

IDENTIDAD DE ENCLAVE MULTIPLATAFORMA

Campo de la invención

Esta divulgación se refiere a los sistemas informáticos seguros.

Antecedentes de la invención

Las regiones aisladas seguras o los entornos de ejecución confiables proporcionan un contenedor seguro, denominado en el presente documento como un enclave, para ejecutar el código confiable en una computadora que también puede tener un código menos confiable en una región fuera de la región aislada. La región aislada de un enclave incluye una porción de la memoria que está protegida durante la ejecución del código que se ubica fuera del enclave. La memoria aislada puede contener tanto código como datos para el enclave, y la protección de esta memoria puede incluir restricciones en la ejecución del código contenido en la memoria de enclave además de las restricciones en la lectura o escritura en la memoria de enclave. Los aspectos de la seguridad de enclave, como el aislamiento de la memoria y las restricciones de ejecución, pueden aplicarse, por ejemplo, por el *hardware* en el procesador informático. La certificación de *software* puede proporcionar confianza en la seguridad de aislamiento de un enclave particular y en el código del enclave que se carga dentro de la región de memoria aislada de ese enclave particular. La certificación también puede proporcionar una prueba de la integridad de la plataforma de *hardware* y *software* en la cual se ejecuta el enclave certificado.

Los sistemas de enclave, como el modo seguro virtual (VSM, por sus siglas en inglés) de Microsoft y extensiones de protección de *software* (SGX, por sus siglas en inglés) de Intel, proporcionan seguridad en parte al aislar un enclave de otro código que se ejecuta tanto en modo de usuario como en modo de núcleo. Las garantías de integridad y confidencialidad pueden proporcionar un enclave con un mayor nivel de confianza en la autenticidad del código que se ejecuta en un enclave y la confianza en la ejecución segura del código de enclave. La certificación de *software* de un enclave en particular puede proporcionar una garantía de integridad. La certificación de *software* puede incluir un valor *hash* firmado criptográficamente de los contenidos (instrucciones y datos) dentro de un enclave y puede combinarse con los datos sobre el entorno del enclave. Cuando se utiliza un enclave en combinación con un módulo de seguridad de *hardware* (HSM, por sus siglas en inglés), como el *hardware* que cumple con el estándar del módulo de plataforma confiable (TPM, por sus siglas en inglés) del grupo de computación confiable (TCG, por sus siglas en inglés), el enclave puede proporcionar un nivel adicional de seguridad y garantías de confidencialidad.

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

confiable proporciona la certificación de un enclave remoto, puede confiarse en un enclave incluso cuando el enclave resida en una computadora desconocida que es propiedad, y está en posesión, de una entidad no confiable. Este comúnmente es el caso, por ejemplo, cuando los recursos informáticos se alquilan en un recurso informático basado en la nube de internet.

Breve descripción de la invención

Se presentan métodos y sistemas para abstraer una plataforma de enclave. El método puede comprender recibir, mediante una plataforma de abstracción de enclave, una primera solicitud para utilizar un enclave desde un cliente de enclave. La primera solicitud puede ajustarse a un protocolo de abstracción de cliente. La primera solicitud puede convertirse en una segunda solicitud que se ajuste a un protocolo nativo de enclave asociado con una plataforma nativa de enclave. La segunda solicitud puede enviarse a la plataforma nativa de enclave. La primera solicitud puede ser, por ejemplo, una solicitud para crear una instancia de un enclave, una solicitud para verificar un informe de certificación de un enclave, una solicitud de invocar a un enclave o una solicitud de asignar la memoria que se comparte tanto con el enclave como con el cliente de enclave. La plataforma nativa puede ajustarse a la arquitectura de enclave de las extensiones de protección de *software* (SGX, por sus siglas en inglés) de Intel y la plataforma nativa puede ajustarse a la arquitectura de enclave del modo seguro virtual (VSM, por sus siglas en inglés) de Microsoft.

Breve descripción de los dibujos

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

La figura 2 muestra un proceso ejemplar para la transmisión de mensajes con una garantía de confidencialidad.

La figura 3 muestra un proceso ejemplar para la transmisión de mensajes con una garantía de integridad.

La figura 4 muestra un proceso ejemplar para la transmisión de mensajes con una garantía de actualización.

La figura 5 muestra un proceso ejemplar para la certificación de *software* de un enclave.

La figura 6 muestra un ejemplo de protocolo de intercambio de claves Diffie-Hellman (DKE, por sus siglas en inglés).

La figura 7 muestra una cadena confiable ejemplar para la certificación 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 muestra un entorno informático de uso general ejemplar.

La figura 12 muestra un diagrama de flujo ejemplar para un método de abstracción de una plataforma de enclave nativa.

La figura 13 muestra un diagrama de flujo ejemplar para un método de abstracción de una plataforma de enclave nativa.

La figura 14 muestra un diagrama de flujo ejemplar para un método para realizar una operación de enclave con la identidad abstracta de enclave.

La figura 15 muestra un diagrama de flujo ejemplar para un método 1500 para realizar 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 muestra un diagrama de flujo ejemplar para procesamiento paralelo con dos enclaves equivalentes.

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

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

La figura 20 es un diagrama de flujo ejemplar para la codificación y decodificación de datos distribuida.

La figura 21 es un diagrama de bloques de un enclave de depósito de claves ejemplares.

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

La figura 23 es un diagrama de flujo ejemplar para la operación de enclave de depósito de claves con una clave bloqueada en el depósito.

Descripción detallada de la invención

Se divulga un modelo de abstracción para los enclaves que simplifica el desarrollo de los clientes de enclave y del *software* que se ejecuta dentro de un enclave. Un modelo de abstracción puede ser una simplificación y unificación de las arquitecturas de plataformas de enclave nativas, 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 el *software* dentro de un enclave y una o más plataformas de enclave nativas, y entre el *software* dentro de un enclave y un cliente de enclave. Tal plataforma de abstracción puede proporcionar la ventaja de permitir que una única versión del *software* de enclave y del *software* de cliente de enclave se ejecute hasta arriba de múltiples plataformas de enclave nativas, como SGX y VSM. Además de simplificar la tarea de escribir el *software* para enclaves y clientes de enclave, permite a los usuarios finales de los enclaves ejecutar el *software* de enclave y de cliente de enclave en una computadora que es compatible con cualquier arquitectura de enclave nativa compatible sin tener que encontrar versiones del *software* de enclave y de cliente de enclave que se adapte a la plataforma de enclave nativa de una computadora específica.

Un modelo de abstracción de enclave puede incluir, por ejemplo, acciones básicas para:

administrar el ciclo de vida de un enclave, certificación local y remota de un enclave, codificación de datos en un enclave, control de transferencia del programa dentro y fuera de un enclave y otras características de seguridad 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 solo valor binario o un solo valor *hash* de los contenidos de un enclave. Las interfaces de componentes de *software*, como una interfaz de programación de aplicaciones (API, por sus siglas en inglés) o una interfaz binaria de aplicaciones (ABI, por sus siglas en inglés) se presentan para el desarrollo de programas de enclaves y de cliente de enclaves utilizando las acciones básicas del modelo de abstracción.

Una identidad abstracta puede incluir una identidad anidada o jerarquía de identidad que puede utilizarse para identificar de forma segura los grupos de instancias de enclave. Una instancia de enclave en el presente 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, podría considerarse una instancia diferente. Estas diferentes instancias también pueden tener la misma identidad en un nivel superior en una jerarquía de identidad. Una identidad de enclave abstraída permite identificar grupos relacionados de elementos binarios de enclave relacionados. Por ejemplo, diferentes versiones del mismo enclave, como las versiones de elementos binarios de enclave anteriores y posteriores a la corrección de un error, pueden recibir el mismo nombre, independientemente de la versión. En una capa superior de abstracción, a todos los enclaves de una familia de enclaves se les puede asignar un solo nombre de familia o identidad. En este caso, todos los enclaves que realizan funciones relacionadas pero diferentes pueden identificarse juntos. A continuación, se describen otras capas de identidad o agrupaciones de identidades.

Puede utilizarse cualquier capa de una identidad abstracta para diversas operaciones criptográficas, como codificación de datos, certificación de un enclave o garantía de actualización de datos (a través de contadores monotónicos). Por ejemplo, al codificar los datos producidos por una instancia de enclave en una identidad de nivel superior, esos datos pueden entonces consumirse de manera segura por una instancia de enclave diferente con la misma identidad de enclave de nivel superior. Al codificar los datos para, por ejemplo, una familia de enclaves, cualquier instancia de enclave que sea miembro de esa familia, y solo los miembros de esa familia, podrá decodificar los datos. La certificación de una identidad familiar a partir de una instancia de enclave proporciona la seguridad 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 actualización relacionadas con todas las instancias de enclave que son miembros de la identidad de abstracción.

El modelo de abstracción descrito incluye interfaces de componentes de *software*, como una interfaz de programación de aplicaciones (API, por sus siglas en inglés) o una interfaz binaria de aplicaciones (ABI, por sus siglas en inglés) que puede simplificar el desarrollo de *software* de enclaves y de servidores de enclaves. Una API es un conjunto de definiciones de subrutinas 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 u operación del componente de *software* independientemente de cualquier implementación particular del componente de *software*. Una API puede definirse en un lenguaje informático de alto nivel, como C, C++, C# y similares. Una definición formal de una API puede facilitar la interacción entre los componentes de *software*, por ejemplo, dos componentes de *software* escritos en diferentes momentos o por diferentes autores. Una API puede formalizarse en parte con un lenguaje de descripción de interfaz (IDL, por sus siglas en inglés), como el lenguaje de definición de interfaz de Microsoft (MIDL, por sus siglas en inglés) o el IDL del grupo de gestión de objetos (OMG, 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 puede ser puntos de entrada (o direcciones de instrucciones) del código objeto que resulta a partir de la compilación de una implementación de código fuente de una API junto con los protocolos para utilizar esos puntos de entrada, como los protocolos que especifican los registros de máquina que contienen los argumentos cuando se invocan los puntos de entrada.

Además de permitir la interacción con diferentes niveles de identidad de enclave tal como se describió anteriormente, una API de enclave puede abstraer las diferencias entre las arquitecturas de la plataforma de enclave, por ejemplo, entre las arquitecturas para la ejecución aislada segura proporcionada por las extensiones de protección de *software* (SGX, por sus siglas en inglés) de Intel, modo seguro virtual (VSM, por sus siglas en inglés) de Microsoft y TrustZone de ARM, virtualización segura encriptada (SEV, por sus siglas en inglés) de AMD y arquitecturas basadas en matrices de puertas programable *in situ* (FPGA, por sus siglas en inglés). Las API incluyen interfaces para una plataforma de enclave que abstrae algunos detalles de las arquitecturas de enclave abstraídas. Estas interfaces de plataforma de enclave incluyen una API de computadora central de enclaves, una API de plataforma de enclave y una API de certificación remota. La API de computadora central de enclaves puede utilizarse por un proceso de computadora central no confiable para gestionar el ciclo de vida de un enclave, así como para proporcionar comunicación hacia y desde un enclave. La plataforma de enclave confiable puede proporcionar la API de plataforma de enclave al enclave, y puede incluir acciones básicas de seguridad para la certificación, codificación y comunicación con el código no confiable que se ejecuta en la computadora que aloja el enclave, así como compatibilidad de tiempo de ejecución central como gestión de memoria y programación de subprocesos. La API de certificación remota puede utilizarse para realizar la certificación remota, donde un enclave y su cliente no están alojados en la misma computadora. Por ejemplo, la API de certificación remota puede utilizarse por un cliente local para verificar que los datos se originaron (o se enviaron) desde un enclave con cierta identidad ejecutándose de forma aislada por una plataforma de enclave en una computadora remota. Más generalmente, la API de certificación remota puede utilizarse para establecer los canales de comunicación seguros entre un cliente local y el enclave remoto.

Los enclaves generalmente proporcionan soluciones a problemas que son específicos y surgen a partir del ámbito de la tecnología informática. Más específicamente, los enclaves proporcionan un mecanismo para separar el código confiable del código no confiable donde los segmentos de código

confiable y no confiable residen dentro del espacio de direcciones de un solo procesador informático. Por ejemplo, los enclaves proporcionan una solución de seguridad para el problema de código potencialmente no confiable (como el código que potencialmente contiene errores o virus) que se ejecuta en la misma computadora de uso general que el código que debe acceder a datos confidenciales o privados. Las modalidades de esta divulgación proporcionan soluciones mejoradas adicionales para los problemas de seguridad que surgen a partir del ámbito de la tecnología informática, que incluyen: simplificar el desarrollo de *software* al permitir la creación de un único enclave o cliente de enclave que se creará para múltiples plataformas nativas de enclaves; simplificar la gestión de computadoras corporativas al reducir la cantidad de componentes de *software* que deben adaptarse a características de *hardware* específicas de una computadora particular; y habilitar nuevas situaciones de computación segura con la codificación distribuida de datos, como la distribución del procesamiento seguro de enclaves a través de los enclaves alojados en múltiples computadoras.

La figura 1 muestra un diagrama de bloques de alto nivel de un sistema de enclave junto con algunas relaciones confiables. El sistema de enclave 100 incluye un enclave 176 (denominado alternativamente contenedor de enclaves o entorno de ejecución seguro), que incluye una región de memoria aislada y segura que contiene el código 180 y los datos 182. El código 180 puede ser público o privado, y los datos 182 pueden ser públicos o privados. Por ejemplo, los datos o el código privados pueden pertenecer (o ser privados a) un propietario de datos 142, mientras los datos o códigos públicos pudieron proporcionarse por otra entidad, como el proveedor de *software* 148. En una modalidad, el código que se ejecuta dentro de un contenedor de enclave 176 puede ser completamente público y no privado, mientras los datos que el código público de enclave utiliza como entrada o produce como salida pueden ser privados. En otra modalidad, lo contrario es posible, donde el código es privado mientras los datos son públicos. En otra modalidad más, tanto el código como los datos de entrada pueden ser públicos, mientras la salida de la ejecución del código con los datos de entrada puede ser privada. Otras combinaciones públicas y privadas de código, datos de entrada y datos de salida son posibles.

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

El cliente de enclave 104 puede ser un proceso o programa fuera del contenedor de enclave para el cual el enclave 176 realiza sus computaciones con el código 180 y los datos 182. En una modalidad de enclave local, el cliente de enclave 104 también puede ejecutarse 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 remota diferente, conectada a la computadora del cliente de

enclave a través de una red. En el caso del enclave local, el proceso del cliente de enclave también puede ser el proceso de la computadora central de enclave del contenedor de enclave 176, ya que el proceso del cliente de enclave puede gestionar la creación del enclave local 176. En el caso del enclave remoto, el enclave 176 puede, por ejemplo, ejecutarse en una computadora en la nube de internet donde el propietario de infraestructura 152 es un proveedor de servicios de computación en la nube, y la computadora en la nube incluye un *hardware* confiable 172 producido por el fabricante 162.

El cliente de enclave 104 puede incluir un método de configuración 106 para configurar un computación solicitado por el enclave 176. El método de configuración 106 puede incluir causar la creación del contenedor seguro del enclave 176, causar la creación de instancias del enclave (por ejemplo, al enviar una solicitud al *software* no confiable 174 para solicitar la creación de instancias del enclave), que puede incluir copiar el código binario en el contenedor seguro y provocar o solicitar una computación en el enclave, por ejemplo, al invocar un método en el código copiado en el contenedor seguro. La computación solicitada puede incluir la ejecución del código 180, y los datos 182 pueden introducirse, o pueden ser el resultado de la computación solicitada. Los datos ingresados a la computación solicitada pueden cifrarse cuando están fuera del enclave, y los datos de entrada cifrados pueden descifrarse antes de usarlos dentro del enclave. Una vez que el enclave 176 ha completado la tarea solicitada, los datos que representan el resultado de la tarea se cifran y se devuelven 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 actualización de los resultados recibidos. Un único proveedor de *software* 148 puede proporcionar tanto el código 180 para ejecutarse dentro del enclave 176, como al menos una porción del método de verificación 108 que se ejecuta como parte del cliente 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 corrección de los resultados producidos por el enclave 176, puede basarse en relaciones confiable. 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. El propietario de los datos puede confiar aún más en el proveedor de *software* 148 del propio enclave. Y el propietario de los datos puede confiar en el fabricante del *hardware* confiable 172. El *hardware* confiable 172 puede adoptar muchas formas dependiendo de la arquitectura de enclave utilizada, y puede incluir un módulo de seguridad de *hardware* (HSM, por sus siglas en inglés), donde el HSM cumple, por ejemplo, con el estándar del módulo de plataforma confiable (TPM, por sus siglas en inglés). El *hardware* confiable 172 puede incluir, por ejemplo, un TPM y, de lo contrario, podría comprender únicamente *hardware*. Por ejemplo, una implementación de *hardware* 172 confiable con la arquitectura de enclave VSM de Microsoft puede incluir un procesador de uso habitual con indicaciones para las instrucciones de virtualización de sistema operativo y un TPM. La arquitectura de enclave VSM de Microsoft puede incluir un hipervisor para gestionar particiones invitadas (procesadores virtuales), y el hipervisor puede exponer las interfaces de llamada al hipervisor a particiones invitadas para permitir que las particiones invitadas interactúen con el hipervisor. Un contenedor de enclave en la arquitectura VSM de Microsoft puede implementarse como un tipo especial de partición invitada. Un ejemplo de *hardware*

172 confiable con la arquitectura de enclave SGX de Intel puede incluir un procesador con instrucciones especiales específicas para el enclave y funcionalidad de seguridad.

Un enclave, como el enclave 176, puede proporcionar un entorno de ejecución aislado que puede proteger el código o los datos, como el código 180 y los datos 182, al proporcionar una región de memoria con restricciones de lectura, escritura o ejecución desde esa región protegida. Esta región de memoria protegida es un contenedor seguro para datos y códigos confidenciales. Las restricciones en la ejecución desde la región de memoria protegida de un enclave pueden incluir restricciones en las transferencias de ejecución, como las instrucciones de invocación o salto, entre el código fuera del enclave y el código dentro del enclave, y viceversa. Pueden imponerse diferentes restricciones entre invocar al enclave desde fuera del enclave e invocar fuera del enclave desde dentro del enclave. El cumplimiento de estas transferencias de ejecución entre el interior y el exterior de un enclave puede proporcionarse por *hardware*, por ejemplo, con tecnología de *hardware* de virtualización de uso habitual o con *hardware* especializado, como la plataforma SGX de Intel.

La figura 2 muestra un proceso ejemplar 200 para la transferencia de mensajes con una garantía de confidencialidad. Una garantía de confidencialidad proporciona cierto nivel de seguridad de que la comunicación entre dos entidades, como Anne 210 y Ben 230 en este ejemplo, permanece oculta para entidades externas cuando los mensajes se transfieren a través de un medio de comunicación público o desprotegido como la red 218. En este ejemplo, Anne 212 desea enviar un mensaje confidencial a Ben 230. Un bloque de mensajes 212 que contiene datos confidenciales se cifra mediante la clave pública 216 mediante una operación de cifrado 214. La operación de cifrado 214 puede ser, por ejemplo, el modo contador/Galois del estándar de cifrado avanzado (AES-CGM, por sus siglas en inglés), pero también puede ser cualquier operación de cifrado conocida para los técnicos en la materia de la criptografía digital. El bloque cifrado 220 puede transferirse a través de una red 218 a Ben, donde el bloque cifrado 234 se descifra con la clave privada 236 para producir el bloque de mensaje no cifrado 232 para Ben. Con una selección cuidadosa de claves y algoritmos de cifrado, tal como se conoce en la técnica del cifrado de datos informáticos, el mensaje puede permanecer confidencial incluso cuando se transfiere a través de una red pública.

