

REPÚBLICA DE COLOMBIA
SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

Señor(a)
LLC MICROSOFT TECHNOLOGY LICENSING

TÍTULO DE LA SOLICITUD: **“APLICACIONES CRIPTOGRAFICAS PARA UN SISTEMA DE CADENA DE BLOQUES”**

NÚMERO DE SOLICITUD INTERNACIONAL: PCT/US2017/035341

FECHA DE PRESENTACIÓN INTERNACIONAL: 1 de junio de 2017.

PRIORIDAD(ES): 06/06/2016, US (ESTADOS UNIDOS DE AMÉRICA), 62/346,431
18/10/2016, US (ESTADOS UNIDOS DE AMÉRICA), 15/296,953

Como resultado del examen de patentabilidad realizado a la solicitud indicada en la referencia, atendiendo a lo consagrado en el artículo 45 de la Decisión 486 de 2000, se le comunica que esta Dirección ha encontrado que la materia objeto de solicitud no cumple con algunos requisitos establecidos en la Decisión, con fundamento en los siguientes aspectos:

1. Objeto de la invención

De acuerdo con lo establecido en el numeral 1.2.2.2 del Capítulo Primero del Título X de la Circular Única para efectos del examen de patentabilidad, cuando la solicitud contenga más de diez (10) reivindicaciones y no se haya pagado la tasa complementaria, sólo se estudiarán las cubiertas por la tasa pagada al momento de realizar el examen de patentabilidad consagrado en el artículo 45 de la Decisión 486. Por lo tanto, se estudiarán las reivindicaciones 1 a 15 del radicado N° NC2018/0013180 del 05 de diciembre de 2018.

El problema planteado en esta solicitud consiste en la necesidad de diseñar e implementar un sistema que permita garantizar la confiabilidad en la información en una cadena de transacciones.

Para resolver este problema técnico, la solicitud en estudio proporciona un método para delegar el comportamiento de un contrato inteligente asociado con una cadena de bloques al código que no forma parte de la cadena de bloques. Un sistema ordena la ejecución por una máquina virtual del contrato inteligente. Durante la ejecución del contrato inteligente, el contrato inteligente envía a un servicio de contenedores de cryptlets, a través de un cryptodelegate, una solicitud para delegar un comportamiento a un cryptlet que se ejecuta en una computadora anfitriona certificada. Durante la ejecución, el servicio de contenedores de cryptlets identifica una computadora anfitriona para ejecutar el código del cryptlet en un contenedor de cryptlets apropiado. El servicio de contenedores de cryptlets ordena a la computadora anfitriona identificada que ejecute el código del cryptlet para realizar el comportamiento delegado. Una vez realizado el comportamiento delegado, el servicio de contenedores de cryptlets recibe del cryptlet una respuesta al comportamiento solicitado. El servicio de contenedores de cryptlets envía la respuesta al contrato inteligente en la cadena de bloques que verifica la cryptodelegate (Resumen).

2. De la solicitud de patente



Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

2.1. Título

El nombre de la invención de que se trata en el artículo 27 literal d) de la Decisión 486 deberá reflejar el objeto y el campo industrial con el cual se relaciona la misma y ser concordante con la materia descrita y las reivindicaciones de la solicitud. El nombre no podrá referirse a nombres personales, marcas de productos o nombres de fantasía.

El solicitante en el escrito radicado bajo el N° NC2017/0001140 el 5 de diciembre de 2018, tituló la invención como: “APLICACIONES CRIPTOGRAFICAS PARA UN SISTEMA DE CADENA DE BLOQUES”. Este título no se relaciona con el objeto de la solicitud contenido en la descripción y las reivindicaciones, por lo cual se sugiere el título siguiente: “SISTEMA Y MÉTODO ASOCIADO PARA APLICACIONES CRIPTOGRAFICAS EN UN SISTEMA DE CADENA DE BLOQUES”.

2.2. Reivindicaciones (Art. 30 Decisión 486)

- Las reivindicaciones 1, 2, 4 a 6, 9, 10 y 13 a 15 no son claras porque le término: “*cryptodelegate*” no es conocido en el estado de la técnica. Se sugiere reemplazarlo por un término o definición más específica.
- Las reivindicaciones 1, 4, 6, 7, 12 a 15 no son claras porque le término: “*cryptlet*” no es conocido en el estado de la técnica. Se sugiere reemplazarlo por un término o definición más específica.

3. Determinación del Estado de la Técnica

La fecha para determinar el estado de la técnica es el 6 de junio de 2016 que corresponde a la fecha de presentación de la solicitud de prioridad.

Consultadas las fuentes de información con que se cuenta, se encontraron los siguientes documentos que anticipan el objeto de esta solicitud:

Ítem	Documento	Título	Fecha de Publicación
D1	Zhang (2016)	“Town Crier: An Authenticated Data Feed for Smart Contracts”	19 de febrero de 2016
D2	Xu (2016)	“The Blockchain as a Software Connector”	25 de mayo de 2016

El documento D1¹ divulga un sistema de alimentación de datos autenticado llamado Town Crier (TC). TC actúa como un puente entre los contratos inteligentes y los sitios web existentes, que ya son de confianza común para aplicaciones que no son blockchain. Combina un front-end de blockchain con un backend de hardware confiable para raspar sitios web habilitados para HTTPS y servir datos autenticados en la fuente para contratos inteligentes confiables. (Resumen, Párr. 2).

El documento D2² divulga fundamentos para apoyar la decisión arquitectónica sobre si emplear una cadena de bloques descentralizada en lugar de otras soluciones de software, como el almacenamiento tradicional de datos compartidos. Además, exploramos las implicaciones específicas del uso de blockchain como conector de software, incluidas las compensaciones de diseño con respecto a los atributos de calidad (Resumen, líneas 11 a 16).

