se aprovechen instalaciones de monitor de PFP para ataques de canal lateral. Al implementarse estas medidas, un posible atacante tendrá que realizar las mismas modificaciones de hardware a una placa con monitorización de PFP que a una sin ella.

Aunque se ha descrito la invención en términos de realizaciones preferidas, el alcance de la invención se define por las reivindicaciones adjuntas.

REIVINDICACIONES

1. Método para realizar una estimación de integridad en tiempo real de ejecución de una rutina en una plataforma de procesamiento informático, que comprende:

(a) para un código de confianza de la rutina:

(i) monitorizar la ejecución de la rutina rastreando el consumo de potencia de un procesador (205, 206, 2620) tomando muestras durante la ejecución de la rutina;

(ii) usar una técnica de caracterización de plataforma que comprende además detectar secciones de los perfiles que muestran la mayor dependencia de transiciones de estado en el procesador (205, 206, 2620); y usar dichas secciones para seleccionar características que portan la mayor parte de la información;

(iii) obtener a partir de una caracterización de características seleccionadas de la rutina contenida en dichas secciones un conjunto de huellas digitales de potencia de confianza de la rutina;

(iv) establecer un umbral para una tasa de falsas alarmas específica basándose en la distribución de probabilidad de distancia a partir de una firma compuesta por dichas huellas digitales de confianza; y

(b) para un código que no es de confianza de la rutina:

(i) comparar una biblioteca de dichas huellas digitales de confianza con características extraídas de perfiles a partir de la ejecución de código que no es de confianza;

(ii) determinar una distancia entre dichas huellas digitales y las características extraídas; y

(iii) notificar una excepción si la distancia supera el umbral.

2. Método según la reivindicación 1, que comprende además sincronizar dicho perfil con la ejecución de la rutina incorporando información de identificación de módulo en la rutina.

3. Método según la reivindicación 2, en el que la información de identificación de módulo es un código de identificación binario escrito en un registro (3324) de IO antes de la ejecución de la rutina.

4. Método según la reivindicación 2, en el que la información de identificación de módulo es una secuencia de instrucciones que produce un patrón de consumo de potencia distintivo.

5. Método según la reivindicación 1, en el que el perfil del consumo de potencia combina señales de una pluralidad de circuitos de procesador.

6. Método según la reivindicación 1, que comprende además potenciar la calidad de dicha notificación de excepción añadiendo a dicha biblioteca firmas de huella digital de malware conocidos.

7. Sistema para realizar una estimación de integridad en tiempo real de ejecución de una rutina en una plataforma (205, 206, 2620) de procesamiento informático, que comprende:

medios para monitorizar la ejecución de la rutina rastreando el consumo de potencia de un procesador (205, 206, 2620) tomando muestras durante la ejecución de la rutina;

medios para usar una técnica de caracterización de plataforma que comprenden además

medios para detectar secciones de los perfiles que muestran la mayor dependencia de transiciones de estado en el procesador (205, 206, 2620);

medios para usar dichas secciones para seleccionar características que portan la mayor parte de la información;

medios para obtener a partir de una caracterización de las características seleccionadas contenidas en dichas secciones un conjunto de huellas digitales de potencia de confianza de la rutina;

medios para establecer un umbral para una tasa de falsas alarmas específica basándose en la distribución de probabilidad de distancia a partir de una firma compuesta por dichas huellas digitales de confianza;

medios para comparar una biblioteca de dichas huellas digitales de confianza con características extraídas de los perfiles a partir de la ejecución de código que no es de confianza

medios para determinar una distancia entre dichas huellas digitales y las características extraídas; y

medios para notificar una excepción si la distancia supera el umbral;

en el que el sistema se configura para llevar a cabo automáticamente un método según una de las reivindicaciones anteriores.

8. Sistema según la reivindicación 7, que comprende además información de identificación de módulo incorporada en la rutina.

9. Sistema según la reivindicación 8, en el que la información de identificación de módulo es un código de identificación binario escrito en un registro (3324) de IO antes de la ejecución de la rutina.

10. Sistema según la reivindicación 8, en el que la información de identificación de módulo es una secuencia de instrucciones que produce un patrón de consumo de potencia distintivo.

11. Sistema según la reivindicación 7, en el que los medios (110) para rastrear el consumo de potencia combinan señales de una pluralidad de circuitos de procesador.

12. Sistema según la reivindicación 7, que comprende además medios para potenciar la calidad de dicha notificación de excepción añadiendo a dicha biblioteca firmas de huella digital de malware conocidos.

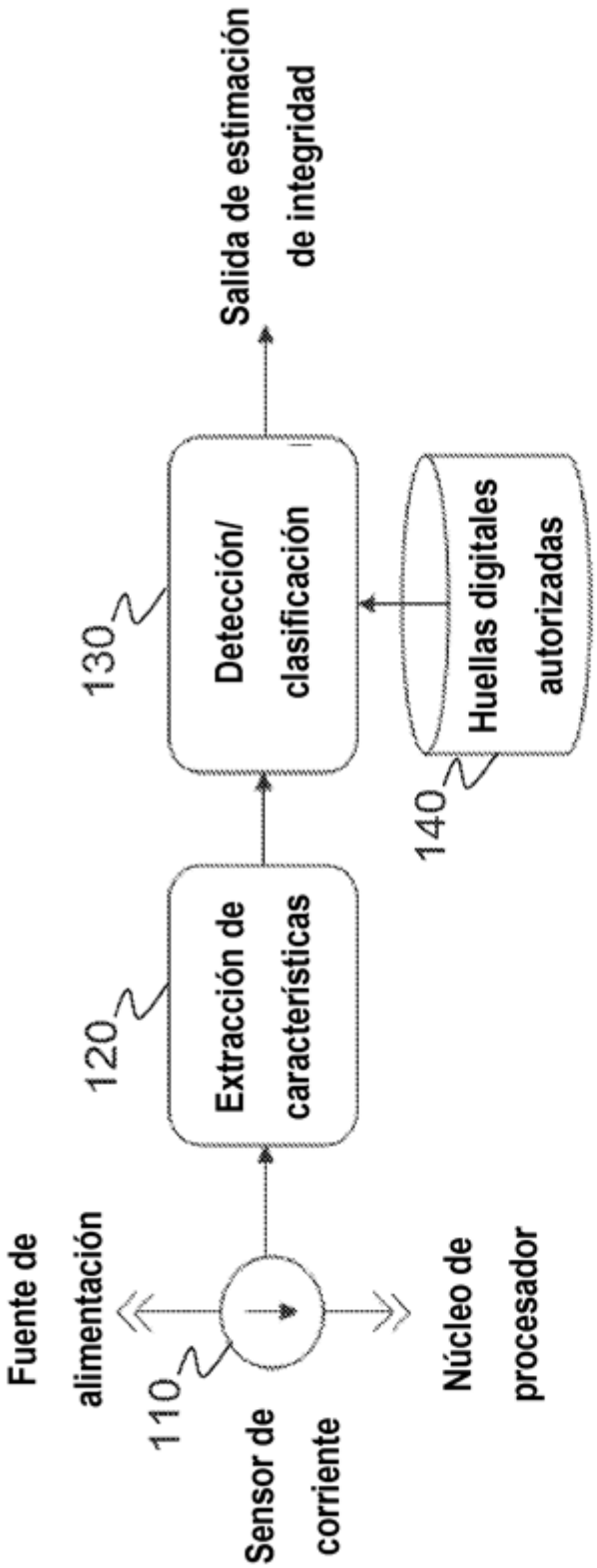


Figura 1 TÉCNICA ANTERIOR

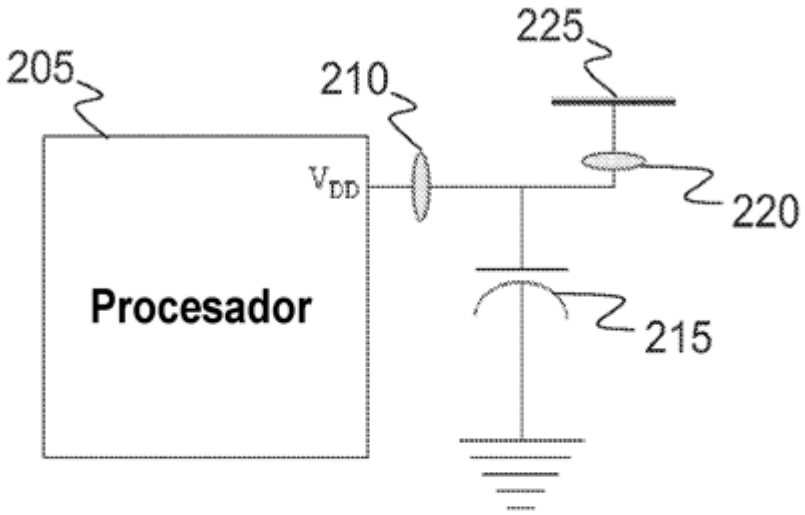


Figura 2

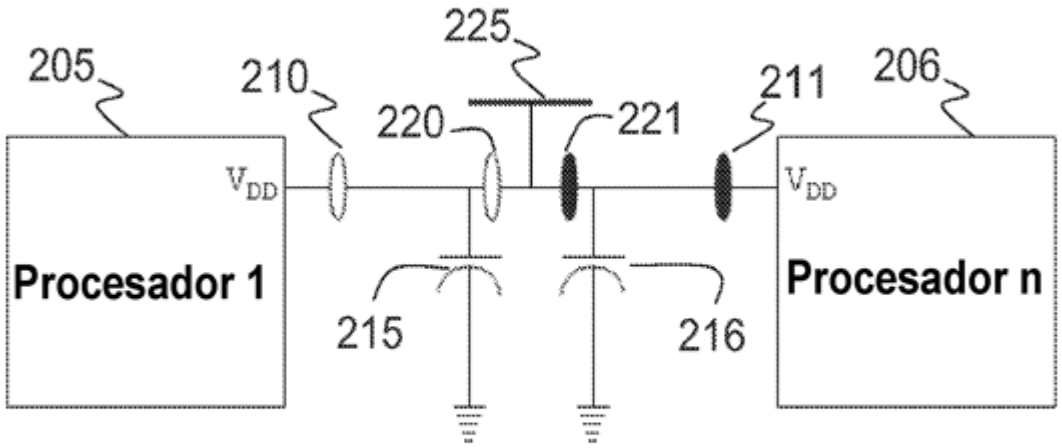


Figura 3

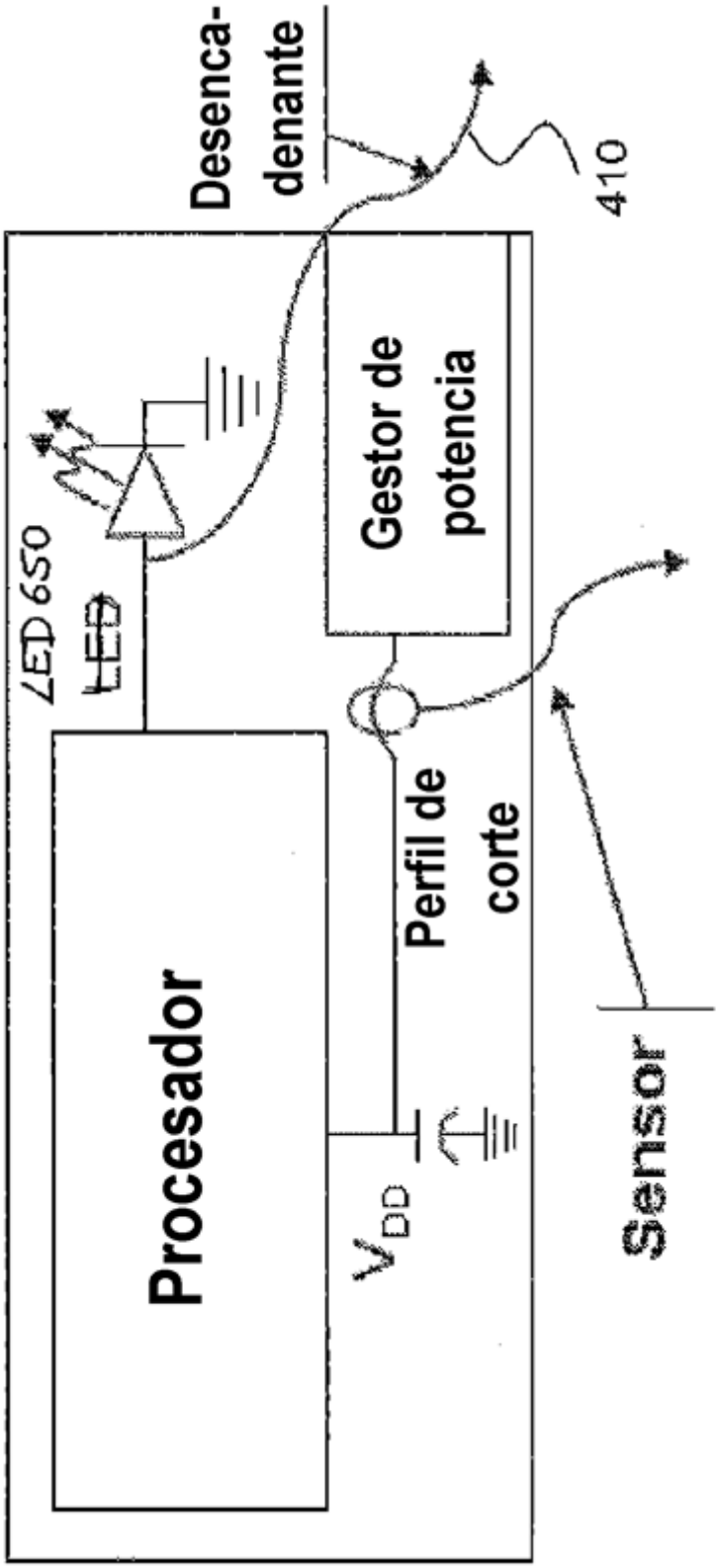

Desenca-
denante
410
LED 650
LED
Gestor de
potencia
Procesador
Perfil de
corte
V_{DD}
Sensor

Figura 4

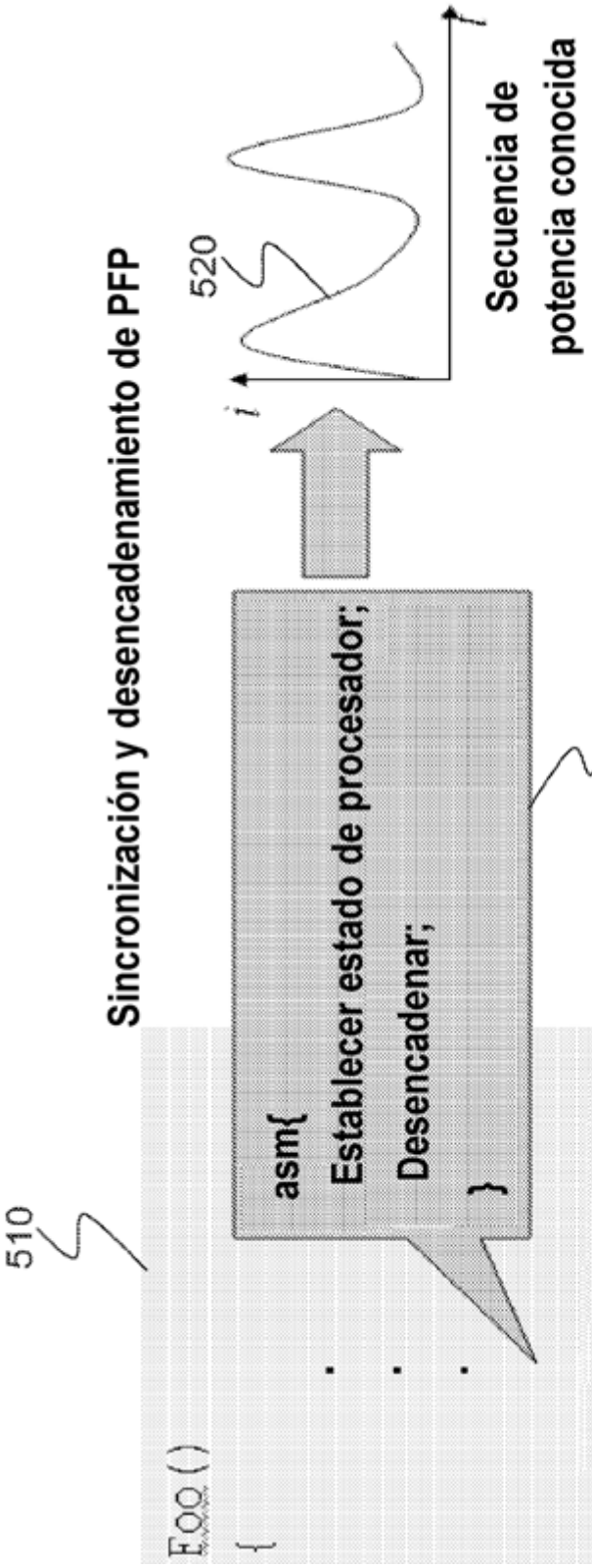

Sincronización y desencadenamiento de PFP
510
Foo ( )
{
.
.
.
asm{
Establecer estado de procesador;
Desencadenar;
}
Lógica original
de software objetivo
crítico
}
515
520
i
t
Secuencia de
potencia conocida

