

SELLADO DE DATOS CON UN ENCLAVE DE SELLADO

CAMPO DE LA INVENCION

Esta divulgación se refiere a sistemas informáticos seguros.

ANTECEDENTES DE LA INVENCION

Las regiones aisladas seguras o entornos de ejecución confiables proporcionan un contenedor seguro, denominado como un enclave en este documento, para ejecutar código confiable en una computadora que puede también tener código de menor confianza en una región fuera de la región aislada. Una región aislada de enclave incluye una porción de memoria que está protegida durante la ejecución de código que reside fuera del enclave. La memoria aislada puede contener tanto código como datos para el enclave, y la protección de esta memoria puede incluir restricciones que ejecutan código contenido en la memoria de enclave además de restricciones sobre la lectura o escritura en la memoria de enclave. Aspectos de seguridad de enclave, tales como aislamiento de memoria y restricciones de ejecución, se pueden hacer cumplir, por ejemplo, por medio de hardware en el procesador de computadora. La afirmación de software puede proporcionar confianza en la seguridad de aislamiento de un enclave particular y en el código de enclave que se carga dentro de la región de memoria aislada de un enclave particular. La afirmación puede proporcionar adicionalmente prueba de la integridad de la plataforma de hardware y software en la cual se ejecuta el enclave afirmado.

Los sistemas de enclave tales como el modo seguro virtual (VSM, Virtual Secure Mode, por sus siglas en inglés) de Microsoft y extensiones de defensa de software (SGX, Software Guard Extensions, por sus siglas en inglés) de Intel proporcionan seguridad en parte aislando un enclave de otro código que se ejecuta en cualquiera de modo de usuario o modo de *kernel*. Las garantías de integridad y confidencialidad pueden proporcionar un enclave con un nivel superior de confianza en la autenticidad del código que se ejecuta en un enclave, y confianza en la ejecución segura del código de enclave. Se puede proporcionar una garantía de integridad por medio de afirmación de software de un enclave particular. La afirmación de software puede incluir un cálculo de clave (*hash*) firmado criptográficamente del contenido (instrucciones y datos) dentro de un enclave y se puede combinar con datos acerca del entorno del enclave. Cuando un enclave se utiliza en combinación con un módulo de seguridad de hardware (HSM, Hardware Security Module, por sus siglas en inglés), tal como hardware que se ajusta a un estándar el grupo de computación confiable (TCG, Trusted Computing Group, por sus siglas en inglés), módulo de plataforma confiable (TPM, Trusted Platform Module, por sus siglas en inglés), el enclave puede proporcionar un nivel adicional de garantías de seguridad y confidencialidad.

Además de la seguridad proporcionada por aislamiento de un enclave local confiable de código local no confiable fuera del aislamiento del enclave, la afirmación de software de un enclave puede habilitar la informática confiable remota. La afirmación de un enclave remoto puede proporcionar confianza tanto en la integridad de la ejecución de las instrucciones en el enclave, como en la confidencialidad de los datos procesados por el enclave. Cuando la afirmación de un enclave remoto es

proporcionada por hardware de un fabricante confiable, un enclave puede ser confiable aun cuando el enclave reside en una computadora desconocida que es propiedad de y mantenida por un tercero no confiable. Este suele ser el caso, por ejemplo, cuando los recursos informáticos son rentados en un recurso informático basado en la nube de Internet.

BREVE DESCRIPCIÓN DE LA INVENCION

Se presentan métodos y sistemas para abstraer una plataforma de enclave. El método puede comprender recibir, por medio de una plataforma de abstracción de enclave, una primera solicitud de utilizar un enclave de un cliente de enclave. La primera solicitud puede ajustarse a un protocolo de abstracción de cliente. La primera solicitud puede convertirse a una segunda solicitud que se ajusta a un protocolo nativo de enclave asociado con una plataforma nativa de enclave. Después, se puede enviar la segunda solicitud a la plataforma nativa de enclave. La primera solicitud puede ser, por ejemplo, una solicitud para instanciar un enclave, una solicitud para verificar un reporte de afirmación de un enclave, una solicitud para llamar al interior de un enclave, o una solicitud para designar memoria que es compartida con el enclave y el cliente de enclave. La plataforma nativa puede ajustarse a la arquitectura de enclave de extensiones de defensa de software (SGX) de Intel y la plataforma nativa puede ajustarse a la arquitectura de enclave de modo seguro virtual (VSM) de Microsoft.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

La figura 1 representa un diagrama de bloques de alto nivel ejemplar de un sistema de enclave.

La figura 2 representa un proceso ejemplar para paso de mensajes con una garantía de confidencialidad.

La figura 3 representa un proceso ejemplar para paso de mensajes con una garantía de integridad.

La figura 4 representa un proceso ejemplar para paso de mensajes con una garantía de frecuencia de actualización.

La figura 5 representa un proceso ejemplar para afirmación de software de un enclave.

La figura 6 representa un protocolo de intercambio de claves de Diffie-Hellman (DKE, Diffie-Hellman Key Exchange, por sus siglas en inglés) ejemplar.

La figura 7 representa una cadena de confianza ejemplar para afirmación de software.

La figura 8 es un diagrama de bloques de interfaces de componentes de software para un sistema de enclave local ejemplar.

La figura 9 es un diagrama de bloques de interfaces de componentes de software para un sistema de enclave local ejemplar con una capa de abstracción.

La figura 10 es un diagrama de bloques de interfaces de componentes de software para un sistema de enclave remoto ejemplar con una capa de abstracción.

La figura 11 representa un entorno informático de propósito general ejemplar.

La figura 12 representa un diagrama de flujo ejemplar de un método para abstraer una plataforma

de enclave nativa.

La figura 13 representa un diagrama de flujo ejemplar de un método para abstraer una plataforma de enclave nativa.

La figura 14 representa un diagrama de flujo ejemplar de un método para llevar a cabo una operación de enclave con identidad abstracta de enclave.

La figura 15 representa un diagrama de flujo ejemplar de un método 1500 para llevar a cabo una operación de enclave con identidad abstracta de enclave.

La figura 16 representa un sistema ejemplar con equivalencia de identidad abstracta de enclave.

La figura 17 representa un diagrama de flujo ejemplar para procesamiento paralelo con dos enclaves equivalentes.

La figura 18 representa un diagrama de flujo ejemplar para procesamiento en serie con dos enclaves equivalentes.

La figura 19 es un diagrama de bloques de un sistema de sellado de datos distribuido ejemplar.

La figura 20 es un diagrama de flujo ejemplar para sellado y desellado de datos distribuidos.

La figura 21 es un diagrama de bloques de un enclave de bóveda de claves ejemplar.

La figura 22 es un diagrama de flujo ejemplar para algunas operaciones de enclave de bóveda de claves.

La figura 23 es un diagrama de flujo ejemplar para operación de enclave de bóveda de claves con una clave de bóveda bloqueada.

DESCRIPCIÓN DETALLADA DE LA INVENCIÓN

Se divulga un modelo de abstracción para enclaves que simplifica el desarrollo de clientes de enclave y del software que se ejecuta dentro de un enclave. Un modelo de abstracción puede ser una simplificación y unificación de arquitecturas de plataforma de enclave nativa, tales como SGX de Intel y VSM de Microsoft. Un componente de software de capa de abstracción puede traducir la comunicación entre un cliente de enclave y una o más plataformas nativas, entre software dentro de un enclave y una o más plataformas de enclave nativas, y entre software dentro de un enclave y un cliente de enclave. Tal plataforma de abstracción puede proporcionar el beneficio de permitir una sola versión de software de enclave y software de cliente de enclave a ejecutarse en múltiples plataformas de enclave nativas, tales como SGX y VSM. Además de simplificar la tarea de escribir software para enclaves y clientes de enclave, permite a los usuarios finales de enclaves ejecutar software de enclave y cliente de enclave en una computadora que soporte cualquier arquitectura de enclave nativa soportada sin tener que encontrar versiones de ambos el software de enclave y de cliente de enclave que estén a la medida para una plataforma de enclave nativa de una computadora específica.

Un modelo de abstracción de enclave puede incluir, por ejemplo, unidades básicas para: gestión del ciclo de vida de un enclave, afirmación local y remota de un enclave, sellado de datos en un enclave, transferencia de control de programa dentro y fuera de un enclave, y otras características de seguridad tales como contadores monotónicos y tiempo confiable. También se presenta una identidad abstracta o

en capas de un enclave que abstrae la identidad de un enclave más allá de un binario simple o un *hash* simple de contenido de un enclave. Se presentan interfaces de componente de software, tales como una interfaz de programación de aplicación (API, application programming interface, por sus siglas en inglés) o interfaz binaria de aplicación (ABI, Application Binary Interface, por sus siglas en inglés) para el desarrollo de enclaves y programas de cliente de enclave que utilizan unidades básicas de modelo de abstracción.

Una identidad abstracta puede incluir una identidad anidada o jerarquía de identidad que se puede utilizar para identificar de manera segura grupos de instancias de enclave. Una instancia de enclave en este documento puede referirse al mismo código binario de enclave cargado en un enclave en la misma máquina y tener la misma identidad. Por otro lado, una nueva versión del código binario, o el mismo código binario cargado en una máquina diferente, se podría considerar como una instancia diferente. Estas instancias diferentes también pueden tener la misma identidad en un nivel superior en una jerarquía de identidad. Una identidad de enclave abstraída permite que se identifiquen como relacionados grupos de binarios de enclave relacionados. Por ejemplo, a diferentes versiones del mismo enclave, tales como las versiones de binarios de enclave antes y después de arreglar un error, se les puede dar el mismo nombre, independientemente de la versión. En una capa superior de abstracción, a todos los enclaves en una familia de enclaves se les puede dar un solo nombre o identidad de familia. En este caso, todos los enclaves que llevan a cabo funciones relacionadas pero diferentes pueden ser identificados conjuntamente. Otras capas de identidad o agrupamientos de identidades se describen a continuación.

Cualquier capa de una identidad abstracta puede ser utilizada para diferentes operaciones criptográficas, tal como sellado de datos, afirmación de un enclave, o garantizar la frecuencia de actualización de los datos (por medio de contadores monotónicos). Por ejemplo, al sellar los datos producidos por una instancia de enclave en una identidad de nivel superior, esos datos pueden ser consumidos de manera segura posteriormente por una instancia de enclave diferente con la misma identidad de enclave de nivel superior. Al sellar los datos en, por ejemplo, una familia de enclave, cualquier instancia de enclave que sea un miembro de esa familia, y solamente los miembros de esa familia, será capaz de desellar los datos. La afirmación de una identidad de familia de una instancia de enclave proporciona certeza de que la instancia de enclave es un miembro de esa familia. Los contadores monotónicos vinculados a una abstracción de identidad pueden proporcionar garantías de frecuencia de actualización relacionadas con todas las instancias de enclave que son miembros de la identidad de abstracción.

El modelo de abstracción divulgado incluye interfaces de componentes de software, tales como una interfaz de programación de aplicación (API) o interfaz binaria de aplicación (ABI), que pueden simplificar el desarrollo de software de enclaves y anfitriones de enclave. Una API es un conjunto de definiciones de subrutina de programación, protocolos, y herramientas para crear software. Una API puede definir las entradas y salidas de un componente de software, tipos de datos utilizados por el componente de software, y la funcionalidad operación del componente de software independientemente

de cualquier implementación particular del componente de software. Una API puede estar definida en un lenguaje de computadora de nivel superior, tal como C, C++, C#, y similares. La definición formalizada de una API puede facilitar la interacción entre componentes de software, por ejemplo dos componentes de software escritos en diferentes momentos o por diferentes autores. Una API se puede formalizar en parte con un lenguaje de descripción de interfaz (IDL, Interface Description Language, por sus siglas en inglés) tal como el lenguaje de definición de interfaz de Microsoft (MIDL, Microsoft Interface Definition Language, por sus siglas en inglés) o el IDL de grupo de gestión de objetos (OMG, Object Management Group, por sus siglas en inglés). Una ABI también es una interfaz entre componentes de software, pero es una interfaz de código de objeto. Por ejemplo, una ABI pueden ser puntos de entrada (con las direcciones de instrucción) de código de objeto que resultan de compilar una implementación de código fuente de una API junto con protocolos para utilizar esos puntos de entrada, tal como protocolos que especifican registros de máquina que contienen argumentos cuando se llama a los puntos de entrada.

Además de permitir la interacción con diferentes niveles de identidad de enclave como se describió anteriormente, una API de enclave puede abstraer las diferencias entre arquitecturas de plataforma de enclave, por ejemplo entre arquitecturas para ejecución aislada segura proporcionada por extensiones de defensa de software (SGX) de Intel, modo virtual seguro (VSM) de Microsoft, y ARM TrustZone, virtualización cifrada segura (SEV, Secure Encrypted Virtualization, por sus siglas en inglés) de AMD, y arquitecturas basadas en matriz de puertas programable en campo (FPGAs, Field Programmable Gate Arrays, por sus siglas en inglés). Las APIs incluyen interfaces para una plataforma de enclave que abstrae algunos detalles de las arquitecturas de enclave abstraídas. Estas interfaces de plataforma de clave incluyen una API de anfitrión de enclave, una API de plataforma de enclave, y una API de afirmación remota. La API de anfitrión de enclave puede ser utilizada por un proceso de anfitrión no confiable para gestionar el ciclo de vida de un enclave así como proporcionar comunicación hacia y desde un enclave. La API de plataforma de enclave puede ser proporcionada por la plataforma de enclave confiable al enclave, y puede incluir unidades básicas de seguridad para afirmación, sellado, y comunicación con el código no confiable que se ejecuta en la computadora que aloja la computadora de enclave, así como soporte de tiempo de ejecución de núcleo tal como gestión de memoria y programación de tareas. La API de afirmación remota puede ser utilizada para llevar a cabo la afirmación remota, cuando un enclave y su cliente no están alojados en la misma computadora. Por ejemplo, la API de afirmación remota puede ser utilizada por un cliente local para verificar que los datos originados (o que fueron enviados) desde un enclave con una cierta identidad ejecutándose bajo aislamiento proporcionado por una plataforma de enclave en una computadora remota. Más generalmente, la API de afirmación remota puede ser utilizada para establecer canales de comunicación seguros entre un cliente local y el enclave remoto.

Los enclave generalmente proporcionan soluciones a problemas que son específicos para, y surgen de, el ámbito de la tecnología informática. Más específicamente, los enclaves proporcionan un mecanismo para segregar código confiable de código no confiable cuando ambos segmentos de código confiable y no confiable residen dentro del espacio de dirección de un solo procesador de computadora.

Por ejemplo, los enclaves proporcionan una solución de seguridad al problema de código potencialmente no confiable (tal como código que contiene potencialmente cualquiera de errores o virus) que se ejecuta en la misma computadora de propósito general que el código que debe acceder a datos sensibles o privados. Las modalidades de esta divulgación proporcionan soluciones mejoradas adicionales a tales problemas de seguridad que surgen del ámbito de la tecnología informática, que incluyen: simplificar el desarrollo de software al permitir que un solo enclave o cliente de enclave esté autorizado para múltiples plataformas de enclave nativas; simplificar la gestión informática corporativa al reducir el número de componentes de software que se deben personalizar a las características de hardware específicas de una computadora particular; y permitir nuevos escenarios informáticos seguros con sellado de datos distribuido, tal como distribuyendo procesamiento de enclave seguro a través de enclaves alojados en múltiples computadoras.

La figura 1 representa un diagrama de bloques de alto nivel de un sistema de enclave junto con algunas relaciones de confianza. El sistema de enclave 100 incluye un enclave 176 (llamado alternativamente un contenedor de enclave o entorno de ejecución seguro), que incluye una región aislada segura de memoria que contiene código 180 y datos 182. El código 180 puede ser público o privado, y los datos 182 pueden ser públicos o privados. Por ejemplo, los datos o código privados pueden pertenecer a (o ser privados para el propietario de los datos 142, mientras los datos o código públicos pueden haber sido proporcionados por otra parte, tal como el proveedor de software 148. En una modalidad, el código que se ejecuta en un contenedor de enclave 176 puede ser completamente público y no privado, mientras los datos que el código de enclave público utiliza como entrada o produce como salida pueden ser privados. En otra modalidad, es posible lo inverso, donde el código es privado mientras los datos son públicos. En todavía otra modalidad, ambos del código y datos de entrada pueden ser públicos, mientras la salida por la ejecución del código con los datos de entrada puede ser privada. Son posibles otras combinaciones de código, datos de entrada, y datos de salida públicos y privados.

El contenedor de enclave 176 está alojado en hardware confiable 172, que simultáneamente puede alojar software no confiable 174. Un propósito principal del sistema de enclave 100 puede incluir al menos un aspecto seleccionado de la lista que consiste: de mantener la integridad del código 180, mantener la confidencialidad del código 180, mantener la integridad de los datos 182, y mantener la confidencialidad de los datos 182. Proteger el contenido del enclave 176 del software no confiable 174 (p.ej., divulgación a software no confiable, modificación por software no confiable, o similar) puede ser una meta. El hardware confiable es construido por el fabricante 162, y es propiedad de y gestionado por el propietario de infraestructura 152.

El cliente de enclave 104 puede ser un proceso o programa fuera del contenedor de enclave para el cual el enclave 176 lleva a cabo sus cálculos con el código 180 y los datos 182. En una modalidad de enclave local, el cliente de enclave 104 también se puede estar ejecutando en el hardware confiable 172. En una modalidad de enclave remoto, el cliente de enclave puede ejecutarse en una computadora mientras el hardware confiable 172 es una computadora diferente, remota, conectada a la computadora del cliente de enclave por medio de una red. En el caso de enclave local, el proceso de cliente de enclave

también puede ser el proceso de anfitrión de enclave del contenedor de enclave 176 de modo que el proceso de cliente de enclave puede gestionar la creación del enclave local 176. En el caso de enclave remoto, el enclave 176 puede ejecutarse, por ejemplo, en una computadora basada en la nube de Internet donde el propietario de infraestructura 152 es un proveedor de servicio informático en la nube, y la computadora basada en la nube incluye hardware confiable 172 que es fabricado por el fabricante 162.

El cliente de enclave 104 puede incluir un método de configuración 106 para configurar un cálculo solicitado por el enclave 176. El método de configuración 106 puede incluir provocar la creación del contenedor seguro del enclave 176, provocando la instanciación del enclave (por ejemplo, al enviar una solicitud para software no confiable 174 para pedir la instanciación del enclave), que puede incluir copiar código binario al contenedor seguro, y provocar o solicitar un cálculo en el enclave, por ejemplo al llamar a un método en el código copiado al contenedor seguro. El cálculo solicitado puede incluir ejecutar código 180, y los datos 182 se pueden introducir en, o pueden ser un resultado de, el cálculo solicitado. La entrada de datos al cálculo solicitado se puede cifrar cuando está fuera del enclave, y los datos de entrada cifrados se pueden descifrar antes de ser utilizados dentro del enclave. Una vez que el enclave 176 ha completado la tarea solicitada, los datos que representan el resultado de la tarea son cifrados y enviados de vuelta al cliente de enclave 104. Cuando el cliente de enclave 104 recibe los resultados cifrados, un método de verificación 108 puede confirmar la integridad y frecuencia de actualización de los resultados recibidos. Un solo proveedor de software 148 puede proporcionar tanto el código 180 para ejecutar dentro del enclave 176, como al menos una porción del método de verificación 108 que se ejecuta como parte del cliente de enclave 104.

La confianza de un propietario de datos en la privacidad de una porción privada de los datos 182 y una porción privada del código 180, así como la confianza en la exactitud de los resultados producidos por el enclave 176, se pueden basar en relaciones de confianza. Por ejemplo, un propietario de datos 142 puede confiar en el cliente de enclave 104, que puede no estar ejecutándose dentro de un contenedor de enclave en sí. El propietario de datos puede además confiar en el proveedor de software 148 del enclave en sí. Y el propietario de datos puede confiar en el fabricante del hardware confiable 172. El hardware confiable 172 puede tomar muchas formas dependiendo de la arquitectura de enclave utilizada, y puede incluir un módulo de seguridad de hardware (HSM), donde el HSM se ajusta, por ejemplo, al estándar de módulo de plataforma confiable (TPM). El hardware confiable 172 puede incluir, por ejemplo, un TPM y podría comprender de otra forma solamente hardware. Por ejemplo, una implementación de hardware confiable 172 que utiliza la arquitectura de enclave VSM de Microsoft puede incluir un procesador de producto básico con instrucciones para instrucciones de virtualización de sistema operativo y un TPM. La arquitectura de enclave VSM de Microsoft puede incluir un hipervisor para gestionar particiones huésped (procesadores virtuales), y el hipervisor puede exponer interfaces de hiperllamada a las particiones huésped para permitir que las particiones huésped interactúen con el hipervisor. Un contenedor de enclave en la arquitectura VSM de Microsoft se puede implementar como un tipo especial de partición huésped. Un ejemplo de hardware confiable 172 con arquitectura de enclave SGX de Intel puede incluir un procesador con instrucciones específicas de enclave y funcionalidad de seguridad especiales.

Un enclave, tal como el enclave 176, puede proporcionar un entorno de ejecución aislado que puede proteger código o datos, tal como el código 180 y los datos 182, al proporcionar una región de memoria con restricciones sobre la lectura, escritura, o ejecución desde esa región protegida. Esta región de memoria protegida es un contenedor seguro para código y datos confidenciales. Las restricciones sobre la ejecución desde una región de memoria protegida del enclave puede incluir restricciones sobre transferencias de ejecución, tales como instrucciones de llamada o salto, entre código fuera del enclave al código dentro del enclave, y viceversa. Se pueden hacer cumplir diferentes restricciones entre llamar al interior del enclave desde el exterior del enclave y llamar al exterior del enclave desde el interior del enclave. El cumplimiento de estas transferencias de ejecución entre el interior y exterior de un enclave se puede proporcionar por medio de hardware, por ejemplo con tecnología de hardware de virtualización de producto básico o con hardware de especialidad tal como la plataforma SGX de INTEL.