¹<https://eprint.iacr.org/2016/168.pdf>

²http://design.inf.usi.ch/sites/default/files/biblio/2016_WICSA_BlockChainSoftwareConnector.pdf



Oficio N° 815 de 22 de enero de 2020

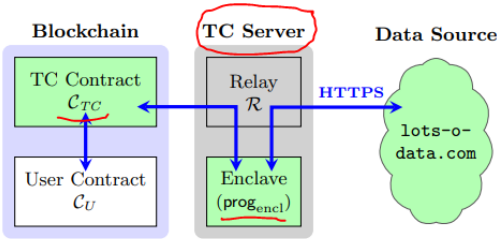
Ref. Expediente N° NC2018/0013180

4. Examen de Patentabilidad (Art. 14 Decisión 486)

4.1. Nivel Inventivo (Art. 18 Decisión 486)

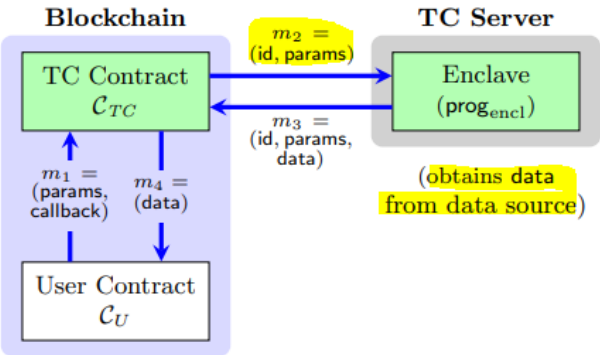
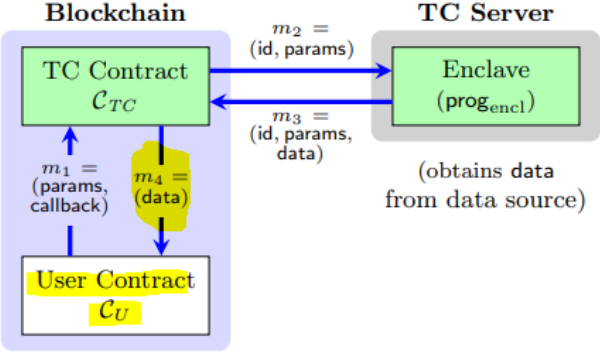
La reivindicación 1 consiste en: “Un sistema informático para delegar el comportamiento asociado con un contrato (...)” (Radicado N° NC2018/0013180 del 5 de diciembre de 2018).

Tabla 1. Comparación de las características técnicas esenciales, con las del estado de la técnica

Características esenciales	D1
Un sistema informático para delegar el comportamiento asociado con un contrato instalado en una cadena de bloques a una aplicación distinta a la cadena de bloques, el sistema informático comprende.	Un sistema de alimentación de datos autenticado llamado Town Crier (TC). TC actúa como un puente entre los contratos inteligentes y los sitios web existentes, que ya son de confianza común para aplicaciones que no son blockchain. Combina un front-end de blockchain con un backend de hardware confiable para raspar sitios web habilitados para HTTPS y servir datos autenticados en la fuente para contratos inteligentes confiables. (Resumen, Párr. 2). Town Crier incluye tres componentes principales: el Contrato TC (C _{TC}), el Enclave (cuyo código se denota por prog _{encl}) y el Relé (R). El enclave y el relé residen en el servidor TC, mientras que el contrato TC reside en la cadena de bloques. Nos referimos a un contrato inteligente que hace uso del servicio Town Crier como solicitante o contrato de confianza, que denominamos C _U , y su propietario (fuera de la cadena) como cliente o usuario (Pág. 3, Párr. 6, líneas 1 a 7).
Uno o más medios de almacenamiento legibles por computadora que almacenan las instrucciones ejecutables por computadora de un cryptodelegate y un servicio de contenedores de cryptlets.	En la Figura 1 se muestra un esquema arquitectónico de TC que muestra su interacción con entidades externas (Pág. 3, Párr. 7).  Figure 1: Basic Town Crier architecture. Trusted components are depicted in green. Además, SGX puede sellar pkOff en un almacenamiento no volátil al tiempo que protege la integridad y la confidencialidad [8,22], lo que nos permite mantener el mismo enlace a través de reinicios del servidor (Pág. 7, Párr. 4).
El cryptodelegate que incluye las instrucciones que recibe del código del contrato que ejecuta en una máquina virtual una identidad de un cryptlet y un comportamiento solicitado que debe realizar el cryptlet.	Una solicitud de datagrama de C _U toma la forma de un mensaje m ₁ = (parámetros, devolución de llamada) a C _{TC} en la cadena de bloques. params especifica el datagrama solicitado, por ejemplo, params: = (url, spec, T), donde url es la fuente de datos de destino, spec especifica el contenido del datagrama que se va a recuperar (por ejemplo, un ticker de stock en un momento determinado), y T especifica el tiempo de entrega del datagrama (iniciado por el raspado de la fuente de datos). La devolución de llamada de parámetro en m ₁ indica el

