UNIVERSIDAD DE VALLADOLID

TRABAJO FIN DE GRADO

Fortificación IoT mediante blockchain y cifrado asimétrico

Autor:
Javier Sanz Peláez

Tutor:
Federico Simmross Wattenberg
Juan Pablo Casaseca de la Higuera

Departamento:

Teoría de la Señal y Comunicaciones e Ingeniería Telemática

14 de junio de 2018

«*Cuando el sabio señala la luna, el tonto mira al dedo.*»

Proverbio chino

Abstract

Fortificación IoT mediante blockchain y cifrado asimétrico

por Javier Sanz Peláez

The deployment of wide networks of sensors for monitoring involves a few problems from the security point of view. It is not a simple issue to identify whether all elements in a network are legitimate or the information transmited is true, specially when talking about a great number of devices. This situation can not be faced in the same way as traditional computing does, due to the limited power of devices in the field of Internet of Things.
In order to solve this problems, a transport system has been developed to ensure the integrity of data through networks of devices. This is achived making use of the recent technology blockchain and asymmetric cryptography techniques. This system pretends to provide security in deployments with a long number of devices with little power.

Resumen

Fortificación IoT mediante blockchain y cifrado asimétrico

por Javier Sanz Peláez

El despliegue de amplias redes de sensores para la monitorización plantea varios problemas desde el punto de vista de seguridad. Identificar si todos los elementos de la red son legítimos y si la información transmitida es veraz no es un asunto trivial cuando se habla de grandes cantidades de dispositivos. Esta situación no puede plantearse de la misma forma que se hace con una informática más tradicional, ya que los elementos con los que se trabaja en el ámbito del *Internet de las Cosas* son mucho menos potentes que un ordenador convencional.

Para resolver esta situación se ha desarrollado un sistema de transporte que garantiza la integridad de los datos a través de las redes de dispositivos mediante la reciente tecnología de blockchain y técnicas de cifrado asimétrico. Este sistema está orientado a proporcionar seguridad a despliegues con un gran número de dispositivos de escasa potencia.

Agradecimientos

Gracias a todos aquellos que siempre me han apoyado, especialmente a mi familia y su infinita paciencia.

Índice general

Índice de figuras

Lista de Tablas

Lista de Abreviaturas

IoT	**I**nternet **o**f **T**hings
SBC	**S**ingle **B**oard **C**omputer
NA	**N**odo **A**ctivo
NM	**N**odo **M**aestro
NE	**N**odo **E**sclavo
NV	**N**odo **V**isualizador
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**gineers
IETF	**I**nternet **E**ngineering **T**ask **F**orce
OCF	**O**pen **C**onnectivity **F**oundation
OMA	**O**pen **M**obile **A**lliance
M2M	**M**achine to **M**achine
PAN	**P**ersonal **A**rea **N**etwork
UWB	**U**ltra **W**ide **B**and
WSN	**W**ireless **S**ensor **N**etwork

Capítulo 1

Introducción

1.1. Motivación

"Internet de las cosas" (IoT[1]) es un paradigma de comunicación reciente que aboga por que todos los objetos de la vida cotidiana estén equipados por microcontroladores y transceptores proporcionando información propia y de su entorno a través de Internet. Tal y como lo recoge la Comisión Europea, "Internet of Things (IoT) representa el siguiente paso hacia la digitalización de nuestra sociedad y economía, donde los objetos y las personas están interconectados a través de redes de comunicación, informando de su estado y/o de su entorno" [1]. La Comisión Europea representa esta interconexión con la figura 1.1 en su página web. La Internet del futuro [2] incluirá un gran numero de objetos que proporcionarán información y servicios a los usuarios finales a través de un sistema de comunicación que, entre otras características, debe ser seguro.

FIGURA 1.1: Ilustración esquemática de IoT [1].

En IoT todo se vuelve virtual. Cada persona y objeto es localizable en Internet de manera que se puede proporcionar interacción con una amplia variedad de dispositivos, como aplicaciones domésticas, cámaras de vigilancia, monitorización de procesos productivos, actuadores, vehículos, etc. Esto posibilita nuevos escenarios para ciudadanos, empresas y administraciones públicas con un potencial enorme de cambiar los aspectos cotidianos de los usuarios. Para los particulares, el efecto más

[1]Internet of Things.

palpable de IoT será en el trabajo y en casa, con la domótica que permitirá optimizar tareas, además de dispositivos para mejorar las condiciones de salud y asistencia a modo de ejemplo. A las empresas les permitirá mejorar procesos productivos optimizando los recursos disponibles, lo cual se traduce en un incremento de los beneficios.