La figura 2 representa un proceso 200 ejemplar para paso de mensajes con una garantía de confidencialidad. Una garantía de confidencialidad proporciona cierto nivel de certeza de que la comunicación entre dos partes, tales como Anne 210 y Ben 230 en este ejemplo, permanezca oculta de terceros cuando se pasan mensajes a través de un medio de comunicación público o no protegido tal como la red 218. En este ejemplo, a Anne 212 le gustaría enviar un mensaje confidencial a Ben 230. Un bloque de mensaje 212 que contiene datos confidenciales se cifra utilizando la clave pública 216 por medio de una operación de cifrado 214. La operación de cifrado 214 puede ser, por ejemplo, el estándar de cifrado avanzado (AES-GCM, Advanced Encryption Standard Galois/Counter Mode, por sus siglas en inglés), pero también puede ser cualquier operación de cifrado conocido por aquellos expertos en la materia de criptografía digital. El bloque cifrado 220 puede pasar a través de una red 218 a Ben donde el bloque cifrado 234 se descifra con una clave privada 236 para producir el bloque de mensaje 232 descifrado para Ben. Con una selección cuidadosa de claves y algoritmos de cifrado, como se conoce en la materia del cifrado de datos de computadora, el mensaje puede permanecer confidencial aun cuando pasa a través de una red pública.

La figura 3 representa un proceso 300 ejemplar para paso de mensajes con una garantía de integridad. Una garantía de integridad proporciona cierto nivel de certeza de que la comunicación entre dos partes, tales como Anne 310 y Ben 350 en este ejemplo, no se manipule o altere de otra manera cuando se pasan mensajes a través de un medio de comunicaciones público o no protegido tal como la red 340. En el ejemplo de la figura 3, Anne 310 envía un mensaje 314 a Ben 350 de modo que haya confianza de que el mensaje 314 no se manipule o corrompa cuando Ben 350 lo reciba. Para proporcionar esta garantía de integridad, un proceso de *hashing* seguro 316 opera en el mensaje 314 para producir un valor *hash* 318. El valor *hash* 318 es después firmado por el proceso de firma 320 utilizando la clave privada de Anne para producir una firma 342. La firma 342 puede ser enviada a través de una red pública 340 u otro proceso de comunicación no protegido junto con una copia del mensaje 314 como el mensaje 344. Ben recibe entonces un mensaje 354 para el cual a Ben le gustaría verificar la integridad, de modo que Ben pueda tener la confianza de que el mensaje 354 sea el mismo que el mensaje 314 que Anne envió después de haber pasado a través de una red no confiable 340. Para

verificar la integridad, el mensaje 354 recibido se procesa por medio del *hashing* seguro 356 que es idéntico al *hashing* seguro 316 para producir el valor *hash* 358. La firma recibida 342 se verifica por medio del proceso de verificación de firma 360 utilizando la clave pública de Anne. El valor *hash* 318 que se extrae de la firma 342 después se compara con el valor *hash* 358 para verificar 380 que las firmas son las mismas. Si son las mismas, el mensaje es aceptado 384 como que tiene integridad. Alternativamente, si el mensaje 314 fue alterado de alguna manera antes de ser recibido como el mensaje 354, entonces la firma no será correcta y el mensaje será rechazado 388.

En algunas modalidades, el *hashing* seguro 316 y el *hashing* seguro 356 pueden ser una función criptográfica de *hash*. Una función criptográfica de *hash* es una función unidireccional que mapea datos de tamaño arbitrario a una cadena de bits de un tamaño fijo (generalmente mucho más pequeño). La salida de una función de *hash* puede ser llamada un valor *hash* o simplemente un *hash*. Una buena función de *hash* será unidireccional de modo que será difícil determinar la entrada dimensionada arbitrariamente dada solamente la salida de *hash*. Con una buena función de *hash*, incluso un pequeño cambio en la entrada producirá un cambio en la salida.

Un sistema de comunicación puede combinar las garantías de confidencialidad e integridad. Por ejemplo, el cifrado de bloque de mensaje de la figura 2 se puede combinar con la firma de un *hash* de mensaje de la figura 3. El sistema de combinación puede requerir dos juegos de pares de claves públicas/privadas, uno para el remitente y uno para el destinatario. Un sistema de combinación puede utilizar una clave privada en el destinatario para descifrar un mensaje (la clave privada de Ben como en la figura 2), mientras utiliza otra clave privada para afirmar el *hash* de mensaje (clave privada de Anne como en la figura 3).

La figura 4 representa un proceso 400 ejemplar para paso de mensajes con una garantía de frecuencia de actualización. Una garantía de frecuencia de actualización proporciona cierto nivel de certeza de que cuando se envían múltiples mensajes de una parte a otra, tal como de Anne 410 a Ben 450 en este ejemplo, el mensaje recibido en el receptor es el mensaje más reciente. Una garantía de frecuencia de actualización se puede construir sobre una garantía de integridad, y puede evitar el ataque por repetición. Con una garantía de integridad, es difícil que un tercero con intenciones nefastas cree su propio mensaje y lo envíe a Ben de tal manera que Ben entienda que el mensaje fue creado por Anne. Sin embargo, un tercero puede tomar un mensaje realmente creado por Anne, tal vez observado en un momento ya que fue enviado a través de una red pública, y el tercero puede reenviarlo a Ben en otro momento posterior (esto es, repetir el mensaje). Ben determinará que el mensaje fue creado realmente por Anne (debido a que así fue), pero Ben no sabrá que Anne no es la persona que lo envió en ese momento. Esto puede ser llamado un ataque por repetición sobre Ben o Anne por el tercero. La figura 4 es una solución ejemplar para evitar los ataques por repetición al proporcionar una garantía de frecuencia de actualización utilizando nonces y marcas de tiempo. Un nonce es un número aleatorio de un solo uso, acoplado con un sistema para asegurar que el mismo número no se vuelva a utilizar como un nonce más de una vez. En algunos sistemas, un nonce puede ser simplemente un número creciente monotónicamente, de tal manera que cada número utilizado es mayor que todos los números anteriores

al mismo.

En la figura 4, el mensaje 414 de Anne puede ser enviado a través de una red pública 430 como el mensaje 436 junto con el nonce 434 y la marca de tiempo 432. El nonce 434 es generado por un generador de número pseudoaleatorio criptográficamente seguro (CSPRNG, Cryptographically Secure Pseudorandom Number Generator, por sus siglas en inglés) 426, y la marca de tiempo 432 se produce por medio de un reloj sincronizado 424. Hay muchos sistemas de CSPRNG que son conocidos por aquellos expertos en la materia de criptografía digital. El reloj sincronizado 424 en el lado de Anne de la red 430 se sincroniza con el reloj sincronizado 480 en el lado de Ben de la red. En el lado de Ben, cuando se recibe un mensaje 454, el nonce 434 de acompañamiento está almacenado en una memoria intermedia (caché) de nonces recientes 470. La frecuencia de actualización de ese mensaje 454 recibido se puede verificar con dos pruebas. Primero, se pone a prueba el nonce 434 en el cuadro 460 comparando el nonce 434 en la caché de nonces recientes 470 para determinar si el nonce 434 recién recibido ha sido visto antes. Si el nonce 434 recibido ha sido visto antes, el mensaje 454 es rechazado como un mensaje repetido en el cuadro 468. Si el nonce 434 recibido no ha sido visto antes, se determina que el mensaje está bien en el cuadro 464 para la primera prueba. En segundo lugar, la marca de tiempo 432 recibida se compara con el reloj sincronizado local 490. Si la marca de tiempo es reciente, se determina que el mensaje 454 es aceptable en el cuadro 494, de otra manera, el mensaje 454 se rechaza como expirado en el cuadro 498. La cantidad de retraso tolerado cuando se determina si una marca de tiempo reciente es de hecho reciente en el cuadro 490 puede depender del sesgo de reloj esperado entre el reloj sincronizado 424 y el reloj sincronizado 480 y los retrasos de tiempo en el procesamiento y transmisión del mensaje a través de la red 430.

Un sistema de comunicación puede combinar una garantía de frecuencia de actualización cualquiera o ambos de una garantía de confidencialidad, como en la figura 2, y una garantía de integridad, como en la figura 3. En un sistema que combina todos de los tres, el mensaje 436 enviado a través de la red 430 será una versión cifrada del mensaje 414 original de Anne, y incluiría una firma que comprende un *hash* firmado del mensaje 414 junto con la marca de tiempo 432, el nonce 434, y el mensaje 436.

La figura 5 representa un proceso 500 ejemplar para afirmación de software de un enclave. La afirmación de software cuando se combina con un protocolo de acuerdo de clave tal como el de la figura 6, puede asegurar un verificador que tiene establecido un secreto compartido con una pieza específica de software que está alojada en un contenedor aislado creado por hardware confiable. En la modalidad de la figura 5, un cliente de enclave 510 (el verificador de afirmación) puede desear utilizar los servicios informáticos seguros de enclave en la plataforma confiable 530. La plataforma confiable 530 está alojada en una computadora (no representada) de tal manera que la plataforma confiable 530 puede comprender un subconjunto de la computadora anfitriona. La plataforma confiable 530 puede comprender elementos de hardware y elementos de software de la computadora anfitriona. El enclave comprende contenedor de enclave seguro 536 y el código y datos dentro del mismo, tales como el código y datos públicos 538 y el código y datos secretos 542.

Se combinan tres procesos en el proceso 500 ejemplar: se llevan a cabo un proceso de intercambio de clave que produce una clave compartida (SK, Shared Key, por sus siglas en inglés); un proceso de afirmación para afirmar al cliente de enclave 510 del enclave en la plataforma confiable 530; y un cálculo seguro. La clave compartida SK del primer proceso se utiliza para comunicar entradas y salidas del cálculo seguro. En el intercambio de clave, el cliente de enclave calcula g^A , almacenada en el cuadro 512, a partir de la clave privada A del cliente de enclave y una función generadora g, por ejemplo como se describe posteriormente para el protocolo de intercambio de clave de Diffie-Hellman (DKE, Diffie-Hellman key exchange, por sus siglas en inglés) de la figura 6. El g^A calculado se envía después en el mensaje 520 a la plataforma confiable 530. El mensaje 520 puede ser enviado de manera segura sin cifrado y antes de que la afirmación esté completa. El software dentro del contenedor de enclave seguro 536 puede utilizar una clave privada de enclave B para calcular g^B utilizando la misma función generadora g. Ambos B y g^B pueden ser almacenados en el contenedor de enclave en el cuadro 540.

Para afirmar la identidad del enclave (para proporcionar certeza acerca del código que se está ejecutando dentro del contenedor de enclave seguro 536), se envía un mensaje de afirmación 522 al cliente de enclave 510. Un mensaje de afirmación puede ser un mensaje especial firmado por integridad como en la figura 3. El mensaje especial puede contener información de identidad acerca del enclave. Cuando se combina con DKE como en la modalidad de la figura 5, el mensaje especial también puede incluir parámetros de intercambio de clave. En la modalidad de la figura 5, el estado inicial 538 del contenedor de enclave seguro 536 del código público y datos públicos se utilizan como una identidad de clave, aunque son posibles otras identidades. En lugar de enviar todo el estado inicial 538 en un mensaje de afirmación, se envía en su lugar un *hash* del estado inicial, $M = \text{Hash}(\text{Estado Inicial})$. El mensaje de afirmación 522 incluye el contenido de mensaje (M y g^B), y una firma del contenido de mensaje ($\text{Sign}_{AK}(\text{Hash}(g^B), M)$). La firma del contenido de mensaje se puede crear, por ejemplo, por medio de software dentro del contenedor de enclave seguro 536 que pide a la plataforma confiable 530 que aloja el enclave que afirme en un *hash* del g^B calculado y la identidad del enclave. La plataforma confiable 530 puede hacer eso al proporcionar una firma que utiliza la clave de afirmación (AK, Attestation Key, por sus siglas en inglés) de plataforma 532 para producir ($\text{Sign}_{AK}(\text{Hash}(g^B), M)$). En este ejemplo, la identidad de enclave es un *hash* M del estado inicial 538, pero son posibles otras declaraciones de identidad. Esta firma de afirmación ($\text{Sign}_{AK}(\text{Hash}(g^B), M)$) une el valor g^B a la identidad de enclave M y también une ambos de g^B y M a la plataforma confiable 530. El cliente de enclave 510 puede entonces verificar el mensaje de afirmación verificando la firma de afirmación y la identidad de enclave. La firma puede ser verificada como en la figura 3 utilizando una clave pública que corresponde a la clave de afirmación AK. Verificar la firma puede proporcionar una garantía de integridad de la identidad de enclave en el mensaje de afirmación. La identidad de enclave se puede verificar comparando la información de identidad en el mensaje de afirmación con un valor de identidad conocido de manera independiente. Por ejemplo, si la información de identidad en el mensaje de afirmación es un *hash* del estado inicial del enclave, el cliente de enclave 510 puede conocer el *hash* del estado inicial, o puede ser capaz de calcular tal *hash* a partir de un estado inicial conocido, y este valor se puede comparar después con el valor de identidad

proporcionado en el mensaje de afirmación.

Para producir la clave compartida SK, tanto el cliente de enclave 510 como el código dentro del contenedor seguro 536 pueden generar g^{AB} (la función generadora g aplicada al producto de A por B) que puede servir como una clave compartida SK. La clave compartida SK se puede utilizar para cifrar mensajes para confidencialidad entre el cliente de enclave 510 y el enclave, por ejemplo para enviar datos de entrada a , y datos de salida d , el código dentro del contenedor de enclave 536. Observar que la clave compartida se produce de manera independiente en cada lado del canal de comunicación en los cuadros 540 y 514 sin que ninguno del cliente de enclave o el enclave sepa la clave privada del otro. Por ejemplo, en la modalidad de la figura 5, el código y datos secretos se pueden proporcionar de manera segura por el cliente de enclave 510 al cifrar el código y datos secretos con la clave compartida SK establecida previamente, produciendo $Enc_{SK}(\text{código/datos secretos})$ antes de enviarlos en un mensaje 524 a la plataforma confiable 530. En otras modalidades, el código y datos secretos 542 ejecutados en o utilizados por un contenedor de enclave seguro 536 pueden originarse desde otras ubicaciones. Se puede llevar a cabo un cálculo seguro dentro del contenedor de enclave seguro 536 utilizando el código y/o datos secretos 542 para producir un resultado de cálculo 544. Los resultados de cálculo 516 se pueden entonces comunicar de manera segura de vuelta al cliente de enclave 510 por el cifrado de los resultados con la clave compartida SK ($Enc_{SK}(\text{resultados})$) antes de enviarlos al cliente de enclave en el mensaje 526.

El proceso de la figura 5 descrito anteriormente proporciona una garantía a un cliente de enclave de que se está comunicando con un enclave “real” de una cierta identidad, y que el enclave está protegido por la plataforma de enclave. No proporciona ninguna garantía al código dentro del contenedor de enclave acerca de la entidad en el otro lado del canal de comunicación. En una modalidad alterna (no representada), tal garantía acerca del cliente de enclave se puede proporcionar ejecutando el cliente de enclave como un enclave en sí. En esta modalidad alterna, el cliente de enclave puede pedir a la plataforma de enclave confiable una firma en el *hash* de g^A , la cual se puede verificar después por el otro enclave.

La afirmación se puede hacer de manera local o remota. En la figura 5, el cliente de enclave 510 puede o no residir en la misma computadora que la plataforma confiable 530, de tal manera que las comunicaciones entre el cliente de enclave 510 y la plataforma confiable 530 pueden ocurrir dentro de una sola computadora (por ejemplo pasando memorias intermedias (buffers) de datos entre diferentes procesos en la misma computadora), o pueden ocurrir a través de una red de ordenadores que conecta el cliente de enclave 510 a la plataforma confiable 530. La afirmación local se puede llevar a cabo cuando el cliente de enclave 510 y la plataforma confiable 530 están alojados en la misma computadora local. El artefacto, o resultado, de la afirmación local es llamado un reporte de afirmación, y es $Sign_{AK}(\text{Hash}(g^A), M)$ en el ejemplo de la figura 5. La afirmación remota puede ocurrir cuando el cliente de enclave 510 y la plataforma confiable 530 están alojados en diferentes computadoras. El artefacto, o resultado, de la afirmación remota es llamado una cita de afirmación, que en algunos casos puede ser muy similar o idéntico a un reporte de afirmación local. En otros casos, una cita de afirmación puede incluir un artefacto

adicional de confianza relacionada con la computadora o plataforma nativa que proporciona la cita. Tal artefacto adicional de confianza puede incluir un certificado de estado del anfitrión tal como un registro de TCG asociado con un TPM. La afirmación de un enclave se puede llevar a cabo en cualquier capa de una identidad de enclave. Un enclave puede tener una identidad anidada o jerárquica, y la afirmación en niveles superiores de identidad puede afirmar que un enclave instanciado es un miembro de un grupo progresivamente más grande de instanciaciones de enclave potenciales conforme aumenta el nivel de identidad. Los niveles superiores pueden corresponder a un superconjunto de las instanciaciones de enclave potenciales de nivel inferior. Una jerarquía ejemplar de identidades puede incluir, desde el nivel más bajo, más específico a los niveles más altos, menos específicos puede ser: ExactHash, InstanceHash, ImageID, FamilyID, y AuthorID.

La identidad de un enclave se puede derivar de los archivos binarios (los binarios de enclave) cargados en el contenedor seguro de enclave. Un binario de enclave puede ser firmado por su autor utilizando una clave privada del autor. Para una afirmación de nivel ExactHash, el estado inicial 538 utilizado para calcular un reporte de afirmación (la entrada a una función de *hash* para producir un reporte de afirmación) puede incluir todo el contenido de cada archivo binario cargado en el contenedor seguro 536, excepto para binarios asociados con la plataforma confiable 530.

La afirmación en el nivel de identidad InstanceHash puede incluir un subconjunto del estado inicial 538. El subconjunto se puede especificar al momento de que los archivos binarios de enclave (los archivos binarios) son cargados en el contenedor seguro 536 donde fueron firmados originalmente por el autor de esos archivos binarios de enclave. Por ejemplo, un primer (o primario) archivo binario de enclave puede incluir una lista de identidades de otros archivos binarios de enclave de los cuales depende el primer archivo binario de enclave. Para cada identidad enlistada, se puede incluir un marcador en el primer archivo binario para indicar si cada archivo binario enlistado es medido o no por la función de *hash* para producir un reporte de afirmación de InstanceHash.

La afirmación en niveles superiores de un ID de enclave puede no incluir ejecutar todo el contenido de cualquier binario de enclave a través de una función de *hash*. En su lugar, sólo se puede ejecutar una estructura de datos de IDs a través de la función de *hash*. Por ejemplo, un archivo binario de enclave puede incluir una lista de identificadores de enclave de nivel superior, tales como identificadores universalmente únicos (UUIDs, Universally Unique Identifiers, por sus siglas en inglés), que indican: un ID de imagen (ImageID) único para ese archivo binario de enclave particular; un ID de familia (FamilyID) único para un grupo de archivos binarios de enclave que incluye ese archivo binario de enclave particular y que son de autoría por el mismo autor; y un ID de autor (AuthorID) único para un grupo de familias de archivos binarios de enclave todos de los cuales son de autoría por el mismo autor. El ImageID, FamilyID, y AuthorID pueden ser asignados por el autor de un binario de enclave al momento en que el binario se firmó originalmente. La suplantación de identidad desde enclave se puede prevenir cuando el cliente de enclave puede acceder a los binarios de enclave y verificar la firma del autor en esos binarios utilizando la clave pública del autor (o una clave pública asociada con el autor). Esto verifica la integridad de los binarios de enclave, incluyendo cualquier identidad de nivel superior asignada por el autor, como

habiendo sido creados por ese autor de enclave.

La figura 6 representa un protocolo de intercambio de claves de Diffie-Hellman (DKE) 600 ejemplar. DKE es un proceso ejemplar para establecer una clave K compartida a través de un canal de comunicación que tiene sólo una garantía de integridad; se pueden utilizar otros procesos para crear claves compartidas conocidos en la materia de criptografía digital. En el ejemplo de DKE de la figura 6, se comparte una clave K secreta entre Anne 610 y Ben 650 sin comunicar nunca de manera explícita K a través de un medio de comunicación público (inseguro) entre Anne 610 y Ben 650. Antes de que el proceso inicie, 1) un número primo grande p y 2) un generador g en Z_p se pueden establecer y conocer por ambos Anne y Ben. El proceso inicia entonces con ambos Anne y Ben escogiendo un número aleatorio entre 1 y p . El número aleatorio de Anne es A , escogido en el cuadro 612, y el número aleatorio de Ben es B , escogido en el cuadro 652. Anne utiliza su número aleatorio para calcular $g^A \bmod p$ en el cuadro 614, y transmite esa cantidad en el cuadro 616 que es recibida por Ben en el cuadro 656. Ben utiliza su número aleatorio para calcular $g^B \bmod p$ en el cuadro 654, que se transmite en el cuadro 656 a Anne y recibe en el cuadro 618. Anne puede producir la clave compartida K como $(g^B \bmod p)^A = g^{AB} \bmod p$ en el cuadro 620 y Ben puede producir la clave K compartida como $(g^A \bmod p)^B = g^{AB} \bmod p$ en el cuadro 660. Con el fin de evitar ataques de hombre en el medio, la comunicación entre Anne y Ben a través de una red no protegida de los cuadros 616 y 658 puede incluir una garantía la integridad de mensaje, por ejemplo creada utilizando un proceso tal como el de la figura 3.