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

	punto de entrada al que se devolverá el datagrama. Mientras que la devolución de llamada no necesita ser en C_U, suponemos que es por simplicidad (Pág. 5, Párr. 2). Ethereum tiene su propia criptomoneda asociada llamada Ether. (Al momento de escribir este artículo, 1 Ether tiene un valor de mercado de poco menos de \$ 15 US [1].) Para evitar ataques DoS, evitar bucles infinitos inadvertidos dentro de los contratos y, en general, controlar el gasto de recursos de red, Ethereum permite la compra basada en Ether recurso llamado gas para contratos de energía. Cada operación, incluido el envío de datos, la ejecución de cálculos y el almacenamiento de datos, tiene un costo fijo de gas (Pág. 3, Párr. 3, líneas 1 a 8). Implementamos C_{TC} como se describe en la Sección 6 en Solidity, un lenguaje de alto nivel con una sintaxis similar a JavaScript que se compila en el código de bytes de Ethereum Virtual Machine, el lenguaje que usan los contratos de Ethereum (Pág. 13, Apéndice 1, Párr. 1).
Proporciona al servicio de contenedores de cryptlets la identidad y el comportamiento solicitado.	C_{TC} genera una nueva identificación única y reenvía $m_2 = (id, params)$ al Enclave (Pág. 5, Párr. 3, líneas 1 y 2).  Figure 2: Data flows in datagram processing.
Recibe del servicio de contenedores de cryptlets una respuesta generada por el cryptlet que realiza el comportamiento solicitado.	En respuesta, recibe $m_3 = (id, parámetros, datos)$ del servicio TC, donde los datos son el datagrama (por ejemplo, el precio de cotización bursátil deseado) (Pág. 5, Párr. 3, líneas 2 a 4).
Y envía la respuesta al código del contrato.	C_{TC} verifica la consistencia de los parámetros en la solicitud y la respuesta y, si coinciden, reenvía los datos al punto de entrada de devolución de llamada en el mensaje m_4 (Pág. 5, Párr. 3, líneas 4 a 6).  Figure 2: Data flows in datagram processing.

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

El servicio de contenedores de cryptlets, que incluye las instrucciones que almacenan la información relacionada con los cryptlets registrados, que incluye las referencias a los cryptlets, recibe del cryptodelegate la identidad y el comportamiento solicitado.	C _{TC} genera una nueva identificación única y reenvía m ₂ = (id, params) al Enclave (Pág. 5, Párr. 3, líneas 1 y 2).
Identifica una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets.	No menciona.
Que autentica el cryptlet.	Específicamente, utilizamos una funcionalidad UC global F _{sgx} (Σ _{sgx}) [prog _{encl} , R] para denotar (una instancia de) una funcionalidad SGX parametrizada por un esquema de firma (grupo) Σ _{sgx} . Aquí prog _{encl} denota el programa de enclave SGX y R el host SGX físico (que suponemos por simplicidad es el mismo que el del relé TC). Como se describe en la Figura 3, tras la inicialización, F _{sgx} ejecuta outp: = prog _{encl} .Initialize () y da fe del código de prog _{encl} y outp. Al reanudar una llamada con (id, params), F _{sgx} se ejecuta y genera el resultado de prog _{encl} .Resume (id, params). El formalismo adicional para F _{sgx} se da en el apéndice de la versión en línea [40] (Pág. 5, Párr. 2, líneas 6 a 16).
Proporciona el comportamiento solicitado al cryptlet.	C _{TC} genera una nueva identificación única y reenvía m ₂ = (id, params) al Enclave (Pág. 5, Párr. 3, líneas 1 y 2).
Recibe la respuesta generada por el cryptlet y envía a la cadena de bloques.	Luego, prog _{encl} recupera el datagrama solicitado y lo devuelve a R junto con los parámetros, id (Pág. 8, Párr. 1, líneas 3 y 4).
Verifica por el cryptodelegate la respuesta.	En respuesta, recibe m ₃ = (id, parámetros, datos) del servicio TC, donde los datos son el datagrama (por ejemplo, el precio de cotización bursátil deseado) (Pág. 5, Párr. 3, líneas 2 a 4).
Uno o más procesadores para ejecutar las instrucciones ejecutables por computadora almacenadas en uno o más medios de almacenamiento legibles por computadora.	SGX. Las Extensiones de protección de software de Intel (SGX) [8, 22, 29, 31] son un conjunto de nuevas instrucciones que confieren protecciones de hardware al código de nivel de usuario. SGX permite la ejecución de procesos en un espacio de direcciones protegido conocido como <i>enclave</i> . El enclave protege la confidencialidad e integridad del proceso de ciertas formas de ataque de hardware y otro software en el mismo host, incluido el sistema operativo. (Pág. 3, Párr. 3).

El documento D1 se considera el estado de la técnica más cercano a la invención definida en la reivindicación 1 porque divulga un sistema de alimentación de datos autenticado llamado Town Crier (TC). TC actúa como un puente entre los contratos inteligentes y los sitios web existentes, que ya son de confianza común para aplicaciones que no son blockchain. Combina un front-end de blockchain con un backend de hardware confiable para raspar sitios web habilitados para HTTPS y servir datos autenticados en la fuente para contratos inteligentes confiables. (Resumen, Párr. 2).

La diferencia entre la invención definida en la reivindicación 1 y sistema informático divulgado de D1 consiste en que identifica una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets.

El efecto que se logra al incluir al identificar una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets es permitir distribuir la carga de trabajo relacionada con diferentes comportamientos que se ejecutan potencialmente en diferentes hosts o anfitriones.

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿Cómo modificar el sistema conocido en D1 con el fin de mejorar la confiabilidad del sistema y su tolerancia a fallos?

Sin embargo, el hecho de identificar una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets para que sea incluida y así mejorar la confiabilidad del sistema es conocido por una persona normalmente versada en la materia³.

En consecuencia, la persona normalmente versada en la materia, estaría motivada a incluir el identificar una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets en el sistema de D1, para así llegar al objeto de la reivindicación 1. Por lo cual se considera obvia.