La figura 3 muestra un proceso 300 ejemplar para la transferencia de mensajes con una garantía de integridad. Una garantía de integridad proporciona cierto nivel de seguridad de que la comunicación entre dos entidades, como Anne 310 y Ben 350 en este ejemplo, no se manipule ni de otro modo se altere cuando los mensajes se transmiten a través de un medio de comunicación público o desprotegido, como la red 340. En el ejemplo de la figura 3, Anne 310 envía un mensaje 314 a Ben 350 de tal manera que exista confianza de que el mensaje 314 no se manipule o de otro modo se corrompa cuando Ben 350 lo reciba. Para proporcionar esta garantía de integridad, en el mensaje 314 se opera un proceso de función *hash* seguro para generar un valor *hash* 318. El valor *hash* 318 entonces se firma mediante el proceso de firma 320 utilizando la clave privada de Anne para producir una firma 342. La firma 342 puede enviarse 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 entonces recibe un mensaje 354 para el cual Ben

desea verificar la integridad, de modo que Ben tenga confianza de que el mensaje 354 es el mismo que el mensaje 314 que Anne envió después de pasar a través de una red no confiable 340. Para verificar la integridad, el mensaje 354 recibido se procesa mediante la función *hash* segura 356 que es idéntico a la función *hash* segura 316 para generar el valor *hash* 358. La firma 342 recibida se verifica mediante el proceso de verificación de firma 360 utilizando la clave pública de Anne. El valor *hash* 318 que se extrae a partir de la firma 342 se compara con el valor *hash* 358 para verificar que las firmas sean las mismas. Si son iguales, se acepta que el mensaje tiene integridad. Alternativamente, si el mensaje 314 se alteró de alguna manera antes de recibirse como el mensaje 354, entonces la firma no será correcta y el mensaje será rechazado 388.

En algunas modalidades, la función *hash* segura 316 y la función *hash* segura 356 pueden ser una función *hash* criptográfica. Una función *hash* criptográfica es una función unidireccional que asigna los datos de tamaño arbitrario a una cadena de bits de un tamaño fijo (generalmente mucho más pequeño). La salida de una función *hash* puede denominarse un valor *hash* o simplemente un *hash*. Una buena función *hash* será unidireccional, ya que será difícil determinar la entrada de tamaño arbitrario proporcionando únicamente la salida de valor *hash*. Con una buena función *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 puede combinarse con la identificación de un valor *hash* de mensaje de la figura 3. El sistema de combinación puede requerir dos conjuntos de pares de claves públicas/privadas, uno para el remitente y otro para el receptor. Un sistema de combinación puede utilizar una clave privada en el receptor para descifrar un mensaje (la clave privada de Ben como en la figura 2), mientras utiliza otra clave privada para firmar el valor *hash* del mensaje (la clave privada de Anne como en la figura 3).

La figura 4 muestra un proceso 400 ejemplar para la transferencia de mensajes con una garantía de actualización. Una garantía de actualización proporciona cierto nivel de seguridad de que cuando se envían múltiples mensajes de una entidad a otra, como desde Anne 410 a Ben 450 en este ejemplo, el mensaje recibido en el receptor es el mensaje más reciente. Una garantía de actualización puede basarse en una garantía de integridad y puede evitar ataques de repetición. Con una garantía de integridad, es difícil para una entidad externa con una intención maliciosa crear su propio mensaje y enviarlo a Ben de modo que Ben entienda que Anne creó el mensaje. Sin embargo, una entidad externa puede tomar un mensaje que Ann realmente creó, tal vez observado en un momento dado cuando se envió a través de una red pública, y la entidad externa puede reenviarlo a Ben en otro momento (es decir, repetir el mensaje). Ben determinará que el mensaje lo creó Anne (debido a que fue así), pero Ben no sabrá que Anne no es la persona que lo envió en ese momento. Esto puede denominarse un ataque de repetición a Ben o Anne por parte de una entidad externa. La figura 4 es un ejemplo de solución para prevenir ataques de repetición al proporcionar una garantía de actualización utilizando códigos únicos descartables y marcas de tiempo. Un código único descartable es un número aleatorio de un solo uso, junto con un sistema para garantizar que el mismo número nunca se utilice como código único

descartable más de una vez. En algunos sistemas, un código único descartable puede ser simplemente un número monótonamente creciente, de modo que cada número utilizado sea mayor que todos los números anteriores.

En la figura 4, el mensaje 414 de Anne puede enviarse a través de una red pública 430 como el mensaje 436 junto con el código único descartable 434 y la marca de tiempo 432. El código único descartable 434 se genera mediante un generador de números pseudoaleatorios (CSPRNG, por sus siglas en inglés) 426 criptográficamente seguro, y la marca de tiempo 432 se produce mediante el reloj sincronizado 424. Existen muchos sistemas CSPRNG que conocen los técnicos en la materia de la criptografía digital. El reloj sincronizado 424 en el lado de la red 430 de Anne está sincronizado con el reloj sincronizado 480 en el lado de la red de Ben. Por el lado de Ben, cuando se recibe un mensaje 454, el código único descartable 434 que lo acompaña se almacena en una memoria caché de los últimos códigos únicos descartables 470. La actualización de ese mensaje 450 recibido puede verificar con dos pruebas. Primero, el código único descartable 434 se prueba en el bloque 460 al comparar el código único descartable 434 con la memoria caché de los últimos códigos únicos descartables 470 para determinar si el código único descartable 434 recibido actualmente se visualizó anteriormente. Si el código único descartable 434 recibido se visualizó anteriormente, el mensaje 454 se rechaza como un mensaje repetido en el bloque 468. Si el código único descartable 434 recibido no se visualizó anteriormente, se determina que el mensaje es correcto en el bloque 464 para esta primera prueba. En segundo lugar, la marca de tiempo recibida 432 se compara con el reloj local sincronizado 490. Si la marca de tiempo es reciente, se determina que el mensaje 454 es aceptable en el bloque 494; de lo contrario, el mensaje 454 se rechaza como caducado en el bloque 498. La cantidad de retraso tolerada cuando se determina si una marca de tiempo reciente es reciente en el bloque 490 puede depender de tendencia de reloj esperada entre el reloj sincronizado 424 y el reloj sincronizado 480 y los retrasos en el procesamiento y transmisión de los mensajes a través de la red 430.

Un sistema de comunicación puede combinar una garantía de actualización con una o ambas garantías de confidencialidad, como en la figura 2, y una garantía de integridad, como en la figura 3. En un sistema que combina las tres garantías, el mensaje 436 enviado a través de la red 430 será una versión cifrada del mensaje original 414 de Anne, y se incluirá una firma que comprenda un valor *hash* firmado del mensaje 414 junto con la marca de tiempo 432, el código único descartable 434, y el mensaje 436.

La figura 5 muestra un proceso 500 ejemplar para la certificación de *software* de un enclave. La certificación de *software*, cuando se combina con un protocolo de gestión de claves como el de la figura 6, puede asegurar a un verificador que ha establecido una conexión oculta compartida con una pieza específica de *software* que se encuentra dentro de un contenedor aislado creado por un *hardware* confiable. En la modalidad de la figura 5, un cliente de enclave 510 (verificador de certificación) puede intentar utilizar los servicios de computación segura del enclave en la plataforma confiable 530. La plataforma confiable 530 está alojada en una computadora (no mostrada) de manera que la plataforma confiable 530 pueda comprender un subconjunto de la computadora de alojamiento. La plataforma

confiable 530 puede comprender elementos de *hardware* y elementos de *software* de la computadora central. El enclave comprende un contenedor de enclave seguro 536 y el código y los datos que contiene, como el código y datos públicos 538 y el código y datos privados 542.

Se combinan tres procesos en el proceso 500 ejemplar: un proceso de intercambio de claves que produce una clave compartida SK; un proceso de certificación para la certificación del cliente de enclave 510 en la plataforma confiable 530; y se realiza una computación segura. La clave compartida SK del primer proceso se utiliza para comunicar las entradas y salidas de la computación segura. En el intercambio de claves, el cliente de enclave calcula g^A , almacenada en el bloque 512, a partir de la clave privada A del cliente de enclave y una función generadora g, por ejemplo, tal como se describe posteriormente para el protocolo de intercambio de claves Diffie-Hellman (DKE, por sus siglas en inglés) de la figura 6. La g^A computada entonces se envía en el mensaje 520 a la plataforma confiable 530. El mensaje 520 puede enviarse de manera segura sin cifrado y antes de que se complete la certificación. El *software* dentro del contenedor de enclave seguro 536 puede utilizar una clave privada B del enclave para calcular g^B utilizando la misma función generadora g. Tanto B como g^B pueden almacenarse en el contenedor de enclave en el bloque 540.

Para certificar la identidad del enclave (para proporcionar seguridad sobre qué código se ejecuta dentro del contenedor de enclave seguro 536), se envía un mensaje de certificación 522 al cliente de enclave 510. Un mensaje de certificación puede ser un mensaje especial firmado por integridad como en la figura 3. El mensaje especial puede contener la información de identidad sobre el enclave. Cuando se combina con DKE como en la modalidad de la figura 5, el mensaje especial también puede incluir parámetros de intercambio claves. En la modalidad de la figura 5, el estado inicial 538 del contenedor seguro de enclave del código público y de los datos públicos se utiliza como una identidad de enclave, aunque son posibles otras identidades. En lugar de enviar todo el estado inicial 538 en un mensaje de certificación, en su lugar se envía un valor *hash* del estado inicial, $M = \text{Hash}(\text{estado inicial})$. El mensaje de certificación 522 incluye el contenido del mensaje (M y g^B), y una firma del contenido de mensaje ($\text{SignAK}(\text{Hash}(g^B), M)$). La firma del contenido de mensaje puede crearse, por ejemplo, mediante un *software* dentro del contenedor de enclave seguro 536 que solicita a la plataforma confiable 530 que aloja el enclave que certifique un valor *hash* de la g^B calculada y la identidad del enclave. La plataforma confiable 530 puede hacerlo proporcionando una firma utilizando la clave de certificación de plataforma (AK) 532 para producir $\text{SignAK}(\text{Hash}(g^B), M)$. En este ejemplo, la identidad del enclave es un valor *hash* M del estado inicial 538, pero son posibles otras declaraciones de identidad. Esta firma de certificación $\text{SignAK}(\text{Hash}(g^B), M)$ vincula el valor g^B a la identidad de enclave M y también vincula tanto g^B como M a la plataforma confiable 530. El cliente de enclave 510 puede verificar el mensaje de certificación al verificar la firma de certificación y la identidad del enclave. La firma puede verificarse como en la figura 3 utilizando una clave pública correspondiente a la clave de certificación AK. La verificación de firma puede proporcionar una garantía de integridad de la identidad del enclave en el mensaje de certificación. La identidad del enclave puede verificarse al comparar la información de identidad en el mensaje de certificación con un valor de identidad conocido de forma independiente. Por ejemplo, si la información de

identidad en el mensaje de certificación es un valor *hash* del estado inicial del enclave, el cliente de enclave 510 puede conocer el valor *hash* del estado inicial, o puede ser capaz de calcular tal valor *hash* desde un estado inicial conocido, y este valor puede compararse con el valor de identidad proporcionado en el mensaje de certificación.

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 veces B) que puede servir como la clave compartida SK. La clave compartida SK puede utilizarse para cifrar los mensajes de confidencialidad entre el cliente de enclave 510 y el enclave, por ejemplo, para enviar los datos de entrada y los datos de salida desde el código dentro del contenedor de enclave 536. Cabe señalar que la clave compartida se produce independientemente en cada lado del canal de comunicación en los bloques 540 y 514 sin que el cliente de enclave o el enclave conozcan la clave privada del otro. Por ejemplo, en la modalidad de la figura 5, el código secreto y los datos pueden proporcionarse de manera segura por el cliente de enclave 510 al cifrar el código y los datos secretos con la clave compartida SK previamente establecida, producir $\text{EncsK}(\text{código}/\text{datos secretos})$ antes de enviarlo en un mensaje 524 a la plataforma confiable 530. En otras modalidades, el código y datos secretos 542 ejecutados o utilizados por un contenedor de enclave seguro 536 pueden originarse desde otras ubicaciones. Puede realizarse una computación segura dentro del contenedor de enclave seguro 536 utilizando el código y/o datos secretos 542 para producir un resultado de computación 544. Los resultados de computación 516 pueden comunicarse de manera segura al cliente de enclave 510 al cifrar los resultados con la clave compartida SK ($\text{Encsk}(\text{resultados})$) 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 cierta identidad, y de que el enclave está protegido por la plataforma de enclave. No proporciona ninguna garantía al código dentro del contenedor de enclave sobre la entidad en el otro lado del canal de comunicación. En una modalidad alternativa (no representada), tal garantía sobre el cliente de enclave puede proporcionarse al ejecutar el cliente de enclave como un enclave en sí. En esta modalidad alternativa, el cliente de enclave puede solicitar a la plataforma de enclave confiable una firma en el valor *hash* de g^A , que entonces puede verificarse por el otro enclave.

La certificación puede realizarse de manera 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 modo que las comunicaciones entre el cliente de enclave 510 y la plataforma confiable 530 pueden ocurrir dentro de una sola computadora (por ejemplo, al transmitir las memorias tampón de datos entre diferentes procesos en la misma computadora), o puede ocurrir a través de una red de computadoras que conecta el cliente de enclave 510 a la plataforma confiable 530. La certificación local puede realizarse cuando el cliente de enclave 510 y la plataforma confiable 530 están alojados en la misma computadora local. El objeto, o resultado, de la certificación local se denomina informe de certificación, y es $\text{Sign}_{AK}(\text{Hash}(g^A), M)$ en el ejemplo de la figura 5. La certificación remota puede ocurrir cuando el cliente enclave 510 y la plataforma

confiable 530 están alojados en diferentes computadoras. El objeto, o resultado, de la certificación remota se denomina una cita de certificación, que en algunos casos puede ser muy similar o idéntica a un informe de certificación local. En otros casos, una cita de certificación puede incluir un objeto confiable adicional relacionado con la computadora o la plataforma nativa que proporciona la cita. Tal objeto adicional confiable puede incluir un certificado de estado de operación de la computadora central, como un registro TCG asociado con un TPM. La certificación de un enclave puede realizarse en cualquier capa de la identidad de un enclave. Un enclave puede tener una identidad subdividida o jerárquica, y la certificación a niveles más altos de identidad puede certificar que un enclave instanciado es un miembro de un grupo cada vez mayor de instancias de enclave potenciales a medida que aumenta el nivel de identidad. Los niveles superiores pueden corresponder a un superconjunto de las instancias de enclave potenciales de nivel inferior. Un ejemplo de jerarquía de identidades puede incluir, desde el nivel más específico más bajo hasta el nivel menos específico más alto que pueden ser: ExactHash, InstanceHash, ImageID, FamilyID y AuthorID.

La identidad de un enclave puede obtenerse a partir de los archivos binarios (elementos binarios de enclave) cargados en el contenedor seguro del enclave. Un elemento binario de enclave puede firmarse por su autor utilizando una clave privada del autor. Para una certificación de nivel ExactHash, el estado inicial 538 utilizado para calcular un informe de certificación (la entrada a una función *hash* para producir un informe de certificación) puede incluir el contenido completo de cada archivo binario cargado en el contenedor seguro 536, excepto los elementos binarios asociados con la plataforma confiable 530.

La certificación a nivel de identidad por InstanceHash puede incluir un subconjunto del estado inicial 538. El subconjunto puede especificarse en el momento en cual los archivos binarios (elementos binarios) de enclave que se cargan en el contenedor seguro 536 se firmaron originalmente por parte del 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 a partir de los cuales depende el primer archivo binario de enclave. Para cada identidad listada, puede incluirse un indicador en el primer archivo binario para indicar si cada archivo binario listado se mide o no mediante la función *hash* para producir un informe de certificación mediante InstanceHash.

La certificación a los niveles superiores de un ID de enclave puede no incluir la ejecución de todo el contenido de cualquier elemento binario de enclave a través de una función *hash*. En su lugar, solo puede ejecutarse una estructura de datos de los identificadores a través de la función *hash*. Por ejemplo, un archivo binario de enclave puede incluir una lista de identificadores de enclave de nivel superior, como identificadores únicos universales (UUID, 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) exclusivo de un grupo de archivos binarios de enclave que incluye ese archivo binario de enclave particular y que se creó por parte del mismo autor; y un ID de autor (AuthorID) exclusivo para un grupo de familias de archivos binarios de enclave donde todos se crean por parte del mismo autor. El autor de un elemento binario de enclave puede asignar ImageID, FamilyID y AuthorID en el momento en el cual el elemento binario se firma originalmente. La suplantación de identidades de enclave puede evitarse cuando el

cliente de enclave puede acceder a los elementos binarios de enclave y verificar la firma del autor en esos elementos binarios utilizando la clave pública del autor (o una clave pública asociada con el autor). Esto verifica la integridad de los elementos binarios de enclave, que incluyen cualquier identidad de nivel superior asignada por parte del autor, tal como se crearon por parte de ese autor de enclave.

La figura 6 muestra un ejemplo del protocolo de intercambio de claves Diffie-Hellman (DKE, por sus siglas en inglés) 600. El DKE es un proceso ejemplar para establecer una clave compartida K a través de un canal de comunicación que solo tiene una garantía de integridad; pueden utilizarse otros procesos para crear claves compartidas conocidas en la técnica de la criptografía digital. En el ejemplo de DKE de la figura 6, una clave secreta K se comparte entre Anne 610 y Ben 650 sin comunicarse nunca K explícitamente a través de un medio de comunicación público (no seguro) entre Anne 610 y Ben 650. Antes de que comience el proceso, 1) un gran número primo p y 2) un generador g en $\mathbb{Z}_p$ puede establecerse y conocerse tanto por parte tanto de Anne como de Ben. El proceso comienza tanto con Anne como con Ben al elegir un número aleatorio entre 1 y p . El número aleatorio de Anne es A , elegido en el bloque 612, y el número aleatorio de Ben es B , elegido en el bloque 652.

Anne utiliza su número aleatorio para computar el modificador de $g^A p$ en el bloque 614, y transmite esa cantidad en el bloque 616 que recibe Ben en el bloque 656. Ben utiliza su número aleatorio para computar el modificador de $g^B p$ en el bloque 654, que se transmite en el bloque 656 a Anne y se recibe en el bloque 618. Anne puede producir la clave compartida K como $(\text{modificador de } g^B p)^A = \text{modificador de } g^{AB} p$ en el bloque 620 y Ben puede producir la clave compartida K como $(\text{modificador de } g^A p)^B = \text{modificador de } g^{AB} p$ en el bloque 660. Con el objetivo de evitar ataques de intermediarios, la comunicación entre Anna y Ben sobre una red no segura de los bloques 616 y 658 puede incluir una garantía de integridad de mensaje, por ejemplo creada al usar un proceso tal como el de la figura 3.