La figura 7 representa una cadena de confianza 700 ejemplar para afirmación de software. La cadena de confianza en la afirmación de software puede estar arraigada en una clave de firma propiedad del fabricante de la plataforma confiable, tal como la plataforma confiable 530 de la figura 5. La plataforma confiable puede incluir componentes de hardware tales como un procesador seguro o un módulo de seguridad de hardware (HSM), y por lo tanto el fabricante puede ser un proveedor de hardware de computadora y también puede proporcionar software para la plataforma confiable. El fabricante puede ser confiable por el verificador 702, y el verificador puede conocer la clave pública PubRK 732 de la clave de raíz del fabricante 736. El cliente de enclave 510 de la figura 5 es un ejemplo de un verificador 702 que puede desear tener certeza acerca de un contenedor seguro 708. El fabricante puede actuar como una autoridad certificada y proporcionar cada instancia de plataforma confiable que produce, por ejemplo cada procesador seguro, con una clave de afirmación 722 única, que se utiliza para producir firmas de afirmación. El fabricante también emite un certificado de aprobación 728 para cada clave de afirmación 722. La clave de raíz del fabricante 736 incluye una clave privada PrivRK 734 que se utiliza para firmar el certificado de aprobación 728. La firma del certificado de aprobación proporciona una garantía de integridad, por ejemplo como se muestra en la figura 3.

El certificado de aprobación 728 incluye la clave pública PubAK 724 de la clave de afirmación 722. El certificado de aprobación 728 puede indicar que la clave de afirmación 722 se ha de utilizar para afirmación de software, y se puede comunicar al verificador 702. El verificador puede ser cualquier entidad que desee verificar una afirmación del contenedor seguro 708, por ejemplo el verificador 702 puede ser el cliente de enclave 510 de la figura 5 que desea asegurar que el cálculo se haga dentro del

contenedor seguro 708. El verificador 702 puede inspeccionar el certificado de aprobación utilizando la PubRK 732 para verificar la integridad y origen del certificado de aprobación. El verificador también puede extraer la PubAK 724 del certificado de aprobación. El certificado de aprobación puede estar asociado con una política de certificación, que puede requerir que la clave de afirmación 722 se utilice solamente para producir firmas de afirmación y que la clave privada PrivAK 726 de la clave de afirmación 722 se mantenga exclusivamente en almacenamiento que esté separado de la memoria de computadora generalmente accesible de la plataforma confiable, tal como en el almacenamiento del hardware a prueba de manipulación 730. El hardware a prueba de manipulación puede ser, por ejemplo, hardware que se ajusta un estándar de módulo de plataforma confiable (TPM).

Un contenedor seguro 708 se puede instanciar en la plataforma confiable 736. La instanciación del contenedor seguro 708 puede incluir definir un espacio de memoria aislado del contenedor seguro que esté restringido del acceso por procesamiento inseguro. El procesamiento inseguro puede incluir, por ejemplo, acceso desde fuera de la plataforma confiable pero en la computadora que aloja la plataforma confiable, o acceso desde dentro de otro contenedor seguro dentro de la plataforma confiable. La instanciación del contenedor seguro 708 también puede incluir cargar código y datos públicos en el contenedor seguro, por ejemplo el estado inicial 535 de la figura 5.

El contenedor seguro 708 instanciado puede intercambiar claves con el verificador 702 para establecer una clave compartida para comunicación confidencial. El proceso de intercambio de clave puede ser el proceso de intercambio de clave de la figura 5 o el proceso de DKE de la figura 6. El verificador envía el mensaje de intercambio de clave 1 704 a la plataforma confiable 736, por ejemplo como en el cuadro 616 de la figura 6, y la plataforma confiable 736 envía el mensaje de intercambio de clave 2 706 de vuelta al verificador 702, por ejemplo como en el cuadro 658 de la figura 6.

La firma de afirmación 710 se puede crear después de que se instancia el contenedor seguro 708 y el intercambio de clave está completo. El contenedor seguro 708 instanciado se puede medir ejecutando una función de *hash* criptográfico en todo o parte del contenedor seguro. Esto puede incluir ejecutar la función de *hash* a través del contenido de la memoria aislada, y archivos binarios que están cargados en la memoria aislada, cualquier otra memoria asociada con la plataforma confiable que se utilice o afecte durante la instanciación del contenedor seguro, o cualquier subconjunto o porción de estos. La salida de ejecutar esta función de *hash* es la medición 712, que es parte de la firma de afirmación 710. Un *hash* criptográfico de los mensajes de intercambio de clave 704 y 706 también se puede incluir con la firma de afirmación 710, representado como los datos 714. La medición 712 y los datos 714 se pueden firmar utilizando la clave privada de afirmación PrivAK 726. La firma de afirmación se puede enviar entonces al verificador 702 junto con la medición 712 y los datos 714. El verificador puede verificar la integridad de la firma de afirmación utilizando la PubAK 724 del certificado de aprobación, lo cual, en el ejemplo de la figura 7, también permite la verificación de la integridad de la medición 712 y los datos 714. El verificador 702 puede verificar la integridad del contenedor seguro 708 comparando la medición 712 contra un resultado esperado (el resultado esperado determinado, por ejemplo, llevando a cabo localmente el mismo *hash* de la medición 712), y verificar que la firma de

afirmación se creó para esta instancia de trayectoria de comunicación de verificador 702 particular inspeccionando los datos 714 (por ejemplo, debido a que el *hash* de los datos 714 está vinculado al mensaje de intercambio de clave 2 706). Después de estas operaciones de verificación y la verificación del certificado de aprobación anterior, el verificador ahora tiene cierta certeza de que puede establecer comunicaciones que tienen tanto confidencialidad como integridad con el contenedor seguro 708 utilizando una clave compartida establecida, que el hardware de plataforma confiable puede ser confiable de acuerdo con su fabricante, y que el estado de software de la plataforma confiable utilizado para crear el contenedor seguro es conocido. El verificador 702 está ahora listo para solicitar procesamiento seguro dentro del contenedor seguro 708 utilizando el código privado y/o datos privados.

Plataforma de abstracción de enclave y unidades básicas

La figura 8 es un diagrama de bloques de interfaces de componentes de software para un sistema de enclave local ejemplar. El sistema de enclave 800 incluye una computadora 810 con plataforma de enclave nativa 812 que aloja el enclave 814 y un cliente 816 del enclave. La plataforma nativa 812 puede ser un componente de hardware y/o software basado en, por ejemplo, SGX de Intel o VSM de Microsoft. El enclave 810 puede ser el enclave 176 de la figura 1. Se puede utilizar un protocolo nativo 844 para enclaves para la comunicación entre el enclave 814, el cliente 816, y la plataforma nativa 812. Como se representa en la figura 8, el protocolo nativo 844 incluye la interfaz 820 en el enclave 814, las interfaces 822 y 824 en la plataforma nativa, y la interfaz 826 en el cliente. Estas interfaces pueden ser APIs o ABIs en componentes de software.

El uso de estas interfaces de software 820, 822, 824, y 826 puede incluir una transferencia de control de ejecución entre componentes de software. Una transferencia de control puede incluir la ejecución de una instrucción de llamada o salto un punto de entrada (una dirección de una instrucción) en el componente de software al cual se está transfiriendo el control. Por ejemplo, si la plataforma nativa 812 es un componente de software, la transferencia de control de la plataforma nativa 812 al cliente 816 puede ocurrir a través de la interfaz de software 826 cuando se ejecuta una instrucción de llamada o salto en la plataforma nativa que especifica una dirección dentro del cliente 816 a la cual llamar o saltar. La dirección especificada dentro del cliente 816 puede ser un punto de entrada para una función o método en la interfaz 826. La transferencia de control se indica como una flecha en la figura 8, por ejemplo: de la plataforma nativa 812 al enclave 814 a través de la interfaz 820; del enclave 814 a la plataforma nativa 812 a través de la interfaz 822; del cliente 816 a la plataforma nativa 812 a través de la interfaz 824, y de la plataforma nativa 812 al cliente 816 a través de la interfaz 826. Un protocolo nativo 844 puede incluir patrones de comunicación a través de las interfaces 820, 822, 824, y 826.

En algunas modalidades, la plataforma nativa 812 se puede implementar al menos en parte como un componente de hardware, por ejemplo con instrucciones de procesador especiales para gestionar un enclave. Tal instrucción de hardware especial se puede ejecutar como parte de un componente de software de la plataforma nativa 812. En modalidades alternas, puede no haber ningún componente de software para algunas o todas de las funciones de la plataforma nativa 812. En estas modalidades

alternas, las interfaces 822 y 824 de la plataforma nativa pueden ser instrucciones de hardware en lugar de puntos de entrada de software, de modo que una función de la plataforma nativa 812 se puede utilizar por el enclave 814 o el cliente 816 o se puede utilizar en su lugar al ejecutar una instrucción de hardware especial en el enclave 814 o el cliente 816, respectivamente, en lugar de ejecutar una instrucción de llamada o salto.

En algunas modalidades, el cliente 816 del enclave 814 puede en sí ser un enclave. Por ejemplo, un cliente de enclave 816 puede utilizar la interfaz 824 para solicitar que se cree el enclave 814. En estas modalidades, la comunicación entre el enclave 814 y el cliente 816 a través de la plataforma nativa 812 es en realidad comunicación entre dos enclaves. Cuando el cliente 816 también es un enclave, el cliente de enclave 816 también puede utilizar la interfaz 822 y exponer una interfaz similar a 820 (no representada).

La figura 9 es un diagrama de bloques de interfaces de componentes de software para un sistema de enclave local ejemplar con una capa de abstracción. El sistema de enclave 900 incluye una plataforma de abstracción 912 para traducir entre el protocolo nativo 944 y los protocolos de abstracción 940, 942. La plataforma nativa 918 puede ser similar a la plataforma de abstracción 812 de la figura 8, y las interfaces 928 y 930 pueden combinar las funciones de las interfaces 820, 822, 824 y 826 de la figura 8. El protocolo de abstracción del enclave 940 incluye las interfaces 920, 922 para el enclave 914, mientras el protocolo de abstracción del cliente 942 incluye las interfaces 924, 926 para el cliente 916. Como en la figura 8, el cliente 916 que se ejecuta en la computadora 910 puede solicitar la creación del enclave 914 a través de la interfaz 924. La capa de abstracción 912 puede provocar la creación del enclave 914 utilizando el protocolo nativo 944 y las interfaces 928, 930 con la plataforma nativa 918. El cliente 916 y el enclave 914 pueden utilizar protocolos de abstracción 940 y 942 cuando la plataforma nativa 918 y el protocolo nativo 944 se basan en diferentes arquitecturas de enclave, tales como SGX de Intel o VSM de Microsoft. Como en la figura 8, el cliente 916 del enclave 914 puede en sí ser un enclave, y la plataforma nativa 918 puede incluir componentes de hardware y/o software.

El enclave 914 y el cliente 916 pueden no comunicarse directamente y pueden en cambio comunicarse a través de la plataforma de abstracción 912. La comunicación directa puede no ser posible o deseable, por ejemplo debido al aislamiento de la memoria del enclave 914. El aislamiento de memoria del enclave puede evitar la lectura desde, escritura en, o ejecución (salto al interior o exterior de) la memoria aislada del enclave.

El enclave 914 puede incluir instrucciones ubicadas dentro de un contenedor seguro de enclave de la computadora 910. El cliente 916 puede incluir instrucciones ubicadas en el espacio de dirección de memoria de la computadora 910, pero fuera del contenedor seguro del enclave 914. La plataforma de abstracción 912 se puede implementar de diferentes formas, incluyendo como instrucciones que están dentro o fuera del contenedor seguro del enclave 914, y también pueden incluir instrucciones ejecutadas desde dentro de y hiperllamadas. En el caso donde la plataforma de abstracción 912 se incluye al menos en parte dentro del contenedor seguro del enclave 914, el código de plataforma de abstracción dentro del contenedor seguro puede ser de autoría separada del resto del código del enclave 914 y puede

interactuar solamente con otro código de enclave a través de APIs/ABIs públicas. Tal código de plataforma de abstracción se puede enlazar estáticamente o enlazar dinámicamente al resto del código dentro del contenedor seguro de enclave. El código de plataforma de abstracción enlazado estáticamente puede ser código de objeto que está asociado con la plataforma de abstracción y incluye (enlace estáticamente), junto con código que es más específico para el enclave 914, en una imagen binaria desde la cual se puede instanciar el enclave 914. En el caso de una plataforma de abstracción enlazada dinámicamente, el código de enclave que es más específico para el enclave 914 y el código asociado más generalmente con la plataforma de abstracción pueden originarse de imágenes binarias separadas. Para un ejemplo enlazado dinámicamente, ver la figura 14.

La figura 10 es un diagrama de bloques de interfaces de componentes de software para un sistema de enclave remoto ejemplar con una capa de abstracción. El sistema de enclave remoto 1000 incluye un enclave 1014 en la computadora 1010, y un cliente 1056 del enclave 1014 en una computadora 1050 separada. La combinación del resguardo del cliente 1016 y plataforma remota de abstracción 1052 puede facilitar la interacción entre el enclave 1014 y el cliente 1056. Muchos elementos en la computadora 1010 pueden ser idénticos o similares a los elementos nombrados de manera idéntica de la computadora 910 de la figura 9. En particular, la plataforma de abstracción 1012, los protocolos 1040, 1042, 1044, y la plataforma nativa 1018 pueden ser similares o idénticos a los elementos correspondientes 912, 940, 942, 944, y 918, respectivamente.

El resguardo del cliente 1016 puede comunicarse con la plataforma remota de abstracción 1052 a través de la comunicación de red 1080. El protocolo de cliente remoto 1082 y las interfaces 1064, 1066 pueden ser similares al protocolo de abstracción del cliente 1042 y las interfaces 1024, 1026. Sin embargo, el protocolo de cliente remoto puede incluir funcionalidad adicional para acciones remotas. Por ejemplo, un método en la interfaz 1064 tal como CreateEnclave para solicitar la creación de un enclave puede incluir adicionalmente la capacidad de especificar una computadora anfitriona de enclave, tal como la computadora 1010, donde se solicita que se cree un enclave. Una cita de afirmación del enclave 1014 proporcionada al cliente 1056 por medio del protocolo de cliente remoto se puede proporcionar en lugar de, o además de, un reporte de afirmación. La computadora 1050 con el cliente 1056 puede o no incluir una plataforma de enclave nativa 1058. Si la plataforma nativa 1058 está presente, ésta puede o no ajustarse a la plataforma nativa 1018 de arquitectura de enclave de muestra, y por lo tanto el protocolo nativo 1044 y el protocolo nativo remoto 1084 pueden no ser iguales.

En una modalidad alterna (no representada), el resguardo del cliente 1016 puede no existir, y la plataforma de abstracción 1012 puede comunicarse directamente con la plataforma remota de abstracción 1052 a través de una red.

Los protocolos de abstracción de enclave, tales como 940, 942, 1040, 1042, 1082 de las figuras 9 y 10, pueden incluir una variedad de métodos de interfaz o puntos de entrada para gestionar y utilizar enclaves que se construyen en múltiples plataformas nativas, tales como SGX de Intel y VSM de Microsoft. Estos métodos pueden proporcionar unidades básicas de enclave que se pueden implementar en múltiples plataformas nativas, proporcionando por lo tanto una “abstracción” de las plataformas

nativas. Las unidades básicas de enclave que se divulgan aquí incluyen gestión de ciclo de vida, afirmación, sellado de datos, transferencia de control, contadores monotónicos, y tiempo de confianza.

Las unidades básicas para gestión de ciclo de vida de enclave pueden incluir métodos para provocar la instanciación o terminación de un enclave tal como el enclave 914. Las unidades básicas de gestión de ciclo de vida pueden ser una parte del protocolo de abstracción del cliente 942, y, más específicamente, se pueden implementar por medio de la plataforma de abstracción 912 como parte de la interfaz 924 para uso por el cliente 916.

Un método para instanciar o crear un enclave puede incluir especificar una imagen ejecutable del código y/o datos que se van a cargar en la memoria aislada del contenedor de enclave seguro. Este código, antes o después de ser cargado en el contenedor de enclave, puede volverse parte del estado inicial utilizado para afirmación del enclave instanciado (como se explicó anteriormente con respecto a la figura 5). Por ejemplo, la imagen ejecutable de un enclave (un binario de enclave) se puede especificar por un cliente de enclave al proporcionar un indicador a un buffer en memoria que contiene la imagen ejecutable. Alternativamente, se puede especificar una imagen de enclave indicando un archivo en un sistema de archivos que contiene el binario de enclave. En algunas modalidades, la imagen de enclave especificada puede estar cifrada; en otras modalidades, el enclave puede no estar cifrado; en otras modalidades, el enclave puede estar cifrado parcialmente. La medición del binario de enclave para afirmación o ocurrir a través de una imagen ejecutable cifrada o después del descifrado.

El código y/o datos que se han de cargar inicialmente en un enclave se pueden indicar especificando un archivo que contiene una imagen primaria de enclave. Además de este código y/o datos, una imagen primaria de enclave puede incluir metadatos adicionales, tal como un tamaño deseado del enclave (la cantidad de memoria requerida dentro del contenedor de enclave), ubicaciones de puntos de entrada dentro del código en el archivo, y una lista de archivos de imágenes dependientes. Los archivos de imágenes dependientes son otros archivos de imagen (no primaria) que también se pueden cargar en el enclave junto con el código y datos en el archivo de imagen primaria. Los archivos de imágenes dependientes pueden en sí contener listas de archivos de imágenes dependientes adicionales. En el caso del sistema de enclave local de la figura 9, las imágenes primaria y dependientes pueden estar archivadas en cualquier dispositivo de almacenamiento accesible, tal como por medio de un sistema de archivos accesible localmente. En el caso del sistema de enclave remoto de la figura 10, el archivo de imagen primaria puede estar en cualquier dispositivo de almacenamiento accesible para cualquiera de la computadora 1010 o la computadora 1050. Si el cliente 1056 solicita la creación de un enclave en la computadora 1010 utilizando una imagen primaria ubicada en la computadora 1050, la plataforma remota de abstracción y el resguardo del cliente 1016 pueden coordinarse para copiar el archivo de imagen primaria especificado en la computadora 1010.

CreateEnclave es un método ejemplar para instanciar enclave. El método CreateEnclave se puede describir con pseudocódigo:

```
HANDLE
```

```
CreateEnclave(
```

```

_In_ PCWSTR    enclavePath,
_In_ DWORD    flEnclaveType,
_In_ DWORD    dwFlags,
__In_reads_bytes_(dwInfoLength)
    PCVOID    enclaveInformation,
_In_ DWORD    dwInfoLength,
_Out_opt_ PDWORD    enclaveError
)

```

El pseudocódigo utilizado para describir los métodos en este documento puede utilizar varias convenciones de pseudocódigo para definir interfaces API. Por ejemplo, los parámetros de función, tales como `enclavePath` anterior, pueden decorarse con “_In_” o “_Out_” para indicar que un parámetro es un parámetro de entrada o salida, respectivamente. “_Out_opt_” puede indicar un parámetro de salida opcional. Las palabras todo indican un tipo de datos. `HANDLE` puede ser un número, tal como un número de 32 bits, utilizado para referirse indirectamente algo. Por ejemplo, el método `CreateEnclave` anterior regresa un `HANDLE` al llamador de `CreateEnclave`, y el `HANDLE` puede ser un indicativo del enclave que fue creado; `PCWSTR` puede ser un indicador a un cierto tipo de cadena de texto; `DWORD` puede ser una cantidad de 32 bits sin firmar; `PCVOID` puede ser un indicador a datos de tipo sin especificar; `BOOL` puede ser un valor binario.

`CreateEnclave` puede permitir que un cliente, tal como el cliente 916, cree un enclave y cargue la imagen primaria dentro del enclave. Cualquier información de configuración de enclave en esta imagen se puede asociar con el enclave instanciado. `CreateEnclave` puede incluir los siguientes parámetros:

`lpEnclaveName`: puede especificar la trayectoria a la imagen primaria de enclave, que en implementaciones puede ser algún otro tipo de identificador para identificar el código y/o datos de la imagen primaria de enclave, tal como un indicativo a un archivo abierto, un identificador uniforme de recursos (URI, Uniform Resource Identifier, por sus siglas en inglés), o un identificador que se utiliza en una búsqueda externa. Por ejemplo, se puede utilizar un identificador globalmente único (GUID, Globally Unique Identifier, por sus siglas en inglés) como una clave en una base de datos de imágenes primarias. En otras implementaciones, este parámetro puede identificar una región de memoria que contiene la imagen primaria de enclave.

`flEnclaveType`: puede especificar el tipo de enclave por crear (en caso de que una imagen de enclave soporte múltiples tipos). Se puede establecer como `DEFAULT` (predeterminado) en caso de que el binario soporte solamente un enclave o el desarrollador haya especificado explícitamente uno predeterminado. `dwFlags`: puede especificar una o más acciones predeterminadas que deben tomarse cuando se crean enclaves y carga la imagen primaria de enclave.

`enclaveInformation`: pueden ser parámetros de entrada opcional para configuración en tiempo de ejecución de enclave.

`lpEnclaveError`: puede especificar un parámetro opcional para regresar un código de error específico de la arquitectura.