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 2: Implementamos C_{TC} como se describe en la Sección 6 en Solidity, un lenguaje de alto nivel con una sintaxis similar a JavaScript que se compila en el código de bytes de Ethereum Virtual Machine, el lenguaje que usan los contratos de Ethereum (Pág. 13, Apéndice 1, Párr. 1).
- Reivindicación 3: Nos referimos a un contrato inteligente que utiliza el servicio Town Crier como solicitante o contrato de confianza, que denominamos C_U (Pág. 3, Párr. 7, líneas 4 a 6). Medimos el tiempo de respuesta del enclave para manejar una solicitud TC, definida como el intervalo entre (1) el relé que envía una solicitud al enclave y (2) el relé que recibe una respuesta del enclave (Pág. 10, Párr. 10, líneas 1 a 4).
- Reivindicación 4: SGX permite que un sistema remoto verifique el software en un enclave y se comunique de forma segura con él. Cuando se crea un enclave, la CPU produce un hash de su estado inicial conocido como medición (Pág. 3, Párr. 5, líneas 1 a 4). Para evitar que el Relé altere los parámetros de la solicitud, el protocolo C_{TC} incluye parámetros como argumento para la entrega que valida contra los valores almacenados. Para reducir este costo para matrices grandes, almacenamos y verificamos SHA3-256 (requestType || timestamp || paramArray). El relé raspa los valores sin procesar para el enclave que calcula el hash y lo incluye como argumento para entregar (Pág. 12, Párr. 4).
- Reivindicación 5: El front-end de contrato inteligente de Town Crier responde a solicitudes por contratos en blockchain con certificaciones del siguiente formulario: "El datagrama X especificado por los parámetros de parámetros es atendido por un sitio web habilitado para HTTPS Y durante un período de tiempo específico T" (Pág. 1, Párr. 5 y Párr. 6).
- Reivindicación 6: Diseñamos e implementamos TC como un sistema completo, altamente escalable y de extremo a extremo que ofrece garantías formales de seguridad a nivel de protocolo criptográfico. TC se ejecuta en host real, habilitado para SGX, en lugar de un emulador (por ejemplo, [10,33]) (Pág. 2, Párr. 2, líneas 1 a 4).

³ https://es.wikipedia.org/wiki/Computaci%C3%B3n_distribuida



Oficio N° 815 de 22 de enero de 2020

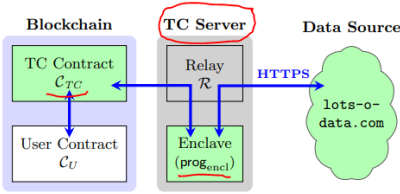
Ref. Expediente N° NC2018/0013180

Las reivindicaciones dependientes 2 a 6 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo sistema de la reivindicación 1, por lo cual se considera que no tienen nivel inventivo.

La reivindicación 7 consiste en: “Un método realizado por un sistema informático, el método comprende: (...)” (Radicado N° NC2018/0013180 del 5 de diciembre de 2018).

Tabla 2. Comparación de las características técnicas esenciales, con las del estado de la técnica

Características esenciales	D1	D2
Un método realizado por un sistema informático, el método comprende:	Un sistema de alimentación de datos autenticado llamado Town Crier (TC). TC actúa como un puente entre los contratos inteligentes y los sitios web existentes, que ya son de confianza común para aplicaciones que no son blockchain. Combina un front-end de blockchain con un backend de hardware confiable para raspar sitios web habilitados para HTTPS y servir datos autenticados en la fuente para contratos inteligentes confiables. (Resumen, Párr. 2).	La máquina de estado replicada [22] es un método general para implementar un servicio tolerante a fallas con un sistema distribuido. Para hacer frente a las fallas, replica el servicio en varios servidores y coordina las solicitudes de servicio emitidas por los clientes. Del mismo modo, la cadena de bloques utiliza la distribución para no depender de ninguna entidad individual (Pág. 10, Col. 1, Párr. 4, líneas 1 a 6).
Recibir de un contrato inteligente que se está ejecutando por una máquina virtual una solicitud para registrar un cryptlet de contrato para realizar el comportamiento en nombre del contrato inteligente, el código del cryptlet de contrato y una indicación de una computadora anfitriona certificada para ejecutar el código del cryptlet de contrato.	Una solicitud de datagrama de C_U toma la forma de un mensaje $m_1 =$ (parámetros, devolución de llamada) a C_{TC} en la cadena de bloques. $params$ especifica el datagrama solicitado, por ejemplo, $params = (url, spec, T)$, donde url es la fuente de datos de destino, $spec$ especifica el contenido del datagrama que se va a recuperar (por ejemplo, un ticker de stock en un momento determinado), y T especifica el tiempo de entrega del datagrama (iniciado por el raspado de la fuente de datos). La devolución de llamada de parámetro en m_1 indica el punto de entrada al que se devolverá el datagrama. Mientras que la devolución de llamada no necesita ser en C_U, suponemos que es por simplicidad (Pág. 5, Párr. 2). Ethereum tiene su propia criptomoneda asociada llamada Ether. (Al momento de escribir este artículo, 1 Ether tiene un valor de mercado de poco menos de \$ 15 US [1].) Para evitar ataques DoS, evitar bucles infinitos inadvertidos dentro de los contratos y, en general, controlar el gasto de recursos de red, Ethereum permite la compra basada en Ether recurso llamado gas para contratos de energía. Cada operación, incluido el envío de datos, la ejecución de cálculos y el almacenamiento de datos, tiene un	No menciona.