La figura 7 muestra un ejemplo de cadena de confianza 700 para la certificación de *software*. La cadena de confianza en la certificación de *software* puede originarse a partir de una clave de firma que es propiedad del fabricante de la plataforma confiable, como la plataforma confiable 530 de la figura 5. La plataforma confiable puede incluir componentes de *hardware* como un procesador seguro o un módulo de seguridad de *hardware* (HSM, por sus siglas en inglés), y, por lo tanto, el fabricante puede ser un proveedor de *hardware* informático y también puede proporcionar *software* para la plataforma confiable. El verificador 702 puede confiar en el fabricante, y el verificador puede conocer la clave pública PubRK 732 de la clave original de fabricante 736. El cliente de enclave 510 de la figura 5 es un ejemplo de un verificador 702 que puede intentar tener garantías sobre un contenedor seguro 708. El fabricante puede actuar como una autoridad de certificación y proporcionar a cada instancia de la plataforma confiable que produce, por ejemplo, cada procesador seguro, una clave de certificación única 722, que se utiliza para producir firmas de certificación. El fabricante también emite un certificado de aprobación 728 para cada clave de certificación 722. La clave original de 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, tal como se muestra en la figura 3.

El certificado de aprobación 728 incluye la clave pública PubAK 724 de la clave de certificación

722. El certificado de aprobación 728 puede indicar que la clave de certificación 722 se utilizará para la certificación de *software*, y puede comunicarse al verificador 702. El verificador puede ser cualquier entidad que tenga la intención de verificar una certificación del contenedor seguro 708, por ejemplo, el verificador 702 puede ser el cliente enclave 510 de la figura 5 que tiene la intención de que se realice una computación segura dentro del contenedor seguro 708. El verificador 702 puede inspeccionar el certificado de aprobación utilizando la PubRK 732 para verificar la integridad y el origen del certificado de aprobación. El verificador también puede extraer la PubAK 724 a partir del certificado de aprobación. El certificado de aprobación puede estar asociado a una política de certificación, lo cual puede requerir que la clave de certificación 722 se utilice solo para producir firmas de certificación y que la clave privada PrivAK 726 de la clave de certificación 722 se mantenga exclusivamente en un almacenamiento que sea independiente de la memoria de computadora generalmente accesible de la plataforma confiable, como en el almacenamiento de *hardware* antimanipulación 770. El *hardware* antimanipulación puede ser, por ejemplo, *hardware* conforme a un estándar de módulo de plataforma segura (TPM, por sus siglas en inglés).

Puede crearse una instancia de un contenedor seguro 708 en la plataforma confiable 736. La creación de instancias del contenedor seguro 708 puede incluir la definición de un espacio de memoria aislado para el contenedor seguro que tiene acceso restringido por un procesamiento no seguro. El procesamiento no seguro puede incluir, por ejemplo, acceso desde fuera de la plataforma confiable, pero en la computadora que aloja la plataforma confiable, o acceso desde otros contenedores seguros dentro de la plataforma confiable. La creación de instancias del contenedor seguro 708 también puede incluir cargar el código y datos públicos en el contenedor seguro, por ejemplo, el estado inicial 535 de la figura 5.

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

La firma de certificación 710 puede crearse después de que se creó una instancia del contenedor seguro 708 y de que se completó el intercambio de claves. El contenedor seguro instanciado 708 puede medirse al ejecutar una función criptográfica *hash* en todo o parte del contenedor seguro. Esto puede incluir ejecutar la función *hash* en el contenido de la memoria aislada y los archivos binarios que se cargan en la memoria aislada, cualquier otra memoria asociada con la plataforma confiable que se utiliza o afecta durante la creación de instancias del contenedor seguro, o cualquier subconjunto o parte de estos. La salida de la ejecución de esta función *hash* es la medida 712, que forma parte de la firma de certificación 710. También se puede incluir un valor criptográfico *hash* de los mensajes de intercambio de claves 704 y 706 con la firma de certificación 710, representado como datos 714. La medición 712 y los datos 714 pueden firmarse utilizando la clave privada de certificación PrivAK 726. La firma de certificación

puede entonces enviarse al verificador 702 junto con la medición 712 y los datos 714. El verificador puede verificar la integridad de la firma de certificación utilizando la PubAK 724 del certificado de aprobación, que, en el ejemplo de la figura 7, también permite la verificación de la integridad de la medición 712 y de los datos 714. El verificador 702 puede verificar la integridad del contenedor seguro 708 al comparar la medición 712 con un resultado esperado (el resultado esperado determinado, por ejemplo, al realizar localmente la misma función *hash* de la medida 712), y verificar que la firma de certificación se creó para esta instancia de ruta de comunicación del verificador 702 particular mediante la inspección de los datos 714 (por ejemplo, debido a que el valor *hash* de los datos 714 está vinculado al mensaje de intercambio de claves 2 706). Después de estas operaciones de verificación y la verificación del certificado de aprobación anterior, el verificador ahora tiene cierta seguridad de que puede establecer comunicaciones con confidencialidad e integridad con el contenedor seguro 708 utilizando una clave compartida establecida, donde *hardware* de plataforma confiable puede ser confiable de acuerdo con su fabricante, y donde se conoce el estado del *software* de la plataforma confiable utilizada para crear el contenedor seguro. El verificador 702 ahora está listo para solicitar un procesamiento seguro dentro del contenedor seguro 708 utilizando un código privado y/o datos privados.

Plataforma de abstracción de enclave y acciones 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 enclaves 800 incluye una computadora 810 con la plataforma de enclave nativa 812 que aloja el enclave 814 y un cliente de enclave 816. La plataforma nativa 812 puede ser un componente de *hardware* y/o *software* con base, por ejemplo, en SGX de Intel o VSM de Microsoft. El enclave 810 puede ser el enclave 176 de la figura 1. Puede utilizarse un protocolo nativo 844 para los enclaves para la comunicación entre el enclave 814, el cliente 816 y la plataforma nativa 812. Tal como se muestra 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 API o ABI en los 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 invocación o salto a un punto de entrada (una dirección de una instrucción) en el componente de *software* al que se transfiere el control. Por ejemplo, si la plataforma nativa 812 es un componente de *software*, la transferencia de control desde la plataforma nativa 812 al cliente 816 puede ocurrir a través de la interfaz de *software* 826 cuando se ejecuta una instrucción de invocación o salto en la plataforma nativa 812, especificando una dirección dentro del cliente 816 para invocar o saltar. La dirección especificada dentro del cliente 816 puede ser un punto de entrada para una función o método en la interfaz 816. La transferencia de control se indica como una flecha en la figura 8, por ejemplo: desde la plataforma nativa 812 hasta el 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 puede implementarse al menos en parte como un componente de *hardware*, por ejemplo, con instrucciones de un procesador especial para gestionar un enclave. Tal instrucción de *hardware* especial puede ejecutarse como parte de un componente de *software* de la plataforma nativa 812. En las modalidades alternativas, puede que no exista ningún componente de *software* para algunas o todas las funciones de la plataforma nativa 812. En estas modalidades alternativas, las interfaces de plataforma nativas 822 y 824 pueden ser instrucciones de *hardware* en lugar de puntos de entrada de *software*, por lo cual una función de la plataforma nativa 812 puede utilizarse por el enclave 814 o el cliente 816 o puede utilizarse ejecutando una instrucción de *hardware* especial en lugar del enclave 814 o el cliente 816, respectivamente, en lugar de ejecutar una instrucción de invocación o salto.

En algunas modalidades, el cliente 816 del enclave 814 puede ser un enclave en sí. 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 la 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 la 820 (no mostrada).

La figura 9 es un diagrama de bloques de las interfaces de componentes de *software* para un sistema de enclave local ejemplar con una capa de abstracción. El sistema de enclaves 900 incluye una plataforma de abstracción 912 para la conversión 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 825 de la figura 8. El protocolo de abstracción de enclave 940 incluye las interfaces 920, 922 para el enclave 914, mientras el protocolo de abstracción de 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 los protocolos de abstracción 940 y 942 cuando la plataforma nativa 918 y el protocolo nativo 944 se basan en diferentes arquitecturas de enclave, como SGX de Intel o VSM de Microsoft. Como en la figura 8, el cliente 916 del enclave 914 puede 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 en su lugar solo pueden 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 la memoria del enclave puede evitar la lectura, escritura o ejecución (saltar dentro o fuera) 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 direcciones

de memoria de la computadora 910, pero fuera del contenedor seguro del enclave 914. La plataforma de abstracción 912 puede implementarse en varias formas, que incluyen las instrucciones que están dentro o fuera del contenedor seguro del enclave 914, y también pueden incluir instrucciones ejecutadas desde dentro de las llamadas al hipervisor. En el caso de que la plataforma de abstracción 912 se incluya al menos en parte dentro del contenedor seguro del enclave 914, el código de la plataforma de abstracción dentro del contenedor seguro puede crearse por separado del resto del código del enclave 914 y solo puede interactuar con otro código del enclave a través de API/ABI públicas. Tal código de plataforma de abstracción puede vincularse estáticamente o vincularse dinámicamente con el resto del código dentro del contenedor seguro del enclave. El código de la plataforma de abstracción estáticamente vinculado puede ser un código objeto asociado con la plataforma de abstracción y está incluido (vinculado estáticamente), junto con el código que es más específico para el enclave 914, en una imagen binaria a partir de la cual puede crearse una instancia del 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 puede provenir de imágenes binarias separadas. Para un ejemplo enlazado dinámicamente, consultar la figura 14.

La figura 10 es un diagrama de bloques de las interfaces de componentes de *software* para un sistema de enclaves remotos ejemplar con una capa de abstracción. El sistema de enclaves remotos 1000 incluye un enclave 1014 en la computadora 1010, y un cliente 1056 del enclave 1014 en una computadora separada 1050. La combinación del módulo ficticio de cliente 1016 y la plataforma de conexión 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 con nombre idéntico 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 módulo ficticio de cliente 1016 puede comunicarse con la plataforma de conexión 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 de cliente 1042 y las interfaces 1024, 1026. Sin embargo, el protocolo de cliente remoto puede incluir una funcionalidad adicional para la conexión remota. Por ejemplo, un método en la interfaz 1064 como CreateEnclave para solicitar la creación de un enclave puede incluir adicionalmente la capacidad de especificar una computadora central de enclave, como la computadora 1010, donde se solicita la creación de un enclave. Puede proporcionarse una cita de certificación del enclave 1014 proporcionada al cliente 1056 a través del protocolo de cliente remoto en lugar de, o además de, un informe de certificación. La computadora 1050 con el cliente 1056 puede incluir o no una plataforma de enclave nativa 1058. Si la plataforma nativa 1058 está presente, puede o no ser compatible con la plataforma nativa de arquitectura de enclave de muestra 1018 y, por lo tanto, el protocolo nativo 1044 y el protocolo nativo remoto 1084 pueden no ser los mismos.

En una modalidad alternativa (no mostrada), el módulo ficticio de cliente 1016 puede no existir, y

la plataforma de abstracción 1012 puede comunicarse directamente con la plataforma de conexión remota de abstracción 1052 a través de una red.

Los protocolos de abstracción de enclave, 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 administrar y utilizar enclaves que se crean en múltiples plataformas nativas, como SGX de Intel y VSM de Microsoft. Estos métodos pueden proporcionar acciones básicas de enclave que pueden implementarse en las múltiples plataformas nativas, por lo que proporcionan una “abstracción” de las plataformas nativas. Las acciones básicas de enclave divulgadas en este punto incluyen la gestión del ciclo de vida del enclave, certificación, codificación de datos, transferencia de control, contadores monotónicos y tiempo confiable.

Las acciones básicas para la gestión del ciclo de vida del enclave pueden incluir métodos para provocar la creación de instancias o la terminación de un enclave como el enclave 914. Las acciones básicas de la gestión del ciclo de vida pueden ser parte del protocolo de abstracción de cliente 942 y, más específicamente, pueden implementarse mediante la plataforma de abstracción 912 como parte de la interfaz 924 para su uso mediante el cliente 916.

Un método para crear una instancia o crear un enclave puede incluir especificar una imagen ejecutable del código y/o los datos que se cargarán en la memoria aislada del contenedor seguro de enclave. Este código, antes o después de que se cargue en el contenedor de enclave, puede formar parte del estado inicial utilizado para la certificación del enclave instanciado (tal como se explicó anteriormente con respecto a la figura 5). Por ejemplo, un cliente de enclave puede especificar la imagen ejecutable de un enclave (un elemento binario de enclave) al proporcionar un indicador hacia una memoria tampón en la memoria que contiene la imagen ejecutable. Alternativamente, una imagen de enclave puede especificarse al indicar un archivo en un sistema de archivos que contiene el elemento 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 parcialmente cifrado. La medición del elemento binario de enclave para la certificación puede ocurrir en una imagen ejecutable cifrada o después del descifrado.

El código y/o los datos que se cargarán inicialmente en un enclave pueden indicarse especificando un archivo que contiene una imagen primaria de enclave. Además de este código y/o datos, una imagen primaria de enclave puede incluir los metadatos adicionales, como un tamaño deseado del enclave (la cantidad de memoria requerida dentro del contenedor de enclave), ubicaciones de los puntos de entrada dentro del código en el archivo, y una lista de archivos de imagen dependientes. Los archivos de imagen dependientes son otros archivos de imagen (no primarios) que también pueden cargarse en el enclave junto con el código y los datos en el archivo de imagen primaria. Los archivos de imagen dependientes pueden contener listas de otros archivos de imagen dependientes. En el caso del sistema de enclave local de la figura 9, las imágenes primarias y dependientes pueden archivarse en cualquier dispositivo de almacenamiento accesible, como a través 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 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 de conexión remota de abstracción y el modulo ficticio de cliente 1016 pueden coordinarse para copiar el archivo de imagen primaria especificado a la computadora 1010.

CreateEnclave es un método ejemplar para crear una instancia de un enclave. El método CreateEnclave puede describir con el 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 el presente documento puede utilizar varias formas de pseudocódigo para definir las interfaces de API. Por ejemplo, los parámetros de función, como enclavePath anterior, pueden complementarse 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 en mayúsculas pueden indicar un tipo de datos. HANDLE puede ser un número, como un número de 32 bits, utilizado para referirse indirectamente a algo. Por ejemplo, el método CreateEnclave anterior devuelve un HANDLE al invocador CreateEnclave, y ese HANDLE puede ser un descriptor del enclave que se creó; PCWSTR puede ser un indicador hacia un cierto tipo de cadena de texto; DWORD puede tener una cantidad de 32 bits sin firmar; PCVOID puede ser un indicador hacia los datos de tipo no especificado; BOOL puede ser un valor binario.

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

lpEnclaveName: puede especificar la ruta a la imagen primaria de enclave, que en las implementaciones puede ser algún otro tipo de firma para identificar el código y/o los datos de la imagen primaria de enclave, como un descriptor de un archivo abierto, un identificador de recursos uniforme (URI, por sus siglas en inglés), o un identificador que se utiliza en una búsqueda externa. Por ejemplo, un identificador único global (GUID, por sus siglas en inglés) puede utilizarse 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 que se creará (en caso de que una imagen de enclave sea compatible con varios tipos). Puede establecerse DEFAULT en caso de que el elemento

binario sea compatible con solo un enclave o el desarrollador especificó explícitamente un valor predeterminado. dwFlags: puede especificar una o más acciones predeterminadas que deben adoptarse al crear enclaves y cargar la imagen primaria de enclave.

enclaveInformation: pueden ser parámetros de entrada opcionales para la configuración en tiempo de ejecución del enclave.

IpEnclaveError: puede especificar un parámetro opcional para devolver un código de error específico de arquitectura.

Al completarse con éxito, CreateEnclave puede devolver un descriptor al enclave. En caso de error, puede devolverse NULL. También pueden devolverse otros identificadores (GUID, URI, etcétera) sin apartarse del alcance de esta divulgación. Para simplificar, esta especificación describirá las API utilizando un descriptor. La creación del enclave puede fallar, por ejemplo, debido a la falta de memoria de enclave, falta de compatibilidad 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 impiden que un enclave de un tipo específico se ejecute en el sistema.

Las implementaciones de CreateEnclave y otros métodos de API descritos a continuación pueden excluir uno o más de los parámetros de método descritos. Por ejemplo, con respecto a CreateEnclave, IpEnclaveName, flEnclaveType, dwFlags y enclaveInformation pueden excluirse, utilizando un valor predeterminado específico para esa API particular. El argumento IpEnclaveError también puede excluirse de la API, y pueden implementarse opcionalmente métodos alternativos para verificar los errores en la invocación a la API.

CreateEnclave puede ser responsable de cargar todos los módulos dependientes tal como se especifica en la imagen primaria de enclave. La imagen primaria de enclave puede ser un archivo de ejecución portátil (PE, por sus siglas en inglés) que especifica otros archivos de imagen binarios de los que depende la imagen primaria. CreateEnclave también puede realizar una inicialización de plataforma nativa específica, como finalizar las mediciones para la certificación, asignar estructuras para la seguridad de la capa de transporte (TLS, por sus siglas en inglés) y/u otros protocolos de gestión y comunicación de claves, etcétera. Las interfaces de protocolo de abstracción de enclave 920, 922 (que incluyen los métodos, por ejemplo, para la codificación y certificación de datos) pueden operarse una vez que se complete la inicialización de enclave.

TerminateEnclave es un método ejemplar para terminar un enclave:

VOID

```
TerminateEnclave(  
    _In_      HANDLE      hEnclave  
)
```

Se puede utilizar TerminateEnclave para destruir un enclave. En las implementaciones, la destrucción de un enclave puede incluir forzar a todos los subprocesos del enclave a regresar a la computadora central o terminar, y/o liberar la memoria asociada con el enclave. Invocar a TerminateEnclave en un enclave en ejecución puede terminarlo y liberar todos los recursos asociados

con el enclave.

La plataforma de abstracción de enclaves 912 puede incluir acciones básicas de transferencia de control de ejecución que pueden utilizarse, por ejemplo, para transferir el control entre un enclave y su cliente. Las acciones básicas de transferencia de control de ejecución pueden habilitar la comunicación entre el enclave 914 y el cliente 916 al iniciar la ejecución del código en un punto de entrada en el otro componente. Las acciones básicas de transferencia de control de ejecución permiten transmitir los datos dentro/fuera de los enclaves permitiendo que los parámetros se asocien con una solicitud de transferencia de control; los parámetros pueden especificar los elementos de datos individuales (los propios parámetros se comunican) o los parámetros pueden ser indicadores hacia las áreas de memoria (se comunican las memorias tampón indicadas por los parámetros). Estas acciones básicas pueden habilitar la transferencia de control a pesar de las limitaciones en la invocación directa o el salto entre el enclave 914 y el cliente 916 debido a las restricciones de seguridad en el contenedor de enclave.