Sin embargo, como con cada nueva tecnología, IoT añade, junto a sus beneficios, nuevos riesgos y desafíos. IoT está estrechamente ligado con la comunicación y la información, y por esto mismo, está justificado considerar la seguridad y privacidad como desafíos presentes y futuros. A primera vista, las similitudes en los desafíos mencionados con una informática más "tradicional" puede hacer que parezca adecuado enfrentarlos de la misma forma, pero son sus diferencias las que señalan que el enfoque no puede hacerse igual que en IoT y requiera un examen más detallado. En adelante de deberá prestar especial atención a algunos aspectos:

- El número de dispositivos conectados a Internet ha sobrepasado ampliamente al de personas. Además, este numero continúa incrementándose y se espera que sea entre 26 mil millones y 50 mil millones para 2020 [3]. Muchos factores están facilitando este desarrollo, como la inclusión del protocolo IPv6, que permite otorgar una dirección única a cada dispositivo facilitando la comunicación entre ellos. Por el contrario, los problemas de privacidad y seguridad no crecen linealmente con el número de dispositivos conectados, sino mucho más rápido. Esto se debe a que el número de canales de comunicación en una red crece mucho más rápido que el número de nodos.

- Las redes de ordenadores son siempre heterogéneas, induciendo problemas de seguridad. Para IoT este problema se acentúa drásticamente al ser todavía más heterogéneas, integrando multitud de dispositivos de diferentes fabricantes, diferentes plataformas y protocolos de comunicación.

- Mientras que los servidores y estaciones de trabajo están protegidos en salas y oficinas con personal cerca, en una configuración IoT, sensores y otros dispositivos pueden estar ubicados en cualquier parte, expuestos a robos, daños e intrusiones. Un potencial atacante tiene una mayor posibilidad de acceder físicamente a los aparatos para encontrar sus vulnerabilidades.

- En IoT es muy habitual, como se menciona en el punto anterior, que al poder estar un dispositivo en cualquier lugar, no tenga una linea de alimentación estable, teniendo en su lugar baterías. En este caso, para que el dispositivo pueda funcionar de forma autónoma el mayor tiempo posible antes de agotar sus recursos, se hace uso de sistemas de bajo consumo energético y baja potencia de proceso que, es posible, que no puedan hacer uso de las librerías de seguridad con la misma eficacia que un ordenador personal, servidor o un sistema más potente. Este inconveniente ha perdurado a través del tiempo que se lleva haciendo implementaciones en IoT dándose avances muy discretos.

- Con el incremento de dispositivos conectados, la cantidad de información generada aumenta y el coste de mantenimiento sube, pudiendo verse reducidas las medidas de redundancia para evitar su perdida.

- Los dispositivos conectados se usan y se integran perfectamente en el mundo que nos rodea. Pueden recopilar datos, comunicarse e interactuar con otros dispositivos sin supervisión, de forma transparente al usuario, dado que ante un potencial número tan elevado de dispositivos se hace imposible comprobar de forma permanente si las interconexiones están comprometidas.

- Si bien hasta los últimos años los ciberataques han amenazado principalmente los sistemas de información, las redes informáticas y los ordenadores personales, IoT escalará los riesgos de seguridad a un nivel diferente. En la era de IoT, como los actuadores y los sistemas de control estarán interconectados con otros sistemas, los atacantes pueden apuntar directamente a los dispositivos conectados y lograr la destrucción física de los equipos e infraestructuras, como automóviles autónomos, casas inteligentes, redes eléctricas, yacimientos petrolíferos , sistemas de transporte o plantas nucleares. Stuxnet [4] fue el primer código malicioso conocido que atacó el sistema de control de una instalación nuclear; sin embargo, con la expansión de IoT, será una amenaza a tener en cuenta en el futuro [4].

- Los dispositivos omnipresentes como los *wearables* pueden unirse y abandonar su red en cualquier momento. Esto, en combinación con las características de comunicación multiprotocolo, hace que las medidas tradicionales de seguridad de la información sean insuficientes para el IoT.

En la tabla 1.1 se refleja el crecimiento que estima la consultora Gartner [2], para la cual habrá más de 26 mil millones de dispositivos para 2020 y en consecuencia, una tecnología basada en Blockchain debería permitir administrar las transacciones entre todos estos dispositivos con el mismo celo que se emplea en las transacciones económicas.

Category	2016	2017	2018	2020
Consumer	3.963,0	5.244,3	7.036,3	12.863,0
Business: Cross-Industry	1.102,1	1.501,0	2.132,6	4.381,4
Business: Vertical-Specific	1.316,6	1.635,4	2.027,7	3.171,0
Grand Total	**6.381,8**	**8.380,6**	**11.196,6**	**20.415,4**

TABLA 1.1: Unidades de IoT instaladas por categorías (millones) [2].