Con la terminación exitosa, CreateEnclave puede regresar un indicativo al enclave. Con error, se puede regresar NULL (nulo). También se pueden regresar otros identificadores (GUID, URI, etc.) sin apartarse del alcance de esta divulgación. Por simplicidad, esta especificación describirá las APIs que utilizan un indicativo. La creación de enclave puede fallar, por ejemplo, debido a la falta de memoria de enclave, falta de soporte para el tipo de enclave especificado en la plataforma de abstracción o plataforma nativa, o la creación puede fallar debido a políticas de configuración explícitas que evitan que se ejecute un enclave de un tipo especificado en el sistema.

Las implementaciones de CreateEnclave y otro método de API descrito posteriormente pueden excluir uno o más de los parámetros de método descritos. Por ejemplo, con respecto a CreateEnclave, los lpEnclaveName, flEnclaveType, dwFlags, y enclaveInformation se pueden excluir, utilizando un valor predeterminado específico para esa API particular. El argumento lpEnclaveError también se puede excluir de la API, y opcionalmente se pueden implementar métodos alternativos para comprobar si hay errores en la llamada de API.

CreateEnclave puede ser responsable de cargar todos los módulos dependientes como se especifica en la imagen primaria de enclave. La imagen primaria de enclave puede ser un archivo de ejecución portátil (PE, Portable Execution, por sus siglas en inglés) que especifica otros archivos de imagen binaria de los cuales depende la imagen primaria. CreateEnclave también puede llevar a cabo inicialización específica de plataforma nativa, tales como finalizar mediciones para afirmación, designar estructuras para seguridad de capa de transporte (TLS, Transport Layer Security, por sus siglas en inglés) y/u otros protocolos de acuerdo y comunicación de clave, etc. Las interfaces 920, 922 de protocolo de abstracción de enclave (incluyendo métodos, por ejemplo, para sellado y afirmación de datos) pueden ser operables una vez que la inicialización de enclave se ha completado.

TerminateEnclave es un método ejemplar para terminar un enclave:

VOID

TerminateEnclave(

 In HANDLE hEnclave

)

TerminateEnclave se puede utilizar para destruir un enclave. En las implementaciones, destruir un enclave puede incluir forzar a todas las tareas de enclave a regresar al anfitrión o terminar, y/o liberar la memoria asociada con el enclave. Llamar a TerminateEnclave en un enclave en ejecución puede terminarlo y liberar todos los recursos asociados con el enclave.

La plataforma de abstracción de enclave 912 puede incluir unidades básicas de transferencia de control de ejecución que se pueden utilizar, por ejemplo, para transferir control entre un enclave y su cliente. Las unidades básicas de transferencia de control de ejecución pueden permitir la comunicación entre el enclave 914 y el cliente 916 al iniciar la ejecución de código en un punto de entrada en el otro componente. Las unidades básicas de transferencia de control de ejecución permiten el paso de datos al interior/exterior de enclaves permitiendo que se asocien parámetros con una solicitud de transferencia de control; los parámetros pueden especificar artículos de datos individuales (los parámetros en sí se

comunican) o los parámetros pueden ser indicadores a áreas de memoria (los buffers indicados por los parámetros se comunican). Estas unidades básicas pueden permitir la transferencia de control a pesar de las limitaciones sobre llamar o saltar directamente entre el enclave 914 y el cliente 916 debido a las restricciones de seguridad en el contenedor de enclave.

Para llamar al interior de un enclave, la interfaz 924 puede incluir mecanismos para permitir que un cliente 916 llame al interior de un enclave 914 a través de la interfaz 920. Por ejemplo, la interfaz 924 puede incluir los métodos GetProcAddress y CallEnclaveIn:

```
typedef PVOID (*ENCPROC)(PVOID);
ENCPROC
GetProcAddress(
    _In_   HMODULE    hEnclave,
    _In_   LPCTSTR    lpProcName
)
BOOL
CallEnclaveIn(
    _In_   ENCPROC    pCallin,
    _In_   PVOID pParameter,
    _Out_  PVOID pReturn
)
```

Un cliente de enclave, tal como el cliente 916, puede llamar al interior de un enclave, tal como el enclave 914, utilizando el indicador de función regresado por GetProcAddress(). El parámetro lpProcName puede relacionar la función exportada en la imagen primaria de enclave. Por ejemplo:

```
// llamar a la función Callin para el enclave.
ENCPROC pCallin = (ENCPROC) GetProcAddress(hEnclave,
    "CallinExample");
PVOID pParameter; // Indicador a memoria
if (NULL != pCallin)
{
    CallEnclaveIn(pCallin, pParameter);
}
```

En otras modalidades de GetProcAddress, lpProcName puede ser otro identificador de la función exportada específica, tal como un número, tal como una selección de una enumeración de puntos de entrada exportados desde una imagen de enclave, u otro identificador no textual correspondiente a la función. Otras modalidades de CallEnclaveIn pueden tomar adicionalmente un parámetro de entrada que especifica el enclave al interior del cual se va a llamar, por ejemplo, el indicativo regresado CreateEnclave.

Cuando se llama al interior de un enclave, se puede suspender una tarea en el proceso del cliente y se puede utilizar una tarea de enclave (con ID de tarea separado) para dar servicio a la solicitud

de llamada al interior. El código de enclave, que se ejecuta en la tarea de enclave, puede entonces tener acceso a la memoria que previamente estaba disponible para el cliente de enclave antes de llamar al interior del enclave. Por ejemplo, el cliente puede poner datos en el buffer indicado por pParameter antes de llamar al método de abstracción CallEnclaveIn, y después el enclave puede tener acceso al buffer indicado por pParameter mientras se da servicio a la solicitud de llamada al interior. Al llamar al exterior, la tarea de llamada original (cliente) se puede utilizar para dar servicio a la llamada al exterior. La re-entrada puede estar soportada (por ejemplo una llamada al exterior en el anfitrión puede llamar al interior del enclave nuevamente).

Para llamar al exterior de un enclave, la interfaz 922 puede incluir métodos relacionados con los métodos CallEnclaveIn anteriores que permiten que un enclave 914 llame al exterior al cliente de enclave 916. Por ejemplo, el enclave 914 puede llamar al exterior a cualquier función en el proceso anfitrión de un tipo particular, por ejemplo el tipo de función ENCPROC. El indicador de función para el mismo se puede pasar utilizando los parámetros de llamada al interior al enclave.

```
BOOL
```

```
CallEnclaveOut(
```

```
    _In_  ENCPROC    pCallout,
```

```
    _In_  PVOID pParameter,
```

```
    _Out_ PVOID pReturn
```

```
)
```

```
// Llamada al exterior a la función en el proceso de anfitrión
```

```
ENCPROC pCallout = (ENCPROC) 0xF00; // dirección a alguna función en el anfitrión
```

```
PVOID pParameter = // Indicador a memoria
```

```
CallEnclaveOut(pCallout, pSharedMemory);
```

La interfaz 920 puede incluir los puntos de entrada registrados como la función “CallinExample” anterior, y la interfaz 926 puede incluir los puntos de entrada registrados como las funciones “Callout” anteriores. Por ejemplo, en el caso donde una imagen primaria de enclave está en un formato de imagen de ejecución portátil (PE), los puntos de entrada de función en la imagen se pueden enlistar como puntos de entrada de “exportación”, y cada uno de tales puntos de entrada exportados puede incluir un nombre textual, tal como “CallinExample”, para identificar y diferenciar los puntos de entrada en esa imagen de DE de enclave; en otras implementaciones, los puntos de entrada de función se pueden marcar con metadatos adicionales, tales como un bit indicando que una función puede ser un punto de entrada para el enclave. En el ejemplo anterior, para llamar al exterior del enclave, la dirección de la función callout está dada como 0xF00 y es solamente un ejemplo. La dirección real de una función callout se puede determinar en una variedad de formas. Por ejemplo, una dirección de función callout dentro de un cliente se puede pasar al enclave como un parámetro para una función call-in. En otro ejemplo, la dirección de una función callout puede ser registrada por el cliente utilizando una función tal como RegisterCallOut:

```
BOOL RegisterCallOut(
```

```
    _In_  ENCPROC    pCallout,
```

```
_In_ LPCTSTR lpProcName)
```

El código dentro del enclave puede obtener la dirección de la función callout llamando una función complementaria tal como GetCallOut:

```
BOOL GetCallOut(  
_Out_ ENCPROC *pCallout,  
_In_ LPCTSTR lpProcName)
```

En otras modalidades, los métodos CallEnclaveIn y CallEnclaveOut pueden ser en realidad el mismo método. Por ejemplo, un solo método CallEnclave se puede utilizar para llamar al interior y llamar al exterior de un enclave. En situaciones donde el cliente de enclave 916 también es un enclave, la llamada al exterior del enclave 914 al cliente 916 también será llamar al interior de un enclave.

La plataforma de abstracción 912 puede proporcionar unidades básicas para sellar datos en un enclave. Por ejemplo, la interfaz 922 puede proporcionar servicios al enclave 914, tales como sellar y desellar datos a una identidad de enclave. Como se explicó anteriormente, un enclave de tener múltiples identidades anidadas, y los datos se pueden sellar a cualquiera de tales identidades. Cuando los datos se sellan a una identidad que corresponde a un conjunto de instanciaciones de enclave posibles, los datos sellados se pueden desellar por medio de cualquiera de ese conjunto correspondiente de instanciaciones de enclave. Por ejemplo:

```
struct SEALING_POLICY  
{  
ENCLAVE_ID_TYPE   enclaveldType;  
};
```

Por cada valor de enclaveldType, el enclave se sellará a un ID de mapeo. Posibles tipos de identidad de enclave (y valores de enclaveldType) incluyen:

```
ENCLAVE_EXACTHASH  
ENCLAVE_INSTANCEHASH:      // sellar utilizando MRENCLAVE para SGX, IMAGE HASH  
para VSM  
ENCLAVE_IMAGEIDS:          // no soportado en SGX, utilizará IMAGE IDS para VSM  
ENCLAVE_FAMILYID:          // utilizará PRODUCTID para SGX, FAMILY ID para VSM  
ENCLAVE_AUTHORID: // utilizará MRSIGNER para SGX, AUTHORID para VSM
```

La plataforma también puede aplicar configuración de depuración adicional (de autor y tiempo de ejecución) a la política de sellado. Para diferentes políticas de depuración, se pueden utilizar diferentes claves de sellado. Por ejemplo, las configuraciones de depuración y liberación pueden utilizar diferentes claves de sellado.

```
DWORD  
EnclaveSeal(  
_In_ SEALING_POLICY sealingPolicy,  
_In_reads_bytes_opt_(dwPlaintextSize) LPCVOID pPlaintext,  
_In_ DWORD dwPlaintextSize,
```

```

_In_reads_bytes_opt_(dwAuthdataSize) LPCVOID pAuthdata,
_In_          DWORD dwAuthdataSize,
_Out_writes_bytes_to_(dwSealedtextSize) LPVOID pSealedtext,
_Inout_      DWORD *dwSealedtextSize
)
DWORD
EnclaveUnseal(
_In_reads_bytes_opt_(dwSealedtextSize) LPCVOID pSealedtext,
_In_          DWORD dwSealedtextSize,
_In_reads_bytes_opt_(dwAuthdataSize) LPCVOID pAuthdata,
_In_          DWORD dwAuthdataSize,
_Out_writes_bytes_to_(dwPlaintextSize) LPCVOID pPlaintext,
_Inout_      DWORD *dwPlaintextSize
)

```

La plataforma de abstracción 912 puede proporcionar unidades básicas para afirmación, tal como para producir reportes y citas de afirmación, y para verificar reportes y citas. Por ejemplo:

```

DWORD
CreateReport(
_In_reads_bytes_opt_(dwTargetInfoSize) PCVOID pTargetInfo,
_In_          DWORD dwTargetInfoSize,
_In_reads_bytes_opt_(dwAuthData) PCVOID pAuthData,
_In_          DWORD dwAuthData,
_Out_writes_bytes_opt_(*pReportSize) PVOID pReport,
_Inout_      PDWORD pReportSize,
_Out_opt_    PDWORD lpEnclaveError
)
DWORD
VerifyReport(
_In_reads_bytes_(dwReportSize) PCVOID pReport,
_In_          DWORD dwReportSize,
_Out_opt_    LPDWORD lpEnclaveError
)

```

VerifyReport() se puede utilizar en cualquier enclave para afirmar la integridad del reporte y que el reporte fue generado por un enclave en la misma máquina.

```

DWORD CreateQuote(
_In_          GUID quoteType,
_In_          DWORD authDataSize,
_In_reads_bytes_opt_(authDataSize) const BYTE* authData,

```

```

_Out_          DWORD* quoteSize,
_Outptr_result_bytebuffer_opt_(*quoteSize) BYTE** quote
)

```

En CreateQuote, quoteType puede mapear a un proveedor de cita (*quote*), que puede ser una fuente de confianza para generar la cita específica. En CreateQuote, authData puede ser un indicador a los datos que son creados por, y en un formato definido por, el llamador de CreateQuote. Tener en cuenta que authData no necesita ser entendido por la plataforma de abstracción 912. El authData se puede empaquetar en la cita resultante. Se puede esperar que los proveedores de cita soporten esto.

```

DWORD VerifyQuote(
_In_          DWORD quoteSize,
_In_reads_bytes_(quoteSize) const BYTE* quote,
_Out_        DWORD* reportSize,
_Outptr_result_bytebuffer_all_(*reportSize) BYTE** report
)

```

Además de las unidades básicas de enclave descritas anteriormente, una plataforma de abstracción de enclave puede proporcionar: gestión de memoria (por ejemplo para designar y liberar memoria, tal como memoria restringida para un enclave o memoria que está compartida entre un enclave su cliente); manejo de recepción (por ejemplo para manejar error, o excepciones, que ocurren mientras se ejecuta código de enclave); sincronización de tarea; y funciones criptográficas (por ejemplo cifrado, funciones de *hash*, y firma).

Las técnicas descritas anteriormente se pueden implementar en uno o más dispositivos o entornos informáticos, como se describe posteriormente. La figura 11 representa un entorno informático de propósito general ejemplar, por ejemplo, que puede incorporar uno o más de hardware confiable 172, plataforma confiable 736, o computadoras 810, 910, 1010, y 1050, en los cuales se pueden incorporar algunas de las técnicas descritas en este documento. El entorno de sistema informático 1102 es solamente un ejemplo de un entorno informático adecuado y no se pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de la materia en cuestión actualmente divulgada. Tampoco se debe interpretar el entorno informático 1102 como que tiene alguna dependencia o requisito relacionado con cualquiera o una combinación de los componentes ilustrados en el entorno operativo 1102 ejemplar. En algunas modalidades, los diferentes elementos informáticos representados pueden incluir circuitería configurada para instanciar aspectos específicos de la presente divulgación. Por ejemplo, el término circuitería utilizado en la divulgación puede incluir componentes de hardware especializado configurados para llevar a cabo función(es) por medio de firmware o interruptores. En otras modalidades ejemplares, el término circuitería puede incluir una unidad de procesamiento de propósito general, memoria, etc., configurados por medio de instrucciones de software que incorporan lógica operable para llevar a cabo función(es). En modalidades ejemplares, donde la circuitería incluye una combinación de hardware y software, un implementador puede escribir código fuente que incorpora lógica y el código fuente se puede compilar en un código legible por máquina que se puede procesar por la unidad de procesamiento de

propósito general. Ya que un experto en la materia puede apreciar que el estado de la técnica ha evolucionado hasta un punto donde hay poca diferencia entre hardware, software, o una combinación de hardware/software, la selección de hardware contra software para efectuar funciones específicas es una elección de diseño que se deja a un implementador. Más específicamente, un experto en la materia puede apreciar que un proceso de software se puede transformar a una estructura de hardware equivalente, y una estructura de hardware se puede transformar en sí a un proceso de software equivalente. Por lo tanto, la selección de una implementación de hardware contra una implementación de software es una elección de diseño y se deja al implementador.

La computadora 1102, puede incluir cualquiera de un dispositivo móvil o teléfono inteligente, tablet, computadora portátil (*laptop*), computadora de escritorio, o colección de dispositivos en red, recursos informáticos en la nube, etc., generalmente incluye una variedad de medios legibles por computadora. Los medios legibles por computadora pueden ser cualquier medio disponible al cual se pueda acceder por medio de la computadora 1102 e incluye medios volátiles y no volátiles, medios removibles y no removibles. La memoria del sistema 1122 incluye medios de almacenamiento legibles por computadora en la forma de memoria volátil y/o no volátil tal como memoria de sólo lectura (ROM, Read Only Memory, por sus siglas en inglés) 1123 y memoria de acceso aleatorio (RAM, Random Access Memory, por sus siglas en inglés) 1160. Un sistema de entrada/salida básico (BIOS, Basic Input/Output System, por sus siglas en inglés) 1124, que contiene las rutinas básicas que ayudan a la transferencia de información entre elementos dentro de la computadora 1102, tal como durante el arranque, generalmente está almacenado en la ROM 1123. La RAM 1160 generalmente contiene datos y/o módulos de programa que son accesibles inmediatamente y/o en los que se opera inmediatamente por la unidad de procesamiento 1159. A manera de ejemplo, y no limitación, la figura 11 ilustra el hipervisor 1130, sistema operativo (OS, Operating System, por sus siglas en inglés) 1125, programas de aplicación 1126, otros módulos de programa 1127 incluyendo un cliente de enclave 1165, y el enclave 1128.

La computadora 1102 también puede incluir otros medios de almacenamiento de computadora removibles/no removibles, volátiles/no volátiles. A manera de ejemplo solamente, la figura 11 ilustra una unidad de disco duro 1138 que lee desde o escribe en medios magnéticos no removibles, no volátiles, una unidad de disco magnético 1139 que lee desde o escribe en un disco magnético removible, no volátil 1154, y una unidad de disco óptico 1104 que lee desde o escribe en un disco óptico removible, no volátil 1153 tal como una CD-ROM u otro medio óptico. Otros medios de almacenamiento de computadora removibles/no removibles, volátiles/no volátiles que se pueden utilizar en el entorno operativo ejemplar incluyen, pero no están limitados a, casetes de cinta magnética, tarjetas de memoria flash, discos versátiles digitales, cinta de video digital, RAM de estado sólido, ron de estado sólido, y similares. La unidad de disco duro 1138 generalmente está conectada al bus del sistema 1121 a través de una interfaz de memoria no removible tal como la interfaz 1134, y la unidad de disco magnético 1139 y la unidad de disco óptico 1104 generalmente se conectan al bus del sistema 1121 por medio de una interfaz de memoria removible, tal como la interfaz 1135 o 1136.

Las unidades y sus medios de almacenamiento de computadora asociados comentados

anteriormente y que se ilustran en la figura 11 proporcionan almacenamiento de instrucciones legibles por computadora, estructuras de datos, módulos de programa y otros datos para la computadora 1102. En la figura 11, por ejemplo, la unidad de disco duro 1138 se ilustra como almacenando el sistema operativo 1158, programas de aplicación 1157, otros módulos de programa 1156, tales como aplicaciones de cliente de enclave y archivos binarios de enclave, y datos de programa 1155. Tener en cuenta que estos componentes pueden ser iguales o diferentes al sistema operativo 1125, programas de aplicación 1126, otros módulos de programa 1127, y datos de programa 1128. El sistema operativo 1158, programas de aplicación 1157, otros módulos de programa 1156, y los datos de programa 1155 tienen diferentes números aquí para ilustrar que, como mínimo, son copias diferentes. Un usuario puede introducir comandos e información en la computadora 1102 a través de dispositivos de entrada tales como un teclado 1151 y dispositivo apuntador 1152, comúnmente denominado como un mouse, bola de desplazamiento o panel táctil. Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, palanca de mando, mando de juego, antena satelital, escáner, escáner de retina, o similares. Estos y otros dispositivos de entrada a menudo se conectan a la unidad de procesamiento 1159 a través de una interfaz de entrada del usuario 1136 que se acopla al bus del sistema 1121, pero se pueden conectar por medio de otra interfaz y estructuras de bus, tales como un puerto paralelo, puerto de juegos o un bus universal en serie (USB, Universal Serial Bus, por sus siglas en inglés). Un monitor 1142 u otro tipo de dispositivo de visualización también se conecta al bus del sistema 1121 por medio de una interfaz, tal como una interfaz de video 1132. Además del monitor, las computadoras también pueden incluir otros dispositivos periféricos de salida tales como bocinas 1144 e impresora 1143, los cuales se pueden conectar a través de una interfaz periférica de salida 1133.

La computadora 1102 puede operar en un entorno de red utilizando conexiones lógicas a una o más computadoras remotas, tal como una computadora remota 1146. La computadora remota 1146 puede ser una computadora personal, un servidor, un ruteador, una PC de red, un dispositivo de pares (*peer*) u otro nodo de red común, y generalmente incluye muchos o todos de los componentes descritos anteriormente con relación a la computadora 1102, aunque solamente se ha ilustrado un dispositivo de almacenamiento de memoria 1147 en la figura 11. Las conexiones lógicas representadas en la figura 11 incluyen una red de área local (LAN, Local Area Network, por sus siglas en inglés) 1145 y una red de área amplia (WAN, Wide Area Network, por sus siglas en inglés) 1149, pero también pueden incluir otras redes. Tales entornos de red son comunes en oficinas, redes de computadoras a nivel empresarial, intranets, Internet, y recursos informáticos en la nube.

Cuando se utiliza en un entorno de red de LAN, la computadora 1102 se conecta a la LAN 1145 a través de una interfaz de red o adaptador 1137. Cuando se utiliza en un entorno de red de WAN, la computadora 1102 generalmente incluye un módem 1105 u otros medios para establecer comunicaciones a través de la WAN 1149, tal como Internet. El módem 1105, el cual puede ser interno o externo, se puede conectar al bus del sistema 1121 a través de la interfaz de entrada del usuario 1136, u otro mecanismo apropiado. En un entorno de red, los módulos de programa representados con relación a la computadora 1102, o porciones de los mismos, se pueden almacenar en el dispositivo de

almacenamiento de memoria remoto. A manera de ejemplo, y no limitación, la figura 11 ilustra programas de aplicación remotos 1148 como que residen en el dispositivo de memoria 1147. Se apreciará que las conexiones de red mostradas son ejemplares y se pueden utilizar otros medios para establecer un enlace de comunicaciones entre las computadoras.