Figura 5

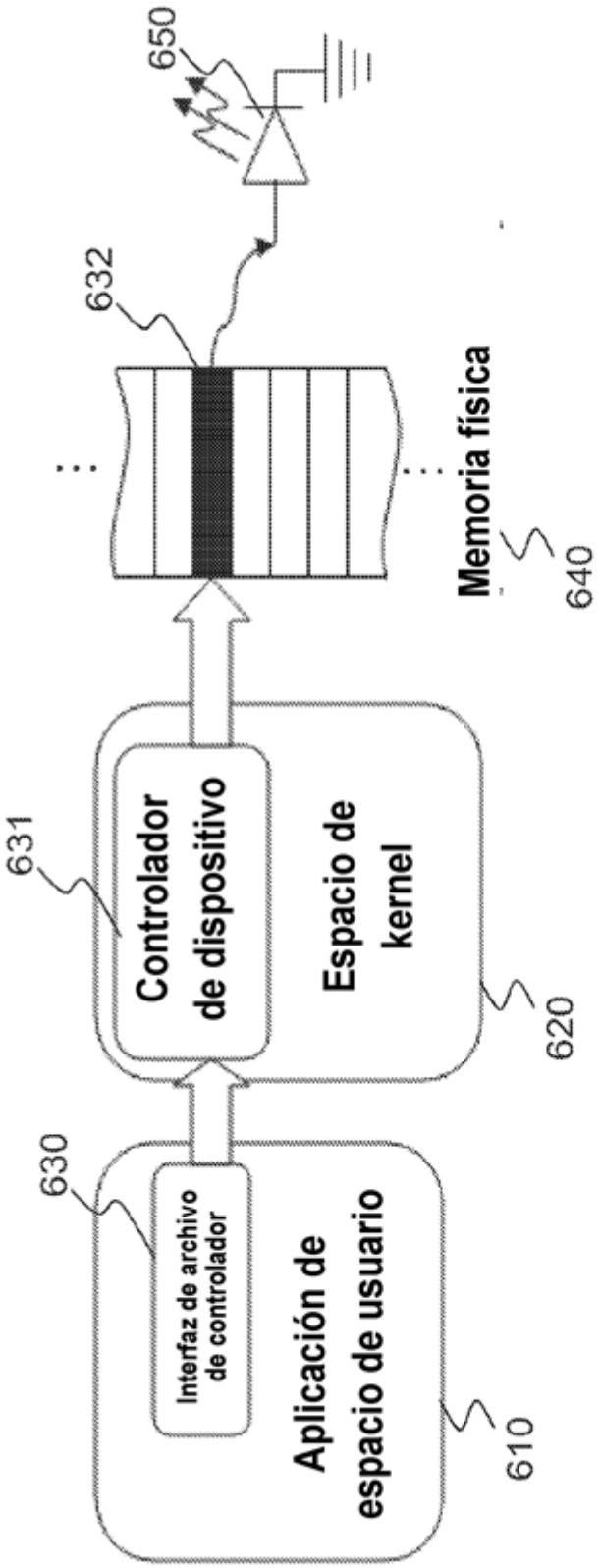

650
632
Memoria física
640
631
Controlador de dispositivo
Espacio de kernel
620
630
Interfaz de archivo de controlador
Aplicación de espacio de usuario
610

Figura 6

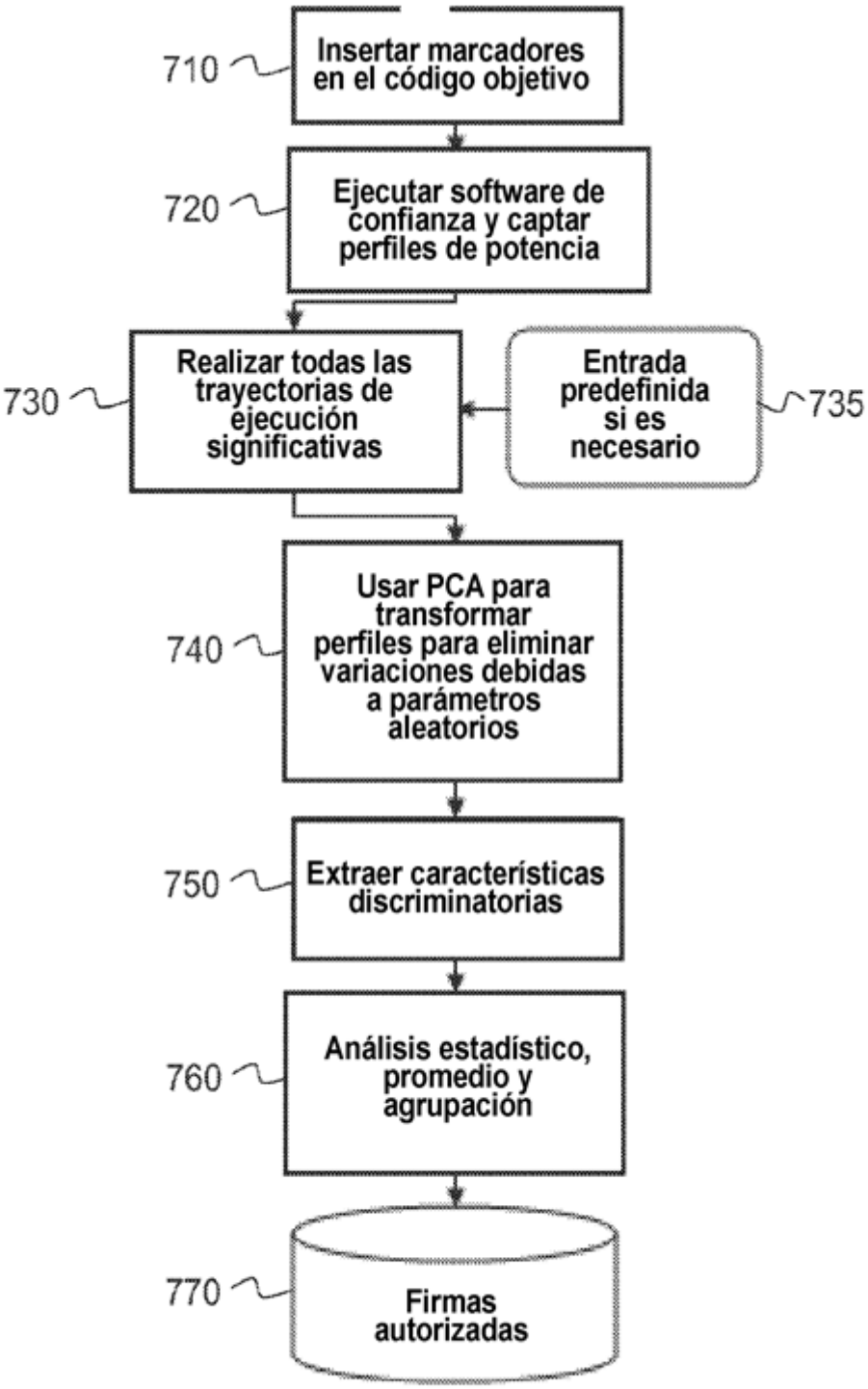

710
Insertar marcadores en el código objetivo
720
Ejecutar software de confianza y captar perfiles de potencia
730
Realizar todas las trayectorias de ejecución significativas
Entrada predefinida si es necesario
735
740
Usar PCA para transformar perfiles para eliminar variaciones debidas a parámetros aleatorios
750
Extraer características discriminatorias
760
Análisis estadístico, promedio y agrupación
770
Firmas autorizadas

Figura 7

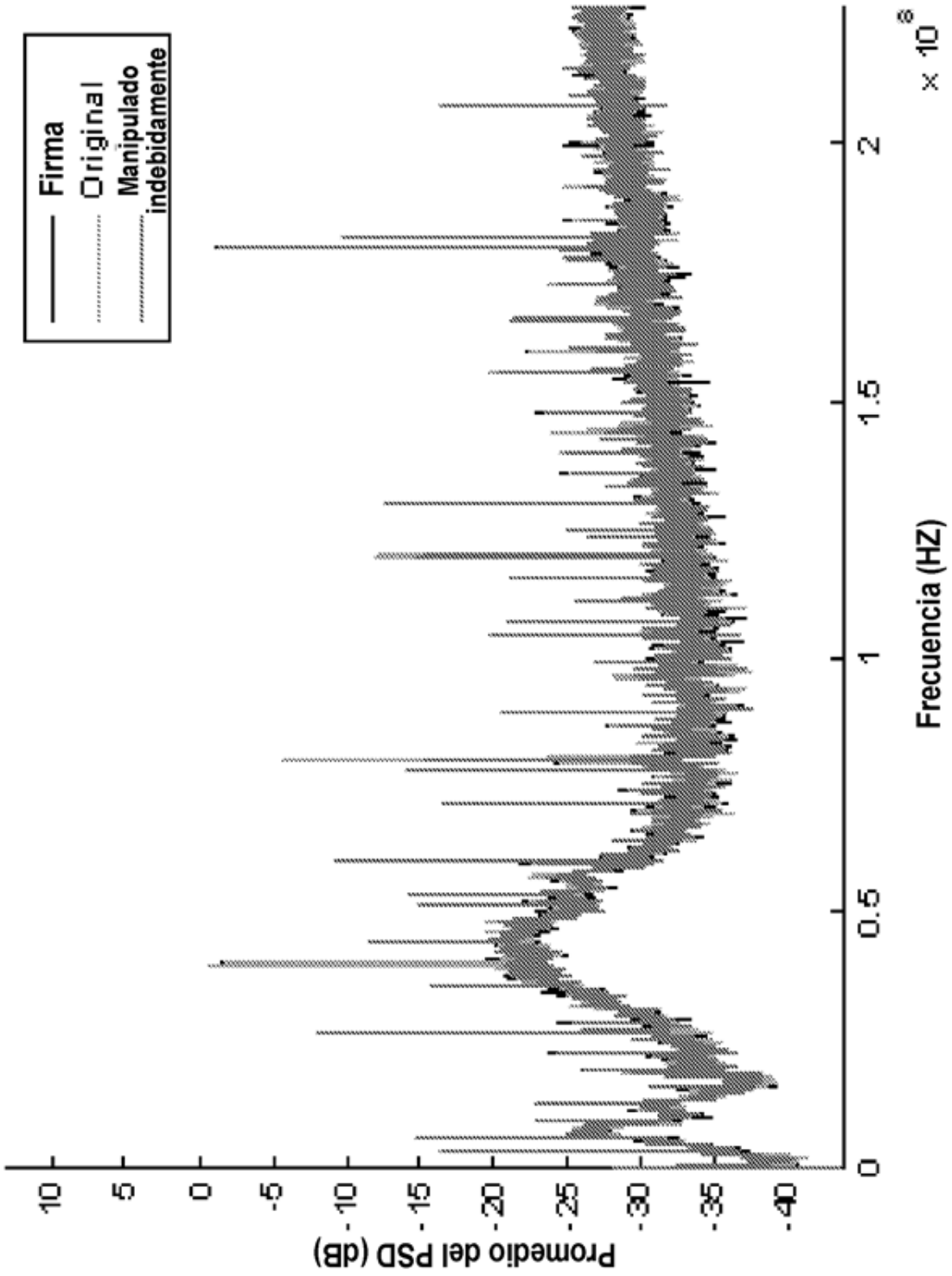

Firma
Original
Manipulado indebidamente
Promedio del PSD (dB)
10
5
0
-5
-10
-15
-20
-25
-30
-35
-40
0
0.5
1
1.5
2
x 10^8
Frecuencia (HZ)

Figura 8

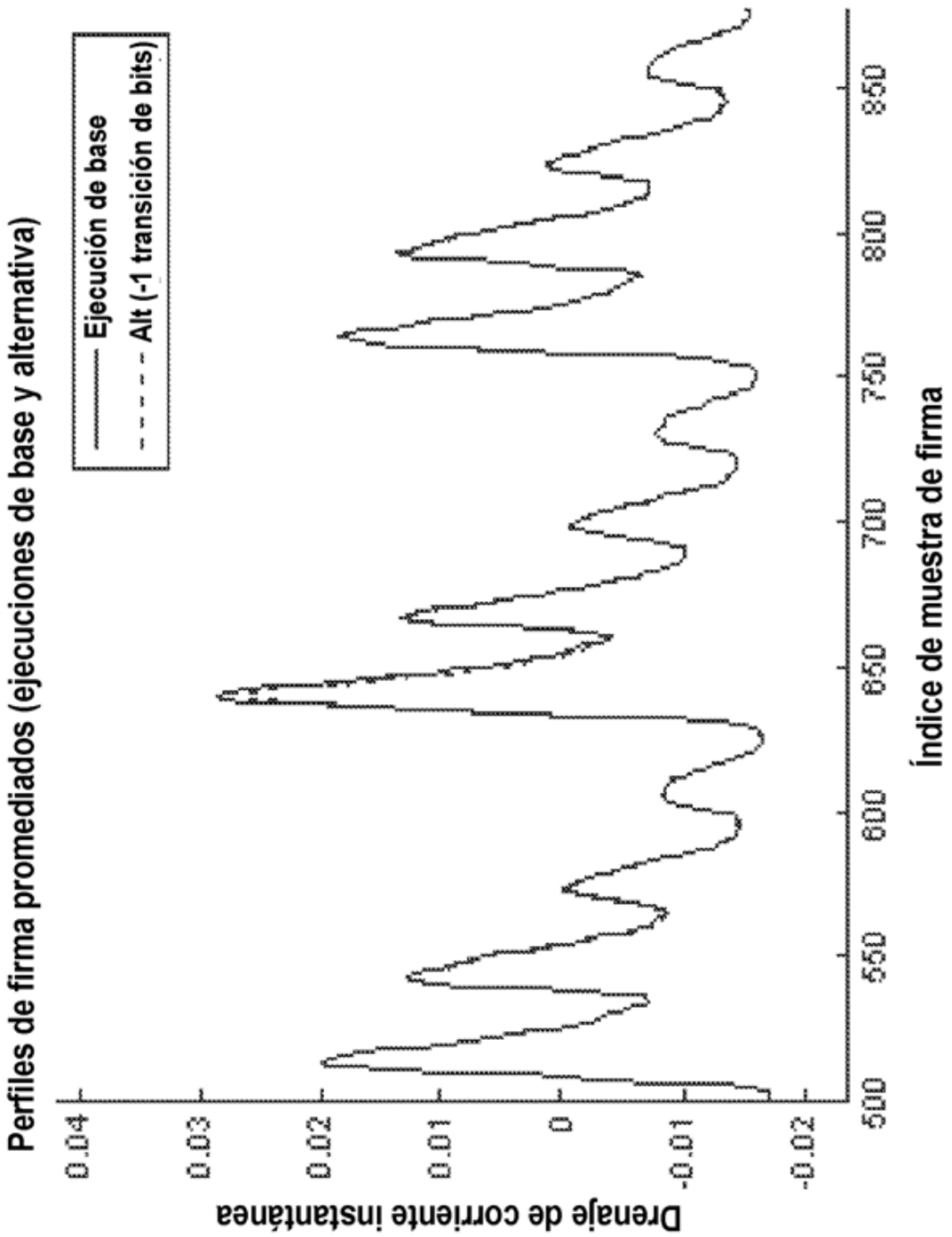

Perfiles de firma promediados (ejecuciones de base y alternativa)
Ejecución de base
Alt (-1 transición de bits)
Drenaje de corriente instantánea
0.04
0.03
0.02
0.01
0
-0.01
-0.02
500
550
600
650
700
750
800
850
Índice de muestra de firma

Figura 9

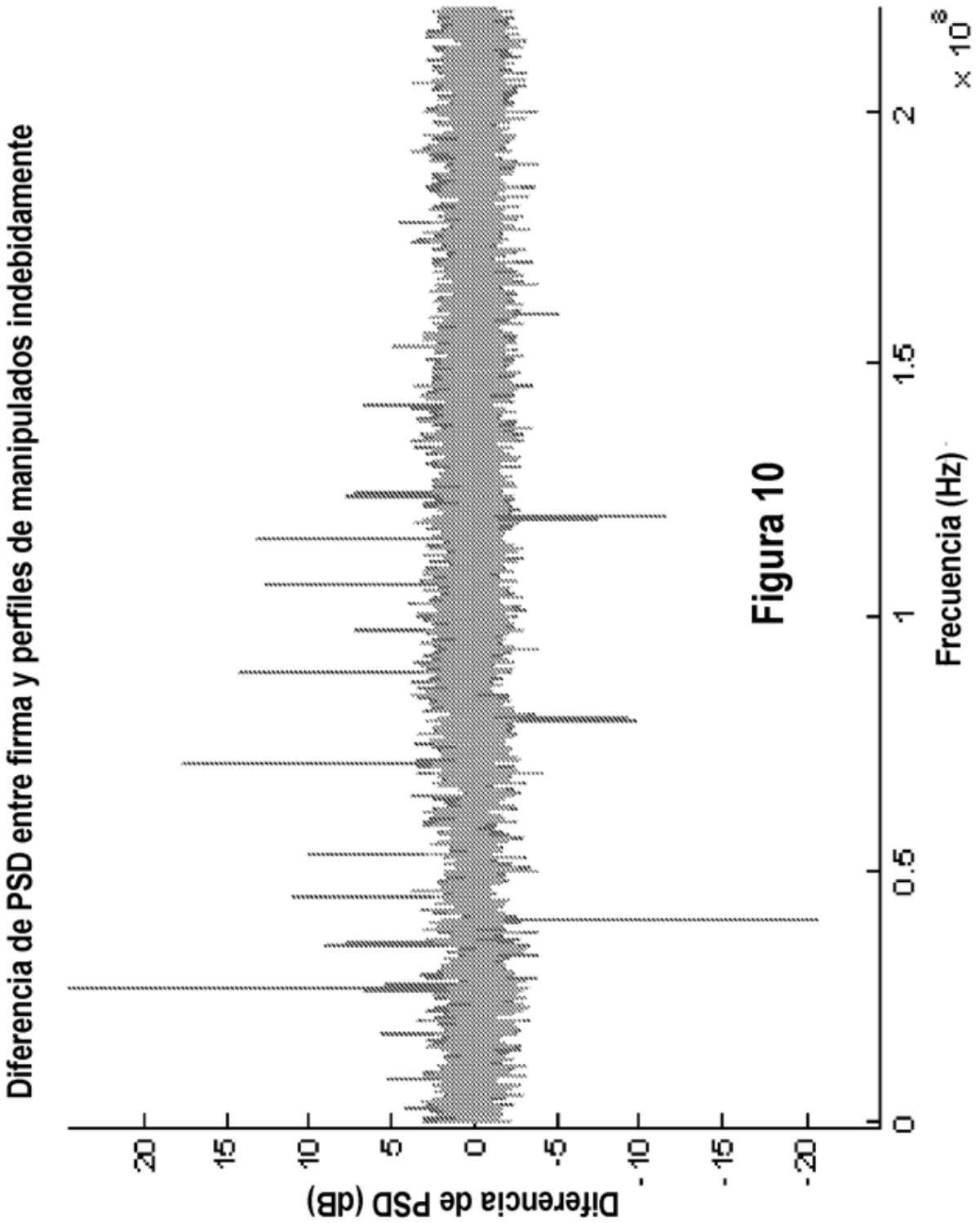

Diferencia de PSD entre firma y perfiles de manipulados indebidamente
Diferencia de PSD (dB)
20
15
10
5
0
-5
-10
-15
-20
0
0.5
1
1.5
2
Frecuencia (Hz)
x 10^8