	costo fijo de gas (Pág. 3, Párr. 3, líneas 1 a 8).	
Verificar el código del cryptlet de contrato.	No menciona.	Los diferentes componentes de la arquitectura pueden coordinar sus cálculos a través de la cadena de bloques. Para hacerlo, es posible enviar transacciones a contratos inteligentes para invocar las funciones definidas en los contratos inteligentes, o utilizar un oráculo de validación para firmar transacciones, cuyo resultado depende del estado externo (Pág. 5, Col. 1, Párr. 2).
Almacenar el código del cryptlet de contrato en una cadena de bloques de registro de cryptlets y almacenamiento de datos.	En la Figura 1 se muestra un esquema arquitectónico de TC que muestra su interacción con entidades externas (Pág. 3, Párr. 7).  Figure 1: Basic Town Crier architecture. Trusted components are depicted in green. Además, SGX puede sellar pk_{Off} en un almacenamiento no volátil al tiempo que protege la integridad y la confidencialidad [8,22], lo que nos permite mantener el mismo enlace a través de reinicios del servidor (Pág. 7, Párr. 4).	Si la validación no puede automatizarse, un árbitro humano puede validar transacciones y firmar transacciones válidas. Si la validación se puede automatizar, un árbitro automatizado podría extraer periódicamente el valor de las variables del almacenamiento del contrato como estado de la aplicación para validar las transacciones (Pág. 5, Col. 2, Párr. 2).
Recibir del contrato inteligente una solicitud para que el cryptlet de contrato realice el comportamiento.	En respuesta, recibe $m_3 = (id, \text{parámetros, datos})$ del servicio TC, donde los datos son el datagrama (por ejemplo, el precio de cotización bursátil deseado) (Pág. 5, Párr. 3, líneas 2 a 4).	No menciona.
Ordenar al cryptlet de contrato que se ejecute en la computadora anfitriona certificada.	Diseñamos e implementamos TC como un sistema completo, altamente escalable y de extremo a extremo que ofrece garantías formales de seguridad a nivel de protocolo criptográfico. TC se ejecuta en host real, habilitado para SGX, en lugar de un emulador (por ejemplo, [10,33]) (Pág. 2, Párr. 2, líneas 1 a 4).	No menciona.
Enviar al cryptlet de contrato que se ejecuta en la computadora anfitriona certificada la solicitud para realizar el comportamiento.	C_{TC} verifica la consistencia de los parámetros en la solicitud y la respuesta y, si coinciden, reenvía los datos al punto de entrada de devolución de llamada en el mensaje m_4 (Pág. 5, Párr. 3, líneas 4 a 6).	No menciona.

Ref. Expediente N° NC2018/0013180

	Blockchain TC Contract C_{TC} $m_1 =$ (params, callback) $m_4 =$ (data) User Contract C_U TC Server Enclave (progencl) (obtains data from data source) $m_2 =$ (id, params) $m_3 =$ (id, params, data) Figure 2: Data flows in datagram processing.	
En donde las comunicaciones con el contrato inteligente y el cryptlet de contrato son a través de canales seguros	Esta disposición permite que el enclave establezca un canal seguro con un servidor web; el enclave mismo realiza el protocolo de enlace TLS con un servidor de destino y realiza todas las operaciones criptográficas internamente, mientras que el proceso no confiable solo actúa como una interfaz de red (Pág. 14, Col. 1, Párr. 5, líneas 5 a 9).	No menciona.

El documento D1 se considera el estado de la técnica más cercano a la invención definida en la reivindicación 1 porque divulga un sistema de alimentación de datos autenticado llamado Town Crier (TC). TC actúa como un puente entre los contratos inteligentes y los sitios web existentes, que ya son de confianza común para aplicaciones que no son blockchain. Combina un front-end de blockchain con un backend de hardware confiable para raspar sitios web habilitados para HTTPS y servir datos autenticados en la fuente para contratos inteligentes confiables. (Resumen, Párr. 2).

La diferencia entre la invención definida en la reivindicación 1 y sistema informático divulgado de D1 consiste en verificar el código del cryptlet de contrato.

El efecto que se logra al incluir el verificar el código del cryptlet de contrato es mejorar la seguridad en la transmisión.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿Cómo modificar el método conocido en D1 con el fin de mejorar la seguridad en el método?

Sin embargo, el incluir la verificación el código del cryptlet de contrato para mejorar la seguridad en el método, ya está divulgado en el documento D2 que se refiere a fundamentos para apoyar la decisión arquitectónica sobre si emplear una cadena de bloques descentralizada en lugar de otras soluciones de software, como el almacenamiento tradicional de datos compartidos. Además, exploramos las implicaciones específicas del uso de blockchain como conector de software, incluidas las compensaciones de diseño con respecto a los atributos de calidad (Resumen, líneas 11 a 16). Los diferentes componentes de la arquitectura pueden coordinar sus cálculos a través de la cadena de bloques. Para hacerlo, es posible enviar transacciones a contratos inteligentes para invocar las funciones definidas en los contratos inteligentes, o utilizar un oráculo de validación para firmar transacciones, cuyo resultado depende del estado externo (Pág. 5, Col. 1, Párr. 2).

En consecuencia, la persona normalmente versada en la materia, estaría motivada a incluir la verificación el código del cryptlet de contrato del documento D2 en el método de D2, para así llegar al objeto de la reivindicación 1. Por lo cual se considera obvia.

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 8: SGX permite que un sistema remoto verifique el software en un enclave y se comunique de forma segura con él. Cuando se crea un enclave, la CPU produce un hash de su estado inicial conocido como medición (Pág. 3, Párr. 5, líneas 1 a 4). Para evitar que el Relé altere los parámetros de la solicitud, el protocolo CTC incluye parámetros como argumento para la entrega que valida contra los valores almacenados. Para reducir este costo para matrices grandes, almacenamos y verificamos SHA3-256 (requestType || timestamp || paramArray). El relé raspa los valores sin procesar para el enclave que calcula el hash y lo incluye como argumento para entregar (Pág. 12, Párr. 4).
- Reivindicación 9: Implementamos C_{TC} como se describe en la Sección 6 en Solidity, un lenguaje de alto nivel con una sintaxis similar a JavaScript que se compila en el código de bytes de Ethereum Virtual Machine, el lenguaje que usan los contratos de Ethereum (Pág. 13, Apéndice 1, Párr. 1).
- Reivindicación 10: El front-end de contrato inteligente de Town Crier responde a solicitudes por contratos en blockchain con certificaciones del siguiente formulario: "El datagrama X especificado por los parámetros de parámetros es atendido por un sitio web habilitado para HTTPS Y durante un período de tiempo específico T" (Pág. 1, Párr. 5 y Párr. 6).