La figura 12 representa un diagrama de flujo ejemplar de un método 1200 para abstraer una plataforma de enclave nativa. Una plataforma de abstracción, tal como 912 de la figura 9, puede recibir una solicitud para una plataforma de enclave en el cuadro 1202. La solicitud puede venir de un cliente de enclave, tal como el cliente 914, o un enclave, tal como el enclave 916.

Una solicitud de un enclave puede ser una solicitud para llevar a cabo una unidad básica de plataforma de abstracción y puede incluir, por ejemplo, una solicitud para: crear un reporte o cita de afirmación de un enclave; sellar datos al enclave; llamar a una función en el cliente de un enclave (llamar al exterior al cliente); leer un contador monotónico (proporcionar el valor actual de un contador monotónico); proporcionar una medición de tiempo confiable; y designar memoria que puede ser compartida entre un enclave y su cliente (por ejemplo, para permitir un indicador a memoria compartida que se pasará como un parámetro cuando se llama al interior o al exterior de un enclave). En algunas modalidades, todo el espacio de memoria virtual de un cliente de enclave se puede compartir con (y ser accesible desde) el enclave, de tal manera que se puede implementar una solicitud para designar memoria compartida como una solicitud para designar memoria para el cliente del enclave. En algunas modalidades, los métodos para designar memoria compartida son disponibles tanto para un enclave como para su cliente.

Una solicitud de un cliente de enclave puede ser una solicitud para llevar a cabo una unidad básica de plataforma de abstracción y puede incluir, por ejemplo, una solicitud para: instanciar un enclave; verificar un reporte de afirmación de un enclave; Llamar a una función dentro de un enclave (llamar al interior de un enclave); y designar memoria que puede ser compartida entre un enclave y su cliente.

Una solicitud de plataforma de abstracción se puede traducir a una solicitud de plataforma nativa en las operaciones 1204-1208. Los parámetros incluidos o implicados en la solicitud recibida se pueden convertir en el paso opcional 1204 si se determina, por ejemplo, que el formato de datos de un parámetro en la solicitud original no es el mismo que el formato de datos para ese parámetro en la plataforma nativa. Por ejemplo, si una solicitud de un enclave o cliente incluye un parámetro derivado de un reporte de afirmación de formato de abstracción, tal como una identidad de abstracción de enclave, se convertirá a un parámetro utilizado en un reporte de afirmación de formato nativo, tal como una identidad de enclave nativo.

Si se determina que la convención de llamada de la plataforma nativa y la solicitud recibida difieren, la convención de llamada se puede convertir en el paso opcional 1206. Una convención de llamada se puede convertir, por ejemplo, reordenando parámetros en una pila de llamada, moviendo parámetros entre registros del procesador y una pila de llamada, y convirtiendo entre métodos de comunicación de condición de error tal como regresando un valor de error y llamando a un manejador de

excepción.

En algunas modalidades, la plataforma nativa puede ser idéntica a la plataforma de abstracción para algunas solicitudes, en cuyo caso, las operaciones de conversión del cuadro 1204 y 1206 se pueden omitir.

En el cuadro 1208, se envía la solicitud convertida a la plataforma nativa para provocar que se lleve a cabo la solicitud por medio de la plataforma nativa. Por ejemplo, en el caso donde la plataforma nativa se ajusta a la arquitectura de enclave de extensiones de defensa de software (SGX) de Intel, la plataforma nativa puede incluir instrucciones de procesador para enclaves. En este caso, enviar la solicitud en el cuadro 1208 puede incluir ejecutar una o más instrucciones de procesador para enclaves. En otro ejemplo, la plataforma nativa puede ajustarse a la arquitectura de enclave de modo seguro virtual (VSM) de Microsoft, que puede incluir un hipervisor con hiperllamadas para enclaves. Una hiperllamada es un desvío de software al código de hipervisor, y una hiperllamada puede provocar un cambio de contexto de procesador un contexto en el cual se pueden permitir operaciones privilegiadas. En este ejemplo de VSM, enviar la solicitud en el cuadro 1208 puede incluir hacer hiperllamadas al hipervisor y/u otros mecanismos para cambiar el contexto de ejecución a un contexto en el cual se pueden permitir operaciones privilegiadas.

Enviar una solicitud a una plataforma nativa aquí significa generalmente llevar a cabo la solicitud utilizando las características de la plataforma nativa. La operación de enviar la solicitud a la plataforma nativa 1208 puede involucrar múltiples operaciones con la plataforma nativa, y puede variar dependiendo de la operación (o unidad básica) solicitada, tal como crear un enclave, afirmación, sellado de datos, transferencia de control, o utilizar contadores monotónicos y tiempo confiable.

La unidad básica CreateEnclave puede ser utilizada para instanciar un enclave. Una solicitud de CreateEnclave para instanciar un enclave puede provocar que una plataforma de abstracción cree un contenedor seguro (por ejemplo designando cierta memoria y estableciendo una frontera de seguridad o aislamiento para esa memoria), copie código de enclave en ese contenedor seguro (por ejemplo de una imagen de enclave), y configure o permita puntos de entrada al código de enclave (por ejemplo de acuerdo con metadatos de punto de entrada en una imagen de enclave).

Enviar una solicitud de CreateEnclave a una plataforma nativa con un hipervisor habilitado con enclave (un hipervisor que proporciona funciones de gestión de enclave, tal como VSM), puede incluir designar memoria y hacer hiperllamadas para configurar tablas de página de procesador para la memoria en una manera que evita que código fuera del contenedor de enclave acceda a esa memoria. Las hiperllamadas de creación de enclave desde la plataforma de abstracción también pueden provocar que el hipervisor configure información de configuración para transferencia de control al enclave en puntos de entrada designados. Después, el código fuera del contenedor seguro puede hacer hiperllamadas de transferencia de control para transferir la ejecución en los puntos de entrada designados dentro del contenedor seguro.

Enviar una solicitud de CreateEnclave a una plataforma nativa con un procesador habilitado con enclave (un procesador con instrucciones de procesador de gestión de enclave, tal como SGX), puede

incluir que la plataforma de abstracción ejecute una instrucción tal como ECREATE para informar a la CPU que se debe crear una cierta área de memoria como un contenedor de enclave seguro, y ejecute una instrucción tal como EADD para agregar páginas de datos a ese contenedor de enclave. También se pueden utilizar instrucciones de procesador especiales para crear páginas especiales en memoria para designar los puntos de entrada en el enclave para transferencia de control al interior del enclave. Después, el código fuera del contenedor seguro puede ejecutar una instrucción tal como EENTER que especifica uno de los puntos de entrada designados para transferir control de ejecución a ese punto de entrada de enclave.

La unidad básica CreateReport puede ser utilizada para crear un reporte de afirmación. Una solicitud de CreateReport para crear un reporte de afirmación de un enclave se puede llevar a cabo por medio de una capa de abstracción como se explicó anteriormente con respecto a las figuras 5 y 7. Con un hipervisor habilitado con enclave, una capa de abstracción puede enviar la solicitud a la plataforma nativa haciendo una hiperllamada que cambie el estado de ejecución a, por ejemplo, un contexto de monitor de seguridad que tenga acceso a una clave secreta tal como PrivAK726 de la figura 7 que pueda ser utilizada para reportes firmados. Esta clave secreta puede sólo estar disponible para el contexto de monitor de seguridad si la computadora arrancó en una configuración en buen estado como verificado con un registro de TCG basado en TPM. El monitor de seguridad puede etiquetar los datos de reporte con una identidad del enclave que está siendo afirmado.

Con un procesador habilitado con enclave, se puede enviar una solicitud de CreateReport a la plataforma nativa ejecutando una instrucción, tal como EREPORT, que genera un reporte y lo envía a un enclave especial que tendrá acceso a una clave privada para firmar reportes.

La unidad básica EnclaveSeal puede ser utilizada para sellar datos a un enclave. Sellar datos a un enclave cifra los datos en una manera o con una clave que se asocia con el enclave. Una solicitud de EnclaveSeal puede ser una solicitud para sellar datos ubicados dentro de un enclave, tal como todo o parte del estado del enclave, al enclave que utiliza una política de sellado. La política de sellado, tal como SEALING_POLICY anterior, puede especificar un tipo de identidad de enclave que indica cuáles enclaves deben ser capaces de desellar los datos. El proceso de sellado en sí puede utilizar una clave de cifrado asociada con la identidad de enclave especificada en la política de sellado. Después, una nueva instanciación de enclave puede ser capaz de desellar los datos si el valor de identidad del nuevo enclave en el tipo de identidad especificado es el mismo que el valor de identidad del enclave de sellado en el tipo de identidad especificado.

El sellado de datos permite que datos de enclave secretos o sensibles se copien de manera segura al almacenamiento no seguro, tal como a una memoria fuera del contenedor seguro del enclave o a un almacenamiento persistente tal como un disco duro. Cuando los datos sellados son datos de estado de enclave, el sellado permite que un enclave se reinicie a un estado previo, y permite que se interrumpa una operación de enclave seguro y continúe posteriormente en otro enclave.

Para reiniciar un estado de enclave, primero se guarda un estado de enclave sellando su estado al enclave. El sellado se puede hacer cifrando los datos de estado con una clave asociada con el

enclave. Posteriormente, tal vez después de que ha cambiado el estado del enclave, los datos de estado sellados se pueden desellar en el mismo enclave descifrando los datos sellados y después reemplazando un estado actual del enclave con los datos descifrados (por ejemplo copiando los datos descifrados al contenedor seguro del enclave).

Para interrumpir una operación segura y continuar en otro enclave, la operación segura inicia ejecutando una operación que comprende múltiples instrucciones de procesador en un primer enclave. Cuando el primer enclave se interrumpe, el estado de ese enclave se puede sellar a una identidad de enclave especificada en la política de sellado, y después los datos sellados se pueden guardar en almacenamiento no seguro, tal como almacenamiento persistente local o basado en la nube. Entonces, el primer enclave se puede terminar (opcionalmente) o iniciar otras operaciones de enclave. Un segundo enclave se puede instanciar o reutilizar para continuar la operación interrumpida al desellar los datos de estados sellados dentro del segundo enclave. La operación interrumpida se puede continuar en el segundo enclave donde se quedó el primer enclave.

Con un hipervisor habilitado con enclave, una capa de abstracción puede enviar una solicitud de EnclaveSeal a la plataforma nativa haciendo una hiperllamada. La hiperllamada puede cambiar el estado de ejecución a un contexto, por ejemplo un contexto de monitor de seguridad, que vendrá acceso a una clave secreta de sellado asociada con el enclave que se puede utilizar para sellar o desellar datos. La clave de sellado se puede derivar de una combinación de una identidad de enclave y una clave secreta de plataforma disponibles solamente para el monitor de seguridad. Esta clave de plataforma puede sólo estar disponible para el monitor de seguridad cuando la máquina se arranca en una configuración en buen estado, y la configuración de arranque se verifica con un registro de TCG basado en un TPM. En esta modalidad de hipervisor habilitado con enclave, el código de enclave nunca tiene acceso a la clave de sellado.

Con un procesador habilitado con enclave, se puede enviar una solicitud de EnclaveSeal a la plataforma nativa ejecutando una instrucción, tal como EGETKEY, para obtener una clave de cifrado. Este algoritmo puede generar una clave de sellado que es única para el enclave. La clave de sellado se puede derivar de una identidad del enclave y un secreto incrustado en el procesador. El código dentro de un enclave puede entonces cifrar los datos con la clave de sellado. Los datos se pueden sellar cifrando con una clave de sellado, por ejemplo por medio de código dentro de un enclave, por medio de una plataforma de abstracción, o por medio de una plataforma nativa. EnclaveUnseal puede utilizar de manera similar EGETKEY para generar la clave de desellado.

Una solicitud de transferencia de control puede ser una solicitud para transferir control de ejecución de procesador de instrucciones dentro de un enclave hacia afuera a un punto de entrada fuera del enclave (por ejemplo CallEnclaveOut), o a la inversa de instrucciones fuera del enclave a un punto de entrada dentro del enclave (por ejemplo CallEnclaveIn). Esto se puede hacer, por ejemplo, para una operación de base de datos segura. Después de instanciar un enclave de base de datos, un cliente de enclave puede solicitar que el enclave lleve a cabo una operación específica, tal como una consulta de base de datos utilizando la unidad básica CallEnclaveIn para transferir control a un punto de entrada

dentro del enclave de base de datos que llevará a cabo la consulta. Después de que el enclave completa la consulta, el resultado de la consulta puede ser regresado (posiblemente después de cifrar el resultado) al cliente con la unidad básica CallEnclaveOut para transferir el control de vuelta al cliente en un punto de entrada en el cliente que puede recibir el resultado de consulta. Las unidades básicas CallEnclaveIn y CallEnclaveOut pueden tomar indicador a un buffer de memoria que se puede compartir entre un enclave y su cliente (el buffer puede ser legible, escribible, y/o ejecutable por cualquiera del enclave o su cliente).

Con un hipervisor habilitado con enclave, una capa de abstracción puede enviar una solicitud de CallEnclaveIn a la plataforma nativa haciendo una hiperllamada. La hiperllamada puede cambiar el estado de ejecución a un contexto, por ejemplo un contexto de monitor de seguridad, que guardará los registros de CPU, restaurará un conjunto guardado previamente de valores de registro de CPU de enclave (posiblemente desde memoria del enclave), cambiará la configuración de tabla de página para permitir el acceso a la memoria protegida del enclave, e invocará un punto de entrada de función dentro del enclave. De manera similar, cuando una plataforma de abstracción recibe una solicitud de CallEnclaveOut, la solicitud se puede enviar a la plataforma nativa por medio de una hiperllamada que guardará los registros de CPU del enclave (posiblemente guardando en memoria del enclave) y restaurará los registros de CPU guardados previamente para un cliente de enclave, cambiará la configuración de tabla de página para evitar el acceso a la memoria del enclave, y transferirá control de procesador a un punto de entrada en el cliente de enclave fuera del enclave.

Con un procesador habilitado con enclave, una solicitud de CallEnclaveIn se puede enviar a la plataforma nativa ejecutando una instrucción, tal como EENTER, que puede provocar que la CPU restaure un conjunto de registros de CPU de enclave (posiblemente desde memoria del enclave) e invoque una función (transferir control a un punto de entrada) dentro del enclave. Una unidad básica CallEnclaveOut puede ejecutar una instrucción, tal como EEXIT, que puede transferir control a instrucciones fuera del enclave y/o provocar una falla que transfiera control fuera del enclave.

El contador monotónico tiene una variedad de usos. Por ejemplo, un enclave puede desear restringir el qué tan atrás se puede revertir su estado. Los contadores monotónicos pueden ser utilizados, por ejemplo, como un nonce para garantizar la frecuencia de actualización de mensajes, como se comentó anteriormente con respecto a la figura 4. Los contadores monotónicos generalmente tienen la capacidad de ser leídos, y ser aumentados, pero no ser disminuidos. Para restringir el retroceso o revertir a un estado de enclave, el código dentro de un enclave puede aumentar un contador monotónico asociado, y después guardar el valor del contador junto con el estado interno del enclave. El estado y valor de contador pueden ser guardados, por ejemplo, con la unidad básica EnclaveSeal. Después, cuando se restaura el estado del enclave, utilizando por ejemplo la unidad básica EnclaveUnseal, el código dentro del enclave puede leer el valor actual del contador monotónico y lo compara con el valor de contador con el estado desellado. Si el valor del contador con el estado desellado es menor que el valor actual del contador, el enclave puede evitar el uso del estado desellado.

Con un hipervisor habilitado con enclave, una capa de abstracción puede enviar una solicitud a la plataforma nativa para leer o aumentar un contador monotónico haciendo una hiperllamada que se

expone al enclave. Cuando se invoca una hiperllamada para leer o aumentar el contador, el procesador cambiará el estado de ejecución a un contexto, tal como un monitor de seguridad, que verificará la identidad del enclave haciendo la hiperllamada, y después leerá de o aumentará, respectivamente, el contador monotónico correspondiente almacenado en, por ejemplo un almacenamiento no volátil tal como un chip de TPM. Alternativamente, el monitor de seguridad puede leer o aumentar un contador almacenado en un servidor confiable remoto o un conjunto de servidores confiables remotos, estableciendo un canal de comunicación seguro contra el servidor y pidiéndole que lea o aumente un contador monotónico especificado. El servidor o servidores confiables remotos pueden mantener el contador dentro de un enclave para aislarlo del resto del código en la computadora anfitriona.

Con un procesador habilitado con enclave, se puede enviar una solicitud a la plataforma nativa ejecutando una instrucción. Con tal procesador, las unidades básicas del contador monotónico se pueden implementar leyendo o aumentando un contador en almacenamiento de memoria no volátil en un chip en la tarjeta madre de la computadora. Alternativamente, estas unidades básicas también se pueden implementar utilizando un servidor remoto confiable como con el hipervisor habilitado con enclave.

La figura 13 representa un diagrama de flujo ejemplar de un método 1300 para abstraer una plataforma de enclave nativa. Una plataforma de abstracción de enclave puede recibir una solicitud de un enclave o cliente de enclave en el cuadro 1302. En el cuadro 1304, la plataforma de abstracción puede determinar si una plataforma nativa incluye opciones de procesador de enclave, por ejemplo, determinando si la plataforma nativa se ajusta a SGX. Si lo hace, las instrucciones del procesador de enclave se ejecutan en el cuadro 1306. En el cuadro 1308, la plataforma de abstracción puede determinar si una plataforma nativa incluye hiperllamadas de enclave, por ejemplo, determinando si la plataforma nativa se ajusta a VSM. Si lo hace, la plataforma nativa hace hiperllamadas de enclave. Los resultados de ejecutar las instrucciones de enclave o llamar las hiperllamadas de enclave se eliminan en el cuadro 1312. La eliminación puede incluir, por ejemplo, convertir los parámetros de salida o el manejo de excepción de las instrucciones del procesador de enclave o las hiperllamadas de enclave al formato o protocolo de la interfaz de la capa de abstracción. Los parámetros de salida convertidos después son regresados al solicitante original (enclave o cliente) en el cuadro 1314.

Identidad abstracta de enclave

La figura 14 representa imágenes binarias de enclave ejemplares utilizadas para instanciar un enclave. Un enclave se puede instanciar creando contenedor seguro, tal como el contenedor de enclave 1490, y copiando porciones de una o más imágenes de enclave al contenedor. El contenedor de enclave 1490 puede haber sido creado por referencia a la imagen primaria de enclave 1410. Una imagen primaria puede incluir referencias a otras imágenes de enclave dependientes. En este ejemplo, la imagen primaria 1410 incluye Dependencia1 y Dependencia2 como referencias a imágenes de enclave dependientes 1420 y 1450, respectivamente. La imagen 1420 incluye dependencias adicionales en las imágenes 1430 y 1440, mientras la imagen 1450 depende de la imagen 1460. Una vez que todas estas imágenes (o porciones de las mismas) son copiadas en el contenedor 1490, el enclave resultante puede

incluir las imágenes no de plataforma 1402, que pueden incluir código y datos que son únicos o específicos para el enclave instanciado, las imágenes de plataforma de abstracción 1440, e imágenes de plataforma nativa 1406.

Cada imagen de enclave, tal como la imagen primaria 1410, puede incluir IDs, dependencias, código, datos, y una firma del autor de la imagen. En el ejemplo de la imagen 1410, se incluyen 2 IDs 1410.1 y 1410.2. Estos IDs pueden ser UUIDs que especifican, por ejemplo, un valor de identidad abstracta que corresponde a un valor de ImageID, FamilyID, o AuthorID que, individualmente o colectivamente se pueden utilizar para identificar cualquier enclave instanciado con la imagen de enclave. Como se representa, la imagen 1410 tiene dos IDs, pero son factibles menos o más IDs. El código en la imagen 1410 puede ser instrucciones binarias ejecutables por el procesador de la computadora que aloja el contenedor de enclave 1490. Los datos en la imagen 1410 pueden ser utilizados por cualquier código cargado en el contenedor 1490. La imagen 1410 también puede incluir una firma Sig1410 para asegurar la integridad de cualquiera o todos de los contenidos de la imagen, tales como IDs, referencias de dependencia, código y datos. Otras imágenes 1420-1460 pueden contener de manera similar IDs, referencias de dependencia, código, datos, y firmas.

Un indicador de dependencia, tal como Dependencia1 y Dependencia2 de la imagen 1410, Dependencia1 y Dependencia2 de la imagen 1420, y Dependencia1 de la imagen 1450, se puede especificar en una variedad de formas. Si las imágenes 1410-1460 se almacenan en la memoria de un sistema de computadora, un indicador de dependencia puede ser simplemente una dirección en la memoria. Si las imágenes de enclave son archivos en un sistema de archivos, las referencias pueden ser nombres de archivo. En algunas modalidades, las referencias pueden ser un identificador lógico. Un identificador lógico podría ser un número único, tal como un UUID, o podría ser otro dato, tal como una cadena de texto, que de otra forma identifica una imagen de dependencia. Por ejemplo, una cadena de texto puede indicar un autor, origen, nombre de producto, familia de producto, y/o número de versión de imagen binaria dependiente. Un identificador lógico incluye una ubicación web o de Internet, tal como una ubicación donde se podría recuperar una copia de un binario dependiente.