Para invocar a un enclave, la interfaz 924 puede incluir mecanismos para permitir que un cliente 916 invoque a un enclave 914 a través de la interfaz 920. Por ejemplo, la interfaz 924 puede incluir los métodos GetProcAddress y CallEnclave:

```
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, como el cliente 916, puede invocar a un enclave, como el enclave 914, utilizando el indicador de función devuelto por GetProcAddress(). El parámetro lpProcName puede coincidir con la función exportada en la imagen primaria de enclave. Por ejemplo:

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

}

En otras modalidades de GetProcAddress, lpProcName puede ser otro identificador de la función exportada específica, como un número, 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 asumir adicionalmente un parámetro de entrada que especifica el enclave al cual se va a invocar, por ejemplo, el descriptor devuelve CreateEnclave.

Cuando se invoca a un enclave, puede suspenderse un subproceso en el proceso de cliente y puede utilizarse un subproceso de enclave (con un ID de subproceso separado) para atender la invocación en la solicitud. El código de enclave, que se ejecuta en el subproceso de enclave, puede tener acceso a la memoria que antes estaba disponible para el cliente de enclave antes de invocar al enclave. Por ejemplo, el cliente puede colocar los datos en la memoria tampón indicada por pParameter antes de invocar al método de abstracción CallEnclaveIn, y después el enclave puede tener acceso a la memoria tampón indicada por pParameter mientras atiende la invocación en la solicitud. Después de la invocación, el subproceso de invocación original (cliente) puede utilizarse para atender la invocación. Puede admitirse el reingreso (por ejemplo, una invocación en la computadora central puede invocar nuevamente al enclave).

Para invocar a un enclave, la interfaz 922 puede incluir los métodos relacionados con los métodos CallEnclaveIn anteriores que permiten que un enclave 914 invoque al cliente de enclave 916. Por ejemplo, el enclave 914 puede invocar a cualquier función en el proceso de computadora central de un tipo particular, por ejemplo, el tipo de función ENCPROC. El indicador de función para el mismo puede transmitirse utilizando la invocación en los parámetros al enclave.

BOOL

CallEnclaveOut(

 In ENCPROC pCallout,

 In PVOID pParameter,

 Out PVOID pReturn

)

// Invoca la función en el proceso de la computadora central

ENCPROC pCallout = (ENCPROC) OxFOO; // dirige a cierta función en la computadora central

PVOID pParameter = // Indicador hacia la 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 de que una imagen primaria de enclave esté en un formato de imagen ejecutable (PE, por sus siglas en inglés) portátil, los puntos de entrada de la función en la imagen pueden aparecer como puntos de entrada de "exportación", y cada punto de entrada exportado puede incluir un nombre textual, como "CallinExample", para identificar y diferenciar los puntos de entrada en esa imagen PE de enclave; en otras implementaciones, los puntos de entrada de la función pueden marcarse con

metadatos adicionales, como un bit que indica que una función puede ser un punto de entrada para el enclave. En el ejemplo anterior para invocar el enclave, la dirección de la función de invocación se proporciona como 0xF00 y es solo un ejemplo. La dirección real de una función de invocación puede determinarse de varias maneras. Por ejemplo, una dirección de función de invocación dentro de un cliente puede transmitirse al enclave como un parámetro para una función de invocación. En otro ejemplo, la dirección de una función de invocación puede registrarse por el cliente utilizando una función como RegisterCallOut:

```
BOOL RegisterCallOut (  
    _In_   ENCPROC    pCallout,  
    _In_   LPCTSTR    lpProcName)
```

El código dentro del enclave puede obtener la dirección de la función de invocación al invocar a una función complementaria 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, puede utilizarse un solo método CallEnclave para invocar dentro de un enclave y para invocar fuera de un enclave. En situaciones en las cuales el cliente de enclave 916 también es un enclave, la invocación del enclave 914 al cliente 916 también invocará al enclave.

La plataforma de abstracción 912 puede proporcionar acciones básicas para codificar datos en un enclave. Por ejemplo, la interfaz 922 puede proporcionar los servicios al enclave 914, como codificar y decodificar los datos para una identidad de enclave. Tal como se explicó anteriormente, un enclave puede tener múltiples identidades subdivididas, y los datos pueden codificarse para cualquier identidad de este tipo. Cuando los datos se codifican para una identidad que corresponde a un conjunto de posibles instancias de enclave, los datos codificados pueden decodificarse por cualquiera de ese conjunto correspondiente de instancias de enclave. Por ejemplo:

```
struct SEALING POLICY  
{  
    ENCLAVE ID TYPE    enclaveldType;  
};
```

Para cada valor de enclaveldType, el enclave se codificará con un ID de asignación. Los posibles tipos de identidad de enclave (y los valores de enclaveldType) incluyen:

```
ENCLAVE EXACTHASH  
ENCLAVE INSTANCEHASH: // codifica utilizando MRENCLAVE para SGX, IMAGE_HASH para  
VSM  
ENCLAVE_IMAGEIDS: // no compatible en SGX, utilizará IMAGE IDS para VSM  
ENCLAVE_FAMILYID: // utilizará PRODUCTID para SGX, FAMILYID para VSM  
ENCLAVE_AUTHORID: // utilizará MRSIGER para SGX, AUTHORID para VSM
```

La plataforma también puede aplicar una configuración de depuración adicional (creada y en tiempo de ejecución) a la política de codificación. Para diferentes políticas de depuración, pueden utilizarse diferentes claves de codificación. Por ejemplo, las configuraciones de depuración y liberación pueden utilizar diferentes claves de codificación.

DWORD

Enclave Seal(

```
_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 acciones básicas para la certificación, como para producir informes y citas de certificación, y para verificar informes 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() puede utilizarse por un enclave para afirmar la integridad del informe y que el informe se generó 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 asignarse a un proveedor de citas, el cual puede ser una fuente confiable para generar la cita específica. En CreateQuote, authData puede ser un indicador hacia los datos que se crean por, y en un formato definido por, el invocador de CreateQuote. Cabe señalar que authData no necesita entenderse por la plataforma de abstracción 912. authData puede incorporarse en la cita resultante. Puede esperarse que los proveedores de citas sean compatibles con 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 acciones básicas de enclave descritas anteriormente, una plataforma de abstracción de enclave puede proporcionar: gestión de memoria (por ejemplo, para asignar y liberar memoria, como la memoria restringida a un enclave o memoria que se comparte entre un enclave y su cliente); manejo de excepciones (por ejemplo, para manejar errores, o excepciones, que ocurren al ejecutar el código de enclave); sincronización de subprocesos; y funciones criptográficas (por ejemplo, cifrado, funciones *hash* y firma).

Las técnicas descritas anteriormente pueden implementarse en uno o más dispositivos o entornos informáticos, tal como se describe posteriormente. La figura 11 muestra un entorno informático de uso 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 las cuales pueden incorporarse algunas de las técnicas descritas en el presente documento. El entorno de sistema informático 1102 es solo un ejemplo de un entorno informático adecuado y no pretende sugerir ninguna limitación en cuanto al alcance de uso o funcionalidad de la materia objeto actualmente divulgada. Tampoco debe interpretarse que el entorno informático 1102 tiene una dependencia o requisito relacionado con algún

componente o una combinación de componentes ilustrados en el entorno operativo 1102 ejemplar. En algunas modalidades, los diversos elementos informáticos representados pueden incluir circuitería configurada para ejemplificar aspectos específicos de la presente divulgación. Por ejemplo, el término circuitería utilizado en la divulgación puede incluir componentes de *hardware* especializados configurados para realizar funciones mediante *firmware* o conmutadores. En otras modalidades ejemplares, el término circuitería puede incluir una unidad de procesamiento de uso general, memoria, etcétera, configurada por instrucciones de *software* que incorporan lógica operable para realizar funciones. En las modalidades ejemplar en las cuales los circuitos incluyen una combinación de *hardware* y *software*, un implementador puede escribir el código fuente que incorpore la lógica y el código fuente puede compilarse en un código legible por máquina que pueda procesarse por la unidad de procesamiento de uso general. Puesto que un técnico en la materia puede apreciar que el estado de la técnica evolucionó hasta un punto en el que existe poca diferencia entre *hardware*, *software* o una combinación de *hardware/software*, la selección de *hardware* en lugar de *software* para realizar funciones específicas es una opción de diseño que es decisión de un implementador. Más específicamente, un técnico en la materia puede apreciar que un proceso de *software* puede transformarse en una estructura de *hardware* equivalente, y una estructura de *hardware* puede transformarse en un proceso de *software* equivalente. Por lo tanto, la selección de una implementación de *hardware* en lugar de una implementación de *software* es una opción de diseño y es decisión del implementador.

La computadora 1102, que puede incluir cualquiera de un dispositivo móvil o teléfono inteligente, tableta, computadora portátil, computadora de escritorio, o un conjunto de dispositivos en red, recursos de computación en la nube, etcétera, incluye comúnmente una variedad de medios legibles por computadora. Los medios legibles por computadora pueden ser cualquier medio disponible al cual pueda acceder la computadora 1102 e incluye tanto medios volátiles como no volátiles, medios extraíbles y no extraíbles. La memoria de sistema 1122 incluye medios de almacenamiento legibles por computadora en forma de memoria volátil y/o no volátil, como la memoria exclusivamente de lectura (ROM, por sus siglas en inglés) 1123 y la memoria de acceso aleatorio (RAM, por sus siglas en inglés) 1160. Un sistema básico de entrada/salida (BIOS, por sus siglas en inglés) 1124, que contiene las rutinas básicas que ayudan a transferir información entre elementos dentro de la computadora 1102, como durante el arranque, se almacena generalmente en la ROM 1123. La RAM 1160 generalmente contiene datos y/o módulos de programa que son accesibles de inmediato y/o se operan actualmente por la unidad de procesamiento 1159. A modo de ejemplo, y no de limitación, la figura 11 ilustra el hipervisor 1130, el sistema operativo (OS, por sus siglas en inglés) 1125, programas de aplicación 1126, otros módulos de programa 1127 que incluyen un cliente de enclave 1165 y el enclave 1128.

La computadora 1102 también puede incluir otros medios de almacenamiento informáticos extraíbles/no extraíbles, volátiles/no volátiles. A modo de ejemplo solamente, la figura 11 ilustra una unidad de disco duro 1138 que lee o escribe en un medio magnético no volátil no extraíble, una unidad de disco magnético 1139 que lee o escribe en un disco magnético no volátil 1154 extraíble y una unidad de disco óptico 1104 que lee o escribe en un disco óptico no volátil y extraíble 1153 como un CD ROM u otro

medio óptico. Otros medios de almacenamiento informático extraíbles/no extraíbles, volátiles/no volátiles que pueden utilizarse en el entorno operativo ejemplar incluyen, entre otros, casetes de cinta magnética, tarjetas de memoria *flash*, discos versátiles digitales, cinta de video digital, RAM de estado sólido, ROM de estado sólido, y similares. La unidad de disco duro 1138 por lo general está conectada al bus de sistema 1121 a través de una interfaz de memoria no extraíble, como la interfaz 1134, y la unidad de disco magnético 1139 y la unidad de disco óptico 1104 están comúnmente conectadas al bus de sistema 1121 mediante una interfaz de memoria extraíble, como la interfaz 1135 o 1136.

Las unidades y sus medios de almacenamiento informático asociados descritos anteriormente e ilustrados 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, se ilustra que la unidad de disco duro 1138 almacena el sistema operativo 1158, programas de aplicación 1157, otros módulos de programa 1156, como aplicaciones de cliente de enclave y archivos binarios de enclave, y datos de programa 1155. Cabe señalar que estos componentes pueden ser iguales o diferente que el 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 datos de programa 1155 tienen números diferentes en el presente documento para ilustrar que como mínimo, son copias diferentes. Un usuario puede ingresar comandos e información en la computadora 1102 a través de dispositivos de entrada como un teclado 1151 y un dispositivo señalador 1152, comúnmente denominado ratón, bola seguidora o panel táctil. Otros dispositivos de entrada (no mostrados) pueden incluir un micrófono, palanca de mando, consola de juegos, antena parabólica, escáner, lector de retina o similares. Estos y otros dispositivos de entrada a menudo están conectados a la unidad de procesamiento 1159 a través de una interfaz de entrada de usuario 1136 que está conectada al bus de sistema 1121, pero pueden estar conectados por otra interfaz y estructuras de bus, como un puerto paralelo, un puerto de juegos o un bus serial universal (USB, por sus siglas en inglés). También se conecta un monitor 1142 u otro tipo de dispositivo de presentación al bus de sistema 1121 a través de una interfaz, como una interfaz de video 1132. Además del monitor, las computadoras también pueden incluir otros dispositivos de salida periféricos, como los altavoces 1144 y la impresora 1143, que pueden conectarse a través de una interfaz de periféricos de salida 1133.

La computadora 1102 puede operar en un entorno de red utilizando conexiones lógicas a una o más computadoras remotas, como una computadora remota 1146. La computadora remota 1146 puede ser una computadora personal, servidor, encaminador, computadora en red, dispositivo homólogo u otro nodo de red común, y comúnmente incluye muchos o todos los elementos descritos anteriormente con relación a la computadora 1102, aunque solo 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, por sus siglas en inglés) 1145 y una red de área amplia (WAN, por sus siglas en inglés) 1149, pero también puede incluir otras redes. Tales entornos de red son comunes en oficinas, redes de computadoras de toda la empresa, intranets, internet y recursos de computación en la nube.

Cuando se utiliza en un entorno de red LAN, la computadora 1102 está conectada a la LAN 1145

a través de una interfaz de red o adaptador 1137. Cuando se utiliza en un entorno de red WAN, la computadora comúnmente incluye un módem 1105 u otros medios para establecer comunicaciones a través de la WAN 1 149, como internet. El módem 1105, que puede ser interno o externo, puede conectarse al bus de sistema 1121 a través de la interfaz de entrada de usuario 1136, u otro mecanismo apropiado. En un entorno de red, los módulos de programa representados con relación a la computadora 1102, o partes de los mismos, pueden almacenarse en el dispositivo de almacenamiento de memoria remota. A modo de ejemplo, y no de limitación, la figura 11 ilustra los programas de aplicación remota 1148 que residen en el dispositivo de memoria 1147. Se apreciará que las conexiones de red mostradas son ejemplos y pueden utilizarse otros medios para establecer un enlace de comunicaciones entre las computadoras.

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

Una solicitud de un enclave puede ser una solicitud para realizar una acción básica de la plataforma de abstracción y puede incluir, por ejemplo, una solicitud para: crear un informe de certificación o cita de un enclave; codificar los datos en el enclave; invocar una función en el cliente de un enclave (invocar al cliente); leer un contador monótono (proporcionar el valor actual de un contador monótono); proporcionar una medida de tiempo confiable; y asignar memoria que puede compartirse entre un enclave y su cliente (por ejemplo, para permitir que un indicador hacia la memoria compartida se transmita como un parámetro cuando se invoca desde o hacia un enclave). En algunas modalidades, todo el espacio de memoria virtual de un cliente de enclave puede compartirse con (y puede accederse a él) desde el enclave, de manera que una solicitud para asignar memoria compartida puede implementarse como una solicitud para asignar memoria para el cliente de enclave. En algunas modalidades, los métodos de asignación de memoria compartida están disponibles tanto para un enclave como para su cliente.

Una solicitud de un cliente de enclave puede ser una solicitud para realizar una acción básica de la plataforma de abstracción y puede incluir, por ejemplo, una solicitud para: crear una instancia de un enclave; verificar un informe de certificación de un enclave; invocar a una función dentro de un enclave (invocar a un enclave); y asignar memoria que puede compartirse entre un enclave y su cliente.

Una solicitud de plataforma de abstracción puede traducirse en una solicitud de plataforma nativa en las operaciones 1204-1208. Los parámetros incluidos o implícitos en la solicitud recibida pueden convertirse en la etapa 1204 opcional si se determina, por ejemplo, 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 obtenido a partir de un informe de certificación de formato de abstracción, como una identidad de abstracción de enclave, se convertirá en un parámetro utilizado en un informe de certificación de formato nativo, como una identidad de enclave nativa.

Si se determina que la práctica de invocación de la plataforma nativa y la solicitud recibida difieren, la práctica de invocación puede convertirse en la etapa 1206 opcional. Una práctica de invocación puede convertirse, por ejemplo, al reordenar los parámetros en una pila de llamadas, mover los parámetros entre los registros de procesador y una pila de llamadas, y convertir entre los métodos de comunicación de condiciones de error, como devolver un valor de error e invocar a un controlador de excepciones.

En algunas modalidades, la plataforma nativa puede ser idéntica a la plataforma de abstracción para algunas solicitudes, en cuyo caso pueden omitirse las operaciones de conversión de los bloques 1204 y 1206.

En el bloque 1208, la solicitud convertida se envía a la plataforma nativa para hacer que la plataforma nativa realice la solicitud. Por ejemplo, en el caso de que la plataforma nativa se ajuste a la arquitectura de enclave de extensiones de protección de *software* (SGX, por sus siglas en inglés) de Intel, la plataforma nativa puede incluir instrucciones de procesador para los enclaves. En este caso, enviar la solicitud en el bloque 1208 puede incluir ejecutar una o más instrucciones de procesador para los enclaves. En otro ejemplo, la plataforma nativa puede ajustarse a la arquitectura de enclave del modo seguro virtual (VSM, por sus siglas en inglés) de Microsoft, que puede incluir un hipervisor con llamadas al hipervisor para los enclaves. Una llamada al hipervisor es una captura por *software* del código de hipervisor, y una llamada al hipervisor puede provocar un cambio en el contexto de procesador a un contexto en el cual pueden permitirse las operaciones privilegiadas. En este ejemplo de VSM, enviar la solicitud en el bloque 1208 puede incluir realizar llamadas al hipervisor y/u otros mecanismos para cambiar el contexto de ejecución a un contexto en el cual puedan permitirse las operaciones privilegiadas.

Enviar una solicitud a una plataforma nativa en este punto generalmente significa realizar 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 acción básica) solicitada, como crear un enclave, certificación, codificación de datos, transferencia de control o uso de contadores monotónicos y tiempo confiable.

La acción básica CreateEnclave puede utilizarse para crear una instancia de un enclave. Una solicitud CreateEnclave para crear una instancia de un enclave puede hacer que una plataforma de abstracción cree un contenedor seguro (por ejemplo, asignando parte de la memoria y estableciendo un límite de seguridad o aislamiento para esa memoria), copiar el código de enclave en ese contenedor seguro (por ejemplo, a partir de una imagen de enclave), y configurar o habilitar los puntos de entrada en el código de enclave (por ejemplo, de acuerdo con los metadatos del punto de entrada en una imagen de enclave).

El envío de una solicitud CreateEnclave a una plataforma nativa con un hipervisor habilitado para enclave (un hipervisor que proporciona funciones de gestión de enclave, como VSM), puede incluir la asignación de memoria y la creación de las llamadas al hipervisor para configurar las tablas de páginas de procesador para la memoria de una manera que evite que el código fuera del contenedor de enclave

acceda a esa memoria. Las llamadas al hipervisor de creación de enclave desde la plataforma de abstracción también pueden hacer que el hipervisor configure la información de configuración para transferir el control al enclave en los puntos de entrada designados. Más adelante, el código fuera del contenedor seguro puede hacer que las transferencias de control se transfieran para transferir la ejecución a los puntos de entrada designados dentro del contenedor seguro.