Asimismo, el documento D2 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 11: Como se muestra en la figura 2, el control de la aplicación fluye a través de transacciones iniciadas desde cuentas de propiedades externas y transferidas entre cuentas contractuales. Los contratos se comportan como agentes autónomos que viven en el entorno de ejecución de la red blockchain. Los contratos se instancian mediante el envío de transacciones con el código fuente de los contratos a la red blockchain. Un contrato define un conjunto de funciones. En el nivel de función, el contrato ejecuta el código de una función cuando recibe una transacción que llama a la función con sus parámetros requeridos (Pág. 5, Col. 1, Párr. 3, líneas 1 a 10).

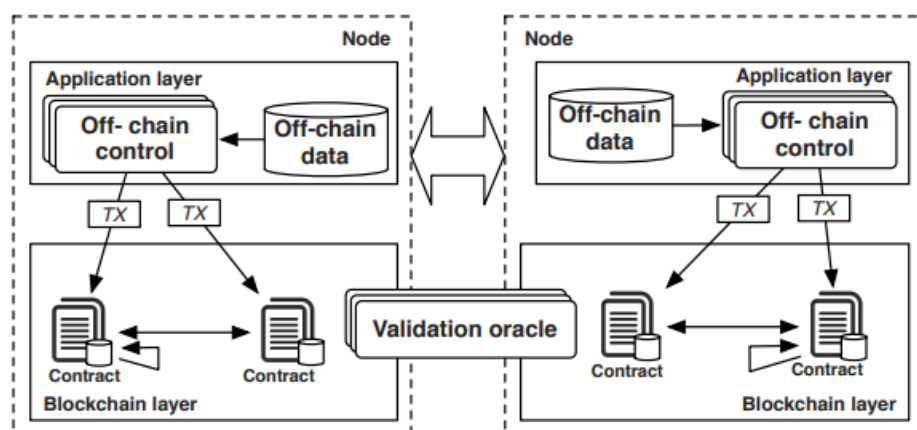


Figure 2: Interaction between applications and blockchain

Ref. Expediente N° NC2018/0013180

- Reivindicación 12: Además, un sistema basado en blockchain puede mantener una cadena única para registrar todos los tipos de transacciones juntas o mantener múltiples cadenas para aislar información de partes separadas o de preocupaciones separadas, por ejemplo, usando una cadena para almacenar transacciones y usando una cadena separada para almacenar información de control de acceso (Decisión de diseño de aplicación 3 en la Tabla II) (Pág. 4, Col. 1, Párr. 2).

Las reivindicaciones dependientes 8 a 12 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo método de la reivindicación 7, por lo cual se considera que no tienen nivel inventivo.

La reivindicación 13 consiste en: “Un método realizado por uno o más sistemas informáticos, el método comprende: (...)” (Radicado N° NC2018/0013180 del 5 de diciembre de 2018).

Tabla 3. Comparación de las características técnicas esenciales, con las del estado de la técnica

Características esenciales	D1
Un método realizado por uno o más sistemas informáticos, el método comprende:	Un sistema de alimentación de datos autenticado llamado Town Crier (TC). TC actúa como un puente entre los contratos inteligentes y los sitios web existentes, que ya son de confianza común para aplicaciones que no son blockchain. Combina un front-end de blockchain con un backend de hardware confiable para raspar sitios web habilitados para HTTPS y servir datos autenticados en la fuente para contratos inteligentes confiables. (Resumen, Párr. 2). Town Crier incluye tres componentes principales: el Contrato TC (C _{TC}), el Enclave (cuyo código se denota por prog _{encl}) y el Relé (R). El enclave y el relé residen en el servidor TC, mientras que el contrato TC reside en la cadena de bloques. Nos referimos a un contrato inteligente que hace uso del servicio Town Crier como solicitante o contrato de confianza, que denominamos C _U , y su propietario (fuera de la cadena) como cliente o usuario (Pág. 3, Párr. 6, líneas 1 a 7).
Ordenar la ejecución mediante una máquina virtual de un contrato inteligente asociado con una cadena de bloques.	Diseñamos e implementamos TC como un sistema completo, altamente escalable y de extremo a extremo que ofrece garantías formales de seguridad a nivel de protocolo criptográfico. TC se ejecuta en host real, habilitado para SGX, en lugar de un emulador (por ejemplo, [10,33]) (Pág. 2, Párr. 2, líneas 1 a 4).
Durante la ejecución del contrato inteligente, enviar a través de un cryptodelegate a un servicio de contenedores de cryptlets una solicitud para delegar un comportamiento a un cryptlet que se ejecuta en una computadora anfitriona certificada.	C _{TC} verifica la consistencia de los parámetros en la solicitud y la respuesta y, si coinciden, reenvía los datos al punto de entrada de devolución de llamada en el mensaje m ₄ (Pág. 5, Párr. 3, líneas 4 a 6).

	Blockchain TC Contract C_{TC} $m_1 =$ (params, callback) $m_4 =$ (data) User Contract C_U TC Server Enclave ($prog_{encl}$) (obtains data from data source) $m_2 =$ (id, params) $m_3 =$ (id, params, data) Figure 2: Data flows in datagram processing.
Durante la ejecución del servicio de contenedores cryptlets, identificar una computadora anfitriona para ejecutar el código del cryptlet.	No menciona.
Indicar a la computadora anfitriona identificada que ejecute el código del cryptlet para realizar el comportamiento delegado.	Ethereum tiene su propia criptomoneda asociada llamada Ether. (Al momento de escribir este artículo, 1 Ether tiene un valor de mercado de poco menos de \$ 15 US [1].) Para evitar ataques DoS, evitar bucles infinitos inadvertidos dentro de los contratos y, en general, controlar el gasto de recursos de red, Ethereum permite la compra basada en Ether recurso llamado gas para contratos de energía. Cada operación, incluido el envío de datos, la ejecución de cálculos y el almacenamiento de datos, tiene un costo fijo de gas (Pág. 3, Párr. 3, líneas 1 a 8).
Recibir del cryptlet una respuesta al comportamiento solicitado.	En respuesta, recibe $m_3 =$ (id, parámetros, datos) del servicio TC, donde los datos son el datagrama (por ejemplo, el precio de cotización bursátil deseado) (Pág. 5, Párr. 3, líneas 2 a 4).
Enviar la respuesta al contrato inteligente en la cadena de bloques validada por el cryptodelegate.	C_{TC} genera una nueva identificación única y reenvía $m_2 =$ (id, params) al Enclave (Pág. 5, Párr. 3, líneas 1 y 2).