En algunas modalidades, un archivo de imagen de enclave se puede ubicar buscando un indicador de dependencia, tal como un identificador lógico, en un registro de imágenes de enclave para encontrar un indicador a la versión actual o copia local de la imagen de enclave referenciada. En algunos casos, se puede utilizar un servicio confiable para resolver un indicador de dependencia en identificación de una imagen de enclave o ubicación de imagen particular.

En algunas modalidades, un indicador de dependencia puede ser un identificador criptográficamente seguro, tal como un *hash* criptográfico de la imagen binaria de enclave dependiente pretendida. Tal *hash* puede incluir todo el binario dependiente, o solamente una porción del mismo. La porción de un binario dependiente incluido en un indicador de dependencia puede incluir valores de identidad abstracta, tales como ID 1410.1 o ID 1420.2, y pueden ser valores de identidad abstracta. Un servicio de resolución para un identificador criptográficamente seguro puede no necesitar ser tan confiable como con un identificador lógico debido a que la entidad que determina las dependencias de

enclave puede ser capaz de verificar que se encontró la imagen dependiente correcta calculando el *hash* del binario dependiente en sí.

La figura 15 representa un diagrama de flujo ejemplar de un método 1500 para llevar a cabo una operación de enclave con identidad abstracta de enclave. Un valor de identidad abstracta para un enclave puede proporcionar una base para determinar la equivalencia entre dos enclaves que tienen alguna característica en común pero no son exactamente idénticos. Un valor de identidad se puede incluir en un reporte de afirmación y se puede asociar con el tipo de identidad abstracta (tal como ExactHash, InstanceHash, ImageID, FamilyID, o AuthorID). Dos enclaves que no son exactamente iguales pueden tener el mismo valor de identidad abstracta para un tipo de identidad abstracta. Adicionalmente, el código de enclave idéntico instanciado en contenedores seguros en dos plataformas de enclave nativas diferentes también puede tener el mismo valor de identidad abstracta. El método 1500 se puede llevar a cabo, por ejemplo, por medio de una capa de plataforma de abstracción entre una plataforma de enclave nativa y cualquiera de un enclave o un cliente de enclave.

En el cuadro 1502, se instancia un enclave a partir de una imagen de enclave, tal como la imagen de enclave primaria 1410 de la figura 14. La imagen de enclave puede ser una imagen primaria que incluye código de enclave, datos, una lista de identidades, una lista de cualquier imagen de enclave dependiente, y una firma. Para asegurar la integridad de las imágenes de enclave, las imágenes se pueden firmar con una clave privada que puede corresponder al autor de la imagen de enclave. La lista de IDs de identidad de enclave en la imagen de enclave puede corresponder a los tipos de identidad abstracta tales como un ImageID, un FamilyID, y un AuthorID, cada uno pretendido para identificar la imagen de enclave colectivamente junto con otras imágenes de enclave relacionadas. Una lista de imágenes de enclave descendientes puede referirse a otras imágenes de enclave que contienen código de enclave del cual depende el código en la imagen de enclave primaria. Una imagen de enclave dependiente puede o no estar autorizada por el mismo autor, y algunas imágenes de enclave dependientes pueden estar asociadas con una plataforma de enclave generalmente (ya sea una plataforma de abstracción o plataforma nativa) en lugar de asociados particularmente con una imagen de enclave primaria o el autor de la imagen de enclave primaria. El enclave se puede instanciar creando un contenedor de enclave seguro utilizando cualquier plataforma de enclave nativa, y copiando todo o una porción de la imagen de enclave y cualquier imagen de enclave dependiente en el contenedor seguro.

En el cuadro 1503, se solicita una operación de enclave, por ejemplo por un enclave o un cliente de enclave, junto con un tipo de identidad de enclave. El tipo de identidad puede especificar un tipo de identidad abstracta, tal como Image ID o AuthorID, y relacionarse con un enclave instanciado particular, pero no especifica el valor de AuthorID para ese enclave. El resto del método 1500 después del cuadro 1503 describe operaciones para llevar a cabo la operación (tal como afirmación, sellado de datos, o uso de un contador monótonico, etc.) con el enclave instanciado utilizando un valor de identidad derivado para ese tipo de identidad del enclave instanciado. La identidad se puede determinar utilizando un *hash* de un subconjunto de la(s) imagen(es) de enclave. Cuál(es) del subconjunto de imagen(es) de enclave se utilizar(n) como entrada al *hash* se puede basar en parte en el tipo de identidad deseado a utilizar en la

operación de enclave.

En el cuadro 1504, se determina una porción de la imagen de enclave, llamada una porción de identidad en este documento, con base en el tipo de identidad. La porción de identidad puede incluir todas de, parte de, o ninguna de las diferentes imágenes binarias de enclave utilizadas para instanciar un enclave en el cuadro 1502. La porción de identidad puede incluir todo o, una porción de, o nada del código de enclave contenido en la imagen de enclave. La porción de identidad también puede incluir cero, uno, o más IDs de identidad enlistados en una porción no de código de las imágenes de enclave incluidas. La porción de identidad puede o no incluir también datos de enclave contenidos en las imágenes de enclave. La porción de identidad puede incluir cualquier combinación de estas diferentes partes de las imágenes de enclave. Por ejemplo, una porción de identidad puede incluir todo el código, nada de los datos, y dos o cuatro IDs de identidad disponibles. En el cuadro opcional 1506, se determina cuáles imágenes de enclave dependientes se van a incluir en la porción de identidad, y se determina una porción de identidad de cada imagen incluida.

La porción de identidad de imágenes dependientes puede o no ser la misma que la porción de identidad de una imagen de enclave primaria. Por ejemplo, todo el código y el ImageID se incluyen en la porción de identidad de una imagen primaria, mientras nada del código y solamente el FamilyID de una imagen dependiente se pueden incluir en la porción de identidad de la imagen dependiente.

Cuando el código de enclave se incluye en la porción de identidad, las porciones de código de enclave en la porción de identidad se pueden determinar por una combinación del tipo de identidad y una indicación de cuáles dependencias se van a incluir en la porción de identidad. El InstanceHash de tipo de identidad puede incluir, por ejemplo, código de enclave en la imagen primaria, pero no imágenes dependientes, mientras el tipo ExactHash puede incluir el código de enclave en las imágenes dependientes que no se consideran parte de una plataforma de enclave. Por ejemplo, todas las imágenes de enclave dependientes que no están firmadas con una clave privada de autor de plataforma de enclave se pueden considerar como que no son parte de la plataforma de enclave. Alternativamente o adicionalmente, la imagen primaria puede incluir una lista de cuáles imágenes de enclave dependiente se van a incluir o excluir en la porción de identidad para los tipos de identidad InstanceHash o ExactHash.

Los IDs de identidad de enclave, que se pueden incluir como metadatos en una imagen de enclave, se pueden incluir en la porción de identidad de la imagen de enclave en lugar de, o además de, el código de enclave. Por ejemplo, la porción de identidad para los tipos de identidad ImageID, FamilyID, y AuthorID pueden incluir metadatos de un ID correspondiente de la imagen de enclave. Cuando los tipos de identidad están anidados o en capas, la porción de identidad para los tipos de nivel inferior pueden incluir los datos de ID para tipos de nivel superior. Por ejemplo, la porción de identidad para ImageID puede incluir los datos de ID para ImageID, FamilyID, y Author ID, mientras la porción de identidad para AuthorID puede solamente incluir los datos de ID para AuthorID.

Los tipos de identidad que incluyen código de enclave, tal como InstanceHash y ExactHash, proporcionan un nivel superior de certeza, por ejemplo al cliente de enclave por medio de afirmación, que cierto código de enclave está ejecutándose dentro de un enclave. Sin embargo, la identidad del enclave

cambiará necesariamente cuando cualquier porción de identidad del código de enclave cambie. Por ejemplo, si una corrección de seguridad u otro fallo es corregido en una nueva versión de una imagen de enclave, también cambiará el valor de identidad resultante basado en el nuevo código. Al proporcionar un mecanismo para que ciertas porciones de código de enclave se excluyan del cálculo de *hash* de identidad, la identidad de un enclave se puede desacoplar de los cambios a la porción excluida del código de enclave. Por ejemplo, cuando el código de enclave del autor depende del código de enclave proporcionado por la plataforma de enclave, la identidad de enclave puede desacoplarse de revisiones a la plataforma dependiente.

En el cuadro 1508, se determina un valor de identidad que puede representar una identidad del enclave instanciado en el cuadro 1502. Un valor de identidad se puede determinar calculando un *hash* sobre la porción de identidad determinada previamente de la imagen o imágenes de enclave (el valor de identidad es la salida de una función de *hash* donde la porción de identidad es la entrada a la función de *hash*). En algunas modalidades, la entrada a la función de *hash* serán porciones de la(s) imagen(es) de enclave original(es), mientras en otras modalidades, la entrada a la función de *hash* serán porciones de un contenedor de enclave después de haber copiado la porción de identidad de la imagen al contenedor (y posiblemente descifrado la porción de identidad en el caso donde una imagen de enclave original está cifrada).

En el cuadro 1510, se puede verificar opcionalmente la integridad del valor de identidad resultante al verificar la integridad de la(s) imagen(es) de enclave original(es). La integridad de una imagen de enclave se puede verificar con una clave pública que corresponde a una clave privada utilizada para firmar la imagen de enclave. Tal par de claves pública/privada pueden estar asociadas, por ejemplo, con el autor de la(s) imagen(es) de enclave, de tal manera que la confianza en el valor de identidad resultante se pueda arraigar en la confianza del autor del enclave.

Finalmente, en el cuadro 1512, se puede llevar a cabo una operación relacionada con el enclave instanciado utilizando el valor de identidad. Por ejemplo: se puede generar o verificar un reporte de afirmación del enclave instanciado para un tipo de identidad; se pueden sellar o desellar los datos del enclave instanciado en una identidad; y se puede utilizar un contador monótonico o un tiempo confiable vinculado al enclave instanciado y tipo de identidad.

Las operaciones de enclave que utilizan tipos de identidad de nivel superior permiten interacciones entre grupos de instanciaciones de enclave posibles. La afirmación a un tipo de identidad de nivel superior puede proporcionar equivalencia de reporte de afirmación para todos los enclaves con la misma identidad de nivel superior. Por ejemplo, un reporte de afirmación para un tipo de identidad AuthorID puede ser equivalente al reporte de afirmación de todos los enclaves instanciados a partir de una imagen primaria que contiene los mismos metadatos de AuthorID. Los datos sellados a un tipo de identidad de nivel superior se pueden desellar por cualquier enclave con el mismo valor de identidad de nivel superior. Por ejemplo, los datos sellados a un enclave instanciado con el tipo de identidad AuthorID se puede desellar por cualquier otro enclave instanciado con los mismos metadatos de AuthorID en su imagen primaria de enclave.

Equivalencia de identidad de enclave

La figura 16 representa un sistema ejemplar con equivalencia de identidad abstracta de enclave. Un cliente de enclave 1602 se puede comunicar con un primer enclave 1612 instanciado en un contenedor de enclave seguro de la primera plataforma nativa 1616 por medio de la plataforma de abstracción 1614, y el cliente 1602 también se puede comunicar con un segundo enclave 1622 instanciado en un contenedor de enclave seguro de la segunda plataforma nativa 1626 por medio de la plataforma de abstracción 1624. La primera plataforma nativa 1616 puede o no residir en la misma computadora que la segunda plataforma nativa. El cliente de enclave 1602 puede residir en la misma computadora que cualquiera de las plataformas nativas, o puede residir en una tercera computadora separada. La primera plataforma nativa 1616 puede no ser la misma que la segunda plataforma nativa 1626. Por ejemplo, la primera plataforma nativa 1616 puede ser una versión más vieja de la segunda plataforma nativa del mismo fabricante de plataforma nativa. Alternativamente, la primera plataforma nativa 1616 y la segunda plataforma nativa 1626 pueden ajustarse a arquitecturas de enclave completamente diferentes, tales como VSM y SGX.

Un cliente de enclave puede determinar de manera segura que los enclaves son equivalentes al comparar los valores de identidad derivados de reportes de afirmación. El cliente de enclave 1602 puede identificar de manera segura cada uno de los enclaves al recibir reportes de afirmación separados del primer enclave 1612 y el segundo enclave 1622. Un valor de identidad se puede incluir en (o derivar de) cada uno de estos reportes de afirmación. Si los valores de identidad son iguales, el cliente de enclave 1602 puede tener la confianza de que el primer enclave 1612 y el segundo enclave 1622 son equivalentes en cierto sentido. Los valores de identidad de los reportes de afirmación pueden ser valores de identidad abstracta que corresponden a un tipo de identidad abstracta particular (tal como ExactHash, InstanceHash, ImageID, FamilyID, o AuthorID), o *hashes* de tales valores de identidad abstracta. En este caso, se puede determinar equivalencia donde los enclaves no son exactamente idénticos. Dos enclaves puede no ser exactamente idénticos, pero aun así determinarse como equivalentes, por ejemplo cuando las imágenes de enclave cargadas en el contenedor de enclave son versiones diferentes de la misma funcionalidad, o las mismas imágenes primarias con diferentes imágenes dependientes, o las mismas imágenes de enclave cargadas en contenedores de enclave de arquitecturas de enclave diferentes.

El primer enclave 1612 se puede considerar equivalente pero no idéntico al segundo enclave 1622 por una variedad de situaciones. En un primer ejemplo, solamente un subconjunto del código cargado inicialmente en los contenedores de enclave es el mismo (por ejemplo, equivalente para tipos de identidad abstracta ExactHash o InstanceHash). En un segundo ejemplo, el autor del código de enclave puede haber incluido un ID idéntico en dos imágenes binarias de enclave diferentes, aunque el código en las dos imágenes binarias sea diferente (por ejemplo, ImageID, FamilyID, o AuthorID). En un tercer ejemplo, el código en cada enclave es completamente el mismo pero se carga (se instancia) en diferentes plataformas nativas. En este tercer ejemplo, la primera plataforma nativa 1616 y la segunda plataforma nativa 1626 pueden ser fabricadas por diferentes fabricantes y por lo tanto la confianza de los diferentes reportes de afirmación se arraiga en las diferentes autoridades de certificación (ver la figura 7,

elemento 738) de los diferentes fabricantes. Un ejemplo donde las dos plataformas nativas son diferentes es en una torre de servidores o en informática en la nube donde los servidores designados para las cargas de trabajo de procesamiento del primer y segundo enclaves son servidores que no soportan la misma plataforma de enclave nativa.

En una modalidad alterna, el primer enclave puede ser el cliente del segundo enclave, de tal manera que los cuadros 1602 y 1612 se combinan. Determinar la equivalencia de enclave en esta modalidad puede incluir determinar, dentro del primer enclave, que un valor de identidad de un reporte de afirmación del segundo enclave es igual que el valor de identidad propio del primer enclave (en un nivel de identidad abstracta particular).

La figura 17 representa un diagrama de flujo ejemplar para procesamiento paralelo con dos enclaves equivalentes. El proceso 1700 se puede llevar a cabo, por ejemplo, por un cliente de dos o más enclaves diferentes. En el cuadro 1702, se instancian dos enclaves en casos de plataforma nativa diferentes, por ejemplo como se representa en la figura 16. Los enclaves se pueden instanciar por un cliente de enclave que especifica una imagen binaria de enclave (tal como la imagen primaria 1410 de la figura 14) por medio de un método de CreateEnclave de las plataformas de abstracción 1614 y 1624. La imagen binaria de enclave especificada para instanciar los dos enclaves puede ser igual o diferente. En el cuadro 1704 se crea un reporte de afirmación para cada enclave instanciado. Los reportes de afirmación se pueden crear, por ejemplo, a petición del cliente de enclave o a petición de los mismos enclaves. Una entidad que desea probar la equivalencia de los dos enclaves, tal como el cliente de enclave, obtiene copias de ambos reportes de afirmación. Los reportes de afirmación se pueden verificar opcionalmente en el cuadro 1706. Por ejemplo, se puede verificar la integridad del reporte verificando la firma de afirmación con un certificado de aprobación (tal como en la figura 7, elemento 728) de la plataforma nativa que produjo el reporte de afirmación. Además, el certificado de aprobación se puede verificar con la clave pública del fabricante de la plataforma nativa (tal como en la figura 7, elemento 732). Los valores de identidad (o un hash de los mismos) se pueden extraer de cada reporte de afirmación en el cuadro 1708, y se puede determinar la equivalencia de los dos enclaves verificando que los valores de identidad extraídos sean iguales para cada enclave. Estos valores de identidad pueden ser valores de identidad abstracta (o *hashes* de los mismos) asociados con un tipo de identidad.

Una vez que un cliente de enclave ha probado la equivalencia de dos instanciaciones de enclave a partir de las operaciones en los cuadros 1708 y 1710, los dos enclaves se pueden utilizar de manera intercambiable, de acuerdo con el tipo de equivalencia mostrado. Los cuadros 1712-1720 representan un método ejemplar para utilizar los enclaves equivalentes para utilizar los dos enclaves instanciados, equivalentes de una manera de procesamiento paralelo. En los cuadros 1712 y 1716, se copia una porción de un conjunto de datos de entrada, tal como la porción de una base de datos o porción de un archivo de imagen digital, en el primer y segundo enclaves. La porción del conjunto de datos copiado puede ser idéntica o diferente de acuerdo con la tarea de procesamiento en cuestión. Una operación de procesamiento se puede llevar a cabo de manera segura en paralelo de manera simultánea al llevar a cabo parcialmente la operación en el primer enclave en el cuadro 1714 y llevar a cabo parcialmente la

operación en el segundo enclave en el cuadro 1718. La operación puede ser, por ejemplo, buscar en la base de datos o llevar a cabo una operación de procesamiento de imagen. El primer enclave puede buscar en la primera mitad de la base de datos o llevar a cabo la operación de procesamiento de imagen en la primera mitad de una imagen, mientras el segundo enclave puede buscar en la segunda mitad de la base de datos o llevar a cabo la operación de procesamiento de imagen de la segunda mitad de la imagen. Finalmente, en el cuadro 1720, se pueden combinar los resultados del procesamiento paralelo en el primer y segundo enclaves, por ejemplo combinando las dos mitades clasificadas de la base de datos, o uniendo de nuevo las dos mitades de imagen.

La figura 18 representa un diagrama de flujo ejemplar para procesamiento en serie con dos enclaves equivalentes. Como se representa en la figura 18, una operación de enclave, tal como una operación de base de datos o una operación de procesamiento de imagen, se hace de manera segura en dos partes secuenciales en dos enclaves separados. El proceso 1800 se puede llevar a cabo, por ejemplo, por el cliente de enclave 1602 de la figura 16. En el cuadro 1802, se crea un primer enclave en una primera plataforma de enclave nativa, y en el cuadro 1804 se crea un reporte de afirmación del primer enclave. Este primer reporte de afirmación (del primer enclave) se puede verificar en el cuadro 1806, por ejemplo como se describió anteriormente con respecto al cuadro 1706 de la figura 17. En el cuadro 1808, se inicia una operación segura en el primer enclave, pero no se completa. Opcionalmente, en el cuadro 1810 se puede sellar el estado del primer enclave para moverse de manera segura fuera del primer enclave. Por ejemplo, el estado del primer enclave se puede sellar a un tipo de identidad del primer enclave. Una vez que se ha guardado el estado de enclave, se puede determinar el primer enclave (no representado).

Un segundo enclave se utiliza iniciando en el cuadro 1812. En el cuadro 1812, se instancia el segundo enclave en una segunda plataforma nativa. Como en las figuras 16 y 17, la segunda plataforma nativa puede o no estar alojada en la misma computadora que la primera plataforma nativa, y la primera y segunda plataformas nativas pueden ser iguales o diferentes. En el cuadro 1814 se crea un reporte de afirmación de la segunda plataforma nativa, y, opcionalmente, este segundo reporte de afirmación se puede verificar en el cuadro 1816. En el cuadro 1818 se puede comparar un valor de identidad del primer y segundo reportes de afirmación para verificar la equivalencia del primer y segundo enclaves. En modalidades alternas, el segundo enclave se puede instanciar y puede verificar la equivalencia (cuadros 1812-1818) antes de iniciar la operación segura en el primer enclave en el cuadro 1808. Para continuar la operación segura iniciada en el primer enclave, se puede copiar el estado sellado del primer enclave al segundo enclave y desellar en el cuadro 1820. En el cuadro 1822, se completa la operación segura en el segundo enclave (utilizando el estado de enclave copiado de manera segura del primer enclave, si se copió el estado).

Sellado de datos distribuido

La figura 19 es un diagrama de bloques de un sistema de sellado de datos distribuido ejemplar. El sellado de datos se puede distribuir a través de múltiples enclaves, donde esos enclaves están alojados

en plataformas de enclave nativas separadas, y/o en computadoras separadas. Como se explicó anteriormente, las unidades básicas de abstracción EnclaveSeal y EnclaveUnseal pueden sellar y desellar datos para un enclave utilizando una clave vinculada a la plataforma de enclave nativa o computadora física en la cual se está ejecutando un enclave. Esto puede restringir el desellado a solamente enclaves alojados en la misma computadora o el mismo caso de plataforma de enclave nativa. La figura 19 representa un sistema de sellado de datos distribuido, donde el sellado o desellado de datos puede ocurrir en una plataforma nativa o computadora diferente a la plataforma nativa y/o computadora que aloja un enclave. El sistema 1900 incluye las computadoras 1910, 1930, 1950, con las redes 1902 conectando las computadoras 1910 y 1930, y la red 1904 conectando las computadoras 1930 y 1950. La computadora 1910 aloja el enclave de origen 1912, desde el cual se pueden asegurar los datos que se van a sellar; la computadora 1930 aloja un enclave de sellado distribuido (DSE, Distributed Sealing Enclave, por sus siglas en inglés) 1932 para dar servicio a solicitudes de sellado y desellado distribuidos; y la computadora 1950 aloja el enclave de destino 1952 donde se desellan los datos previamente sellados. Como se explicó anteriormente con respecto a la figura 9, los enclaves 1912, 1932, 1952 pueden comunicarse con las plataformas de abstracción 1914, 1934, 1954, respectivamente, por medio de un protocolo de abstracción de enclave, y las plataformas de abstracción 1914, 1934, 1954 pueden comunicarse con las plataformas nativas 1916, 1936, y 1956, respectivamente, por medio de un protocolo nativo. En modalidades alternas, uno o más enclaves 1912, 1932, 1950 pueden alojarse directamente en las plataformas nativas 1961, 1936, 1956 sin una plataforma de abstracción intermediaria. Los datos sellados 1938 pueden ser datos sellados al DSE 1932 utilizando una clave asociada con el DSE 1932 o su plataforma nativa 1936 de alojamiento. Los datos sellados 1938 se pueden almacenar en una ubicación menos protegida, tal como en la computadora 1930 fuera del contenedor de enclave seguro de DSE 1932, por ejemplo en otra parte en el espacio de memoria de la computadora 1930 o en un sistema de archivos de un disco duro.