Figura 10

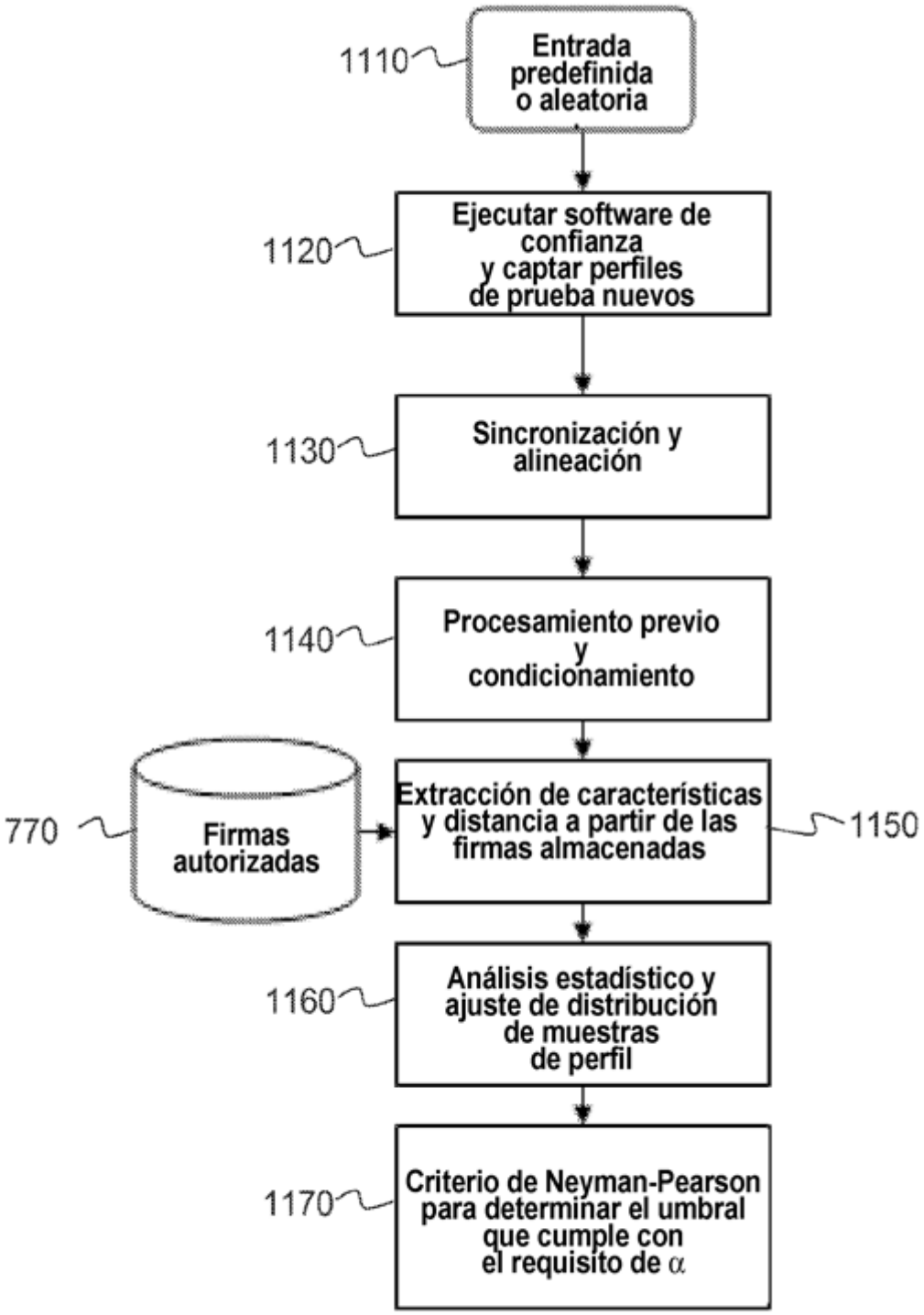

1110
Entrada predefinida o aleatoria
1120
Ejecutar software de confianza y captar perfiles de prueba nuevos
1130
Sincronización y alineación
1140
Procesamiento previo y condicionamiento
770
Firmas autorizadas
Extracción de características y distancia a partir de las firmas almacenadas
1150
1160
Análisis estadístico y ajuste de distribución de muestras de perfil
1170
Criterio de Neyman-Pearson para determinar el umbral que cumple con el requisito de α

Figura 11

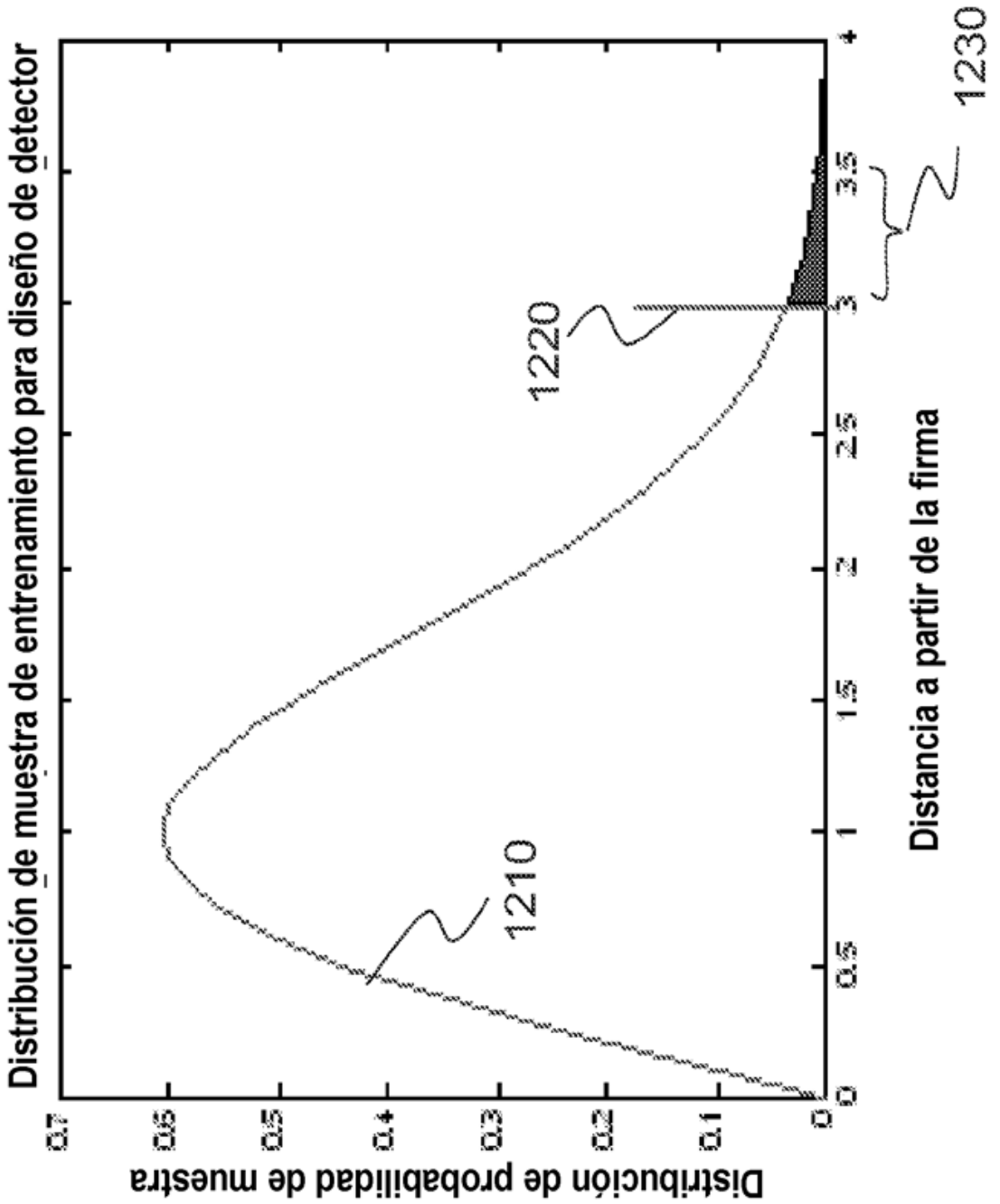

Distribución de muestra de entrenamiento para diseño de detector
Distribución de probabilidad de muestra
Distancia a partir de la firma
1210
1220
1230
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.5
1
1.5
2
2.5
3
3.5
4

Figura 12

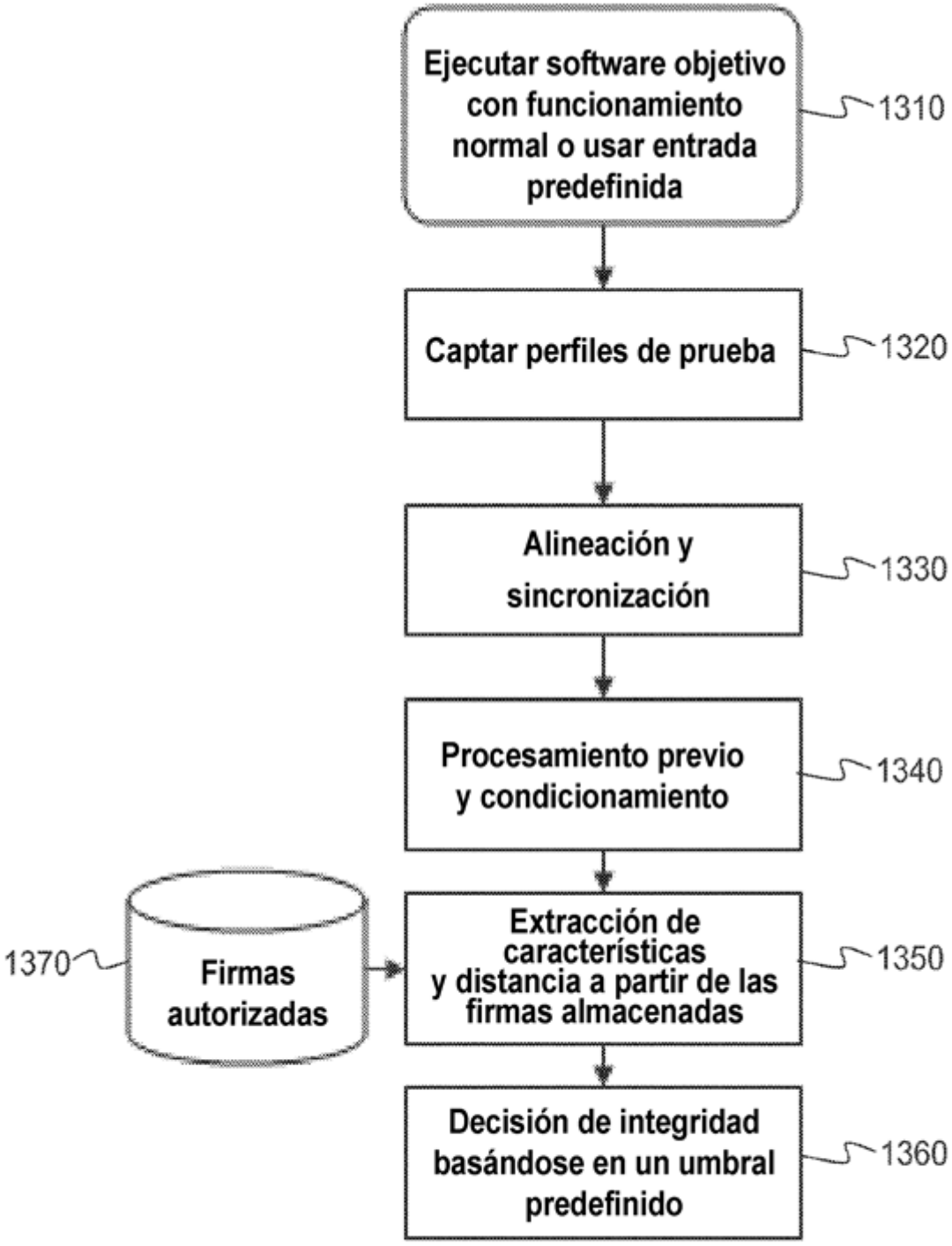

Ejecutar software objetivo con funcionamiento normal o usar entrada predefinida
1310
Captar perfiles de prueba
1320
Alineación y sincronización
1330
Procesamiento previo y condicionamiento
1340
1370
Firmas autorizadas
Extracción de características y distancia a partir de las firmas almacenadas
1350
Decisión de integridad basándose en un umbral predefinido
1360

Figura 13

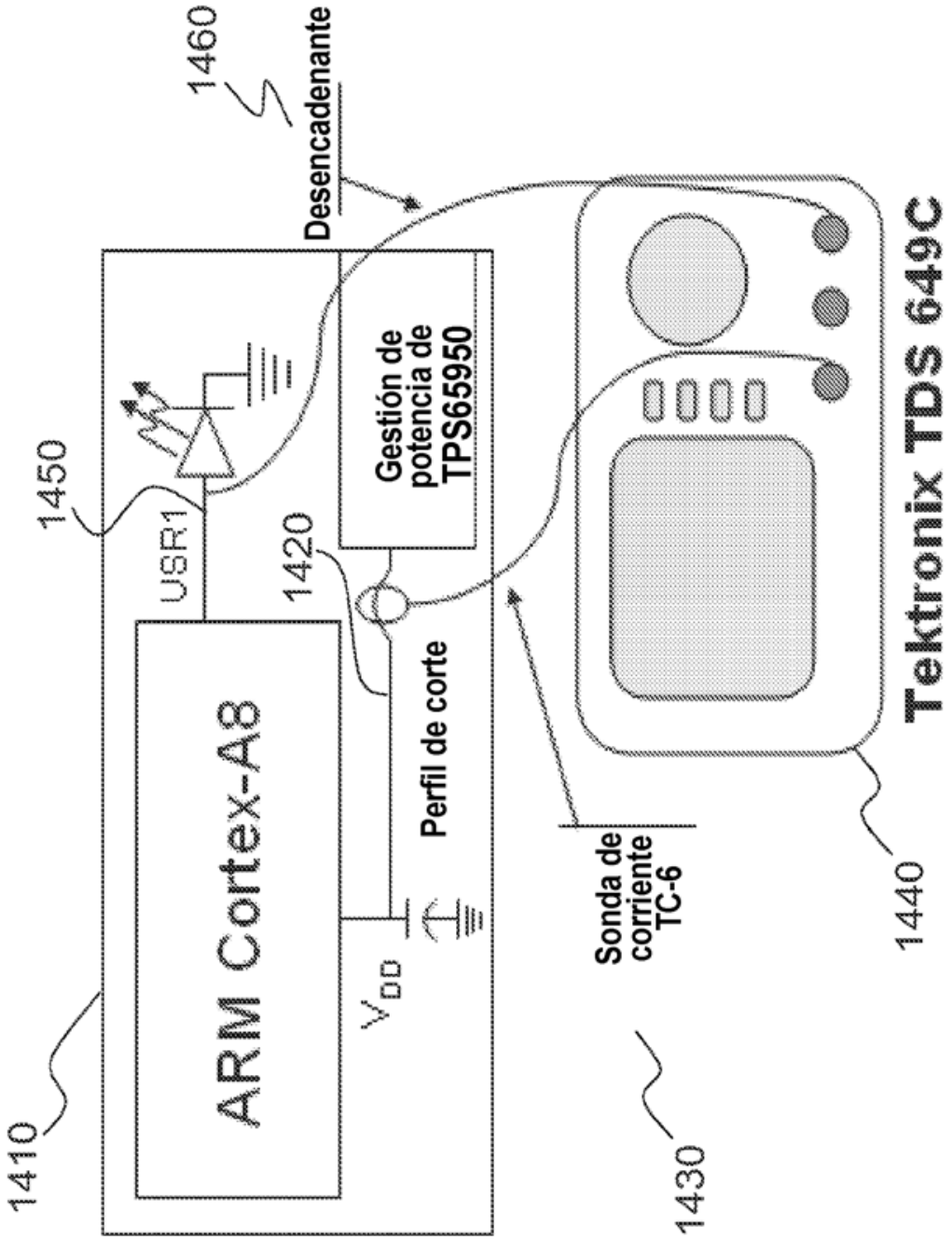

1460
Desencadenante
1450
USR1
1420
Gestión de potencia de TPS65950
ARM Cortex-A8
Perfil de corte
V_{DD}
Sonda de corriente TC-6
Tektronix TDS 649C
1410
1430
1440

Figura 14

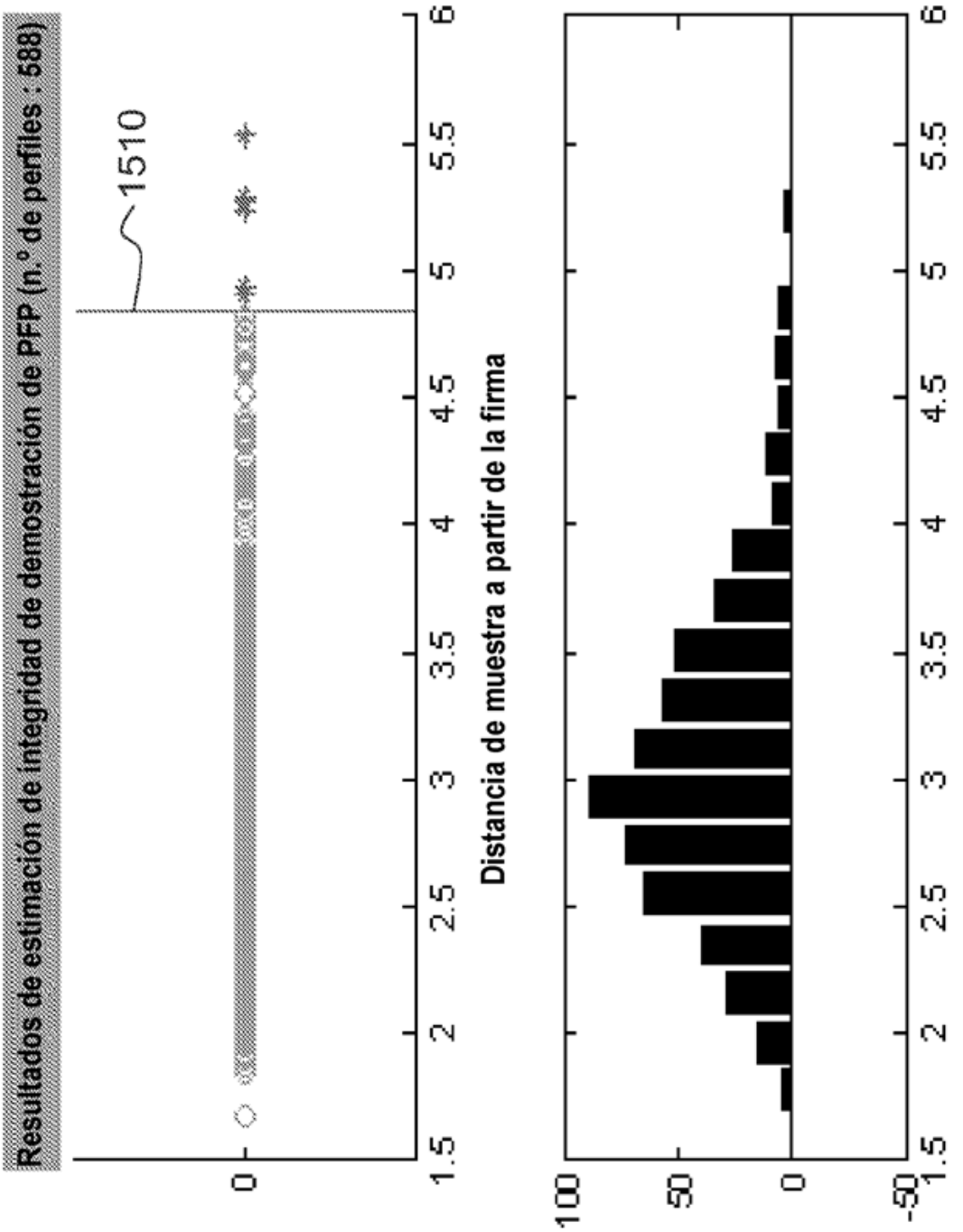

Resultados de estimación de integridad de demostración de PFP (n.º de perfiles : 588)
1510
0
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
Distancia de muestra a partir de la firma
100
50
0
-50
1.5
2
2.5
3
3.5
4
4.5
5
5.5
6
Historia de muestra