El envío de una solicitud CreateEnclave a una plataforma nativa con un procesador habilitado para enclave (un procesador con instrucciones de procesador de gestión de enclave, como SGX), puede incluir la plataforma de abstracción que ejecuta una instrucción como ECREATE para informar a la CPU que el área de memoria debe crearse como un contenedor de enclave seguro y ejecutar una instrucción como EADD para agregar páginas de datos a ese contenedor de enclave. Las instrucciones de procesador especiales también pueden utilizarse para crear páginas especiales en la memoria para designar los puntos de entrada en el enclave para la transferencia de control al enclave. Posteriormente, el código fuera del contenedor seguro puede ejecutar una instrucción como EENTER que especifica uno de los puntos de entrada designados para transferir el control de ejecución a ese punto de entrada de enclave.

La acción básica CreateReport puede utilizarse para crear un informe de certificación. Una capa de abstracción puede realizar una solicitud CreateReport para crear un informe de certificación de un enclave tal como se explicó anteriormente con respecto a las figuras 5 y 7. Con un hipervisor habilitado para el enclave, una capa de abstracción puede enviar la solicitud a la plataforma nativa al realizar una llamada al hipervisor que cambia el estado de ejecución a, por ejemplo, un contexto de monitorización de seguridad que tiene acceso a una clave secreta como PrivAK726 de la figura 7 que puede utilizarse para firmar los informes. Esta clave secreta solo puede estar disponible para el contexto de monitorización de seguridad si la computadora se inició en una configuración correcta, tal como se verificó con un registro de TCG con base en un TPM. El monitor de seguridad puede etiquetar los datos de informe con una identidad del enclave que se certifica.

Con un procesador habilitado para el enclave, puede enviarse una solicitud CreateReport a la plataforma nativa mediante la ejecución de una instrucción, como EREPORT, que genera un informe y lo envía a un enclave especial que tendrá acceso a una clave privada para firmar los informes.

La acción básica EnclaveSeal puede utilizarse para codificar los datos en un enclave. La codificación de datos en un enclave cifra los datos de una manera o con una clave que está asociada con el enclave. Una solicitud EnclaveSeal puede ser una solicitud para codificar los datos ubicados dentro de un enclave, como todo o parte del estado del enclave, en el enclave utilizando una política de codificación. La política de codificación, como la SEALING_POLICY anterior, puede especificar un tipo de identidad de enclave que indica qué enclaves deberían poder decodificar los datos. Los procesos de codificación en sí pueden utilizar una clave de cifrado asociada con la identidad de enclave especificada en la política de codificación. Posteriormente, una nueva creación de instancia del enclave puede decodificar 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 codificación en el tipo de identidad especificado.

La codificación de datos permite que los datos secretos o confidenciales de enclave se copien de manera segura en un almacenamiento no seguro, como en la memoria fuera del contenedor seguro del enclave o en un almacenamiento persistente, como un disco duro. Cuando los datos codificados son datos de estado de enclave, la codificación permite que un enclave se restablezca a un estado anterior, y permite que una operación segura del enclave se interrumpa y después se continúe en otro enclave.

Para restablecer un estado de enclave, primero se guarda el estado de un enclave al codificar su estado en el enclave. La codificación puede realizarse al cifrar los datos de estado con una clave asociada con el enclave. Posteriormente, quizás después de que el estado de enclave haya cambiado, los datos de estado codificados pueden decodificarse en el mismo enclave al descifrar los datos codificados y después al reemplazar un estado actual del enclave con los datos descifrados (por ejemplo, al copiar los datos descifrados en el contenedor de enclave seguro).

Para interrumpir una operación segura y continuar en otro enclave, la operación segura comienza al ejecutar una operación que comprende múltiples instrucciones de procesador en un primer enclave. Cuando se interrumpe el primer enclave, el estado de ese enclave puede codificarse a una identidad de enclave especificada en la política de codificación, y los datos codificados pueden guardarse en un almacenamiento no seguro, como el almacenamiento persistente local o en la nube. El primer enclave puede terminar (opcionalmente) o iniciar otras operaciones de enclave. Un segundo enclave puede instanciarse o reutilizarse para continuar la operación interrumpida al decodificar los datos de estado codificados en el segundo enclave. La operación interrumpida puede continuar en el segundo enclave donde se dejó el primer enclave.

Con un hipervisor habilitado para el enclave, una capa de abstracción puede enviar una solicitud EnclaveSeal a la plataforma nativa al realizar una llamada al hipervisor. La llamada al hipervisor puede cambiar el estado de ejecución a un contexto, por ejemplo, un contexto de monitorización de seguridad, que tendrá acceso a una clave secreta de codificación asociada con el enclave que puede utilizarse para codificar o decodificar los datos. La clave de codificación puede obtenerse a partir de una combinación de una identidad de enclave y una clave de plataforma secreta disponible solo para el monitor de seguridad. Esta clave de plataforma solo puede estar disponible para el monitor de seguridad cuando la máquina se inicia en una configuración correcta y la configuración de inicio se verifica con un registro TCG con base en un TPM. En esta modalidad de hipervisor habilitada para el enclave, el código de enclave nunca tiene acceso a la clave de codificación.

Con un procesador habilitado para el enclave, puede enviarse una solicitud EnclaveSeal a la plataforma nativa al ejecutar una instrucción, como EGETKEY, para obtener una clave de cifrado. Este algoritmo puede generar una clave de codificación que es única para el enclave. La clave de codificación puede obtenerse a partir de una identidad de enclave y una clave secreta incorporada en el procesador. El código dentro de un enclave puede entonces cifrar los datos con la clave de codificación. Los datos pueden codificarse mediante el cifrado con la clave de codificación, por ejemplo, mediante un código dentro de un enclave, una plataforma de abstracción o una plataforma nativa. EnclaveUnseal también puede utilizar EGETKEY para generar la clave de decodificación.

Una solicitud de transferencia de control puede ser una solicitud para transferir el control de ejecución de procesador desde las instrucciones dentro de un enclave a un punto de entrada fuera del enclave (por ejemplo, `CallEnclaveOut`), o lo contrario a partir de las instrucciones fuera del enclave a un punto de entrada dentro del enclave (por ejemplo, `CallEnclaveIn`). Esto puede realizarse, por ejemplo, para una operación de base de datos segura. Después de crear una instancia de un enclave de base de datos, un cliente de enclave puede solicitar que el enclave realice una operación específica, como una consulta en la base de datos utilizando la acción básica `CallEnclaveIn` para transferir el control a un punto de entrada dentro del enclave, lo contrario es posible, donde el código es privado mientras los datos son accesibles. Después de que el enclave complete la consulta, el resultado de consulta puede devolverse (posiblemente después de cifrar el resultado) al cliente con la acción básica `CallEnclaveOut` para transferir el control de regreso al cliente en un punto de entrada en el cliente que puede recibir el resultado de consulta. Las acciones básicas `CallEnclaveIn` y `CallEnclaveOut` pueden llevar un indicador a una memoria tampón de la memoria que puede compartirse entre un enclave y su cliente (la memoria tampón puede ser legible, escribible y/o ejecutable por el enclave o su cliente).

Con un hipervisor habilitado para el enclave, una capa de abstracción puede enviar una solicitud `CallEnclaveIn` a la plataforma nativa al realizar una llamada al hipervisor. La llamada al hipervisor puede cambiar el estado de ejecución a un contexto, por ejemplo, un contexto de monitorización de seguridad, que guardará los registros de CPU, restaurará un conjunto guardado previamente de valores de registro de CPU de enclave (posiblemente desde la memoria de enclave), cambiará la configuración de tabla de páginas 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 puede enviarse a la plataforma nativa mediante una llamada al hipervisor que guardará los registros de CPU de enclave (posiblemente guardando en la memoria de enclave) y restaurará los registros de CPU previamente guardados para un cliente de enclave, cambiará la configuración de tabla de páginas para evitar el acceso a la memoria de enclave, y transferirá el control de procesador a un punto de entrada en el cliente de enclave fuera del enclave.

Con un procesador habilitado para el enclave, puede enviarse una solicitud `CallEnclaveIn` a la plataforma nativa al ejecutar una instrucción, como `EENTER`, que puede hacer que la CPU restaure un conjunto de registros de CPU de enclave (posiblemente desde la memoria de enclave) e invocar una función (control de transferencia a un punto de entrada) dentro del enclave. Una acción básica `CallEnclaveOut` puede ejecutar una instrucción, como `EEXIT`, que puede transferir el control a las instrucciones fuera del enclave y/o provocar una falla que transfiera el control fuera del enclave.

El contador monótono tiene una variedad de usos. Por ejemplo, un enclave puede intentar restringir hasta qué punto puede revertir su estado. Los contadores monótonos pueden utilizarse, por ejemplo, como un código único descartable para garantizar la actualización de los mensajes, tal como se abordó anteriormente con respecto a la figura 4. Los contadores monótonos generalmente tienen la capacidad de leerse e incrementarse, pero no pueden disminuirse. Para restringir la reversión o revertir el estado de un enclave, el código dentro de un enclave puede incrementar un contador monótono

asociado y después guardar el valor del contador junto con el estado interno del enclave. El estado y el valor del contador pueden guardarse, por ejemplo, con la acción básica EnclaveSeal. Posteriormente, al restaurar el estado del enclave, por ejemplo, utilizando la acción básica EnclaveUnseal, el código dentro del enclave puede leer el valor actual del contador monótono y compararlo con el valor del contador con el estado no codificado. Si el valor del contador con el estado sin codificar es menor que el valor actual del contador, el enclave puede evitar el uso del estado sin codificar.

Con un hipervisor habilitado para el enclave, una capa de abstracción puede enviar una solicitud a la plataforma nativa para que lea o aumente un contador monótono al realizar una llamada al hipervisor que esté expuesta al enclave. Cuando se invoca una llamada al hipervisor para leer o incrementar el contador, el procesador cambiará el estado de ejecución a un contexto, como un monitor de seguridad, que verificará la identidad del enclave que realiza llamada al hipervisor, y después leerá o incrementará, respectivamente, el correspondiente contador monótono almacenado en, por ejemplo, un almacenamiento no volátil, como un chip de TPM. Alternativamente, el monitor de seguridad puede leer o incrementar un contador almacenado en un servidor remoto confiable o un conjunto de servidores remotos confiables, al establecer un canal de comunicación seguro con tal servidor y solicitarle que lea o incremente un contador monótono específico. El servidor o servidores remotos confiables pueden mantener el contador dentro de un enclave para aislarlo del resto del código en la computadora central.

Con un procesador habilitado para el enclave, puede enviarse una solicitud a la plataforma nativa al ejecutar una instrucción. Con tal procesador, las acciones básicas de contador monótono pueden implementarse al leer o incrementar un contador en el almacenamiento de memoria no volátil en un chip en la placa base de la computadora. Alternativamente, estas acciones básicas también pueden implementarse utilizando un servidor remoto confiable como con el hipervisor habilitado para el enclave.

La figura 13 muestra un diagrama de flujo ejemplar para 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 bloque 1302. En el bloque 1304, la plataforma de abstracción puede determinar si una plataforma nativa incluye las instrucciones de procesador de enclave, por ejemplo, al determinar si la plataforma nativa se adapta a SGX. Si lo hace, las instrucciones de procesador de enclave se ejecutan en el bloque 1306. En el bloque 1308, la plataforma de abstracción puede determinar si una plataforma nativa incluye las llamadas al hipervisor de enclave, por ejemplo, al determinar si la plataforma nativa se adapta a VSM. Si lo hace, la plataforma nativa realiza las llamadas al hipervisor de enclave. Los resultados de la ejecución de las instrucciones de enclave o de invocar a las llamadas al hipervisor de enclave se depuran en el bloque 1312. La depuración puede incluir, por ejemplo, la conversión de los parámetros de salida o el manejo de excepciones de las instrucciones de procesador de enclave o las llamadas al hipervisor de enclave en el formato o protocolo de la interfaz de capa de abstracción. Los parámetros de salida convertidos entonces se devuelven al solicitante original (enclave o cliente) en el bloque 1314.

Identidad de enclave abstracta

La figura 14 muestra las imágenes binarias de enclave ejemplares utilizadas para crear una

instancia de un enclave. Puede crearse una instancia de un enclave al crear un contenedor seguro, como el contenedor de enclave 1490, y al copiar partes de una o más imágenes de enclave en el contenedor. El contenedor de enclave 1490 pudo crearse por referencia a la imagen de enclave primaria.1410 Una imagen primaria puede incluir referencias a otras imágenes de enclave dependientes. En este ejemplo, la imagen primaria 1410 incluye Dependence1 y Dependence2 como referencias a las imágenes de enclave dependientes 1420 y 1450, respectivamente. La imagen 1420 incluye las dependencias adicionales de las imágenes 1430 y 1440, mientras la imagen 1450 depende de la imagen 1460. Una vez que todas estas imágenes (o porción de las mismas) se copian en el contenedor 1490, el enclave resultante puede incluir imágenes no pertenecientes a la plataforma 1402, que pueden incluir código y datos que son únicos o específicos para el enclave instanciado, las imágenes de plataforma de abstracción 1404 y las imágenes de plataforma nativa 1406.

Cada imagen de enclave, como la imagen primaria 1410, puede incluir ID, dependencias, código, datos y una firma del autor de la imagen. En el ejemplo de la imagen 1410, se incluyen dos ID 1410.1 y 1410.2. Estos ID pueden ser UUID que especifican, por ejemplo, un valor de identidad abstracta correspondiente a un valor ImageID, FamilyID o AuthorID que, individual o colectivamente, puede utilizarse para identificar cualquier enclave instanciado con esa imagen de enclave. Tal como se muestra, la imagen 1410 tiene dos ID, pero son posibles más o menos ID. 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 utilizarse por cualquier código cargado en el contenedor 1410. La imagen 1410 también puede incluir una firma Sig 1410 para garantizar la integridad de cualquiera o todos los otros contenidos de la imagen, como ID, referencias de dependencia, código y datos. Otras imágenes 1420-1460 también pueden contener ID, referencias de dependencia, código, datos y firmas.

Un indicador de dependencia, como Dependence1 y Dependence2 o la imagen 1410, Dependence1 y Dependence2 de la imagen 1420, y Dependence1 de la imagen 1450, pueden especificarse de varias maneras. Si las imágenes 1410-1460 se almacenan en la memoria de un sistema informático, un indicador de dependencia puede simplemente ser una dirección en la memoria. Si las imágenes de enclave son archivos en un sistema de archivos, las referencias pueden ser nombres de archivos. En algunas modalidades, las referencias pueden ser un identificador lógico. Un identificador lógico puede ser un número único, como un UUID, o puede ser otro tipo de datos, como una cadena de texto, que de otra manera identifica una imagen de dependencia. Por ejemplo, una cadena de texto puede indicar el autor, fuente, nombre del producto, familia de productos y/o número de versión de una imagen binaria dependiente. Un identificador lógico incluye una ubicación web o de internet, como una ubicación donde puede recuperarse una copia de un elemento binario dependiente.

En algunas modalidades, un archivo de imagen de enclave puede ubicarse al buscar un indicador de dependencia, como un identificador lógico, en un registro de imágenes de enclave para encontrar un indicador hacia la versión actual o copia local de la imagen de enclave a la cual se hace referencia. En algunos casos, puede utilizarse un servicio confiable para resolver un indicador de dependencia en la

identificación de una imagen de enclave particular o ubicación de imagen.

En algunas modalidades, un indicador de dependencia puede ser un identificador criptográficamente seguro, como un valor *hash* criptográfico de la imagen binaria de enclave dependiente pretendida. Tal valor *hash* puede incluir todo el elemento binario dependiente, o solo una porción del mismo. La porción de un elemento binario dependiente incluido en un indicador de dependencia puede incluir valores de identidad abstracta, como ID 1410.1 o ID 1420.2, y puede ser valores de identidad abstracta. Un servicio de resolución para un identificador criptográficamente seguro puede no necesitar ser tan confiable como 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 al computar el valor *hash* del propio elemento binario dependiente.

La figura 15 muestra un diagrama de flujo ejemplar para un método 1500 para realizar una operación de enclave con la 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. Puede incluirse un valor de identidad en un informe de certificación y puede asociarse con un tipo de identidad abstracta (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. Además, el código de enclave idéntico creado en contenedores seguros en dos plataformas de enclave nativas diferentes también puede tener el mismo valor de identidad abstracta. El método 1500 puede realizarse, por ejemplo, mediante una capa de plataforma de abstracción entre una plataforma de enclave nativa y ya sea un enclave o un cliente de enclave.

En el bloque 1502, se crea una instancia de un enclave a partir de una imagen de enclave, como la imagen de enclave primaria 1410 de la figura 14. La imagen de enclave puede ser una imagen primaria que incluye el código de enclave, datos, una lista de identidades, una lista de las imágenes de enclave dependientes y una firma. Para garantizar la integridad de las imágenes de enclave, las imágenes pueden firmarse con una clave privada que puede corresponder al autor de la imagen de enclave. La lista de ID de identidad de enclave en la imagen de enclave puede corresponder a tipos de identidad abstracta como ImageID, FamilyID y AuthorID, cada uno destinado a identificar la imagen de enclave colectivamente junto con otras imágenes de enclave relacionadas. Una lista de imágenes de enclave dependientes puede referirse a otras imágenes de enclave que contienen un código de enclave del que depende el código en la imagen de enclave primaria. Una imagen de enclave dependiente puede o no autorizarse por parte del mismo autor, y algunas imágenes de enclave dependientes pueden estar asociadas con una plataforma de enclave en general (ya sea una plataforma de abstracción o una plataforma nativa) en lugar de estar particularmente asociadas con una imagen de enclave primaria o el autor de la imagen de enclave primaria. Puede crearse una instancia del enclave al crear un contenedor de enclave seguro utilizando cualquier plataforma de enclave nativa, y al copiar toda o una porción de la imagen primaria y cualquier imagen de enclave dependiente en el contenedor seguro.

En el bloque 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, como ImageID 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 bloque 1503 describe las operaciones para realizar la operación (como certificación, codificación de datos o uso de un contador monótonico, etcétera) con el enclave instanciado utilizando un valor de identidad obtenido para ese tipo de identidad del enclave instanciado. La identidad puede determinarse utilizando un valor *hash* de un subconjunto de imágenes de enclave. El subconjunto de imágenes de enclave que se utiliza como entrada al valor *hash* puede basarse en parte en el tipo de identidad que se pretende utilizar en la operación de enclave.