El sellado de datos distribuido puede incluir autenticación de DSE 1930 enclave seguro, por ejemplo por afirmación de DSE 1932 a través de la red 1902. Una vez que el enclave de origen 1912 confía en el DSE 1932, el enclave de origen 1912 puede enviar datos sensibles a través de un canal de comunicación seguro al DSE 1932 junto con una política de sellado para sellar el DSE 1932. El DSE 1932 puede entonces sellar los datos del enclave 1912 en sí mismo y almacenar los datos sellados en almacenamiento no seguro. Después, el enclave de destino 1952 puede solicitar los datos sellados previamente. Para desellar los datos, el DSE 1932 puede autenticar el enclave de destino 1952, por ejemplo por afirmación a través de la red 1904, y verificar que el desellado para el enclave de destino 1952 está permitido de acuerdo con la política de sellado proporcionada por el enclave de origen 1912. El DSE 1932 puede desellar los datos previamente sellados del enclave de origen 1912, y después enviar los datos desellados a través de un canal de comunicación seguro al enclave de destino 1952. Los datos de enclave se pueden comunicar de manera segura hacia y desde el DSE 1932 cifrando los datos de enclave a través de las redes 1902 y 1904. Por ejemplo, los datos de enclave enviados a través de la red 1902 se pueden cifrar con una clave generada durante la afirmación del DSE 1932 al enclave de origen

1912, y los datos enviados a través de la red 1904 se pueden cifrar con una clave generada durante la afirmación del enclave de destino 1952 al DSE 1932. Otros canales de comunicación seguros son posibles, tal como cifrado con una clave pública del destino, por ejemplo una clave pública asociada con el DSE o una clave pública asociada con el enclave de destino.

Las identidades de enclave utilizadas en sellado y desellado distribuidos pueden o no ser identidades abstractas de enclave. Por ejemplo, en algunas modalidades con una capa de plataforma de abstracción, una política de sellado, tal como una especificada por un enclave de origen y cumplimentada por un DSE, puede identificar identidades de enclave de desellado permitidas donde las identidades de enclave de desellado permitidas son, por ejemplo, una lista de identidades abstractas de enclave, o una lista de tipos de identidad abstracta en combinación con los valores de identidad abstracta del enclave de origen. En otras situaciones, se puede utilizar una identidad no abstracta. Por ejemplo, en algunas modalidades, se puede implementar un DSE con código disponible públicamente, de tal manera que la confianza en el DSE es confianza en el conocimiento de su código contrario a la confianza en el autor de su código. En este ejemplo, la afirmación de un DSE puede ser un *hash* firmado de todo del código público del DSE, y la entrada a la función de *hash* pueden no incluir valores de identidad abstracta asignados por el autor.

En algunas modalidades, las plataformas nativas 1916, 1936, 1956 son plataformas nativas separadas debido a que están alojadas en diferentes computadoras 1910, 1930, 1950, aun si las plataformas nativas 1916, 1936, 1956 se ajustan a la misma versión de la misma arquitectura de plataforma de enclave relativa. En otras modalidades, las plataformas nativas 1916, 1936, 1956 pueden ajustarse a diferentes arquitecturas de plataforma o diferentes versiones de la misma arquitectura de plataforma de enclave nativa. El uso de identidades abstractas en la política de sellado puede facilitar el alojamiento de enclaves de origen y de destino en diferentes arquitecturas de plataforma nativa.

En otras modalidades de sellado de datos distribuido no representadas en la figura 19, puede no haber computadoras separadas (tales como las computadoras separadas 1910, 1930, 1950). Por ejemplo, los enclaves de origen y destino pueden estar en una computadora (y tal vez en una sola plataforma nativa), mientras el DSE está en una computadora separada. Alternativamente, un DSE puede estar alojado en la misma computadora que cualquiera de la computadora que aloja el enclave de origen o la computadora que aloja el enclave de destino. En estas modalidades de sellado de datos distribuido, el sellado y desellado de datos no es completamente local a una sola computadora, como se describió anteriormente con respecto a las unidades básicas de abstracción EnclaveSeal y EnclaveUnseal.

El sellado de datos distribuido se puede implementar en una API de capa de abstracción, tal como por las plataformas de abstracción 1914, 1934, 1954. Por ejemplo, las unidades básicas DistributedEnclaveSeal y DistributedEnclaveUnseal son similares a las unidades básicas de sellado de datos local EnclaveSeal y EnclaveUnseal comentadas anteriormente.

DWORD

DistributedEnclaveSeal (

In SEALING_POLICY sealingPolicy,

```

    _In_reads_bytes_opt_(dwPlaintextSize) LPCVOID pPlaintext,
    _In_                DWORD dwPlaintextSize,
    _In_reads_bytes_opt_(dwAuthdataSize) LPCVOID pAuthdata,
    _In_                DWORD dwAuthdataSize,
    _Out_writes_bytes_to_(dwSealedtextSize) LPVOID pSealedtext,
    _Inout_             DWORD dwSealedtextSize,
    Set<EnclaveIdentity> SetOfTargetEnclaves
)

```

DistributedEnclaveSeal extiende EnclaveSeal tomando un parámetro SetOfTargetEnclaves adicional, que permite un enclave de llamada, tal como el enclave 1910, para especificar un conjunto de identidades de enclave que están autorizadas para desellar los datos proporcionados por medio del parámetro pPlaintext. Si no se proporciona ninguna identidad por medio de SetOfTargetEnclaves, se puede suponer que una identidad de enclave autorizada por defecto es una identidad del enclave de sellado, por ejemplo ExactHash o InstanceHash del enclave de sellado.

La implementación de DistributedEnclaveSeal, por ejemplo como un método de la plataforma de abstracción 1914 en la computadora del enclave de origen, puede incluir establecer un canal de comunicación seguro con un DSE, tal como cifrando el mensaje a través de la red 1902. La(s) clave(s) para este cifrado puede(n) ser generada(s), por ejemplo, durante un proceso de afirmación, como se describió anteriormente, o puede(n) ser cualquier clave pública asociada con el DSE 1932.

DistributedEnclaveSeal se puede generalizar adicionalmente tomando un parámetro adicional KeyForData (no mostrado en la función DistributedEnclaveSeal prototipo anterior). En algunas modalidades, múltiples conjuntos de datos se pueden mantener sellados simultáneamente para un solo enclave o una sola identidad de enclave. En este caso, KeyForData permite la especificación de cuál conjunto de datos se está sellando. KeyForData puede ser cualquier tipo de identificador de datos, tal como una cadena, un número, un conjunto de propiedades. En algunas modalidades, KeyForData puede ser un parámetro de entrada para DistributedEnclaveSeal y ser generado por el enclave de sellado, permitiendo de manera efectiva que el enclave de sellado nombre el conjunto de datos. En otras modalidades, KeyForData puede ser un parámetro de salida, donde el DSE genera el identificador de KeyForData conforme los datos son sellados.

```

    DWORD
    DistributedEnclaveUnseal(
    _In_reads_bytes_opt_(dwSealedtextSize) LPCVOID pSealedtext,
    _In_                DWORD wSealedtextSize,
    _In_reads_bytes_opt_(dwAuthdataSize) LPCVOID pAuthdata,
    _In_                DWORD dwAuthdataSize,
    _Out_writes_bytes_to_(dwPlaintextSize) LPCVOID pPlaintext,
    _Inout_             DWORD dwPlaintextSize
    Clave    KeyForData,

```

Identidad EnclaveIdentity

)

DistributedEnclaveUnseal se puede implementar en la plataforma de abstracción 1954, y operar en respuesta a una solicitud de un enclave de destino 1952. DistributedEnclaveUnseal puede establecer un canal de comunicación seguro a DSE 1932, por ejemplo, pero cifrando mensajes con una clave generada durante la afirmación del enclave de destino 1952 al DSE en 1932, o cifrando mensajes enviados al enclave de destino con una clave pública del enclave de destino. El DSE puede verificar una identidad del enclave solicitante (destino) tal como por afirmación, desellar los datos sellado solicitados, y enviar de manera segura los datos desellados al enclave solicitante. En modalidades donde el enclave solicitante tiene múltiples identidades, se puede especificar una identidad particular en el parámetro de identidad. En modalidades donde se almacenan múltiples conjuntos de datos de enclave para una sola identidad de enclave, el parámetro KeyForData puede especificar cuál conjunto de datos sellados (para la identidad especificada) se solicita utilizando el mismo valor KeyForData utilizado en DistributedEnclaveSeal cuando se selló el conjunto de datos.

En algunas modalidades, las identidades de enclaves autorizados para desellar datos se pueden especificar (tal como en el parámetro SetOfTargetEnclaves) por medio de claves públicas de los enclaves objetivo autorizados. En esta modalidad, la afirmación del enclave de destino al DSE puede no ser necesaria, pero los datos desellados pueden entonces proporcionarse solamente como cifrados utilizando una de las claves públicas especificadas. Suponiendo que solamente los enclaves objetivo tienen acceso a las claves privadas correspondientes para descifrar, solamente los enclaves objetivo tendrán acceso a los datos desellados.

En modalidades no representadas en la figura 19, las funciones del enclave de sellado distribuido (DSE) 1932 pueden en sí distribuirse a través de múltiples DSEs. Por ejemplo, la funcionalidad de DSE se puede distribuir a través de múltiples DSEs en múltiples computadoras para redundancia y tolerancia a fallos. En este ejemplo, cualquier DSE replicado puede ser capaz de dar servicio a una solicitud de sellado o desellado. Observar que los datos sellados 1938, una vez que son sellados/cifrados, se pueden almacenar de manera segura en cualquier lado, incluyendo replicarse a través de servidores de almacenamiento en la nube.

El sellado de datos distribuido puede permitir el movimiento de cargas de trabajo de enclave entre computadoras. Por ejemplo, los datos de enclave de origen sellados por un DSE pueden ser datos de estado del enclave de origen en un primer servidor en la nube, Los cuales se pueden cargar en el enclave de destino en un segundo servidor en la nube después del desellado. Esto se puede hacer de manera similar a la descrita anteriormente con respecto a la figura 18. Una operación segura puede iniciar la ejecución en el enclave de origen. Después, tal vez después de que la ejecución en el enclave de origen es interrumpida, se puede sellar el estado del enclave de origen a un DSE, y después desellar a un enclave de destino cuando el enclave de destino está listo para continuar la operación segura que se inició en el enclave de origen.

La figura 20 es un diagrama de flujo ejemplar para sellado y desellado de datos distribuidos,

como se podría llevar a cabo por un enclave de sellado o DSE. Los cuadros 2002-2006 corresponden al sellado de datos distribuido, mientras los cuadros 2008-2010 corresponden al desellado de datos distribuido. En respuesta a una solicitud de sellar un conjunto de datos de enclave que se origina desde un enclave de origen, el enclave de sellado (o DSE) puede afirmarse a sí mismo al enclave de origen, enviando un reporte o cita de afirmación al enclave de origen en el cuadro 2002. El enclave de origen puede verificar la identidad del enclave de sellado como un enclave de sellado genuino y confiable, al inspeccionar un valor y firme de identidad en el reporte de afirmación del enclave de sellado. En el cuadro 2004, el enclave de sellado recibe una lista permitida y los datos de enclave que se van a sellar. Estos pueden recibirse a través de un canal seguro como se describió anteriormente con respecto a la figura 19. En el cuadro opcional 2006, el enclave de sellado puede sellar los datos de enclave de origen a sí mismo, por ejemplo si los datos están almacenados fuera del contenedor seguro del enclave de sellado, tal como en un sistema de archivos de computadora. Para desellar los datos para un enclave de destino, el enclave de destino puede afirmarse a sí mismo al enclave de sellado, tal como proporcionando un reporte o cita de afirmación, en el cuadro 2008. En el cuadro 2010, se puede verificar una identidad del enclave de destino, tal como inspeccionando el reporte de afirmación del enclave de destino. En el cuadro 2012, el enclave de sellado determina si el enclave de destino tiene permitido desellar datos del enclave de origen verificando que se incluye una identidad autenticada del enclave de destino en la lista permitida recibida con los datos. Una vez que se ha confirmado el permiso, se pueden desellar los datos de enclave si estaban sellados, y enviar entonces al enclave de destino a través de un canal seguro en el cuadro 2014.

Enclave de bóveda de claves

Las bóvedas de claves se pueden implementar con los enclaves. Una bóveda de claves contiene de manera segura claves, tales como claves de un sistema de cifrado para cifrar y descifrar datos, para clientes de la bóveda de claves. Una bóveda de claves puede llevar a cabo adicionalmente operaciones con la clave, tal como cifrar y descifrar datos, firmar datos, y derivar nuevas claves a partir de una clave existente. Una bóveda de claves, cuando se implementa como un enclave, puede proporcionar almacenamiento muy seguro de ir procesamiento con claves secretas de cifrado. Adicionalmente, la afirmación de software de un enclave de bóveda de claves puede proporcionar altos niveles de certeza a un cliente de bóveda de que se está comunicando con una bóveda de claves auténtica y confiable. Sistemas altamente seguros se pueden construir en un enclave de bóveda de claves con una clave de bóveda bloqueada, por lo cual una clave almacenada dentro de una bóveda de claves nunca se libera a ningún cliente fuera de la bóveda de claves, y en algunos casos, la clave de bóveda bloqueada puede que sólo exista tal como está almacenada dentro de (o posiblemente sellada a) el enclave de bóveda de claves.

La figura 21 es un diagrama de bloques de un enclave de bóveda de claves ejemplar. El enclave 2122 es una bóveda de claves dentro de un contenedor de enclave seguro de la segunda plataforma de enclave nativa 2126. En el ejemplo de la figura 21, el cliente 2112 del enclave de bóveda de claves 2122

también es un enclave, y alojado dentro de un contenedor de enclave seguro de la primera plataforma de enclave nativa 2116. Los enclaves 2112, 2122 pueden interactuar con sus plataformas nativas 2116, 2126 respectivas a través de las plataformas de abstracción de enclave 2114, 2124 respectivas. En otras modalidades, una o ambas plataformas de abstracción 2114, 2124 puede no existir cuando los enclaves 2112 y/o 2122 interactúan directamente con las plataformas nativas 2116, 2126.

El enclave de bóveda de claves 2122 puede comunicarse con el cliente de bóveda 2112 a través del canal de comunicaciones 2150. En algunas modalidades, el canal de comunicaciones 2112 puede ser un canal de comunicaciones seguro que proporciona certeza de confidencialidad, integridad, y/o frecuencia de actualización de mensajes enviados a través del canal de comunicación 2150. La confidencialidad e integridad de tal canal de comunicaciones seguras se pueden establecer, por ejemplo, con cifrado y firmas, como en las figuras 2 y 3, utilizando claves compartidas generadas como parte de un proceso de afirmación, como en las figuras 5 y 6.

La afirmación de software proporciona seguridad en parte al proporcionar certeza de la identidad de la entidad en el otro lado de un canal de comunicación. Al afirmar el enclave de bóveda de claves 2122 a un cliente de bóveda, el cliente puede adquirir certeza de que el enclave de bóveda de claves 2122 es quien dice ser antes de enviar un secreto, tal como una clave u otros datos de texto sin cifrar, a la bóveda de claves. Lo inverso también es cierto para clientes que también son enclaves, como se representa en la figura 21. Al afirmar el enclave de bóveda 2112 a un enclave de bóveda de claves 2122, la bóveda de claves puede adquirir certeza de que el cliente es quien dice ser antes de revelar un secreto, tal como una clave u otros datos de texto sin cifrar, al cliente.

Los sistemas de bóveda de claves con claves de bóveda bloqueada y claves derivadas, particularmente cuando las claves de cifrado se derivan de una clave de bóveda bloqueada, se pueden utilizar para construir un sistema de seguridad que sea flexible y muy seguro. Una función de derivación de clave, que puede o no ser pública, se puede utilizar para generar múltiples claves a partir de una primera clave. La primera clave (un secreto de raíz) puede ser de bóveda bloqueada para el nivel de seguridad más alto, y las claves derivadas a partir de la primera clave se pueden utilizar para fines de cifrado. Si una clave derivada se ve comprometida, se puede generar una nueva clave derivada en un sistema existente sin tener que acceder o cambiar la bóveda de claves que contiene la primera clave.

Un enclave de bóveda de claves (KVE, Key Vault Enclave, por sus siglas en inglés) ejemplar es un sistema de bóveda de claves basado en la nube que proporciona almacenamiento, generación, derivación, distribución, cifrado, descifrado, y firmas de clave utilizando enclaves. El KVE puede ser identificado por su *hash* exacto (un *hash* del contenido de su contenedor seguro), o por un identificador arbitrario asignado por o asociado con su creador. En el último caso, el enclave puede firmarse con la clave privada de su creador para evitar conflictos y violaciones de seguridad debido a colisiones del identificador.

Un cliente de bóveda de claves puede interactuar con un sistema de bóveda de claves utilizando las siguientes unidades básicas ejemplares. Un prototipo de función StoreKey ejemplar es:

StoreKey([in] Keyname, [in] KeyType, [in] KeyValue, [in] Policy)

StoreKey almacena una clave dada en la bóveda de claves y la asocia con un nombre dado. El tipo de clave también se asocia con la clave y restringe lo que se puede hacer con la clave. Por ejemplo, algunas claves solo deben utilizarse para cifrado, otras para firmas, etc. Y claves específicas pueden sólo utilizarse con algoritmos criptográficos específicos. La política puede especificar políticas para restringir adicionalmente el uso de la clave. Por ejemplo, puede especificar un conjunto de identidades de enclave que son permitidas para recuperar la clave y/o utilizar la clave. También puede especificar propiedades temporales, por ejemplo, que la clave debe ser destruida en una cierta fecha, o que se debe limitar la tasa de operaciones utilizando la clave.

Un prototipo de función GenerateKey ejemplar es:

GenerateKey([in] keyName, [in] keyType, [in] Policy)

Generate Key genera una nueva clave de un cierto tipo y la mantiene almacenada dentro de la bóveda de claves, esto es, la clave nunca abandona la bóveda de claves.

Un prototipo de función GetKey ejemplar es:

GetKey([in] KeyName, [out] KeyValue)

GetKey obtiene una clave almacenada dentro de la bóveda de claves. Estas unidades básicas se implementan generalmente a través de un canal de comunicación seguro y el código que llama a las unidades básicas se ejecuta generalmente en un entorno confiable. En tal contexto, a menudo es aceptable recuperar una clave de la bóveda de claves.

Un prototipo de función DeleteKey ejemplar es:

DeleteKey([in] keyName)

DeleteKey elimina una clave de la bóveda de claves.

Un prototipo de función DeriveKey ejemplar es:

DeriveKey([in] newKeyName, [in] KeyName, [in] kdfIdentifier, [in] AdditionalData)

DeriveKey utiliza una función de derivación de clave (KDF, Key Derivation Function, por sus siglas en inglés) criptográfica identificada por kdfIdentifier para derivar una nueva clave con base en la clave identificada por keyName y los datos pasados en AdditionalData.

Un prototipo de función Encrypt ejemplar es:

Encrypt([in] KeyName, [in] algorithm, [in] data, [out] encryptedData)

Encrypt cifra los datos con la clave identificada por KeyName, utilizando el algoritmo solicitado.

Un prototipo de función Decrypt ejemplar es:

Decrypt([in] KeyName, [in] algorithm, [in] encryptedData, [out] data)

Decrypt descifra los datos con la clave identificada por KeyName, utilizando el algoritmo solicitado.

Un prototipo de función Sign ejemplar es:

Sign([in] KeyName, [in] algorithm, [in] data, [out] signature)

Sign firma los datos con la clave identificada por KeyName, utilizando el algoritmo solicitado.

Un prototipo de función VerifySignature ejemplar es:

VerifySignature([in] KeyName, [in] algorithm, [in] signature, [out] bool signatureIsCorrect)

VerifySignature verifica la firma con la clave identificada por KeyName, utilizando el algoritmo solicitado.

Todas de las unidades básicas de bóveda de claves anteriores se pueden implementar estableciendo un canal seguro con el KVE. El canal se puede establecer utilizando afirmación y llevando a cabo un intercambio de claves de Diffie-Hellman como se describió anteriormente con respecto a las figuras 5 y 6. Después de que se establece el canal de comunicación, la solicitud se envía de manera segura a través del canal y la respuesta se lee desde el canal. El canal puede proporcionar garantías de confidencialidad e integridad de los datos intercambiados.