Figura 15

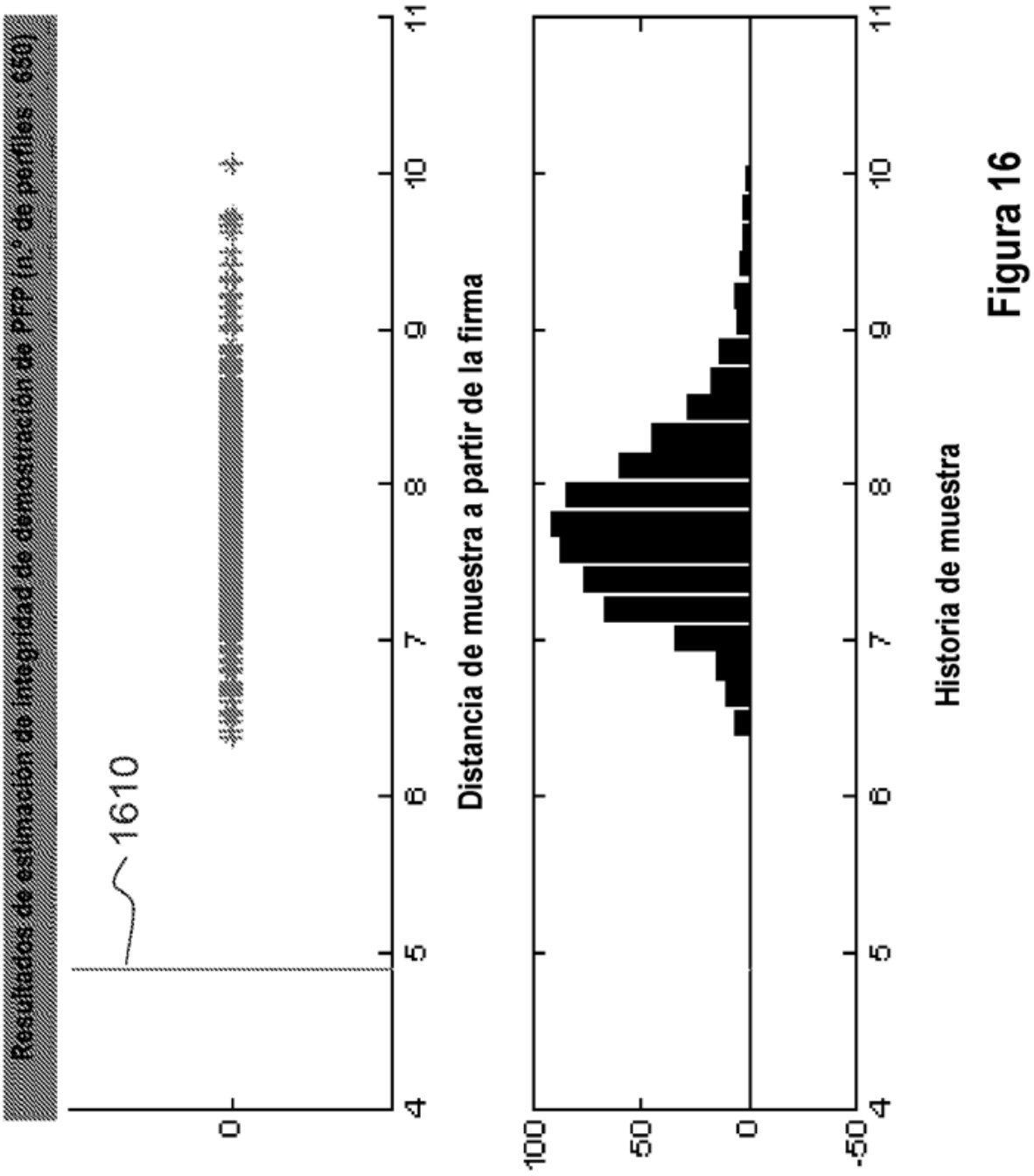

Resultados de estimación de integridad de demostración de PFP (n.º de perfiles : 650)
1610
0
4
5
6
7
8
9
10
11
Distancia de muestra a partir de la firma
100
50
0
-50
4
5
6
7
8
9
10
11
Historia de muestra

Figura 16

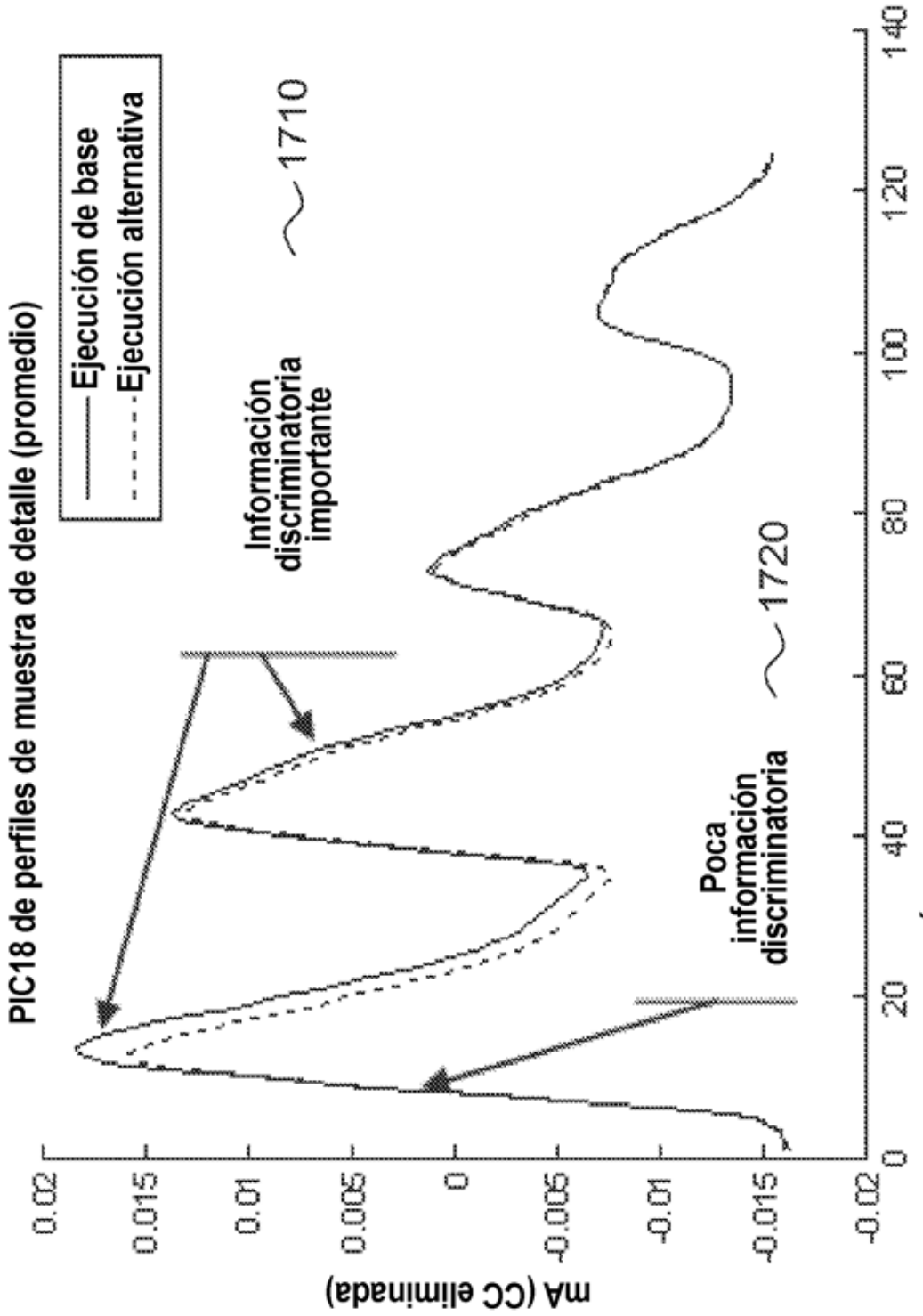

PIC18 de perfiles de muestra de detalle (promedio)
Ejecución de base
Ejecución alternativa
Información discriminatoria importante
1710
Poca información discriminatoria
1720
mA (CC eliminada)
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
0
20
40
60
80
100
120
140
Índice de muestra de tiempo (500 Msps)

Figura 17

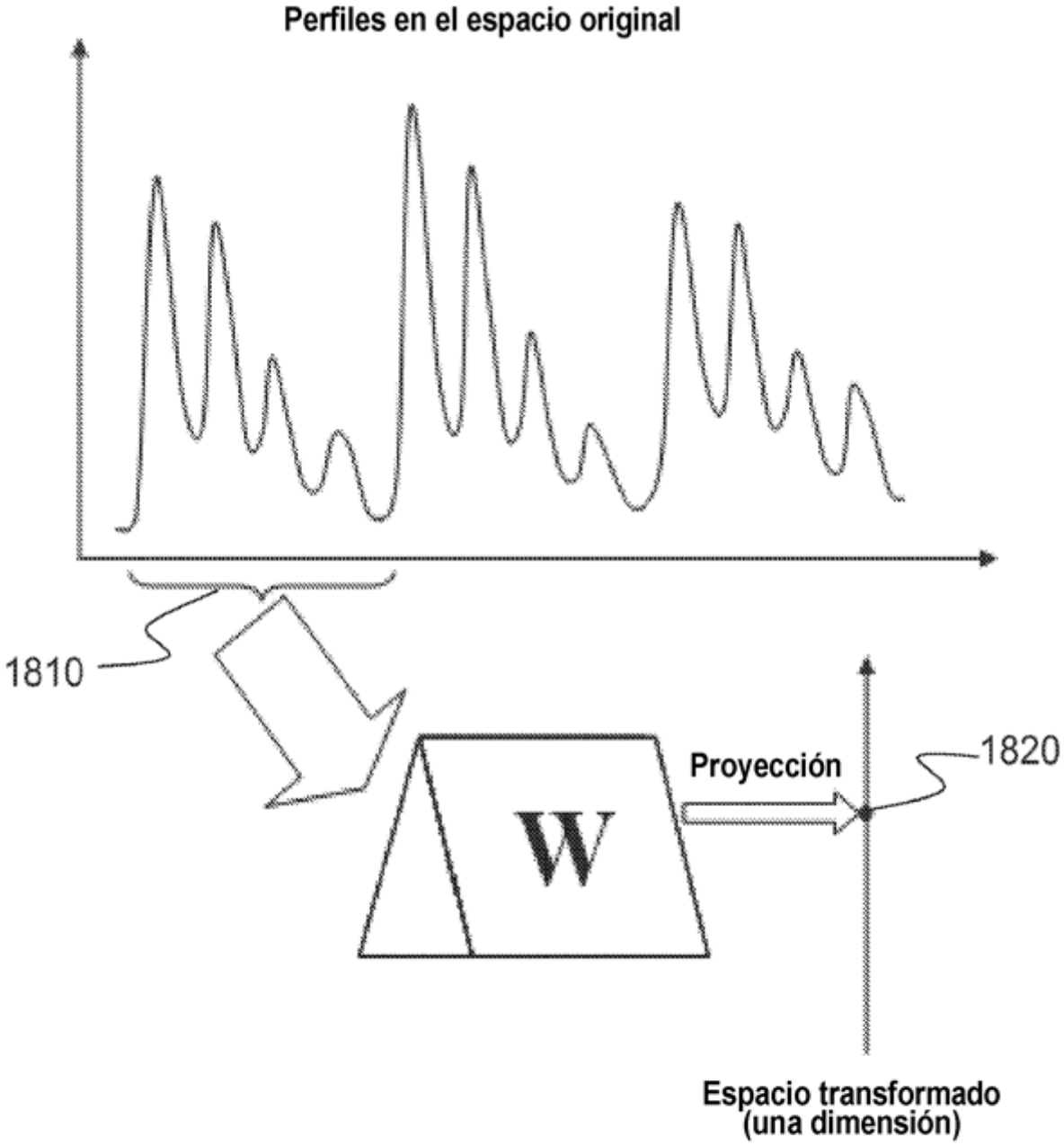

Perfiles en el espacio original
1810
W
Proyección
1820
Espacio transformado
(una dimensión)

Figura 18

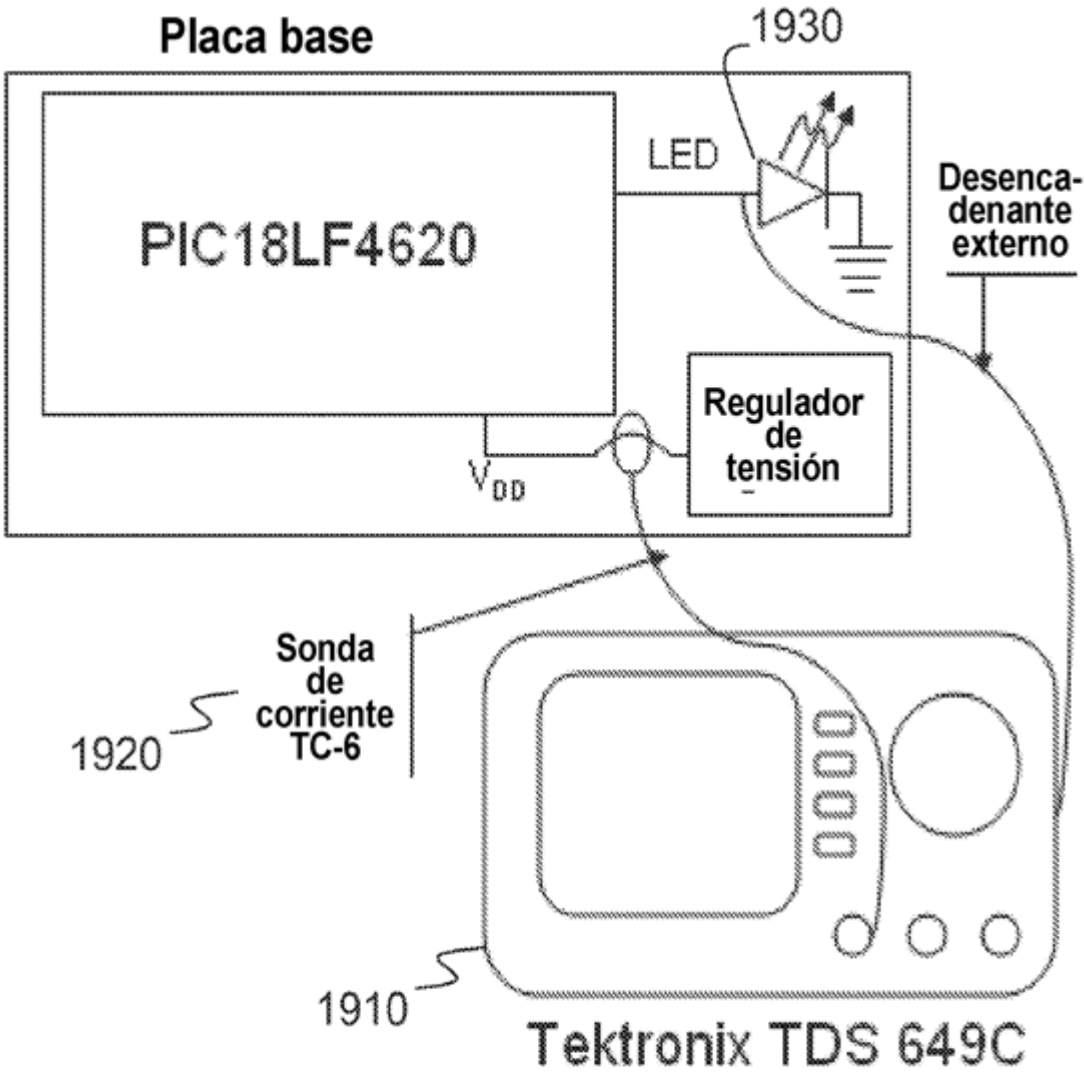

Placa base
1930
LED
PIC18LF4620
Desenca-
denante
externo
Regulador
de
tensión
V_{DD}
Sonda
de
corriente
TC-6
1920
1910
Tektronix TDS 649C

Figura 19

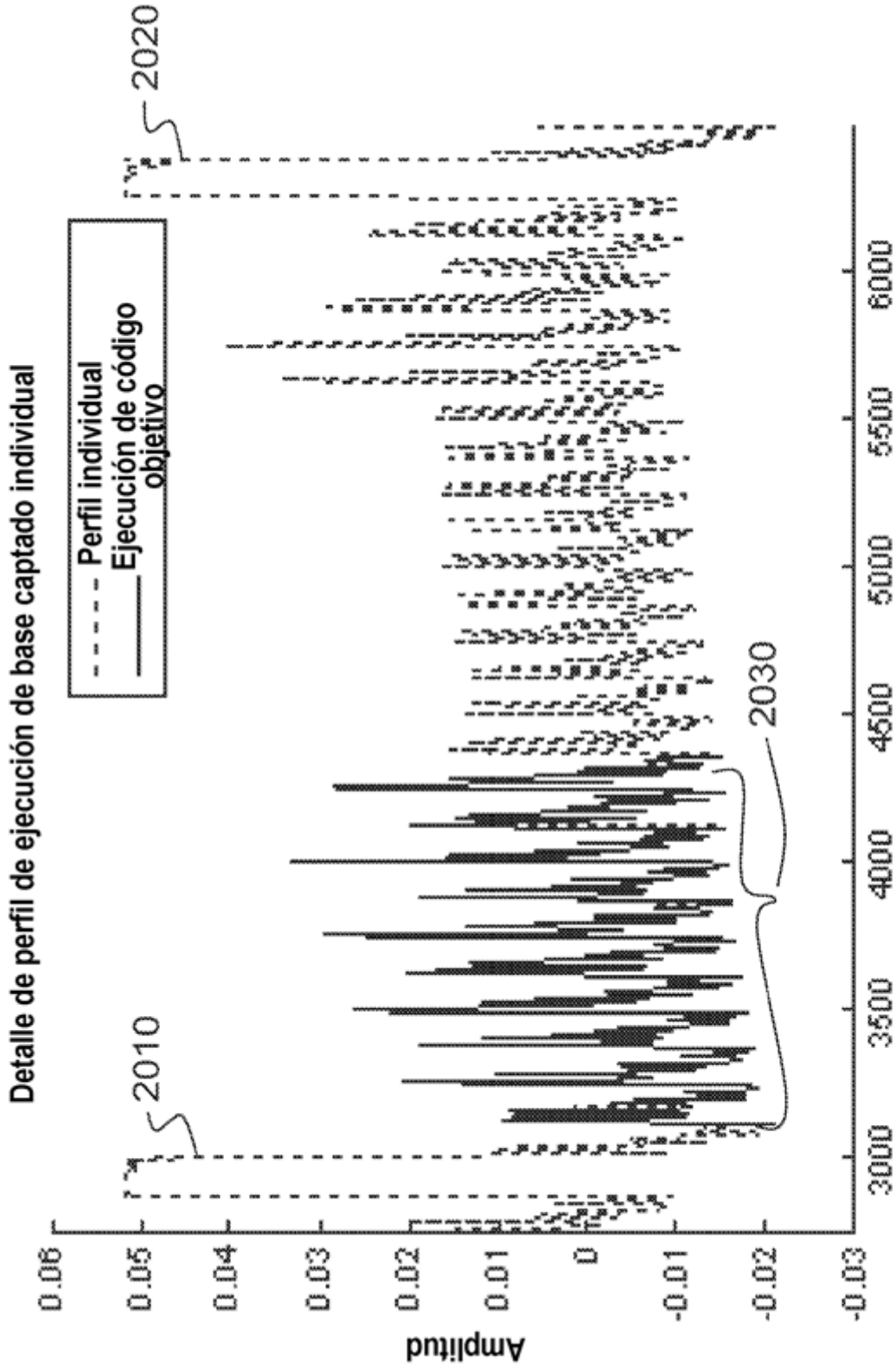

Detalle de perfil de ejecución de base captado individual
Perfil individual
Ejecución de código objetivo
2010
2020
2030
Amplitud
0.06
0.05
0.04
0.03
0.02
0.01
0
-0.01
-0.02
-0.03
3000
3500
4000
4500
5000
5500
6000
Índice de muestra (n)