En el bloque 1504, una porción de la imagen del enclave, denominada porción de identidad en el presente documento, se determina con base en el tipo de identidad. La porción de identidad puede incluir todas, parte o ninguna de las diversas imágenes binarias de enclave utilizadas para crear una instancia de un enclave en el bloque 1502. La porción de identidad puede incluir todo, una porción o ninguno del código de enclave contenido en el enclave imagen. La porción de identidad también puede incluir cero, uno o más ID de identidad enumerados en una porción sin código de las imágenes de enclave incluidas. La porción de identidad puede o no incluir los datos de enclave que contienen las imágenes de enclave. La porción de identidad puede incluir cualquier combinación de estas diversas porciones de las imágenes de enclave. Por ejemplo, una porción de identidad puede incluir todo el código, ninguno de los datos y dos o cuatro ID de identidad disponibles. En el bloque opcional 1506, se determinará qué imágenes de enclave dependiente se incluirán en la porción de identidad y se determinará una porción de identidad de cada imagen incluida.

La porción de identidad de las imágenes dependientes puede o no ser la misma que la porción de identidad de una imagen de enclave primaria. Por ejemplo, todo el código e ImageID se incluyen en la porción de identidad de una imagen primaria, mientras ningún código y solo FamilyID de una imagen dependiente pueden incluirse en la porción de identidad de la imagen dependiente.

Cuando el código de enclave se incluye en la porción de identidad, las porciones del código de enclave en la porción de identidad pueden determinarse mediante una combinación del tipo de identidad y una indicación de qué dependencias deben incluirse en la porción de identidad. El tipo de identidad InstanceHash puede incluir, por ejemplo, el código de enclave en la imagen primaria, pero no las imágenes dependientes, mientras el tipo de identidad ExactHash puede incluir el código de enclave en todas 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 del autor de una plataforma de enclave pueden considerarse como parte de la plataforma de enclave. Además, o como una alternativa, la imagen primaria puede incluir una lista de las imágenes de enclave dependientes que se incluirán o excluirán en la porción de identidad para los tipos de identidad de InstanceHash o ExactHash.

Los ID de identidad de enclave, que pueden incluirse como metadatos en una imagen de enclave, pueden incluirse 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 los metadatos de ID correspondientes de la imagen de enclave. Cuando los tipos de identidad están subdivididos o en capas, la porción de identidad para los tipos de nivel inferior puede incluir los datos de ID para los tipos de nivel superior. Por ejemplo, la porción de identidad para ImageID puede incluir los datos de ID para ImageID, FamilyID y AuthorID, mientras la porción de identidad para AuthorID solo puede incluir los datos de ID para AuthorID.

Los tipos de identidad que incluyen el código de enclave, como InstanceHash y ExactHash, proporcionan un nivel superior de seguridad, por ejemplo, al cliente de enclave a través de la certificación, donde cierto código de enclave se está ejecutando dentro de un enclave. Sin embargo, la identidad de enclave cambiará necesariamente cuando cambie alguna porción de identidad del código de enclave. Por ejemplo, si una solución de seguridad u otro error se corrige en una nueva versión de una imagen de enclave, el valor de identidad resultante con base en el nuevo código también cambiará. Al proporcionar un mecanismo para que ciertas porciones del código de enclave se excluyan del cálculo de valor *hash* de identidad, la identidad de un enclave puede desasociarse de los cambios a la porción excluida del código de enclave. Por ejemplo, cuando el código de enclave de un autor depende del código de enclave proporcionado por la plataforma de enclave, la identidad de enclave puede desasociarse de las revisiones en la plataforma dependiente.

En el bloque 1508, se determina un valor de identidad que puede representar una identidad del enclave instanciado en el bloque 1502. Puede determinarse un valor de identidad al calcular un valor *hash* con respecto a la porción de identidad determinada previamente de la imagen o imágenes de enclave (el valor de identidad es la salida de una función *hash* donde la porción de identidad es la entrada a la función *hash*). En algunas modalidades, la entrada a la función *hash* será las porciones de las imágenes de enclave originales, mientras en otras modalidades, la entrada a la función *hash* será las porciones de un contenedor de enclave después de copiar la porción de identidad de la imagen en el contenedor (y posiblemente descifrar la porción de identidad en el caso en que se cifre una imagen de enclave original).

En el bloque 1510, la integridad del valor de identidad resultante puede verificarse opcionalmente al verificar la integridad de las imágenes de enclave originales. La integridad de una imagen de enclave puede verificar con una clave pública correspondiente a una clave privada utilizada para firmar la imagen de enclave. Tal par de claves públicas/privadas puede asociarse, por ejemplo, con el autor de las imágenes de enclave, de modo que la confianza en el valor de identidad resultante puede originarse en la confianza del autor del enclave.

Finalmente, en el bloque 1512, puede realizarse una operación relacionada con el enclave instanciado utilizando el valor de identidad. Por ejemplo: un informe de certificación del enclave instanciado puede generarse o verificarse para un tipo de identidad; los datos pueden codificarse o decodificarse a partir del enclave instanciado en una identidad; y puede utilizarse un contador monotónico o un tiempo confiable vinculado al enclave instanciado y al tipo de identidad.

Las operaciones de enclave que utilizan los tipos de identidad de nivel superior permiten

interacciones entre grupos de posibles instanciaciones de enclave. La certificación de un tipo de identidad de alto nivel puede proporcionar una equivalencia de informe de certificación para todos los enclaves con la misma identidad de alto nivel. Por ejemplo, un informe de certificación de un tipo de identidad AuthorID puede ser equivalente al informe de certificación de todos los enclaves instanciados a partir de una imagen primaria que contenga los mismos metadatos AuthorID. Los datos codificados en un tipo de identidad de alto nivel pueden decodificarse por cualquier enclave con el mismo valor de identidad de alto nivel. Por ejemplo, los datos codificados en un enclave instanciado con el tipo de identidad AuthorID pueden decodificarse por cualquier otro enclave instanciado con los mismos metadatos AuthorID en su imagen de enclave primaria.

Equivalencia de identidad de enclave

La figura 16 representa un sistema ejemplar con equivalencia de identidad abstracta de enclave. Un cliente de enclave 1602 puede comunicarse con un primer enclave 1612 instanciado en un contenedor de enclave seguro de la primera plataforma nativa 1616 a través de la plataforma de abstracción 1614, y el cliente 1602 también puede comunicarse con un segundo enclave 1622 instanciado en un contenedor de enclave seguro de la segunda plataforma nativa 1626 a través de 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 anterior de la segunda plataforma nativa del mismo fabricante de la plataforma nativa. Alternativamente, la primera plataforma nativa 1616 y la segunda plataforma nativa 1626 pueden ajustarse a arquitecturas de enclave completamente diferentes, como VSM y SGX.

Un cliente de enclave puede determinar de forma segura que los enclaves son equivalentes al comparar los valores de identidad obtenidos a partir de los informes de certificación. El cliente de enclave 1602 puede identificar de forma segura cada uno de los enclaves al recibir los informes de certificación separados a partir del primer enclave 1612 y del segundo enclave 1622. Un valor de identidad puede incluir (u obtenerse a partir de) cada uno de estos informes de certificación. Si los valores de identidad son los mismos, el cliente de enclave 1602 puede tener confianza de que el primer enclave 1612 y el segundo enclave 1622 son equivalentes en algún sentido. Los valores de identidad de los informes de certificación pueden ser valores de identidad abstracta correspondientes a un tipo de identidad abstracta particular (como ExactHash, InstanceHash, ImageID, FamilyID o AuthorID), o los valores *hash* de tales valores de identidad abstracta. En este caso, la equivalencia puede determinarse cuando los enclaves no son exactamente idénticos. Es posible que dos enclaves no sean exactamente idénticos, pero aún así se determine que son equivalentes, por ejemplo, cuando las imágenes de enclave cargadas en el contenedor de enclave son versiones diferentes de la misma funcionalidad, o mismas imágenes primarias con imágenes dependientes diferentes, o mismas imágenes de enclave cargadas en contenedores de enclave de diferentes arquitecturas de enclave.

El primer enclave 1612 puede considerarse equivalente pero no idéntico al segundo enclave 1622 por una variedad de situaciones. En un primer ejemplo, solo un subconjunto de código cargado inicialmente en los contenedores de enclave es el mismo (por ejemplo, equivalente para los tipos de identidad abstracta `ExactHash` o `InstanceHash`). En un segundo ejemplo, el autor del código de enclave pudo incluir un ID idéntico en dos imágenes binarias de enclave diferentes, aunque el código de las dos imágenes binarias es diferente (por ejemplo, equivalente para los tipos de identidad `ImageID`, `FamilyID` o `AuthorID`). En un tercer ejemplo, el código en cada enclave es completamente el mismo, pero se carga (crea una instancia) en diferentes plataformas nativas. En este tercer ejemplo, la primera plataforma nativa 1616 y la segunda plataforma nativa 1626 pueden producirse por diferentes fabricantes y, por lo tanto, la confianza de los diferentes informes de certificación se basa en las diferentes autoridades de certificación (consultar la figura 7, elemento 738) de los diferentes fabricantes. Un ejemplo en el cual las dos plataformas nativas son diferentes es en un parque de servidores o en computación en la nube donde los servidores asignados para las cargas de trabajo de procesamiento del primer y segundo enclaves son servidores que no son compatibles con la misma plataforma de enclave nativa.

En una modalidad alternativa, el primer enclave puede ser el cliente del segundo enclave, de manera que los bloques 1602 y 1612 se combinen. La determinación de la equivalencia de enclave en esta modalidad puede incluir determinar, dentro del primer enclave, que un valor de identidad de un informe de certificación del segundo enclave es el mismo que el valor de identidad propio del primer enclave (en un nivel de identidad abstracta particular).

La figura 17 muestra un diagrama de flujo ejemplar para el procesamiento paralelo con dos enclaves equivalentes. El proceso 1700 puede realizarse, por ejemplo, por un cliente de dos o más enclaves diferentes. En el bloque 1702, se crean instancias de dos enclaves en diferentes instancias de plataforma nativa, por ejemplo, tal como se muestra en la figura 16. Pueden crearse instancias de los enclaves mediante un cliente de enclave que especifique una imagen binaria de enclave (como la imagen primaria 1410 de la figura 14) a través de un método `CreateEnclave` de las plataformas de abstracción 1614 y 1624. La imagen binaria de enclave especificada para crear instancias de los dos enclaves puede ser igual o diferente. Se crea un informe de certificación para cada enclave instanciado en el bloque 1704. Los informes de certificación pueden crearse, por ejemplo, a petición del cliente de enclave o a petición de los propios enclaves. Una entidad que pretende probar la equivalencia de los dos enclaves, como el cliente de enclave, obtiene copias de ambos informes de certificación. Los informes de certificación pueden verificarse opcionalmente en el bloque 1706. Por ejemplo, la integridad del informe puede verificarse al verificar la firma de certificación con un certificado de aprobación (como la figura 7, elemento 728) de la plataforma nativa que produjo el informe de certificación. Además, el certificado de aprobación puede verificarse con la clave pública del fabricante de la plataforma nativa (como la figura 7, elemento 732). Los valores de identidad (o un valor *hash* de los mismos) pueden extraerse de cada informe de certificación en el bloque 1708, y la equivalencia de los dos enclaves puede determinarse al verificar que los valores de identidad extraídos sean los mismos para cada enclave. Estos valores de identidad pueden ser valores de identidad abstracta (o valores *hash* de los mismos) asociados con un

tipo de identidad.

Una vez que un cliente de enclave ha demostrado la equivalencia de dos instanciaciones de enclave de las operaciones en los bloques 1708 y 1710, los dos enclaves pueden utilizarse indistintamente, de acuerdo con el tipo de equivalencia que se muestra. Los bloques 1712-1720 representan un método ejemplar para utilizar los enclaves equivalentes para utilizar los dos enclaves equivalentes instanciados en una manera de procesamiento paralelo. En los bloques 1712 y 1716, una porción de un conjunto de datos de entrada, como una porción de una base de datos o una porción de un archivo de imagen digital, se copia en el primer y segundo enclave. La porción del conjunto de datos copiada puede ser idéntica o diferente de acuerdo con la tarea de procesamiento en cuestión. Una operación de procesamiento puede realizarse de manera segura en paralelo al realizar parcialmente de manera simultánea la operación en el primer enclave en el bloque 1714 y realizar parcialmente la operación en el segundo enclave en el bloque 1718. La operación puede ser, por ejemplo, buscar en la base de datos o realizar una operación de procesamiento de imágenes. El primer enclave puede buscar en la primera mitad de la base de datos o realizar la operación de procesamiento de imágenes en la primera mitad de una imagen, mientras el segundo enclave puede buscar en la segunda mitad de la base de datos o realizar la operación de procesamiento de imágenes de la segunda mitad de la imagen. Finalmente, en el bloque 1720, los resultados del procesamiento paralelo en el primer y segundo enclave pueden combinarse, por ejemplo, al combinar las dos mitades clasificadas de la base de datos, o al volver a unir las dos mitades de la imagen.

La figura 18 muestra un diagrama de flujo ejemplar para el procesamiento en serie con dos enclaves equivalentes. Tal como se muestra en la figura 18, una operación de enclave, como una operación de base de datos o una operación de procesamiento de imágenes, se realiza de manera segura en dos partes secuenciales en dos enclaves separados. El proceso 1800 puede realizarse, por ejemplo, por el cliente de enclave 1602 de la figura 16. En el bloque 1802, se crea un primer enclave en una primera plataforma de enclave nativa, y se crea un informe de certificación del primer enclave en el bloque 1804. Este primer informe de certificación (del primer enclave) se puede verificar en el bloque 1806, por ejemplo, tal como se describe anteriormente con respecto al bloque 1706 de la figura 17. En el bloque 1808, se inicia una operación segura en el primer enclave, pero no se completa. El estado del primer enclave puede codificarse opcionalmente para que se mueva de manera segura fuera del primer enclave en el bloque 1810. Por ejemplo, el primer estado de enclave puede codificarse en un tipo de identidad del primer enclave. Una vez que se guarde el estado de enclave, el primer enclave puede terminarse (no mostrado).

Se utiliza un segundo enclave a partir del bloque 1812. En el bloque 1812, se crea una instancia del segundo enclave en una segunda plataforma nativa. Tal 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 la segunda plataforma nativa pueden ser iguales o diferentes. Se crea un informe de certificación de la segunda plataforma nativa en el bloque 1814 y, opcionalmente, este segundo informe de certificación puede verificarse en el bloque 1816. El valor de identidad del primer y segundo informe

de certificación puede compararse en el bloque 1818 para verificar la equivalencia del primer y segundo enclave. En las modalidades alternativas, puede crearse una instancia del segundo enclave y verificarse la equivalencia (bloques 1812-1818) antes de que se inicie la operación segura en el primer enclave en el bloque 1808. Para continuar la operación segura iniciada en el primer enclave, el estado codificado del primer enclave puede copiarse en el segundo enclave y decodificarse en el bloque 1820. En el bloque 1822, la operación segura se completa en el segundo enclave (utilizando el estado de enclave copiado de forma segura desde el primer enclave, si el estado se copió).

Codificación de datos distribuida

La figura 19 es un diagrama de bloques de un sistema de codificación de datos distribuida ejemplar. La codificación de datos puede distribuirse a través de múltiples enclaves, donde esos enclaves se alojan en plataformas de enclaves nativas independientes y/o en computadoras separadas. Tal como se explicó anteriormente, las acciones básicas de abstracción EnclaveSeal y EnclaveUnseal pueden codificar y decodificar los 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 la decodificación solo a los enclaves alojados en la misma computadora o la misma instancia de plataforma de enclave nativa. La figura 19 muestra un sistema de codificación de datos distribuida, donde la codificación y decodificación de los datos pueden ocurrir en una plataforma nativa o computadora diferente a la plataforma nativa y a la computadora que aloja un enclave. El sistema 1900 incluye computadoras 1910, 1930, 1950, con la red 1902 que conecta las computadoras 1910 y 1930, y la red 1904 que conecta las computadoras 1930 y 1950. La computadora 1910 aloja el enclave de origen 1912, a partir del cual pueden obtener los datos que se codifican; la computadora 1930 aloja un enclave de codificación distribuida (DSE, por sus siglas en inglés) 1932 para atender las solicitudes de codificación y decodificación distribuidas; y la computadora 1950 aloja el enclave de destino 1952 donde los datos previamente codificados están decodificados. Tal 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, a través 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, a través de un protocolo nativo. En las modalidades alternativas, uno o más enclaves 1912, 1932, 1950 pueden alojarse directamente en las plataformas nativas 1961, 1936, 1956 sin una plataforma de abstracción intermedia. Los datos codificados 1938 pueden ser datos codificados en el DSE 1932 utilizando una clave asociada con el DSE 1932 o su plataforma nativa de alojamiento 1936. Los datos codificados 1938 pueden almacenarse en una ubicación menos protegida, como en la computadora 1930 fuera del contenedor de enclave seguro del DSE 1932, por ejemplo, en otro lugar en el espacio de memoria de la computadora 1930 o en un sistema de archivos de un disco duro.

La codificación de datos distribuida puede incluir la autenticación del DSE 1930 en el enclave de origen, por ejemplo, mediante la certificación del 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 los datos confidenciales a través de un canal de comunicación seguro al DSE 1932, junto con una política de

codificación para la codificación por el DSE 1932. El DSE 1932 puede entonces codificar los datos del enclave 1912 en sí mismo y almacenar los datos codificados en un almacenamiento no seguro. Posteriormente, el enclave de destino 1952 puede solicitar los datos previamente codificados. Para desbloquear los datos, el DSE 1932 puede autenticar el enclave de destino 1952, por ejemplo, mediante la certificación a través de la red 1904, y verificar que se permita la decodificación para el enclave de destino 1952 de acuerdo con la política de codificación provista por el enclave de origen 1912. El DSE 1932 puede decodificar los datos previamente codificados del enclave de origen 1912, y después enviar los datos decodificados a través de un canal de comunicación seguro al enclave de destino 1952. Los datos de enclave pueden comunicarse de manera segura hacia y desde el DSE 1932 mediante el cifrado de 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 pueden estar cifrados con una clave generada durante la certificación de DSE 1932 en el enclave de origen 1912, y los datos enviados a través de la red 1904 pueden cifrarse con una clave generada durante la certificación del enclave de destino 1952 en el DSE 1932. Otros canales de comunicación segura son posibles, como el 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 la codificación y decodificación distribuidas pueden ser o no identidades de enclave abstractas. Por ejemplo, en algunas modalidades con una capa de plataforma de abstracción, una política de codificación, tal como una especificada por un enclave de origen y aplicada por un DSE, puede identificarse las identidades de enclave de decodificación permitidas donde las identidades de enclave de decodificación permitidas son, por ejemplo, una lista de identidades de enclave abstractas, o una lista de tipos de identidad abstracta en combinación con los valores de identidad abstracta del enclave de origen. En otras situaciones puede utilizarse una identidad no abstracta. Por ejemplo, en algunas modalidades, un DSE puede implementarse con un código disponible públicamente, de manera que la confianza en el DSE es la confianza en el conocimiento de su código en lugar de confiar en el autor de su código. En este ejemplo, la certificación de un DSE puede ser un valor *hash* firmado de todo el código público del DSE, y la entrada a la función *hash* puede no incluir los valores de identidad abstracta asignados por parte del 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, incluso si las plataformas nativas 1916, 1936, 1956 conforman la misma versión de la misma arquitectura de plataforma de enclave nativa. En otras modalidades, las plataformas nativas 1916, 1936, 1956 pueden adaptarse 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 codificación puede facilitar el alojamiento de enclaves de origen y destino en diferentes arquitecturas de plataforma nativa.