El documento D1 se considera el estado de la técnica más cercano a la invención definida en la reivindicación 1 porque divulga un sistema de alimentación de datos autenticado llamado Town Crier (TC). TC actúa como un puente entre los contratos inteligentes y los sitios web existentes, que ya son de confianza común para aplicaciones que no son blockchain. Combina un front-end de blockchain con un backend de hardware confiable para raspar sitios web habilitados para HTTPS y servir datos autenticados en la fuente para contratos inteligentes confiables. (Resumen, Párr. 2).

La diferencia entre la invención definida en la reivindicación 13 y el método informático divulgado de D1 consiste en que identifica una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets.

El efecto que se logra al incluir al identificar una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets es permitir distribuir la carga de trabajo relacionada con diferentes comportamientos que se ejecutan potencialmente en diferentes hosts o anfitriones.

Por lo tanto, el problema técnico objetivo que pretende resolver esta invención se puede formular así: ¿Cómo modificar el método conocido en D1 con el fin de mejorar la confiabilidad del método y su tolerancia a fallos?

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

Sin embargo, el hecho de identificar una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets para que sea incluida y así mejorar la confiabilidad del método es conocido por una persona normalmente versada en la materia⁴.

En consecuencia, la persona normalmente versada en la materia, estaría motivada a incluir el identificar una computadora anfitriona para ejecutar el cryptlet y el contenedor de cryptlets en el método de D1, para así llegar al objeto de la reivindicación 13. Por lo cual se considera obvia.

Asimismo, el documento D1 divulga las características de las reivindicaciones dependientes así:

- Reivindicación 14: Esta disposición permite que el enclave establezca un canal seguro con un servidor web; el enclave mismo realiza el protocolo de enlace TLS con un servidor de destino y realiza todas las operaciones criptográficas internamente, mientras que el proceso no confiable solo actúa como una interfaz de red (Pág. 14, Col. 1, Párr. 5, líneas 5 a 9).

Las reivindicaciones dependientes 14 no mencionan alguna característica que, junto con las características de la reivindicación independiente, aporte nivel inventivo sistema de la reivindicación 13, por lo cual se considera que no tienen nivel inventivo.

5. Conclusión con respecto a novedad y/o nivel inventivo

Los requisitos de Patentabilidad de la presente solicitud están afectados así:

Ítem	Documento	Reivindicaciones afectadas	Requisito que afecta
D1	Zhang (2016)	1 a 10, 13 y 14	Nivel Inventivo
D2	Xu (2016)	7, 11 y 12	Nivel inventivo

En cumplimiento del artículo 45 de la Decisión 486 de 2000, se le indica que debe dar respuesta dentro del término de sesenta (60) días contados a partir de la fecha de notificación de la presente comunicación. En caso de no dar respuesta al presente requerimiento, o si a pesar de la respuesta subsistieran los impedimentos para la concesión, se denegará la patente.

Si como respuesta a este requerimiento, el solicitante modifica: i) el capítulo reivindicatorio, ii) la descripción de la invención a proteger, iii) o presenta un nuevo capítulo reivindicatorio deberá observar las instrucciones contenidas en el numeral 1.2.2.1.1., del Capítulo Primero del Título X de la Circular Única y la Dirección de Nuevas Creaciones seguirá el procedimiento descrito en el numeral 1.2.2.5.1.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 6.2 del Capítulo Sexto del Título I de la Circular Única, informándoles que contra el mismo no procede recurso alguno en actuación administrativa.

f. and f. and f.

4 https://es.wikipedia.org/wiki/Computaci%C3%B3n_distribuida

Oficio N° **815** de **22 de enero de 2020**

Ref. Expediente N° NC2018/0013180

MARÍA JOSÉ LAMUS BECERRA
DIRECTORA DE NUEVAS CREACIONES



Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

INFORME DE BÚSQUEDA
EXAMEN DE PATENTABILIDAD

Solicitud No.		Examen de patentabilidad							
NC2018/0013180		1°	X	2°		3°		Revoca	
Solicitud Internacional No.		Fecha solicitud internacional: (DD/MM/AÑO)							
PCT/US2017/035341		1 de junio de 2017							
Fecha de determinación estado de la técnica (DD/MM/AÑO)				Prioridad		Presentación			
6/6/2016				X					
Solicitante									
LLC MICROSOFT TECHNOLOGY LICENSING									
Reivindicaciones que no son objeto de búsqueda por:									
Art. 14	-	Art. 15	-	Art. 20	-	Art. 25 (Falta de unidad de invención)		-	
CRITERIOS DE BÚSQUEDA									
Reivindicaciones objeto de búsqueda				1 a 15					
Palabras clave utilizadas				Method, delegate, behavior, smart, contract, blockchain, system, virtual, machine, cryptlet, container, service, cryptodelegate, executes, host, identifies, verify.					
Clasificaciones utilizadas para la búsqueda				CIP:	G06F 21/51, G06F 21/53, H04L 9/32				
DOCUMENTOS CONSIDERADOS RELEVANTES (Documentos patente, Literatura no patente (LNP): Artículos, catálogos, etc.)									
Informes de búsqueda / Bases de datos consultadas	N° de solicitud de patente / N° de patente / Autor LNP ⁽¹⁾	Fecha de Publicación /Presentación (DD/MM/AAAA)	Reivindicaciones /Secuencias afectadas	Citas relevantes del estado de la técnica				Categoría del documento citado ⁽²⁾	
SIPI	NC2018/0004116	18/04/2018	-	-				A	
ISA	Thomas (2014) ¹	17/07/2014	-	-				A	
	GitHub (2014) ²	29/11/2014	-	-				A	
	Zhang (2016) ³	19/02/2016	1 a 10, 13 y 14	1 Resumen, Párr. 2; Pág. 3, Párr. 6, líneas 1 a 7; Pág. 3, Párr. 7; Pág. 7, Párr. 4; Pág. 5, Párr. 2; Pág. 3, Párr. 3, líneas 1 a 8; Pág. 13, Apéndice 1, Párr. 1; Pág. 5, Párr. 3, líneas 1 y 2; Figura 2; Pág. 5, Párr. 3, líneas 2 a 4; Pág. 5, Párr. 3, líneas 4 a 6; Pág. 5, Párr. 3, líneas 1 y 2; Pág. 5, Párr. 2, líneas 6 a 16; Pág. 5, Párr. 3, líneas 1 y 2; Pág. 5, Párr. 3, líneas 2 a 4; Pág. 3, Párr. 3				Y	
	Xu (2016) ⁴	25/05/2016	7, 11 y 12	2 Pág. 13, Apéndice 1, Párr. 1.				Y	
				3 Pág. 10, Párr. 10, líneas 1 a 4.					
			4 Pág. 12, Párr. 4.						
			5 Pág. 1, Párr. 5 y Párr. 6.						
			6						