Figura 20

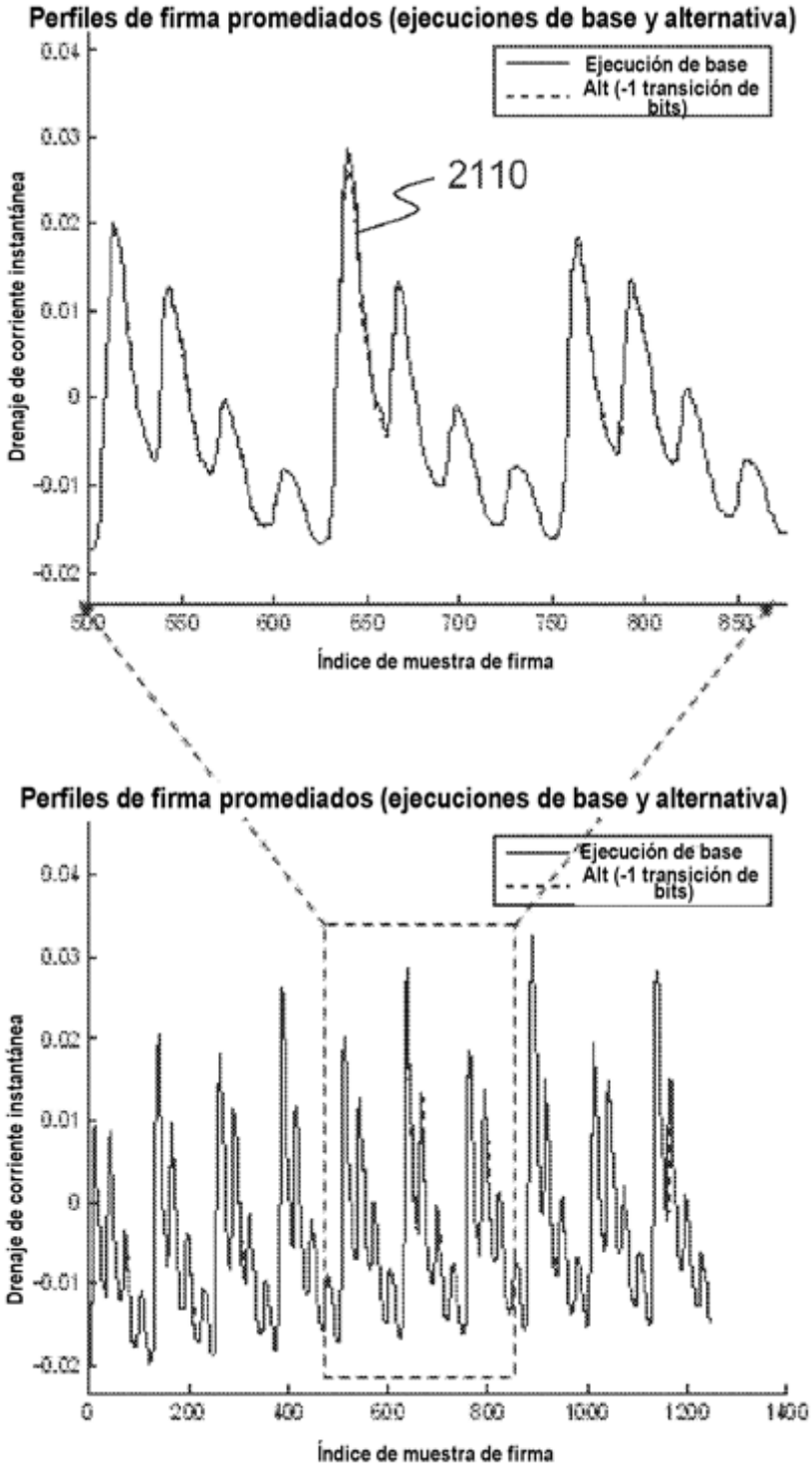

Perfiles de firma promediados (ejecuciones de base y alternativa)
Ejecución de base
Alt (-1 transición de bits)
2110
Drenaje de corriente instantánea
Índice de muestra de firma
Perfiles de firma promediados (ejecuciones de base y alternativa)
Ejecución de base
Alt (-1 transición de bits)
Drenaje de corriente instantánea
Índice de muestra de firma

Figura 21

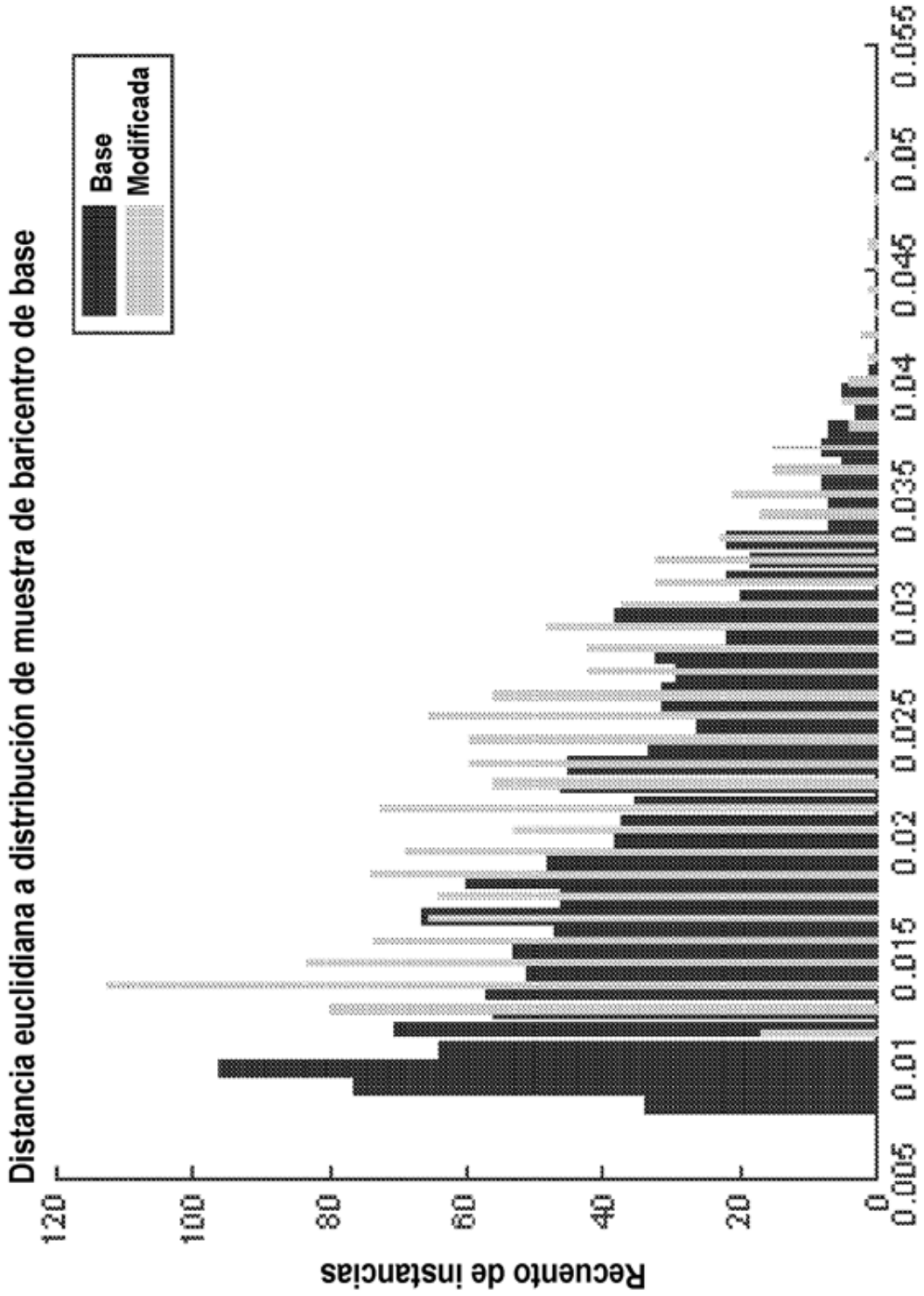

Distancia euclidiana a distribución de muestra de baricentro de base
Base
Modificada
Recuento de instancias
120
100
80
60
40
20
0
0.005
0.01
0.015
0.02
0.025
0.03
0.035
0.04
0.045
0.05
0.055
Distancia euclidiana a firma de base

Figura 22

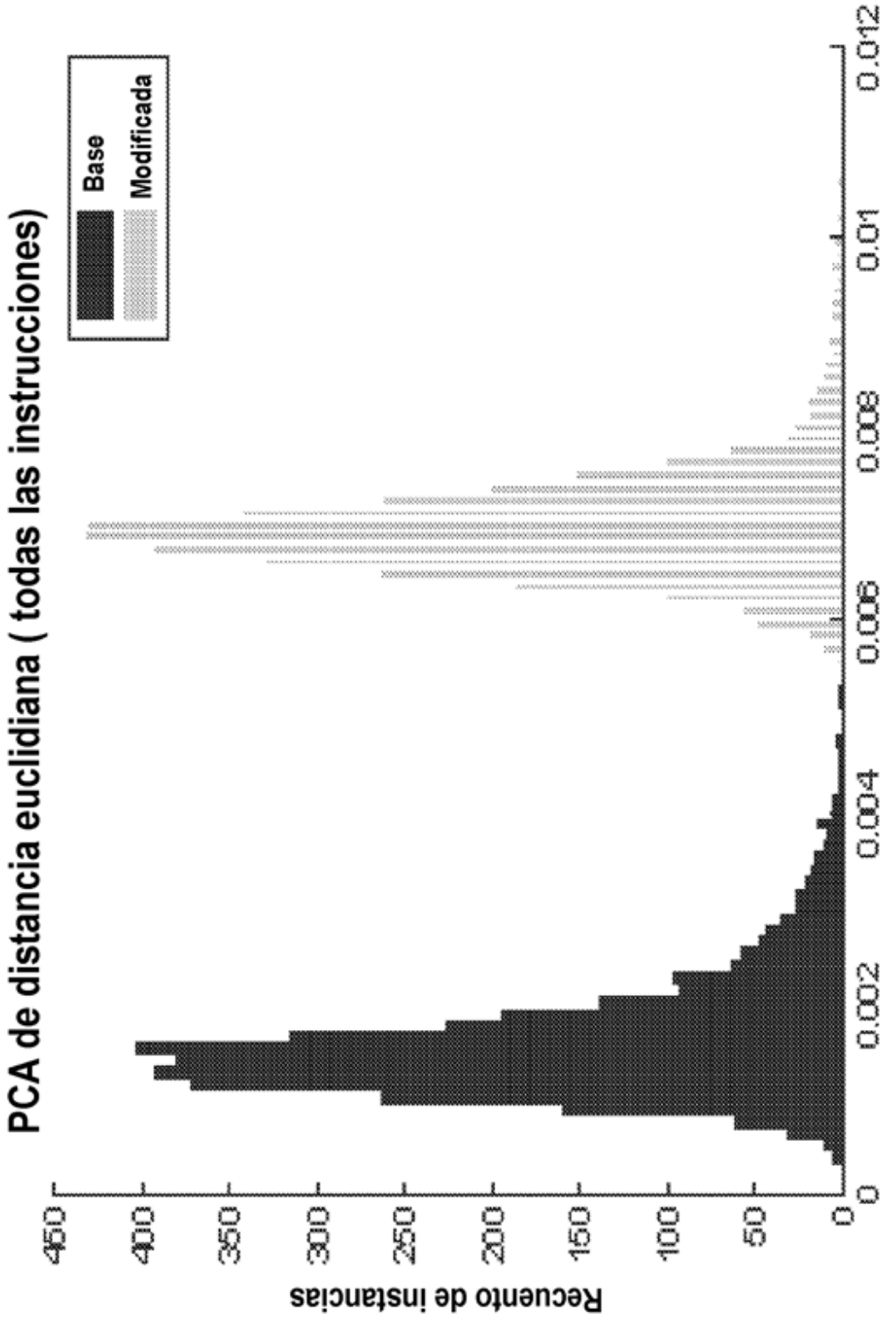

PCA de distancia euclidiana (todas las instrucciones)
Base
Modificada
Recuento de instancias
0
50
100
150
200
250
300
350
400
450
0
0.002
0.004
0.006
0.008
0.01
0.012
Distancia euclidiana a firma de base

Figura 23

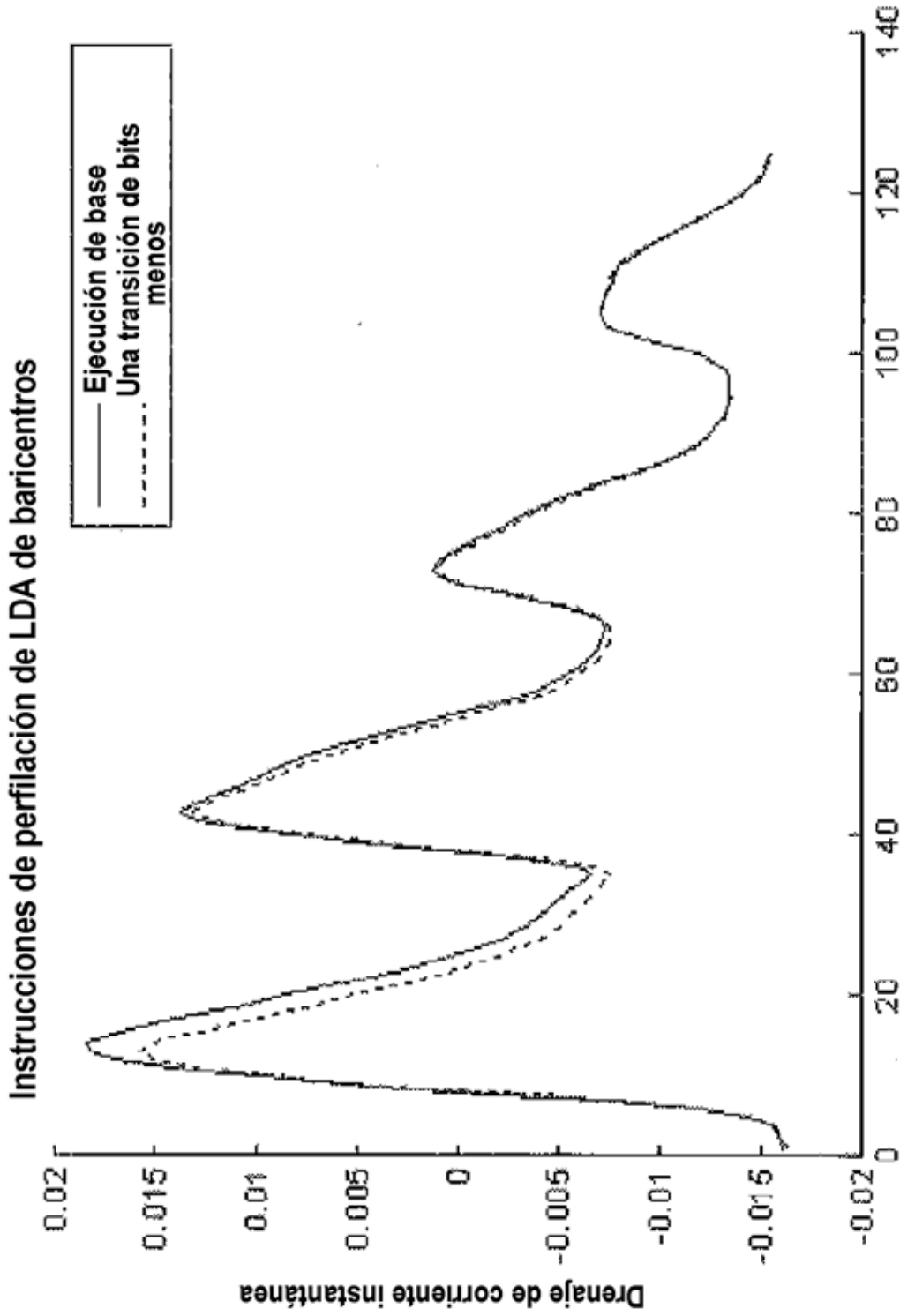

Instrucciones de perfilación de LDA de baricentros
Ejecución de base
Una transición de bits menos
Drenaje de corriente instantánea
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
0
20
40
60
80
100
120
140
Índice de muestra