En otras modalidades de la codificación de datos distribuida no representadas en la figura 19, puede que no existan tres computadoras separadas (como computadoras separadas 1910, 1930, 1950). Por ejemplo, los enclaves de origen y destino pueden estar en una computadora (y quizás en una sola plataforma nativa), mientras el DSE está en una computadora separada. Alternativamente, un DSE puede

estar alojado en la misma computadora ya sea en la computadora que aloja el enclave de origen o en la computadora que aloja el enclave de destino. En estas modalidades de codificación de datos distribuida, la codificación y la decodificación de datos no es completamente local en una sola computadora, tal como se describió anteriormente con respecto a las acciones básicas de abstracción `EnclaveSeal` y `EnclaveUnseal`.

La codificación de datos distribuida puede implementarse en una API de capa de abstracción, como mediante las plataformas de abstracción 1914, 1934, 1954. Por ejemplo, las acciones básicas `DistributedEnclaveSeal` y `DistributedEnclaveUnseal` son similares a las acciones básicas locales de codificación de datos `EnclaveSeal` y `EnclaveUnseal` abordadas 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` al considerar un parámetro `SetOfTargetEnclaves` adicional, que permite que un enclave invocador, como el enclave 1910, especifique un conjunto de identidades de enclave que están autorizados para decodificar los datos proporcionados a través del parámetro `pPlaintext`. Si no se proporcionan identidades a través de `SetOfTargetEnclaves`, puede suponerse que una identidad de enclave autorizada por defecto es una identidad del enclave de codificación, por ejemplo, `ExactHash` o `InstanceHash` del enclave de codificación.

La implementación de `DistributedEnclaveSeal`, por ejemplo, como un método de la plataforma de abstracción 1914 en la computadora del enclave de origen, puede incluir el establecimiento de un canal de comunicación seguro con un DSE, como el mensaje cifrado en la red 1902. Las claves para este cifrado pueden, por ejemplo, generarse durante un proceso de certificación, tal como se describió anteriormente, o puede ser cualquier clave pública asociada con el DSE 1932.

`DistributedEnclaveSeal` puede generalizarse adicionalmente al considerar un parámetro adicional `KeyForData` (no mostrado en el prototipo de función `DistributedEnclaveSeal` anterior). En algunas modalidades, múltiples conjuntos de datos pueden mantenerse codificados simultáneamente para un solo enclave o una sola identidad de enclave. En este caso, `KeyForData` permite especificar qué conjunto de datos se codifica. `KeyForData` puede ser cualquier tipo de identificador de datos, como una cadena, un número o un conjunto de propiedades. En algunas modalidades, `KeyForData` puede ser un parámetro de entrada para `DistributedEnclaveSeal` y generado por el enclave de codificación, lo cual permite que el

enclave de codificación asigne un nombre al conjunto de datos. En otras modalidades, KeyForData puede ser un parámetro de salida, donde el DSE genera el identificador KeyForData a medida que se codifican los datos.

```
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
    Key KeyForData,
    EnclaveIdentity Identity
)
```

DistributedEnclaveUnseal puede implementarse 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 para DSE 1932, por ejemplo, pero cifrando mensajes con una clave generada durante la certificación del enclave de destino 1952 para el DSE 1932, o cifrando los mensajes enviados al enclave de destino con una clave pública del enclave de destino. El DSE puede verificar una identidad del enclave solicitante (destino), por ejemplo, mediante la certificación, decodificar los datos codificados solicitados y enviar de forma segura los datos decodificados al enclave solicitante. En las modalidades en las cuales el enclave solicitante tiene múltiples identidades, puede especificarse una identidad particular en el parámetro de identidad. En las modalidades en las cuales se almacenan múltiples conjuntos de datos de enclave para una sola identidad de enclave, el parámetro KeyForData puede especificar qué conjunto de datos codificados (para la identidad especificada) se solicita utilizando el mismo valor de KeyForData utilizado en DistributedEnclaveSeal cuando se codificó el conjunto de datos.

En algunas modalidades, las identidades de los enclaves autorizados para decodificar los datos pueden especificarse (como en el parámetro SetOfTargetEnclaves) mediante claves públicas de los enclaves objetivo autorizados. En esta modalidad, la certificación del enclave de destino para el DSE puede no ser necesaria, pero los datos decodificados solo pueden proporcionarse como cifrados utilizando una de las claves públicas especificadas. Suponiendo que solo los enclaves seleccionados tengan acceso a las claves privadas correspondientes para descifrar, solo los enclaves seleccionados tendrán acceso a los datos decodificados.

En las modalidades no representadas en la figura 19, las funciones del enclave de codificación distribuida (DSE, por sus siglas en inglés) 1932 pueden distribuirse a través de múltiples DSE. Por ejemplo, la funcionalidad de DSE puede distribuirse a través de múltiples DSE en múltiples computadoras por redundancia y tolerancia a fallas. En este ejemplo, cualquier DSE replicado puede ser capaz de

atender una solicitud de codificación o decodificación. Cabe señalar que los datos codificados 1938, una vez que se codifican/cifran, pueden almacenarse de forma segura en cualquier lugar, que incluye la replicación en servidores de almacenamiento en la nube.

La codificación de datos distribuida puede permitir el movimiento de las cargas de trabajo de enclave entre computadoras. Por ejemplo, los datos de enclave de origen codificados por un DSE pueden ser datos de estado del enclave de origen en un primer servidor en la nube, que pueden cargarse en el enclave de destino en un segundo servidor en la nube después de decodificarse. Esto puede realizarse de manera similar a como se describió anteriormente con respecto a la figura 18. Una operación segura puede iniciar la ejecución en el enclave de origen. Posteriormente, tal vez después de que se interrumpa la ejecución en el enclave de origen, el estado del enclave de origen puede codificarse en un DSE y después decodificarse en un enclave de destino cuando el enclave de destino esté listo para continuar la operación segura que se inició en el enclave origen.

La figura 20 es un diagrama de flujo ejemplar para la codificación y decodificación distribuidas de datos, como podría realizarse por un enclave de codificación o DSE. Los bloques 2002-2006 corresponden a la codificación distribuida de datos, mientras los bloques 2008-2010 corresponden a la codificación de datos distribuida. En respuesta a una solicitud para codificar un conjunto de datos de enclave que se origina en un enclave de origen, el enclave de codificación (o DSE) puede certificar el enclave de origen, al enviar un informe de certificación o una cita al enclave de origen en el bloque 2002. El enclave de origen puede verificar la identidad del enclave de codificación como un enclave de codificación genuino y confiable, mediante la inspección de un valor de identidad y su firma en el informe de certificación del enclave de codificación. En el bloque 2004, el enclave de codificación recibe una lista permitida y los datos de enclave que se codificarán. Estos pueden recibirse a través de un canal seguro tal como se describió anteriormente con respecto a la figura 19. En el bloque opcional 2006, el enclave de codificación puede codificar los datos del enclave de origen en sí mismo, por ejemplo, si los datos se almacenan fuera del contenedor seguro del enclave de codificación, como en un sistema de archivos de computadora. Para decodificar los datos de un enclave de destino, el enclave de destino puede certificar el enclave de codificación, por ejemplo, al proporcionar un informe de certificación o cita, en el bloque 2008. En el bloque 2010, puede verificarse una identidad del enclave de destino, como al inspeccionar el informe de certificación del enclave de destino. En el bloque 2012, el enclave de codificación determina si el enclave de destino tiene permiso para decodificar los datos del enclave de origen al verificar que se incluyó una identidad autenticada del enclave de destino en la lista permitida recibida con los datos. Una vez que se confirmó el permiso, los datos de enclave pueden decodificarse si se codificaron y después enviarse al enclave de destino a través de un canal seguro en el bloque 2014.

Enclave de depósito de claves

Los depósitos de claves pueden implementarse con los enclaves. Un depósito de claves guarda de forma segura las claves, como las claves de un sistema de cifrado para cifrar y descifrar los datos, para los clientes del depósito de claves. Un depósito de claves puede además realizar operaciones con la clave, como cifrar y descifrar los datos, firmar datos y obtener nuevas claves a partir de una clave

existente. Un depósito de claves, cuando se implementa como un enclave, puede proporcionar almacenamiento y procesamiento muy seguros con claves de cifrado secretas. Además, la certificación de *software* de un enclave de depósito de claves puede proporcionar altos niveles de seguridad a un cliente de depósito de claves que se está comunicando con un depósito de claves auténtica y confiable. Los sistemas altamente seguros pueden construirse en un enclave de depósito de claves con una clave bloqueada en el depósito, por lo cual una clave almacenada dentro de un depósito de claves nunca se libera a ningún cliente fuera del depósito de claves, y en algunos casos, la clave bloqueada en el depósito solo puede existir como almacenada dentro (o posiblemente codificada) del enclave de depósito de claves.

La figura 21 es un diagrama de bloques de un ejemplo de enclave de depósito de claves. El enclave 2122 es un depósito 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 depósito de claves 2122 también es un enclave y está alojado dentro de un contenedor de enclave seguro de la primera plataforma de enclave nativa 2116. Los enclaves 2112, 2122 pueden interactuar con sus respectivas plataformas nativas 2116, 2126 a través de las respectivas plataformas de abstracción de enclave 2114, 2124. En otras modalidades, una o ambas plataformas de abstracción 2114, 2124 pueden no existir donde los enclaves 2112 y/o 2122 interactúan directamente con las plataformas nativas 2116, 2126.

El enclave de depósito de claves 2122 puede comunicarse con el cliente de depósito de claves 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 seguridad de confidencialidad, integridad y/o actualización de los mensajes enviados a través del canal de comunicaciones 2150. La confidencialidad y la integridad de tal canal de comunicaciones seguras pueden establecerse, por ejemplo, con cifrado y firmas, como en las figuras 2 y 3, utilizando las claves compartidas generadas como parte de un proceso de certificación, como en las figuras 5 y 6.

La certificación de *software* proporciona seguridad en parte al garantizar la identidad de la entidad en el otro tamaño de un canal de comunicación. Al certificar el enclave de depósito de claves 2122 para un cliente de depósito de claves, el cliente puede obtener la seguridad de que el enclave de depósito de claves 2122 es el cual se afirma antes de enviar un elemento secreto, como una clave u otros datos en texto sin formato, al depósito de claves. Lo contrario también se aplica para los clientes que también son enclaves, tal como se muestra en la figura 21. Al certificar el enclave de depósito de claves 2112 en un enclave de depósito de claves 2122, el depósito de claves puede obtener la seguridad de que el cliente el cual se afirma antes de revelar un elemento secreto, como una clave u otros datos en texto sin formato, al cliente.

Los sistemas de depósito de claves con claves bloqueadas en el depósito y claves obtenidas, particularmente cuando las claves de cifrado se obtienen a partir de una clave bloqueada en el depósito, pueden utilizarse para generar un sistema de seguridad que sea adaptable y seguro. Una función de obtención de claves, que puede o no ser pública, puede utilizarse para generar múltiples claves a partir

de una primera clave. La primera clave (un elemento secreto original) puede estar bloqueada en el depósito para el nivel superior de seguridad, y las claves obtenidas a partir de la primera clave pueden utilizarse con fines de cifrado. Si se compromete una clave derivada, puede generarse una nueva clave obtenida en un sistema existente sin tener que acceder o cambiar el depósito de claves que contiene la primera clave.

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

Un cliente de depósito de claves puede interactuar con un sistema de recuperación de claves utilizando las siguientes acciones básicas ejemplares. Un ejemplo de prototipo de la función StoreKey es:

StoreKey ([in] Keyname, [in] Key Type, [in] Key Value, [in] Policy)

StoreKey almacena una clave dada en el depósito de claves y la asocia con un nombre dado. El tipo de clave también está asociado con la clave y restringe lo que puede realizarse con la clave. Por ejemplo, algunas claves solo deben utilizarse para cifrado, otras para firmas, etcétera. Y las claves específicas solo pueden utilizarse con algoritmos criptográficos específicos. La política puede especificar políticas para restringir aún más el uso de la clave. Por ejemplo, puede especificar un conjunto de identidades de enclave que tienen permitido recuperar la clave y/o utilizar la clave. También puede especificarse propiedades temporales, por ejemplo, que la clave debe destruirse en una fecha determinada, o que la velocidad de las operaciones que utilizan la clave debe estar limitada.

Un prototipo de función GenerateKey ejemplar es:

GenerateKey ([in] keyName, [in] key Type, [in] Policy)

GenerateKey genera una nueva clave de un cierto tipo y la mantiene almacenada dentro del depósito de claves, es decir, la clave nunca sale del depósito de claves.

Un prototipo de función GetKey ejemplar es:

GetKey ([in] KeyName, [out] Key Value)

GetKey obtiene una clave almacenada dentro del depósito de claves. Estas acciones básicas se implementan normalmente a través de un canal de comunicación seguro y el código que invoca la acción básica normalmente se ejecuta en un entorno confiable. En tal contexto, a menudo es aceptable recuperar una clave en el depósito de claves.

Un prototipo de función DeleteKey ejemplar es:

DeleteKey ([in] keyName)

DeleteKey elimina una clave en el depósito de claves.

Un prototipo de función DeriveKey ejemplar es:

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

DeriveKey utiliza una función de obtención de clave criptográfica (KDF, por sus siglas en inglés) identificada por `kdfIdentifier` para obtener una nueva clave con base en la clave identificada por `keyName` y los datos transmitidos 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 las acciones básicas de depósito de claves anteriores pueden implementarse al establecer un canal seguro con el KVE. El canal puede establecerse utilizando la certificación y realizando un intercambio de claves Diffie-Hellman tal como se describió anteriormente con respecto a las figuras 5 y 6. Una vez establecido el canal de comunicación, la solicitud se envía de forma 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 el KVE, genera un par de claves públicas/privadas 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 pueden distribuirse a todos los sistemas/códigos que deseen utilizar el depósito de claves. En este caso, la implementación de las acciones básicas anteriores verifica la cotización para asegurarse de que está tratando con un KVE genuino, y después cifra las solicitudes utilizando la clave pública del KVE. Como parte de la solicitud, la implementación de las acciones 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 bidireccional seguro sin certificación.

La figura 22 es un diagrama de flujo ejemplar para algunas operaciones de enclave de depósito de claves. El proceso 2200 inicia el buzón de entrada 2202 al almacenar de manera segura, dentro del enclave de depósito de claves, una clave utilizada en un sistema de cifrado. La clave puede utilizarse, por ejemplo, para cifrar o descifrar los datos, generar una firma criptográfica, o solo puede utilizarse como una clave original a partir de la cual se obtienen otras claves. La clave puede almacenarse de forma segura en el enclave de depósito de claves, por ejemplo, al almacenar la clave dentro del espacio de

memoria del contenedor seguro del enclave. En otras modalidades, la clave puede mantener segura fuera del contenedor de enclave seguro al codificar los datos de clave en el enclave de depósito de claves, o puede mantenerse segura al codificarse de forma remota con un enclave de codificación distribuida tal como se describe con respecto a las figuras 19 y 20.

En el bloque 2204, el enclave de depósito de claves realiza un proceso de certificación para certificar la identidad del enclave de depósito de claves para el cliente de depósito de claves. Esto puede brindar al cliente la seguridad de que el depósito de claves no es apócrifo y puede confiarle los elementos secretos como la clave o los datos que se cifrarán. La certificación del enclave de depósito de claves puede incluir el envío, al cliente de depósito de claves, un informe de certificación o una cita de certificación del enclave de depósito de claves. El cliente de depósito de claves puede entonces verificar la integridad del informe de certificación al verificar una firma en el informe de certificación con una clave pública asociada con la plataforma de enclave nativa del enclave de depósito de claves. Por ejemplo, el informe de certificación del depósito de claves 2122 puede generarse por la segunda plataforma nativa 2126, y el cliente de depósito de claves 2112 puede verificar la firma en el informe utilizando una clave pública asociada con la segunda plataforma nativa 2126. Este proceso de certificación también puede generar claves utilizadas para un canal de comunicación seguro entre el cliente de depósito de claves y el enclave de depósito de claves, por ejemplo, tal como se muestra en las figuras 5 y 6. El informe de certificación puede incluir una identidad del enclave de depósito de claves que puede determinarse de varias maneras tal como se describió anteriormente, por ejemplo, con respecto a las figuras 14 y 15. La identidad puede, por ejemplo, basarse en un valor *hash* de todo el contenido del contenedor seguro del enclave de depósito de claves, un valor *hash* de solo un identificador único asignado por parte del autor/creador del enclave de depósito de claves, o un valor *hash* de una combinación de una porción del contenido del contenedor y un identificador único.

Algunas operaciones de enclave de depósito de claves también pueden requerir la seguridad de la identidad del cliente del depósito de claves. Por ejemplo, descifrar datos o divulgar una clave (como con las acciones básicas Decrypt o GetKey) puede requerir tal seguridad. En estas situaciones, si un cliente de depósito de claves también es un enclave, el bloque opcional 2208 incluye un proceso de certificación para verificar, mediante el enclave de depósito de claves, la identidad del cliente de depósito de claves. El proceso de certificación del bloque 2208 puede incluir recibir, en el enclave de depósito de claves, un informe de certificación o una cita del cliente de depósito de claves.

En el bloque opcional 2210, puede establecerse un canal de comunicaciones seguro entre el depósito de claves y el enclave de depósito de claves. La comunicación segura puede requerirse para transmitir los elementos secretos entre el cliente de depósito de claves y el enclave de depósito de claves, como las claves o los datos que se cifrarán. El proceso de certificación del bloque 2004 o 2008 puede generar claves que pueden utilizarse para crear un canal de comunicación seguro entre el cliente de depósito de claves y el enclave de depósito de claves, por ejemplo, tal como se muestra en las figuras 5 y 6. Alternativamente, cualquier clave pública conocida del destino de un mensaje puede utilizarse para enviar un mensaje de forma segura.

En el bloque 2212, puede realizarse una operación para las claves, como una de las acciones básicas de depósito de claves descritas anteriormente, dentro del enclave de depósito de claves. Durante esta operación, los datos de claves pueden almacenarse solo en el espacio de direcciones del contenedor seguro del enclave de depósito de claves. Los ejemplos de acciones básicas incluyen DeriveKey, Decrypt, Sign y otras.

El proceso 2200 supone que un enclave de depósito de claves ya conoce la clave. Cabe señalar que, para algunas operaciones o acciones básicas del enclave de depósito de claves, como StoreKey o GenerateKey, el orden de las operaciones puede ser diferente del cual se muestra en el proceso 2200. Por ejemplo, para GenerateKey, se producirá la operación de generación de clave (como en el bloque 2212) antes de la operación de almacenamiento seguro del bloque 2202. Tal orden de operaciones se representa en la figura 23, bloques 2302-2308.