En otra modalidad, la primera vez que se ejecuta KVE, genera un par de clave pública/privada y genera una cita para la clave pública. Después escribe la cita y la clave pública, mientras mantiene la clave privada dentro del enclave. La clave pública y la cita después se pueden distribuir a todos los sistemas/código que deseen utilizar la bóveda de claves. En este caso, la implementación de las unidades básicas anteriores verifica la cita para asegurar que se está hablando con un KVE genuino, y después cifra las solicitudes utilizando la clave pública del KVE. Como parte de la solicitud, la implementación de las unidades básicas puede incluir una clave para cifrar y proteger la integridad de los resultados enviados desde el KVE. Esta modalidad puede proporcionar un canal de comunicaciones seguro de dos vías sin afirmación.

La figura 22 es un diagrama de flujo ejemplar para algunas operaciones de enclave de bóveda de claves. El proceso 2200 inicia en el cuadro 2202 al almacenar de manera segura, dentro del enclave de bóveda de claves, una clave utilizada en un sistema de cifrado. La clave puede ser utilizada, por ejemplo, para cifrar o descifrar datos, generar una firma criptográfica, o se puede utilizar solamente como una clave de raíz de la cual derivar otras claves. La clave se puede almacenar de manera segura en el enclave de bóveda de claves al, por ejemplo, almacenar la clave dentro del espacio de memoria del contenedor seguro del enclave. En otras modalidades, la clave se puede mantener segura fuera del contenedor de enclave seguro al sellar los datos de clave al enclave de bóveda de claves, o se puede mantener segura al sellar remotamente con un enclave de sellado distribuido, como se describió con respecto a las figuras 19 y 20.

En el cuadro 2204, el enclave de bóveda de claves lleva a cabo un proceso de afirmación para afirmar la identidad del enclave de bóveda de claves al cliente de bóveda. Esto puede dar al cliente la certeza de que la bóveda de claves no es un impostor y le pueden confiar secretos tales como una clave o datos que se van a cifrar. La afirmación del enclave de bóveda de claves puede incluir enviar, al cliente de bóveda, un reporte de afirmación o cita de afirmación del enclave de bóveda de claves. El cliente de bóveda de claves puede después verificar la integridad del reporte de afirmación al verificar una firma en el reporte de afirmación con una clave pública asociada con la plataforma de enclave nativa del enclave de bóveda de claves. Por ejemplo, el reporte de afirmación de la bóveda de claves 2122 se puede generar por medio de una segunda plataforma nativa 2126, y el cliente de bóveda 2112 puede verificar la firma en el reporte utilizando una clave pública asociada con la segunda plataforma nativa 2126. Este proceso de afirmación también puede generar claves utilizadas para un canal de comunicación seguro

entre el cliente de bóveda y el enclave de bóveda de claves, por ejemplo como se muestra en las figuras 5 y 6. El reporte de afirmación debe incluir una identidad del enclave de bóveda de claves que se puede determinar de varias formas como se describió anteriormente, por ejemplo con respecto a las figuras 14 y 15. La identidad se puede basar, por ejemplo, en un *hash* de todo el contenido del contenedor seguro del enclave de bóveda de claves, un *hash* de solamente un identificador único asignado por el autor/creador del enclave de bóveda de claves, o un *hash* de una combinación de una porción del contenido del contenedor y un identificador único.

Algunas operaciones de enclave de bóveda de claves también pueden requerir certeza de la identidad del cliente de bóveda. Por ejemplo, descifrar datos o divulgar una clave (tal como con las unidades básicas Decrypt o GetKey) puede requerir tal certeza. En estas situaciones, si un cliente de bóveda también es un enclave, el cuadro opcional 2208 incluye un proceso de afirmación para verificar, por medio del enclave de bóveda de claves, la identidad del cliente de bóveda. El proceso de afirmación del cuadro 2208 puede incluir recibir, en el enclave de bóveda de claves, un reporte o cita de afirmación del cliente de bóveda.

En el cuadro opcional 2210, se puede establecer un canal de comunicaciones seguro entre la bóveda de claves y el enclave de bóveda de claves. Se puede requerir comunicación segura para pasar secretos entre el cliente de bóveda y el enclave de bóveda de claves, tales como claves o datos que se van a cifrar. El proceso de afirmación del cuadro 2004 o 2008 puede generar claves que se pueden utilizar para crear un canal de comunicación seguro entre el cliente de bóveda y el enclave de bóveda de claves, por ejemplo como se muestra en las figuras 5 y 6. Alternativamente, se puede utilizar cualquier clave pública conocida de un destino de mensaje para enviar un mensaje de manera segura.

En el cuadro 2212, se puede llevar a cabo una operación de clave, tal como una de las unidades básicas de bóveda de claves descritas anteriormente, dentro del enclave de bóveda de claves. Durante esta operación, los datos de clave se pueden almacenar solamente en el espacio de dirección del contenedor seguro del enclave de bóveda de claves. Unidades básicas ejemplares incluyen DeriveKey, Decrypt, Sign, y otras.

El proceso 2200 presume que un enclave de bóveda de claves ya conoce la clave. Observar que para algunas operaciones o funciones básicas de enclave de bóveda de claves, tales como StoreKey o GenerateKey, el orden de operaciones puede ser diferente al que se representa en el proceso 2200. Por ejemplo, para GenerateKey, la operación de generación de clave (como en el cuadro 2212) ocurrirá antes de la operación de almacenamiento seguro del cuadro 2202. Tal orden de operaciones se representa en la figura 23, cuadros 2302-2308.

La figura 23 es un diagrama de flujo ejemplar para crear y utilizar un enclave de bóveda de claves con una clave de bóveda bloqueada. En los cuadros 2302-2308 del proceso 2300, se deriva una nueva clave dentro del enclave de bóveda de claves. En los cuadros 2310-2316, se utiliza la clave recién derivada para llevar a cabo una operación de descifrado. Este es un uso ejemplar de la clave de bóveda bloqueada, por lo cual todas las operaciones de clave se llevan a cabo con el enclave de bóveda de claves y la clave nunca se proporciona a un cliente de bóveda. Además, la nueva clave en este ejemplo

puede nunca existir fuera del enclave de bóveda de claves, debido a que fue creada (derivada) desde dentro del mismo enclave de bóveda de claves, y nunca se proporcionó al enclave de bóveda de claves desde un cliente de bóveda u otra parte. Para algunas modalidades y políticas de uso de clave, una clave de bóveda bloqueada puede ser efímera de modo que nunca abandona el contenedor seguro del enclave de bóveda de claves, incluso después de sellar la clave al enclave de bóveda de claves. Tal clave efímera, tal como podría ocurrir con una clave derivada utilizada para asegurar temporalmente un canal de comunicaciones, puede dejar de existir en cualquier parte cuando el contenedor del enclave de bóveda de claves es destruido o terminado. Mientras el proceso de la figura 23 ilustra cómo se puede utilizar una clave de bóveda bloqueada, el proceso de la figura 23 también se puede utilizar con una clave que no es de bóveda bloqueada, por ejemplo si la política de uso de clave permite que la clave regrese al cliente que solicitó su creación.

En el cuadro 2302, el enclave de bóveda de claves se afirma a sí mismo al cliente de bóveda. Esto puede ser requerido por el cliente debido a que el cliente proporcionará un secreto que será cifrado en el cuadro 2312. En el cuadro 2304, el enclave de bóveda de claves puede recibir, por ejemplo del cliente de bóveda, una indicación de una política de uso de clave. La indicación puede ser, por ejemplo, una estructura de datos que especifica la política, o puede ser un identificador que será utilizado con un registro de políticas de uso de clave. La política de uso de clave en sí puede indicar que esta clave nunca debe ser proporcionada a ningún cliente de bóveda. En el cuadro 2306, se deriva una nueva clave a partir de una clave de raíz conocida previamente, por ejemplo con la unidad básica DeriveKey descrita anteriormente. Se puede recibir una solicitud (no representada) para derivar la nueva clave por el enclave de bóveda de claves de, por ejemplo, el cliente de bóveda. En el cuadro 2308, la clave recién derivada puede ser almacenada de manera segura de acuerdo con la política de uso de clave recibida.

El cliente de bóveda puede afirmarse a sí mismo al enclave de bóveda de claves en el cuadro 2310. Un proceso de afirmación puede incluir recibir, en el enclave de bóveda de claves, un reporte o cita de afirmación del cliente de bóveda. La política de uso de clave recibida puede restringir algunos o todos los usos de la nueva clave a solicitudes de los solicitantes que son autenticados por medio de afirmación de software. En los cuadros 2312-2316, se lleva a cabo una operación de descifrado, tal como para la unidad básica Decrypt anterior, utilizando la clave derivada en el cuadro 2306. En otras modalidades, se pueden llevar a cabo otras operaciones con una clave de bóveda bloqueada, tal como cifrado, firma, verificación de una firma y derivación de otra nueva clave a partir de la clave derivada en el cuadro 2306 (derivando una clave de segunda generación a partir de la clave de raíz). En el cuadro 2312, se recibe un buffer de datos cifrados del cliente de bóveda. Los datos cifrados recibidos son descifrados con la clave derivada en el cuadro 2306, y los datos descifrados resultantes (en un buffer de datos descifrados) son enviados al cliente de bóveda a través del canal de comunicaciones seguro en el cuadro 2316.

En una modalidad, un método para sellar datos de enclave comprende:

- enviar, a un enclave de origen alojado en una primera plataforma de enclave nativa, un primer reporte de afirmación de un enclave de sellado alojado en una segunda plataforma de enclave nativa;

- recibir una lista permitida y datos de enclave asociados del enclave de origen, en donde la lista

permitida incluye una lista de una o más identidades de enclave permitidas para desellar los datos de enclave;

almacenar de manera segura los datos de enclave y la lista permitida; y

restringir el acceso de los datos de enclave enclaves con identidades autenticadas se permite de acuerdo con la lista permitida.

En una modalidad, la primera plataforma de enclave nativa está en una primera computadora y la segunda plataforma de enclave nativa está alojada en una segunda computadora.

En una modalidad, la lista permitida es una lista de tipos de identidad abstracta; y que comprende además:

recibir un reporte de afirmación de origen del enclave de origen; y

derivar un valor de identidad permitida a partir del reporte de afirmación de origen y un tipo de identidad abstracta en la lista permitida.

En una modalidad, se cifran los datos de enclave recibidos, y que comprende además:

ejecutar un proceso de afirmación entre el enclave de origen y el enclave de sellado; y

descifrar los datos de enclave cifrados con una clave generada durante el proceso de afirmación.

En una modalidad, almacenar de manera segura los datos de enclave incluye:

cifrar los datos de enclave con una clave asociada con el enclave de sellado para crear datos de enclave sellados;

almacenar los datos de enclave sellados en almacenamiento persistente; y

después de determinar que se permite el acceso a una identidad autenticada, descifrar los datos de enclave sellados.

En una modalidad, un método para sellar datos de enclave comprende:

recibir, en un enclave de origen alojado en una primera plataforma de enclave nativa, un primer reporte de afirmación de un enclave de sellado alojado en una segunda plataforma de enclave nativa;

verificar la confianza en el enclave de sellado; y

enviar una lista permitida y datos de enclave asociados hacia el enclave de sellado, en donde la lista permitida incluye una lista de una o más identidades de enclave permitidas para desellar los datos de enclave.

En una modalidad, el método además comprende:

un proceso de afirmación entre el enclave de sellado y el enclave de origen;

generar datos de enclave cifrados al cifrar los datos de enclave con una clave generada durante el proceso de afirmación; y

en donde los datos de enclave enviados al enclave de sellado son los datos de enclave cifrados.

En una modalidad, la primera plataforma de enclave nativa está en una primera computadora y la segunda plataforma de enclave nativa está alojada en una segunda computadora.

En una modalidad, la lista permitida es una lista de tipos de identidad abstracta; y que comprende además:

enviar un reporte de afirmación de origen del enclave de origen al enclave de sellado, en donde el

reporte de afirmación de origen incluye al menos un valor de identidad del enclave de origen que corresponde a un tipo de identidad abstracta en la lista permitida.

En una modalidad, el método además comprende:

completar parcialmente una operación de procesamiento seguro en el enclave de origen; y
en donde los datos de enclave incluyen datos de estado del enclave de origen suficientes para permitir que otra instanciación de enclave continúe la operación de procesamiento seguro completada parcialmente.

En una modalidad, un sistema comprende al menos un procesador y memoria que almacena en la misma instrucciones que, cuando son ejecutadas por el sistema, provocan al menos:

enviar, a un enclave de origen alojado en una primera plataforma de enclave nativa, un primer reporte de afirmación de un enclave de sellado alojado en una segunda plataforma de enclave nativa;

recibir una lista permitida y datos de enclave asociados del enclave de origen, en donde la lista permitida incluye una lista de una o más identidades de enclave permitidas para desellar los datos de enclave;

almacenar de manera segura los datos de enclave y la lista permitida; y

restringir el acceso de los datos de enclave enclaves con identidades autenticadas se permite de acuerdo con la lista permitida.

En una modalidad, la primera plataforma de enclave nativa está en una primera computadora y la segunda plataforma de enclave nativa está alojada en una segunda computadora.

En una modalidad, la lista permitida es una lista de tipos de identidad abstracta, y las instrucciones provocan además al menos:

recibir un reporte de afirmación de origen del enclave de origen; y

derivar un valor de identidad permitida a partir del reporte de afirmación de origen y un tipo de identidad abstracta en la lista permitida.

En una modalidad, se cifran los datos de enclave recibidos, y las instrucciones provocan además al menos:

ejecutar un proceso de afirmación entre el enclave de origen y el enclave de sellado; y

descifrar los datos de enclave cifrados con una clave generada durante el proceso de afirmación.

En una modalidad, almacenar de manera segura los datos de enclave incluye:

cifrar los datos de enclave con una clave asociada con el enclave de sellado para crear datos de enclave sellados;

almacenar los datos de enclave sellados en almacenamiento persistente; y

después de determinar que se permite el acceso a una identidad autenticada, descifrar los datos de enclave sellados.

En una modalidad, un sistema comprende al menos un procesador y memoria que almacena en la misma instrucciones que, cuando son ejecutadas por el sistema, provocan al menos:

recibir, en un enclave de origen alojado en una primera plataforma de enclave nativa, un primer reporte de afirmación de un enclave de sellado alojado en una segunda plataforma de enclave nativa;

verificar la confianza en el enclave de sellado; y

enviar una lista permitida y datos de enclave asociados hacia el enclave de sellado, en donde la lista permitida incluye una lista de una o más identidades de enclave permitidas para desellar los datos de enclave.

En una modalidad, las instrucciones provocan además al menos:

un proceso de afirmación entre el enclave de sellado y el enclave de origen;

generar datos de enclave cifrados al cifrar los datos de enclave con una clave generada durante el proceso de afirmación; y

en donde los datos de enclave enviados al enclave de sellado son los datos de enclave cifrados.

En una modalidad, la primera plataforma de enclave nativa está en una primera computadora y la segunda plataforma de enclave nativa está alojada en una segunda computadora.

En una modalidad, la lista permitida es una lista de tipos de identidad abstracta; y en donde, las instrucciones provocan además al menos:

enviar un reporte de afirmación de origen del enclave de origen al enclave de sellado, en donde el reporte de afirmación de origen incluye al menos un valor de identidad del enclave de origen que corresponde a un tipo de identidad abstracta en la lista permitida.

En una modalidad, las instrucciones provocan además al menos:

completar parcialmente una operación de procesamiento seguro en el enclave de origen; y

en donde los datos de enclave incluyen datos de estado del enclave de origen suficientes para permitir que otra instanciación de enclave continúe la operación de procesamiento seguro completada parcialmente.

Cada uno de los procesos, métodos, y algoritmos descritos en las secciones anteriores se pueden incorporar en, y automatizar completa o parcialmente por, módulos de código ejecutados por una o más computadoras o procesadores de computadora. Los módulos de código se pueden almacenar en cualquier tipo de medio legible por computadora no transitorio o dispositivo de almacenamiento por computadora, tal como discos duros, memoria de estado sólido, disco óptico y/o similares. Los procesos y algoritmos se pueden implementar parcial o totalmente en circuitos de aplicación específica. Los resultados de los procesos y pasos de proceso divulgados se pueden almacenar, persistentemente o de otra manera, en cualquier tipo de almacenamiento por computadora no transitorio tal como almacenamiento volátil o no volátil. Las diferentes características y procesos descritos anteriormente se pueden utilizar independientemente unos de otros, o se pueden combinar de diferentes formas. Se pretende que todas las combinaciones y subcombinaciones posibles caigan dentro del alcance de esta divulgación. Adicionalmente, ciertos métodos o bloques de proceso se pueden omitir en algunas implementaciones. Los métodos y procesos descritos en este documento tampoco se limitan a ninguna secuencia particular, y los bloques o estados relacionados con los mismos se pueden llevar a cabo en otras secuencias que sean apropiadas. Por ejemplo, los bloques o estados descritos se pueden llevar a cabo en un orden diferente al que se divulga específicamente, o se pueden combinar múltiples bloques o estados en un solo bloque o estado. Los bloques o estados ejemplares se pueden llevar a cabo en serie,

en paralelo o de alguna otra manera. Los bloques o estados se pueden agregar a o remover de las modalidades ejemplares divulgadas. Los sistemas y componentes ejemplares descritos en este documento se pueden configurar de manera diferente a la descrita. Por ejemplo, elementos se pueden agregar a, remover de, o reacomodar en comparación con las modalidades ejemplares divulgadas.

También se apreciará que los diferentes artículos se ilustran como siendo almacenados en memoria o en almacenamiento mientras se utilizan, y que estos artículos o porciones de los mismos se pueden transferir entre memoria y otros dispositivos de almacenamiento para fines de gestión de memoria e integridad de datos. Alternativamente, en otras modalidades algunos o todos de los módulos de software y/o sistemas pueden ejecutarse en memoria en otro dispositivo y comunicarse con los sistemas informáticos que se ilustran por medio de comunicación inter-computadora. Además, en algunas modalidades, algunos o todos de los sistemas y/o módulos se pueden implementar o proporcionar de otras formas, tal como al menos parcialmente en firmware y/o hardware, incluyendo, pero no limitado a, uno o más circuitos integrados de aplicación específica (ASICs, Application-Specific Integrated Circuits, por sus siglas en inglés), circuitos integrados estándar, controladores (p.ej., que ejecutan instrucciones apropiadas, e incluyendo microcontroladores y/o controladores incrustados), matrices de puertas programables en campo (FPGAs, Field Programmable Gate Arrays, por sus siglas en inglés), dispositivos lógicos programables complejos (CPLDs, Complex Programmable Logic Devices, por sus siglas en inglés), etc. Algunos o todos de los módulos, sistemas y estructuras de datos también se pueden almacenar (p.ej., como instrucciones de software o datos estructurados) en un medio legible por computadora, tal como un disco duro, una memoria, una red o un artículo de medio portátil que se leerá por medio de una unidad apropiada o por medio de una conexión apropiada. Para los fines de esta especificación y las reivindicaciones, la frase “medio de almacenamiento legible por computadora” y variaciones del mismo, no incluye ondas, señales, y/u otros medios de comunicación transitorios y/o intangibles. Los sistemas, módulos y estructuras de datos también se pueden transmitir como señales de datos generadas (p.ej., como parte de una onda portadora u otra señal propagada análoga o digital) en una variedad de medios de transmisión legibles por computadora, incluyendo medios inalámbricos y alámbricos, y pueden tomar una variedad de formas (p.ej., como parte de una señal análoga simple o multiplexada, o como múltiples paquetes o tramas digitales discretas). Tales productos de programa de computadora también pueden tomar otras formas en otras modalidades. En consecuencia, la presente divulgación se puede practicar con otras configuraciones de sistema de computadora.

El lenguaje condicional utilizado en este documento, tal como, entre otros, “poder”, “podría”, “puede”, “pueden”, “p.ej.” y similares, a menos que se indique específicamente lo contrario, o se entienda lo contrario dentro del contexto en el cual se utiliza, se entiende generalmente para transmitir que ciertas modalidades incluyen, mientras otras modalidades no incluyen, ciertas características, elementos, y/o pasos. Por lo tanto, tal lenguaje condicional no se pretende generalmente para implicar que características, elementos y/o pasos se requieren de ninguna forma para una o más modalidades o que una o más modalidades incluyen necesariamente lógica para decidir, con o sin entrada de autor o indicaciones, si estas características, elementos y/o pasos se incluyen o se van a llevar a cabo en alguna

modalidad particular. Los términos “que comprende”, “que incluye”, “que tiene” y similares son sinónimos y se utilizan de manera inclusiva, en una forma abierta, y no excluyen elementos, funciones, acciones, operaciones, etc. adicionales. También, el término “o” se utiliza en su sentido inclusivo (y no en su sentido exclusivo) de modo que cuando se utiliza, por ejemplo, para conectar una lista de elementos, el término “o” significa uno, algunos o todos de los elementos en la lista.

Mientras se han descrito ciertas modalidades ejemplares, estas modalidades se han presentado a manera de ejemplo solamente y no se pretenden para limitar el alcance de las invenciones divulgadas en este documento. Por lo tanto, nada en la descripción anterior se pretende para implicar que alguna función, característica, paso, módulo o bloque particular es necesario o indispensable. De hecho, los métodos y sistemas novedosos descritos en este documento se pueden incorporar en una variedad de otras formas; además, se pueden hacer diferentes omisiones, sustituciones y cambios en la forma de los métodos y sistemas descritos en este documento sin apartarse del espíritu de las invenciones divulgadas en este documento. Las reivindicaciones de acompañamiento y sus equivalentes se pretenden para cubrir tales formas o modificaciones como caerían dentro del alcance y espíritu de ciertas de las invenciones divulgadas en este documento.