Figura 24

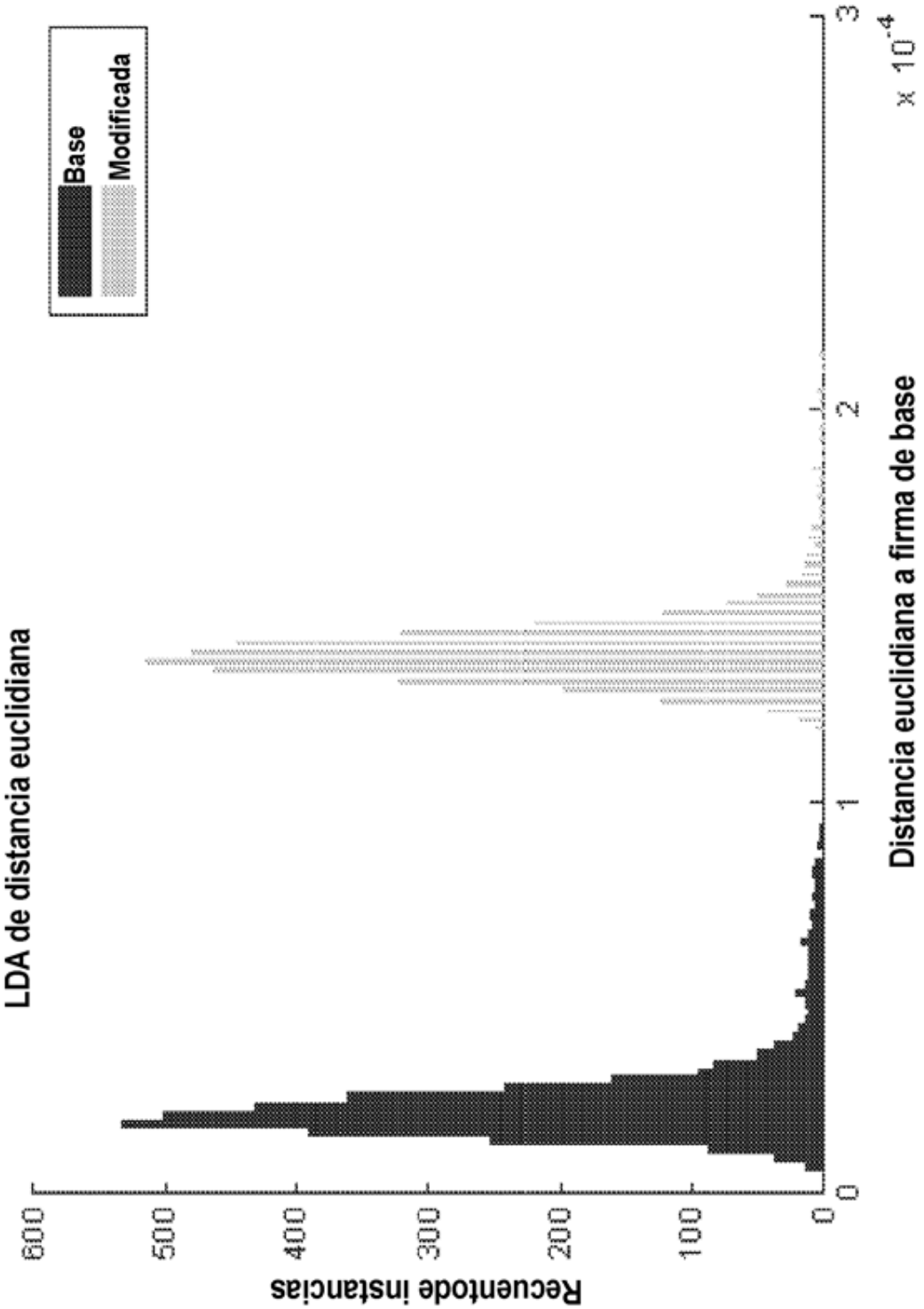

LDA de distancia euclidiana
Base
Modificada
Recuentode instancias
600
500
400
300
200
100
0
0
1
2
3
x 10^-4
Distancia euclidiana a firma de base

Figura 25

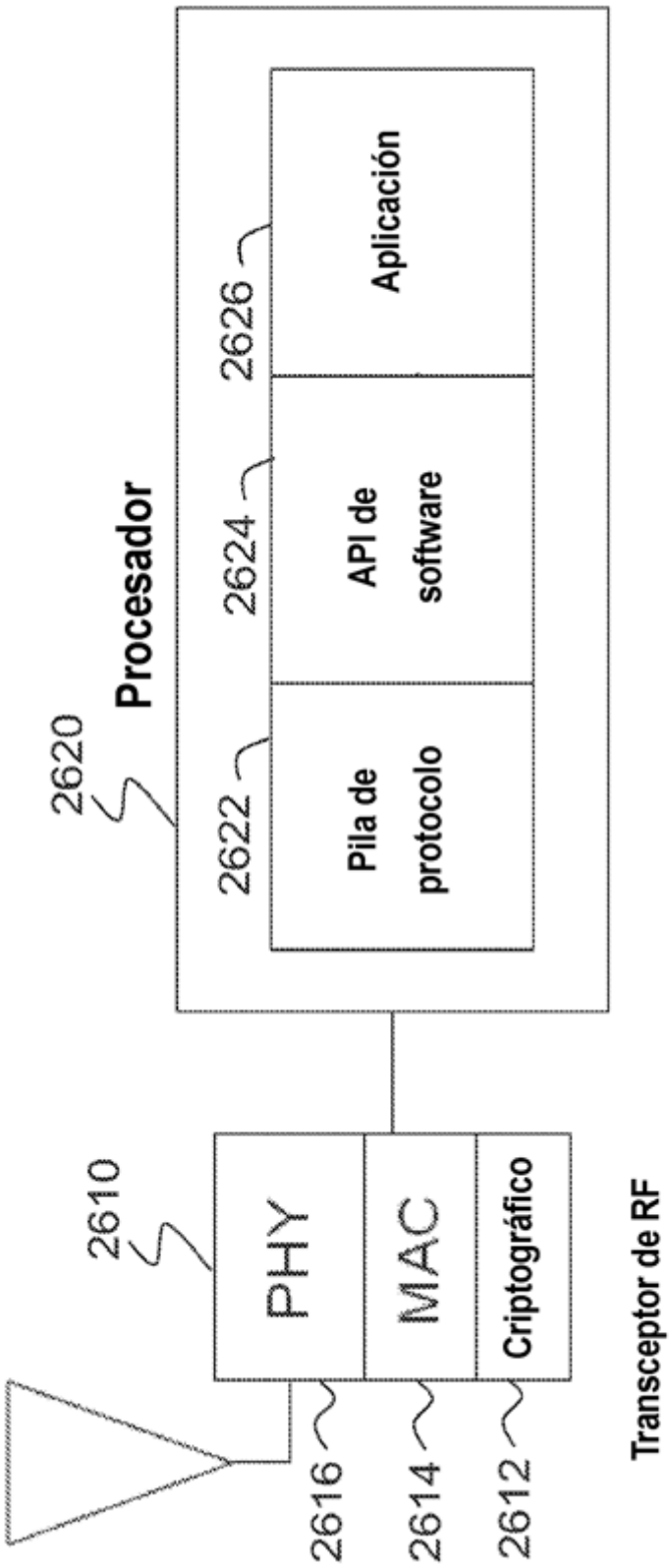

2620
Procesador
2622
Pila de protocolo
2624
API de software
2626
Aplicación
2610
PHY
MAC
Criptográfico
2616
2614
2612
Transceptor de RF

Figura 26

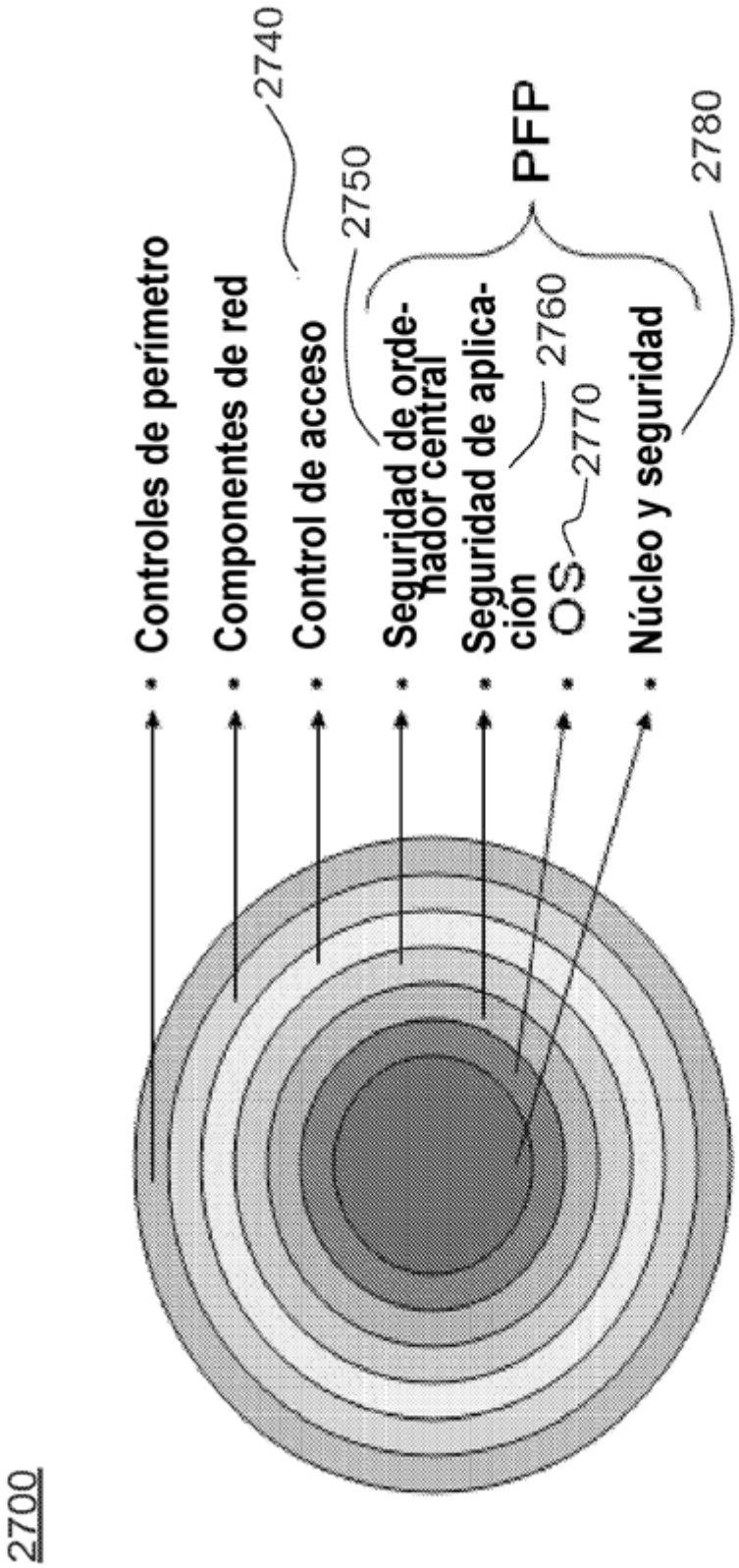

2700
Controles de perímetro
Componentes de red
Control de acceso
Seguridad de orde-nador central
Seguridad de aplica-ción
OS
Núcleo y seguridad
2740
2750
2760
2770
2780
PFP

Figura 27

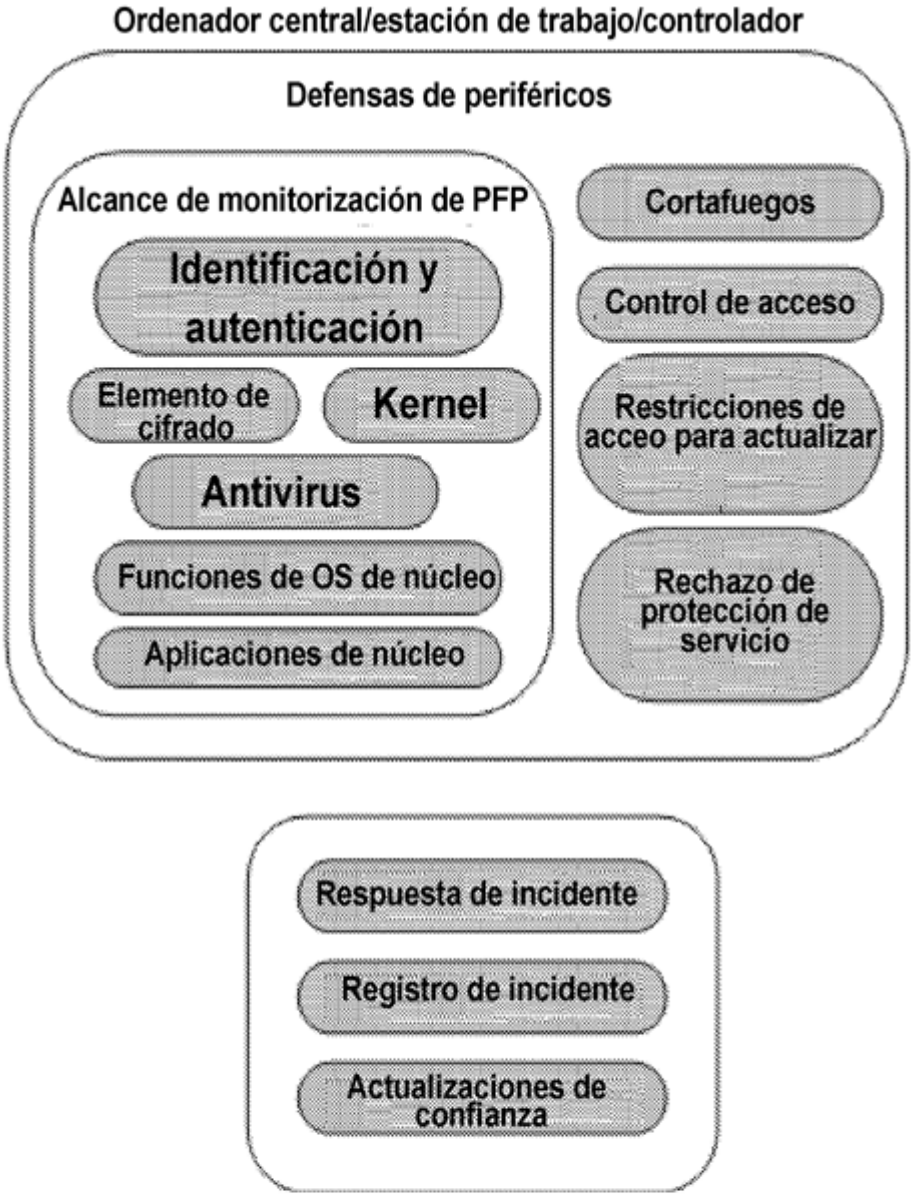

Ordenador central/estación de trabajo/controlador
Defensas de periféricos
Alcance de monitorización de PFP
Identificación y autenticación
Elemento de cifrado
Kernel
Antivirus
Funciones de OS de núcleo
Aplicaciones de núcleo
Cortafuegos
Control de acceso
Restricciones de acceo para actualizar
Rechazo de protección de servicio
Respuesta de incidente
Registro de incidente
Actualizaciones de confianza
Gestor de seguridad de IT / usuario

Figura 28

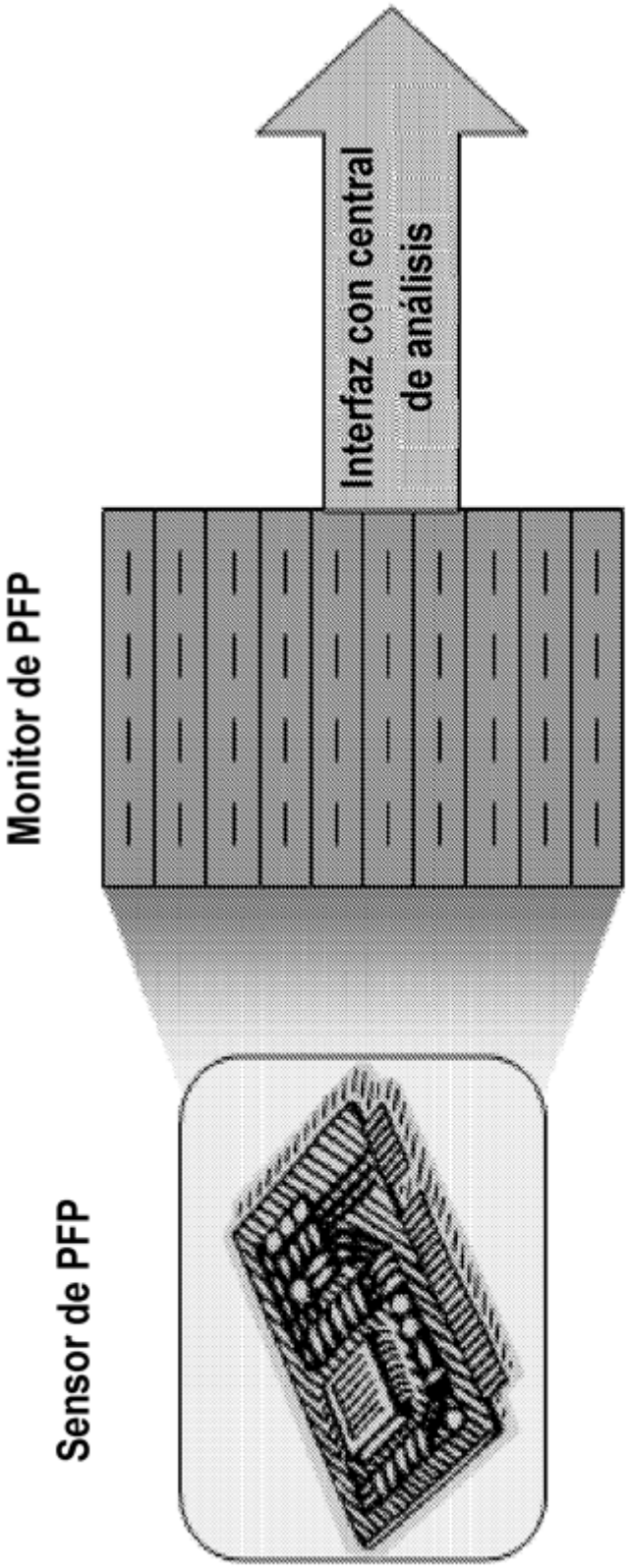
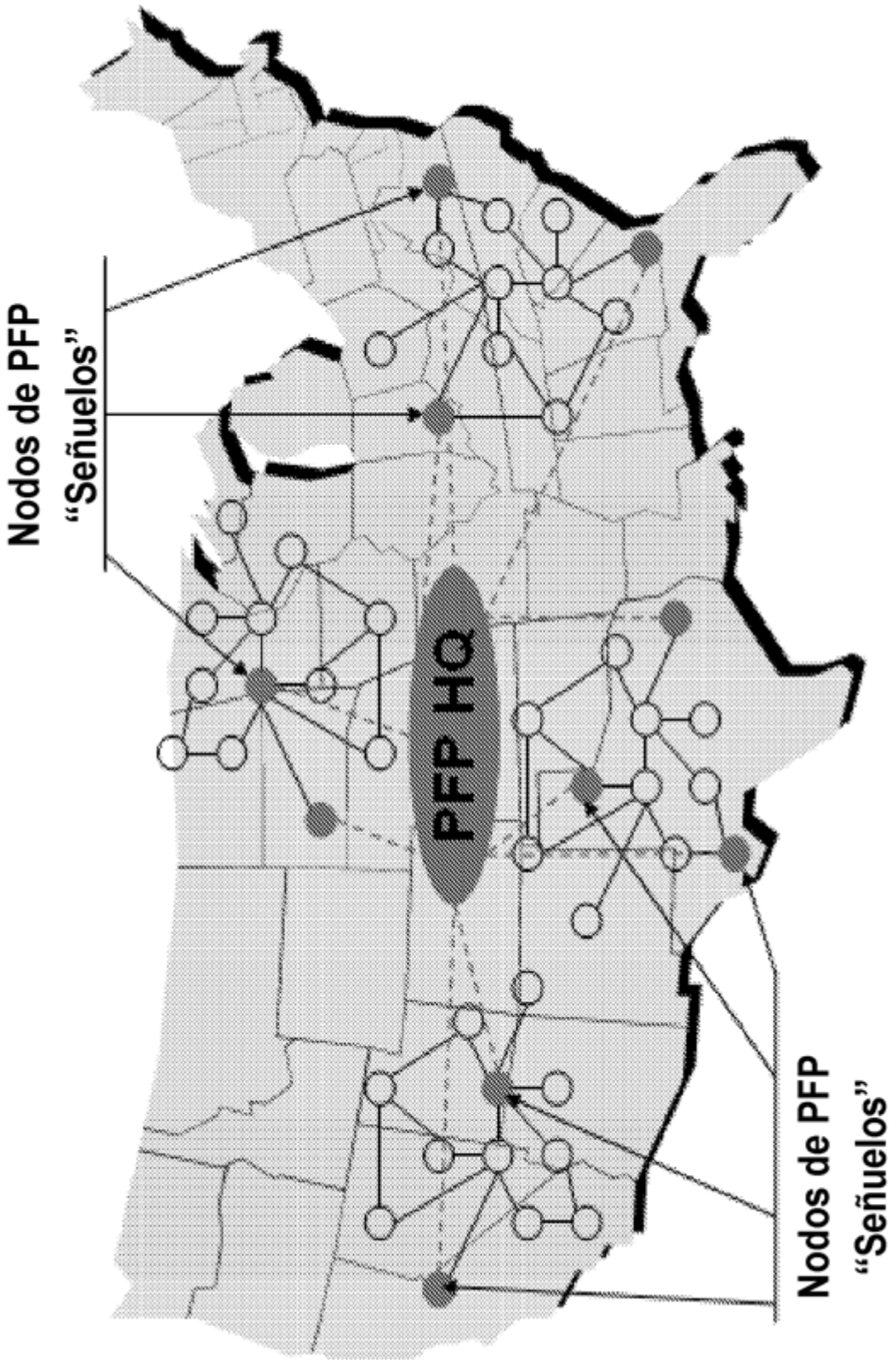