La figura 23 es un diagrama de flujo ejemplar para crear y utilizar un enclave de depósito de claves con una clave bloqueada en el depósito. En los bloques 2302-2308 del proceso 2300, se obtiene una nueva clave dentro del enclave de depósito de claves. En los bloques 2310-2316, la nueva clave obtenida se utiliza para realizar una operación de descifrado. Este es un ejemplo de uso de la clave bloqueada en el depósito de claves, en el cual todas las operaciones de clave se realizan con el enclave de depósito de claves y la clave nunca se proporciona a un cliente de depósito de claves. Además, es posible que la nueva clave en este ejemplo nunca exista fuera del enclave de depósito de claves, ya que se creó (obtuvo) desde el propio enclave de depósito de claves, y nunca se proporcionó al enclave de depósito de claves desde un cliente de depósito de claves o en otro lugar. Para algunas modalidades y políticas de uso de claves, una clave bloqueada en la bóveda puede ser efímera en el sentido de que nunca abandona el contenedor seguro del enclave de depósito de claves, incluso después de codificar la clave en el enclave de depósito de claves. Tal clave efímera, como lo que puede ocurrir con una clave obtenida que se utiliza para proteger temporalmente un canal de comunicaciones, puede dejar de existir en cualquier lugar cuando el contenedor del enclave de depósito de claves se destruya o termine. Aunque el proceso de la figura 23 ilustra cómo puede utilizarse una clave bloqueada en el depósito, el proceso de la figura 23 también puede utilizarse con una clave que no esté bloqueada en el depósito, por ejemplo, si la política de uso de claves permitió que la clave regresara al cliente que solicitó su creación.

En el bloque 2302, el enclave de depósito de claves se certifica para el cliente de depósito de claves. Esto puede requerirse por el cliente debido a que el cliente proporcionará un elemento secreto para cifrarse en el bloque 2312. En el bloque 2304, el enclave de depósito de claves puede recibir, por ejemplo, del cliente de depósito de claves, una indicación de una política de uso de claves. La indicación puede ser, por ejemplo, una estructura de datos que especifique la política, o puede ser un identificador que se utilizará con un registro de políticas de uso de claves. La política de uso de claves en sí puede indicar que esta clave nunca debe proporcionarse a ningún cliente de almacenamiento. En el bloque 2306, una nueva clave se obtiene a partir de una clave original conocida anteriormente, por ejemplo, con la acción básica DenveKey descrita anteriormente. El enclave de depósito de claves puede recibir una solicitud (no mostrada) para obtener la nueva clave, por ejemplo, desde el cliente de depósito de claves.

En el bloque 2308, la clave recientemente obtenida puede almacenarse de manera segura de acuerdo con la política de uso de claves recibida.

El cliente de depósito de claves puede certificar al enclave de depósito de claves en el bloque 2310. Un proceso de certificación puede incluir recibir, en el enclave de depósito de claves, un informe de certificación o una cita del cliente de depósito de claves. La política de uso de claves recibida puede restringir algunos o todos los usos de la nueva clave a las solicitudes de los solicitantes que se autentican a través de la certificación de *software*. En los bloques 2312-2316, una operación de descifrado, como para la acción básica Decrypt anterior, se realiza utilizando la clave obtenida en el bloque 2306. En otras modalidades, otras operaciones pueden realizarse con una clave bloqueada en el depósito, como el cifrado, firma, verificación de una firma y obtención de otra nueva clave a partir de la clave obtenida en el bloque 2306 (obtener una clave de segunda generación a partir de la clave original). En el bloque 2312, se recibe una memoria tampón de datos cifrados desde el cliente de depósito de claves. Los datos cifrados recibidos se descifran con la clave obtenida en el bloque 1314, y los datos descifrados resultantes (en una memoria tampón de datos descifrados) se envían al cliente de depósito de claves a través del canal de comunicaciones seguro en el bloque 2316.

En una modalidad, un método para determinar la equivalencia de enclave comprende:

- recibir un primer informe de certificación a partir de un primer enclave instanciado en una primera plataforma de enclave nativa;

- extraer un primer valor de identidad de enclave asociado con un tipo de identidad de enclave a partir del primer informe de certificación;

- recibir un segundo informe de certificación a partir de un segundo enclave instanciado en una segunda plataforma de enclave nativa, donde la primera plataforma de enclave nativa y la segunda plataforma de enclave nativa son diferentes;

- extraer un segundo valor de identidad de enclave asociado con el tipo de identidad de enclave del segundo informe de certificación;

- determinar que el primer valor de identidad de enclave es el mismo que el segundo valor de identidad de enclave; y

- realizar, con base en la determinación, una operación de computación al completar parcialmente la operación de computación en el primer enclave y al completar parcialmente la operación de computación en el segundo enclave.

En una modalidad, la operación de computación se realiza en serie con el primer y segundo enclave, y además, comprende:

- codificar los datos de estado de enclave del primer enclave después de completar parcialmente la operación de computación en el primer enclave para producir los datos codificados, donde los datos de estado se codifican para el tipo de identidad de enclave;

- copiar los datos codificados desde el primer enclave al segundo enclave; y

- decodificar los datos codificados en el segundo enclave.

En una modalidad, la operación de computación se realiza simultáneamente con el primer y

segundo enclave, y además, comprende:

- copiar una primera porción de un conjunto de datos en el primer enclave;

- copiar una segunda porción de un conjunto de datos en el segundo enclave; y

donde la operación de computación se completa parcialmente en el primer enclave utilizando la primera porción y se completa parcialmente en el segundo enclave utilizando la segunda porción.

En una modalidad, completar parcialmente la operación de computación en el primer enclave incluye la lectura, mediante el primer enclave, de un contador monotónico asociado con el valor de identidad; y

completar parcialmente la operación de computación en el segundo enclave incluye la lectura, mediante el segundo enclave, del contador monotónico asociado con el valor de identidad.

En una modalidad, completar parcialmente la operación de computación en el primer enclave incluye la lectura, por el primer enclave, de un tiempo confiable asociado con el valor de identidad; y

completar parcialmente la operación de computación en el segundo enclave incluye la lectura, mediante el segundo enclave, del tiempo confiable asociado con el valor de identidad.

En una modalidad, el método comprende, además:

crear una firma para el primer informe de certificación al realizar al menos una primera operación de plataforma nativa; y

crear una firma para el segundo informe de certificación al realizar al menos una segunda operación de plataforma nativa.

En una modalidad, la primera plataforma de enclave nativa se adapta a una arquitectura de enclave de extensiones de protección de *software* (SGX, por sus siglas en inglés) de Intel y la segunda plataforma de enclave nativa se adapta a una arquitectura de enclave de modo seguro virtual (VSM, por sus siglas en inglés) de Microsoft.

En una modalidad, el método comprende, además:

- generar el primer informe de certificación al realizar al menos una operación nativa de SGX; y

- generar el segundo informe de certificación al realizar al menos una operación nativa de VSM.

En una modalidad, la primera plataforma de enclave nativa y la segunda plataforma de enclave nativa son versiones diferentes de la misma arquitectura de enclave.

En una modalidad, el método comprende, además:

verificar una primera firma del primer informe de certificación con una clave pública asociada con la primera plataforma de enclave nativa; y

verificar una segunda firma del segundo informe de certificación con una clave pública asociada con la segunda plataforma de enclave nativa, donde la primera clave pública y la segunda clave pública son diferentes.

En una modalidad, el método comprende, además:

verificar un primer certificado de aprobación de la primera plataforma nativa con una primera clave pública asociada con un fabricante de la primera plataforma de enclave nativa; y

verificar un segundo certificado de aprobación de la segunda plataforma de enclave nativa con

una segunda clave pública asociada con el fabricante de la segunda plataforma de enclave nativa, donde la primera clave pública y la segunda clave pública no son las mismas.

En una modalidad, el método comprende, además:

generar el primer informe de certificación al computar un valor *hash* con respecto a una primera porción de la memoria en un contenedor seguro del primer enclave, donde el contenido de la primera porción de la memoria es independiente de la primera plataforma nativa; y

generar el segundo informe de certificación al computar un valor *hash* con respecto a una segunda porción de memoria en un contenedor seguro del segundo enclave, donde el contenido de la segunda porción de memoria es independiente de la segunda plataforma nativa.

En una modalidad, un sistema comprende al menos un procesador y una memoria que almacena en la misma las instrucciones que, cuando se ejecutan por el sistema, producen al menos las etapas de:

recibir un primer informe de certificación a partir de un primer enclave instanciado en una primera plataforma de enclave nativa;

extraer un primer valor de identidad de enclave asociado con un tipo de identidad de enclave a partir del primer informe de certificación;

recibir un segundo informe de certificación a partir de un segundo enclave instanciado en una segunda plataforma de enclave nativa, donde la primera plataforma de enclave nativa y la segunda plataforma de enclave nativa son diferentes;

extraer un segundo valor de identidad de enclave asociado con el tipo de identidad de enclave a partir del segundo informe de certificación;

comparar el primer valor de identidad de enclave con el segundo valor de identidad de enclave; y

realizar una operación de computación al completar parcialmente la operación de computación en el primer enclave y al completar parcialmente la operación de computación en el segundo enclave.

En una modalidad, la operación de computación se realiza en serie con el primer y segundo enclave, y donde las instrucciones producen, además, al menos las etapas de:

codificar los datos de estado de enclave del primer enclave después de completar parcialmente la operación de computación en el primer enclave para producir los datos codificados, donde los datos de estado se codifican para el tipo de identidad de enclave;

copiar los datos codificados desde el primer enclave al segundo enclave; y

decodificar los datos codificados en el segundo enclave.

En una modalidad, la operación de computación se realiza simultáneamente con el primer y segundo enclave, y donde las instrucciones producen, además, al menos las etapas de:

copiar una primera porción de un conjunto de datos en el primer enclave;

copiar una segunda porción de un conjunto de datos en el segundo enclave; y

donde la operación de computación se completa parcialmente en el primer enclave utilizando la primera porción y se completa parcialmente en el segundo enclave utilizando la segunda porción.

En una modalidad, completar parcialmente la operación de computación en el primer enclave incluye la lectura, por el primer enclave, de un contador monotónico asociado con el valor de identidad; y

completar parcialmente la operación informática en el segundo enclave incluye la lectura, por el segundo enclave, del contador monotónico asociado con el valor de identidad.

En una modalidad, completar parcialmente la operación de computación en el primer enclave incluye la lectura, por el primer enclave, de un tiempo confiable asociado con el valor de identidad; y

completar parcialmente la operación de computación en el segundo enclave incluye la lectura, por el segundo enclave, del tiempo de confianza asociado con el valor de identidad.

En una modalidad, las instrucciones producen, además, al menos las etapas de:

crear una firma para el primer informe de certificación al realizar al menos una primera operación de plataforma nativa; y

crear una firma para el segundo informe de certificación al realizar al menos una segunda operación de plataforma nativa.

En una modalidad, la primera plataforma de enclave nativa se adapta a una arquitectura de enclave de extensiones de protección de *software* (SGX, por sus siglas en inglés) de Intel y la segunda plataforma de enclave nativa se adapta a una arquitectura de enclave de modo seguro virtual (VSM, por sus siglas en inglés) de Microsoft, y donde las instrucciones producen, al menos las etapas de:

generar el primer informe de certificación al realizar al menos una operación nativa de SGX; y

generar el segundo informe de certificación al realizar al menos una operación nativa de VSM.

En una modalidad, la primera plataforma de enclave nativa y la segunda plataforma de enclave nativa son versiones diferentes de la misma arquitectura de enclave.

En una modalidad, las instrucciones producen, además, al menos las etapas de:

verificar una primera firma del primer informe de certificación con una clave pública asociada con la primera plataforma de enclave nativa; y

verificar una segunda firma del segundo informe de certificación con una clave pública asociada con la segunda plataforma de enclave nativa, donde la primera clave pública y la segunda clave pública son diferentes.

En una modalidad, las instrucciones producen, además, al menos las etapas de:

verificar un primer certificado de aprobación de la primera plataforma nativa con una primera clave pública asociada con un fabricante de la primera plataforma de enclave nativa; y

verificar un segundo certificado de aprobación de la segunda plataforma de enclave nativa con una segunda clave pública asociada con el fabricante de la segunda plataforma de enclave nativa, donde la primera clave pública y la segunda clave pública no son las mismas.

En una modalidad, las instrucciones producen, además, al menos las etapas de:

generar el primer informe de certificación al computar un valor *hash* con respecto a una primera porción de memoria en un contenedor seguro del primer enclave, donde el contenido de la primera porción de memoria es independiente de la primera plataforma nativa; y

generar el segundo informe de certificación al computar un valor *hash* con respecto a una segunda porción de memoria en un contenedor seguro del segundo enclave, donde el contenido de la segunda porción de memoria es independiente de la segunda plataforma nativa.

En una modalidad, un sistema comprende:

medios para recibir un primer informe de certificación desde un primer enclave instanciado en una primera plataforma de enclave nativa;

medios para extraer un primer valor de identidad de enclave asociado con un tipo de identidad de enclave a partir del primer informe de certificación;

medios para recibir un segundo informe de certificación desde un segundo enclave instanciado en una segunda plataforma de enclave nativa, donde la primera plataforma de enclave nativa y la segunda plataforma de enclave nativa son diferentes;

medios para extraer un segundo valor de identidad de enclave asociado con el tipo de identidad de enclave a partir del segundo informe de certificación;

medios para comparar el primer valor de identidad de enclave con el segundo valor de identidad de enclave; y

medios para realizar una operación de computación al completar parcialmente la operación de computación en el primer enclave y al completar parcialmente la operación de computación en el segundo enclave.

Cada uno de los procesos, métodos y algoritmos descritos en las secciones anteriores puede incorporarse y estar automatizado de manera completa o parcial por módulos de código ejecutados por una o más computadoras o procesadores informáticos. Los módulos de código pueden almacenarse en cualquier tipo de medio permanente legible por computadora o dispositivo de almacenamiento informático, como discos duros, memoria de estado sólido, disco óptico y/o similares. Los procesos y algoritmos pueden implementarse de manera parcial o completa en circuitería de aplicación específica. Los resultados de los procesos descritos y las etapas del proceso pueden almacenarse, de forma persistente o de otro modo, en cualquier tipo de almacenamiento informático permanente, como el almacenamiento volátil o no volátil. Las diversas características y procesos descritos anteriormente pueden utilizarse de forma independiente, o pueden combinarse de varias maneras. Se pretende que todas las combinaciones y subcombinaciones posibles estén dentro del alcance de esta divulgación. Además, ciertos métodos o bloques de procesos pueden omitirse en algunas implementaciones. Los métodos y procesos descritos en el presente documento tampoco están limitados a ninguna secuencia particular, y los bloques o estados relacionados con los mismos pueden realizarse en otras secuencias que sean apropiadas. Por ejemplo, los bloques o estados descritos pueden realizarse en un orden diferente al descrito específicamente, o pueden combinarse múltiples bloques o estados en un solo bloque o estado. Los bloques o estados ejemplares pueden realizarse en serie, simultáneamente o de alguna otra manera. Pueden añadirse o eliminarse bloques o estados de las modalidades ejemplares divulgadas. Los sistemas y componentes ejemplares descritos en el presente documento pueden configurarse de manera diferente a la descrita. Por ejemplo, los elementos pueden agregarse, eliminarse o reorganizarse en comparación con las modalidades ejemplares divulgadas.

También se apreciará que se ilustran diversos elementos almacenados en la memoria o en el almacenamiento mientras se utilizan, y que estos elementos o porciones de los mismos pueden

transferirse entre la memoria y otros dispositivos de almacenamiento con el fin de gestionar la memoria y la integridad de los datos. Alternativamente, en otras modalidades, algunos o todos los módulos de *software* y/o sistemas pueden ejecutarse en la memoria de otro dispositivo y comunicarse con los sistemas informáticos ilustrados a través de la comunicación entre computadoras. Además, en algunas modalidades, algunos o todos los sistemas y/o módulos pueden implementarse o proporcionarse de otras maneras, como al menos parcialmente en *firmware* y/o *hardware*, que incluyen, entre otros, uno o más circuitos integrados de aplicación específica (ASIC, por sus siglas en inglés), circuitos integrados estándar, controladores (por ejemplo, mediante la ejecución de las instrucciones apropiadas, y que incluyen microcontroladores y/o controladores integrados), matriz de puertas programable *in situ* (FPGA, por sus siglas en inglés), dispositivos lógicos programables complejos (CPLD, por sus siglas en inglés), etcétera. Algunos o todos los módulos, sistemas y estructuras de datos también pueden almacenarse (por ejemplo, como instrucciones de *software* o datos estructurados) en un medio legible por computadora, como un disco duro, memoria, red o artículo de soporte portátil que se leerá por una unidad apropiada o a través de una conexión apropiada. Para los propósitos de esta especificación y reivindicaciones, la frase “medio de almacenamiento legible por computadora” y variaciones de la misma, no incluye ondas, señales y/u otros medios temporales y/o intangibles. Los sistemas, módulos y estructuras de datos también pueden transmitirse como señales de datos generadas (por ejemplo, como parte de una onda portadora u otra señal de propagación analógica o digital) en una variedad de medios de transmisión legibles por computadora, que incluyen medios inalámbricos y alámbricos/por cable, y pueden adoptar una variedad de formas (por ejemplo, como parte de una señal analógica única o multiplexada, o como múltiples paquetes o estructuras digitales discretas). Tales productos de programas informáticos también pueden adoptar otras formas en otras modalidades. Por consiguiente, la presente divulgación puede ponerse en práctica con otras configuraciones de sistemas informáticos.

El lenguaje condicional utilizado en el presente documento, como, entre otros, “poder”, “podría”, “pudo”, “puede”, “por ejemplo” y similares, a menos que se indique lo contrario, o se entienda de otra manera dentro del contexto tal como se utiliza, en general, pretende dar a entender que ciertas modalidades incluyen, mientras otras modalidades no incluyen, ciertas características, elementos y/o etapas. Por lo tanto, tal lenguaje condicional generalmente no pretende implicar que las características, elementos y/o etapas se requieran de alguna manera para una o más modalidades o que una o más modalidades necesariamente incluyan la lógica para decidir, con o sin la aportación o indicación del autor, si estas características, elementos y/o etapas están incluidos o se realizarán en cualquier modalidad particular. Los términos “comprender”, “incluir”, “tener” y similares son sinónimos y se utilizan inclusivamente, de manera abierta, y no excluyen elementos, características, acciones, operaciones adicionales, etcétera. Además, 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 los elementos de la lista.

Aunque se han descrito ciertas modalidades ejemplares, estas modalidades se han presentado solo a modo ejemplar y no pretenden limitar el alcance de las invenciones descritas en el presente

documento. Por lo tanto, nada en la descripción anterior pretende implicar que cualquier función, característica, etapa, módulo o bloque particular sea necesario o indispensable. De hecho, los nuevos métodos y sistemas descritos en el presente documento pueden incorporarse en una variedad de otras formas; además, pueden realizarse varias omisiones, sustituciones y cambios en la forma de los métodos y sistemas descritos en el presente documento sin apartarse de la esencia de las invenciones descritas en el presente documento. Las reivindicaciones adjuntas y sus equivalentes tienen el objetivo de abarcar tales formas o modificaciones que se encontrarían dentro del alcance y esencia de algunas de las invenciones divulgadas en el presente documento.