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

				Pág. 2, Párr. 2, líneas 1 a 4 7 Resumen, Párr. 2; Pág. 5, Párr. 2; Pág. 3, Párr. 3, líneas 1 a 8; Pág. 3, Párr. 7; Pág. 7, Párr. 4; Pág. 5, Párr. 3, líneas 2 a 4; Pág. 2, Párr. 2, líneas 1 a 4; Pág. 5, Párr. 3, líneas 4 a 6; Pág. 14, Col. 1, Párr. 5, líneas 5 a 9. 8 Pág. 12, Párr. 4. 9 Pág. 13, Apéndice 1, Párr. 1. 10 Pág. 1, Párr. 5 y Párr. 6. 13 Pág. 3, Párr. 6, líneas 1 a 7; Pág. 2, Párr. 2, líneas 1 a 4; Pág. 5, Párr. 3, líneas 4 a 6; Pág. 3, Párr. 3, líneas 1 a 8; Pág. 5, Párr. 3, líneas 2 a 4; Pág. 5, Párr. 3, líneas 1 y 2. 14 Pág. 14, Col. 1, Párr. 5, líneas 5 a 9. 7 Pág. 10, Col. 1, Párr. 4, líneas 1 a 6; Pág. 5, Col. 1, Párr. 2; Pág. 5, Col. 2, Párr. 2. 11 Pág. 5, Col. 1, Párr. 3, líneas 1 a 10. 12 Pág. 4, Col. 1, Párr. 2.	
USPTO	US2016335533 US2015358312 US2016055553 US2017011460	17/11/2016 10/12/2015 25/02&2016 12/01/2017	- - - -	- - - -	E A A E
Patbase	US2019116174	18/04/2019	-	-	E
Orbit	WO2019/010063	10/01/2019	-	-	E
Googlepatent	JP6364132B2	25/07/2018	-	-	E
(1) Cita completa literatura no patente (LNP)		1 Stefan Thomas ET AL. “Smart Oracles: A simple, Poerful Approach to Smart Contracts . codius/codius Wiki . GitHub”. 2014. XP055395127.			
Apellido, A. A. (Fecha). Título del artículo. Nombre de la revista. Volumen(Número), pp-pp.		2 GitHub. Wiki “Orisi White Paper” 2014. XP055395377.			
O		3 Fan Zhang ET AL. “Town Crier: An Authenticated Data Feed for Smart Contracts”. International Association for Cryptologic Research. Vol. 20160330:013429.			
Apellido, A. A. (Fecha). Título del artículo. Nombre de la revista. Recuperado de http://xxxxxxx.		4 Xiwei Xu ET AL. “The Blockchain as a Software Connector”. DOI: 10.1109/WICSA.2016.21			
(2) Categorías de documentos citados					
“A” Documento que define el estado general de la técnica no considerado como particularmente relevante.			“T” Documento publicado en fecha posterior a la de presentación o la de prioridad de la solicitud internacional y no entra en conflicto con dicha solicitud, pero puede ser útil para la mejor comprensión del principio o teoría en que se basa la invención, o para demostrar que el razonamiento o los hechos en los que se basa dicha invención son incorrectos.		
“E” Solicitud de patente o patente anterior pero publicada en la fecha de presentación internacional o en fecha posterior.					
“L” Documento citado para llamar la atención sobre por ejemplo, dudas acerca de una reivindicación de prioridad o para determinar la fecha de publicación de otra cita, siempre van acompañados de una explicación, por la cual se citan.			“X” Documento particularmente relevante; la invención reivindicada no puede considerarse nueva o que implique una actividad inventiva por referencia al documento aisladamente considerado.		
			“Y” Documento particularmente relevante; la invención reivindicada no puede considerarse que implique una actividad inventiva cuando el		

Oficio N° 815 de 22 de enero de 2020

Ref. Expediente N° NC2018/0013180

“O” Documento del tipo actas de conferencia, este tipo de documentos siempre estará acompañado por otra letra que indique la pertinencia del documento, por ejemplo, O,X, O,Y o bien O,A. “P” Documento publicado antes de la fecha de presentación internacional pero con posterioridad a la fecha de prioridad reivindicada.	documento se asocia a otro u otros documentos de la misma naturaleza, cuya combinación resulta evidente para un experto en la materia.
Fecha del reporte de búsqueda: 12/11/2019	
Examinador: FREDDY ALEXANDER VELANDIA VIVAS	