Interfaz con central de análisis
Monitor de PFP
Sensor de PFP

Figura 29


Nodos de PFP
"Señuelos"
PFP HQ
Nodos de PFP
"Señuelos"

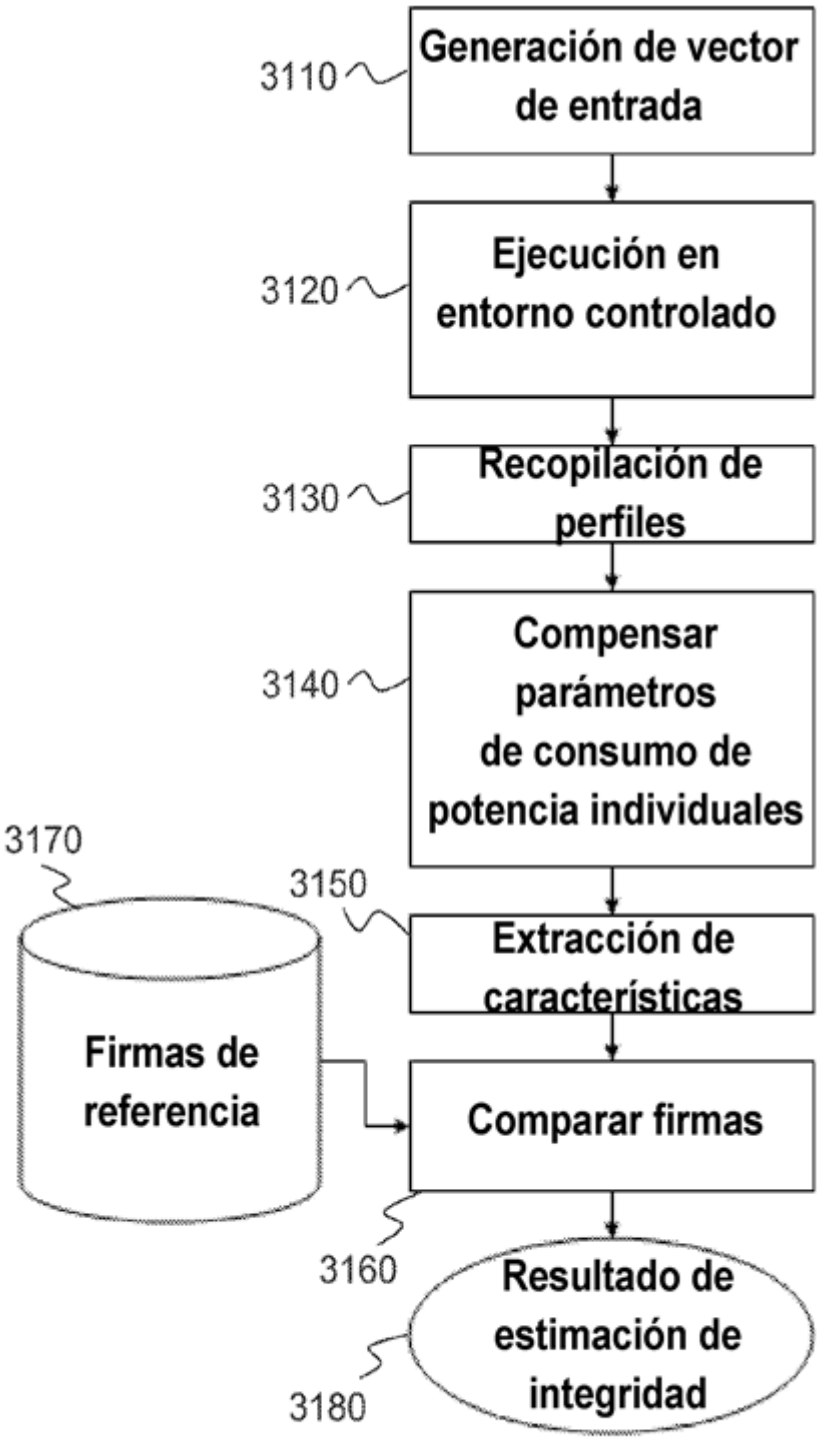

3110
Generación de vector de entrada
3120
Ejecución en entorno controlado
3130
Recopilación de perfiles
3140
Compensar parámetros de consumo de potencia individuales
3170
3150
Extracción de características
Firmas de referencia
Comparar firmas
3160
Resultado de estimación de integridad
3180

Figura 31

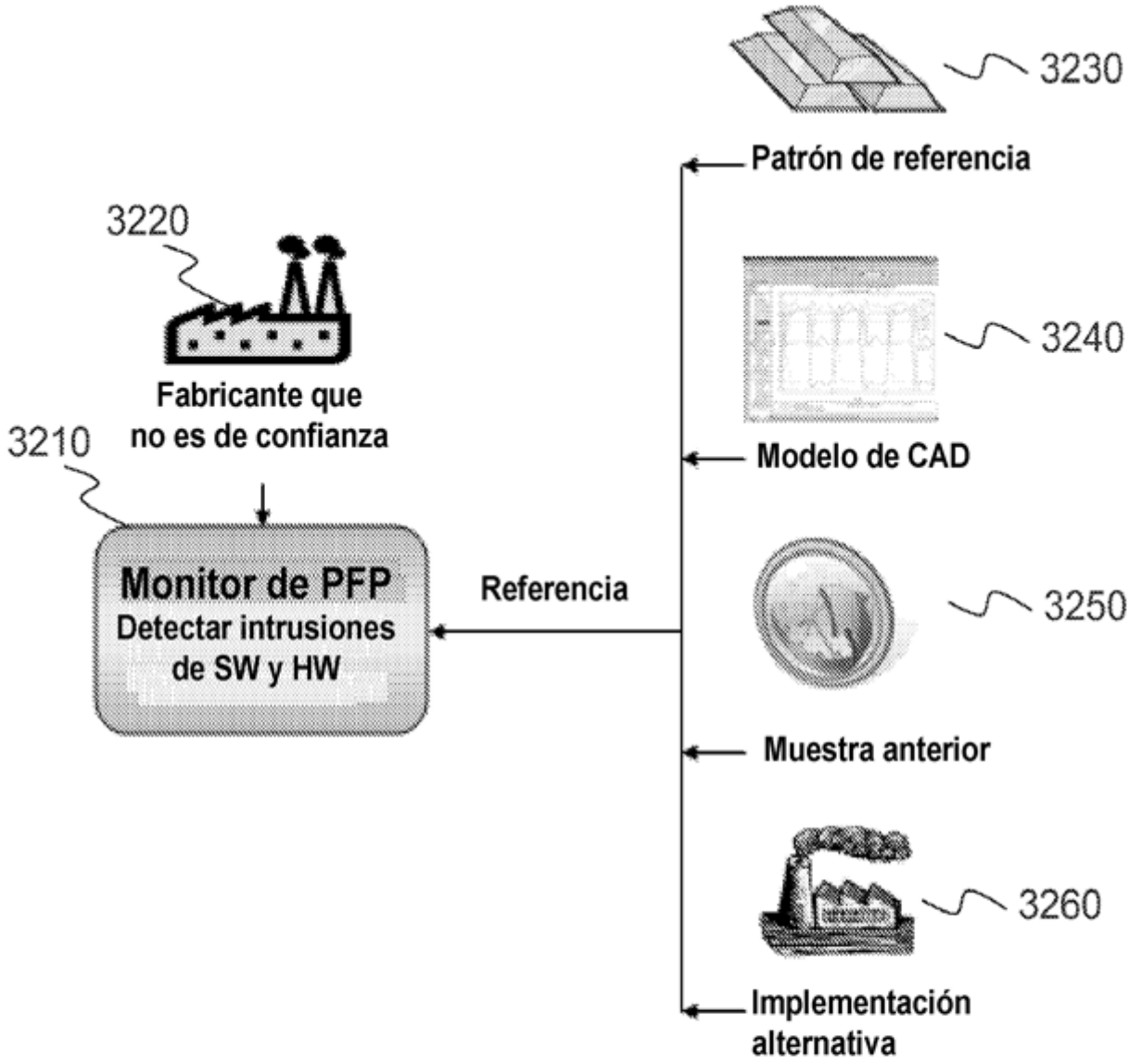

3230
Patrón de referencia
3220
Fabricante que
no es de confianza
3240
Modelo de CAD
3210
Monitor de PFP
Detectar intrusiones
de SW y HW
Referencia
3250
Muestra anterior
3260
Implementación
alternativa

Figura 32

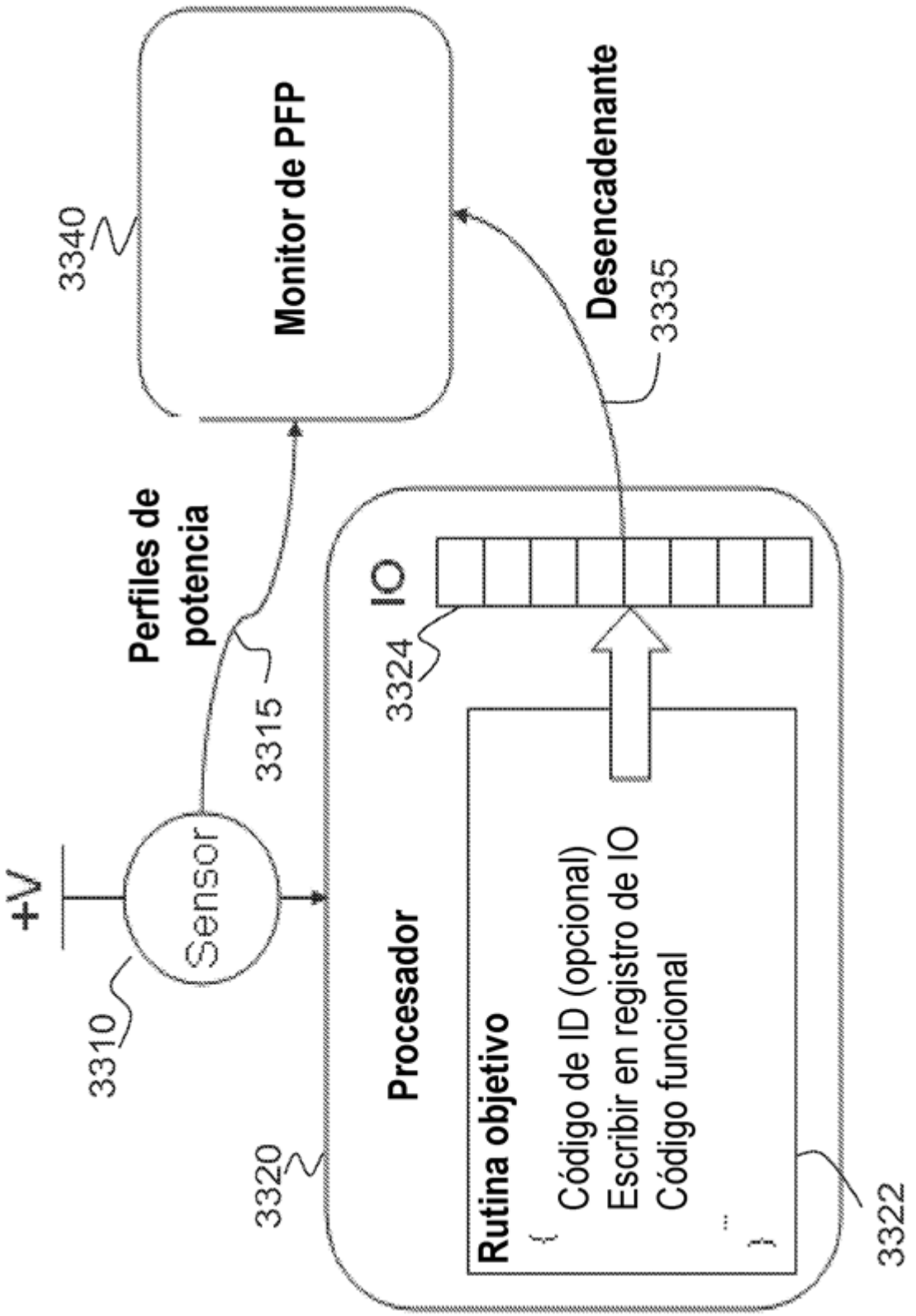

3340
Monitor de PFP
Desencadenante
3335
Perfiles de potencia
3315
IO
3324
+V
Sensor
3310
Procesador
3320
Rutina objetivo
{
Código de ID (opcional)
Escribir en registro de IO
Código funcional
...
}
3322

Figura 33

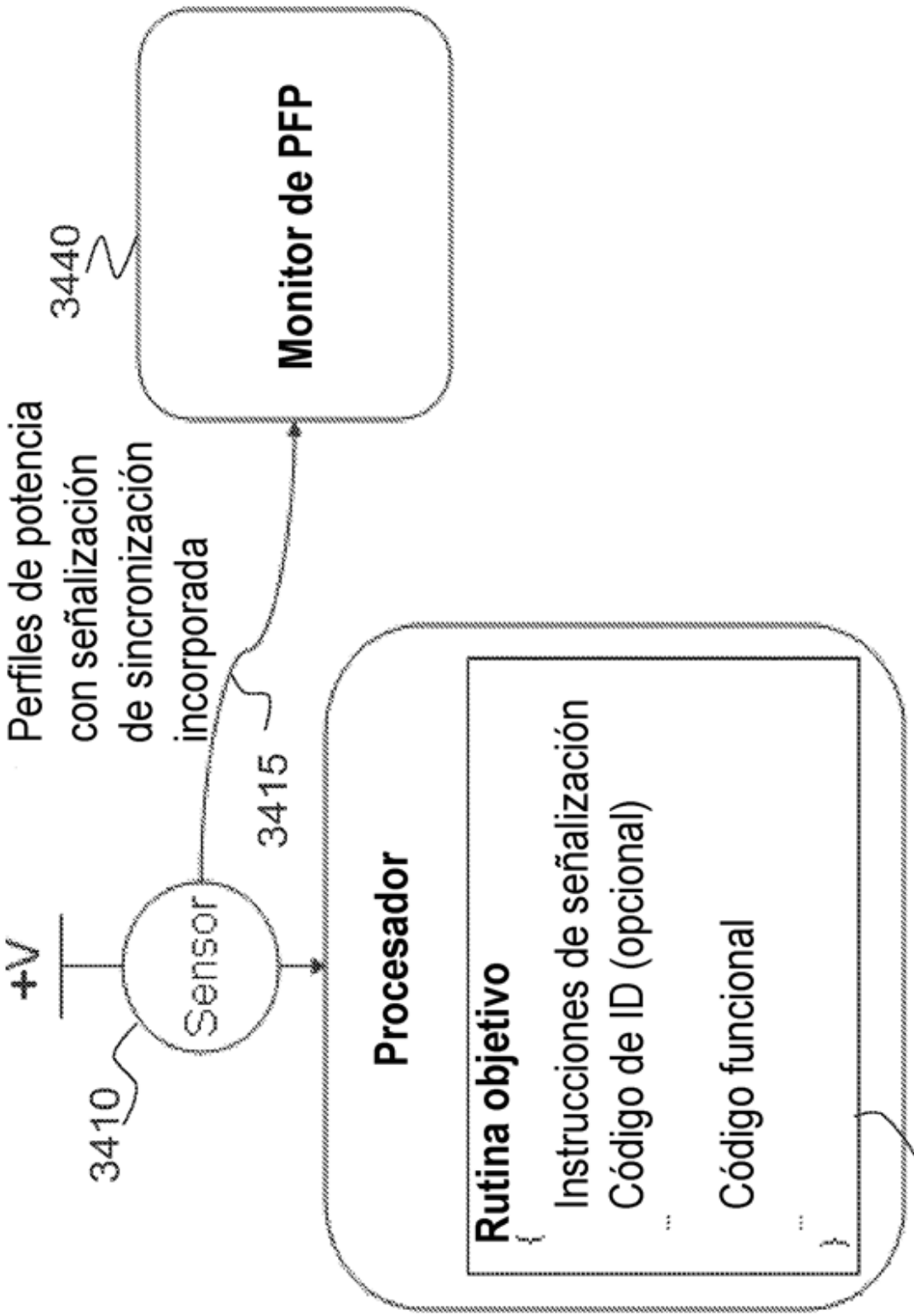

+V
3410
Sensor
3415
Perfiles de potencia con señalización de sincronización incorporada
3440
Monitor de PFP
Procesador
Rutina objetivo
Instrucciones de señalización
Código de ID (opcional)
...
Código funcional
...
3422

Figura 34

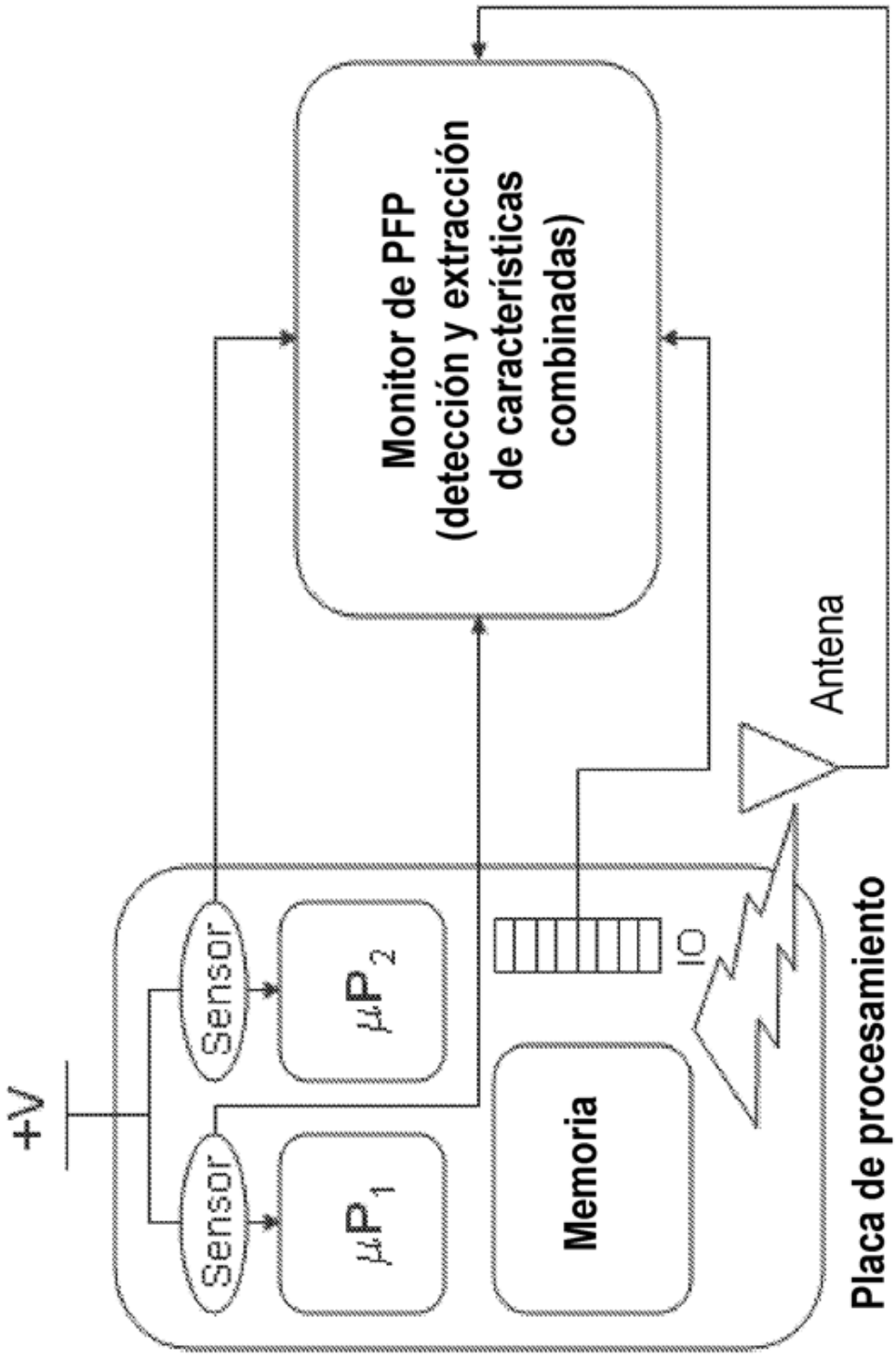

Monitor de PFP
(detección y extracción
de características
combinadas)
Antena
+V
Sensor
Sensor
μP_2
μP_1
IO
Memoria
Placa de procesamiento

Figura 35

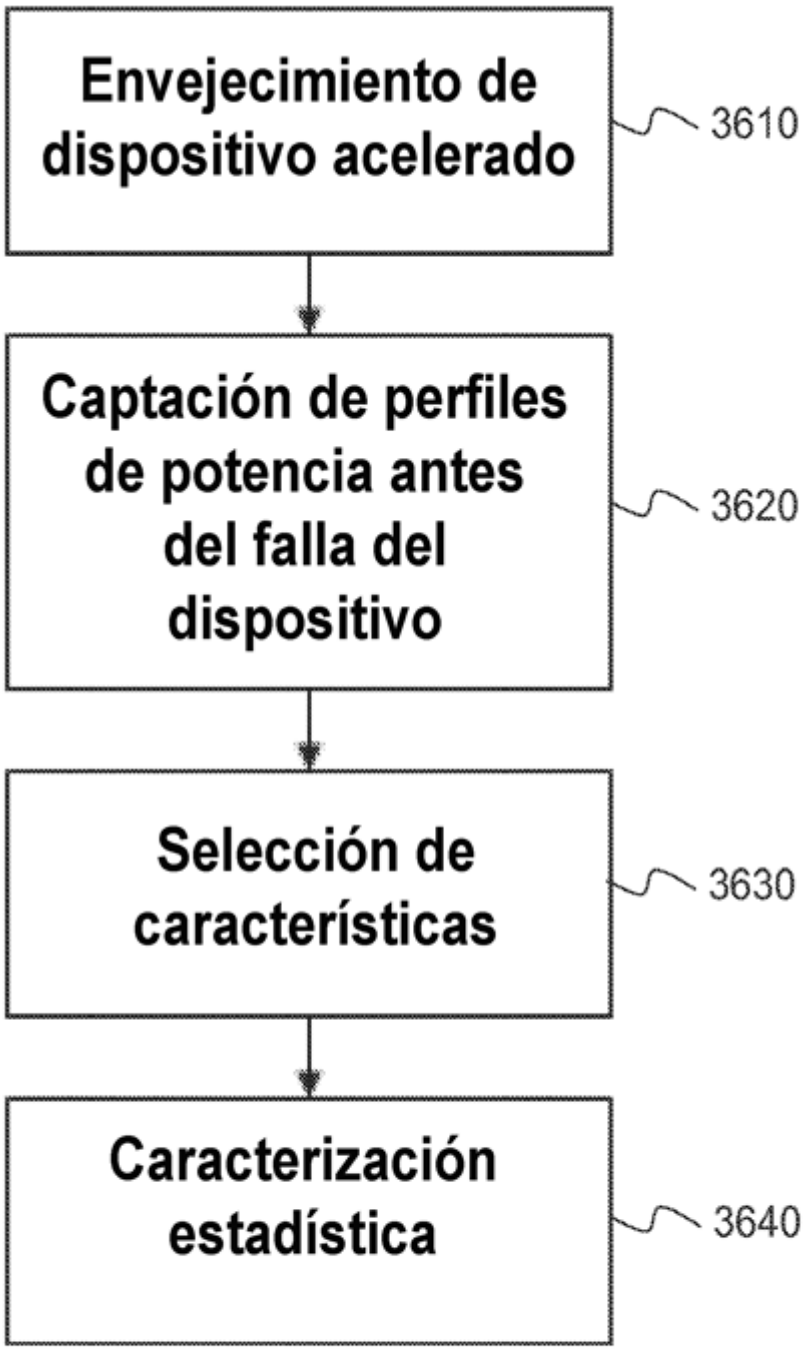

Envejecimiento de dispositivo acelerado
3610
Captación de perfiles de potencia antes del falla del dispositivo
3620
Selección de características
3630
Caracterización estadística
3640

Figura 36

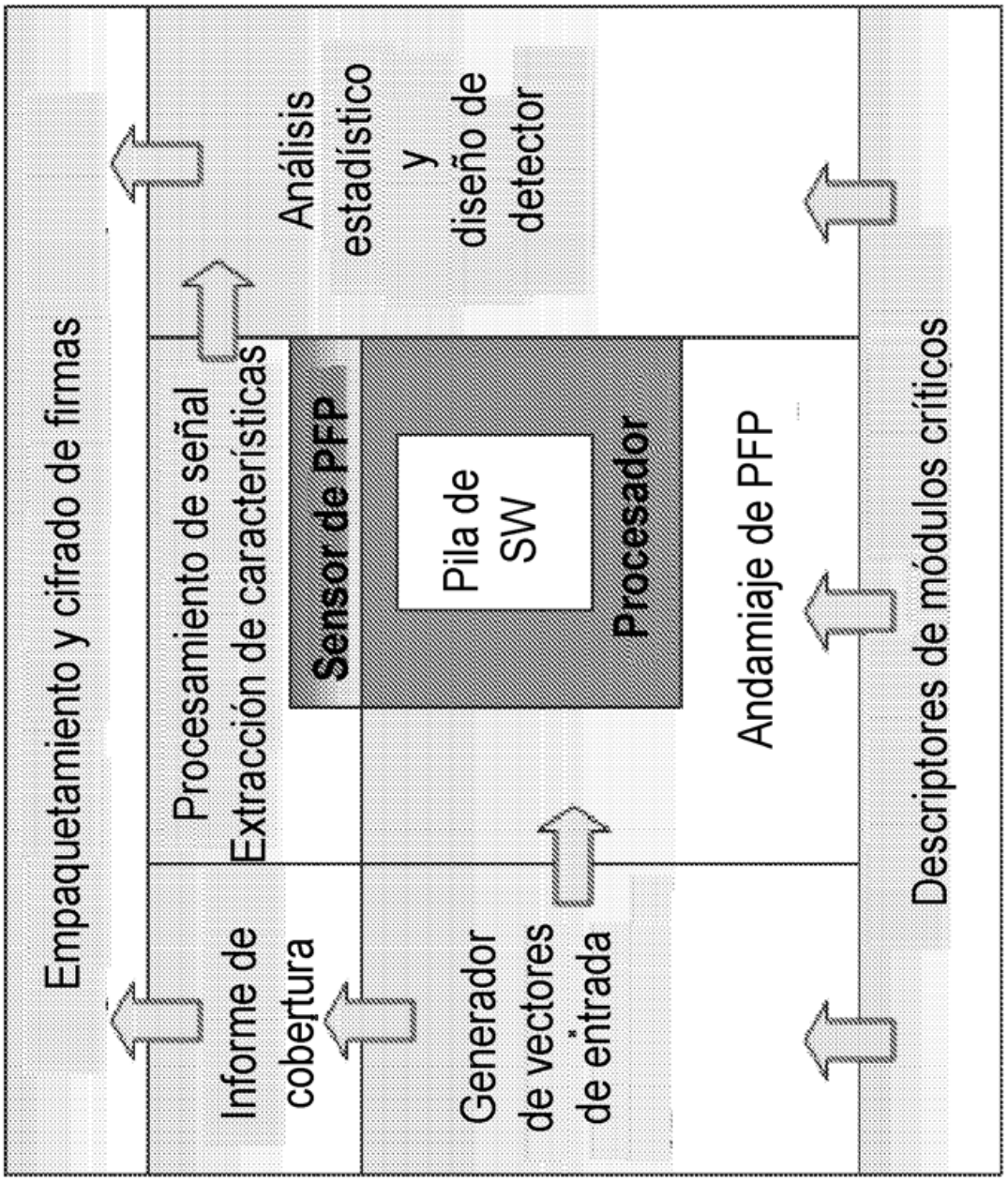

Empaquetamiento y cifrado de firmas
Procesamiento de señal
Extracción de características
Análisis
estadístico
y
diseño de
detector
Informe de
cobertura
Sensor de PFP
Pila de
SW
Procesador
Andamiaje de PFP
Generador
de vectores
de entrada
Descriptores de módulos críticos

Figura 37

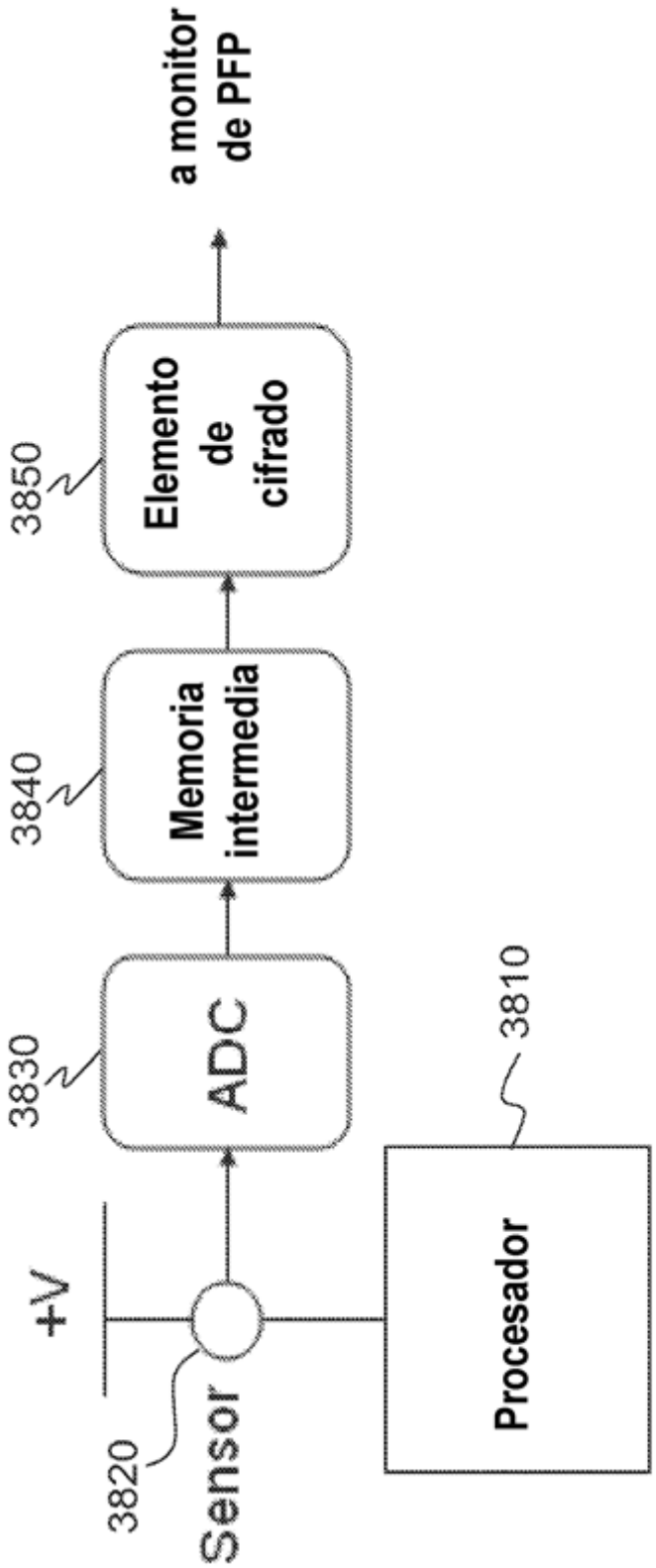

a monitor de PFP
3850
Elemento de cifrado
3840
Memoria intermedia
3830
ADC
3810
Procesador
+V
3820
Sensor

Figura 38

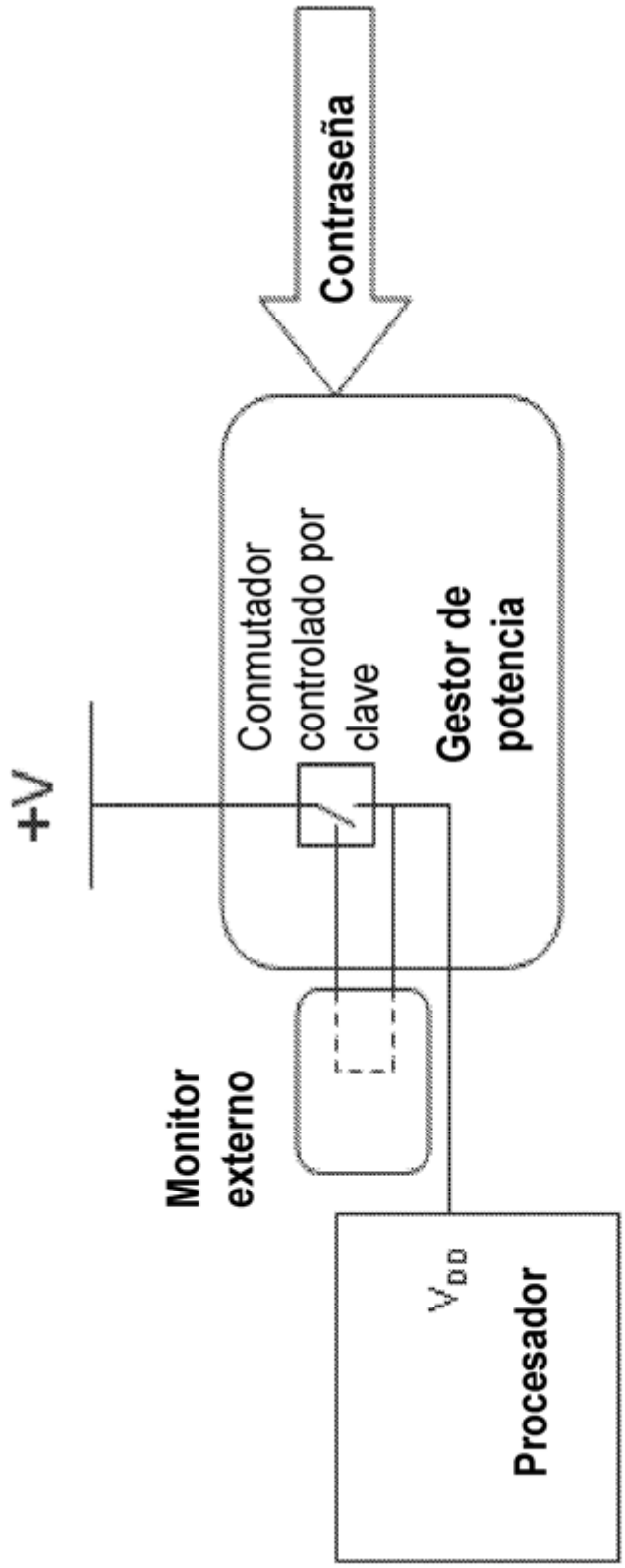

Contraseña
Conmutador controlado por clave
Gestor de potencia
+V
Monitor externo
V_{DD}
Procesador

Figura 39