Cómo se ha visto en los puntos anteriores, las implementaciones IoT tienen una serie de desafíos críticos a los que enfrentarse antes de que se hagan realidad las previsiones de cara a la seguridad.

Blockchain fue empleado por primera vez en 2008 por una o varias personas que emplean el pseudonimo de Satoshi Nakamoto para la primera moneda electrónica. Durante años este sistema pasó desapercibido para la sociedad, pero después de la irrupción de la moneda Bitcoin, muchos desarrolladores se interesaron por la tecnología, permitiendo a Bitcoin convertirse en la divisa digital más popular del mercado.

Blockchain se basa en el almacenamiento de operaciones en un libro de registros llamado ”ledger” el cual se define como «base de datos distribuida, compartida y encriptada que funciona como un repositorio de información irreversible e incorruptible» [5]. Cada diez minutos aproximadamente, crece de forma constante añadiendo nuevos bloques a la cadena. Los usuarios denominados ”mineros” almacenan y validan las entradas más recientes en los bloques que componen la cadena, estos bloques quedan ordenados a su vez cronológicamente. Una vez que los bloques de información han entrado a formar parte de la cadena, no pueden ser borrados o cambiados. La tecnología Blockhain se comporta a la vez como red y base de datos, aportando seguridad e integridad a los datos [6]. Las normas que deben cumplir las transacciones se basan en reglas definidas matemáticamente [7]. Es importante destacar que Blockchain no es una definición de estándar ni un protocolo; su implementación es

tremendamente flexible, siendo principalmente un paradigma que asegura las ventajas de seguridad e integridad allá donde se aplique.

Una forma de comprender Blockchain es verlo como la nueva capa de aplicaciones para la pila de protocolos de Internet, ya que Blockchain puede permitir tanto transacciones económicas inmediatas como a largo plazo, y contratos financieros más complicados. Puede ser una capa para transacciones de diferentes tipos de activos: información, monedas o contratos financieros. Además, un sistema de registro e inventario para registrar, rastrear, supervisar y tramitar todos los activos podría gestionarse con Blockchain. En consecuencia, Blockchain se puede utilizar para cualquier ámbito, incluido el area de Internet de las Cosas.

Es importante considerar los conceptos técnicos de Blockchain para comprender las consecuencias de las diversas arquitecturas con respecto a la seguridad, el rendimiento y la privacidad. Existe una variedad de diferentes tecnologías basadas en Blockchain que fueron desarrolladas para resolver varios problemas. Por lo tanto, para diferentes necesidades existen tecnologías disponibles más o menos diferentes.

En general, Blockchain es una plataforma digital que mantiene el historial completo de todas las transacciones entre usuarios a través de la red. Además, Blockchain es una base de datos para proporcionar transacciones, su aplicación más popular se da en la moneda digital, como las redes Bitcoin y Ethereum, de las que hablaremos más adelante. Todas las transacciones que se crean entre usuarios se verifican mediante algoritmos criptográficos para luego agruparlos en bloques que se agregan a Blockchain. Nadie puede cambiar la información en bloques porque están encadenados entre sí. Con respecto a Bitcoin, cada nodo de la red tiene su propia copia de Blockchain, sincronizada con otros nodos utilizando un protocolo de peer-to-peer [2]. Esto demuestra la ausencia de necesidad de una autoridad central y, en consecuencia, conduce a la confianza de los participantes generando integridad desde cualquier entidad individual. Blockchain permite procesar diferentes transacciones y llegar de forma segura al consenso sin terceros.

1.2. Objetivo

El objetivo de este Trabajo Fin de Grado es estudiar y desarrollar una solución basada en la tecnología de Blockchain que permita a una red de dispositivos de bajo coste ser escalable, conservar la integridad de los datos que transporta y almacena, y por último, albergar la capacidad de cifrar la información enviada superando los desafíos planteados para la seguridad en el ámbito IoT mencionados anteriormente.

Para ello se emplea un escenario base, tal como se puede ver en la Figura 1.2. Posteriormente se crean diferentes situaciones en las que se intenta corromper el funcionamiento normal de la red comprobando si realmente la red deja de funcionar de manera correcta. Con los datos recogidos de esta simulación se planteará de forma teórica si la implementación de la solución desarrollada podría ser capaz de corregir las deficiencias encontradas. Finalmente se aplicará la solución planteada y de forma práctica será analizado el comportamiento, valorando las mejoras y los inconvenientes surgidos, así como las conclusiones y las lineas futuras para la mejora.

[2]Protocolo de red entre pares.

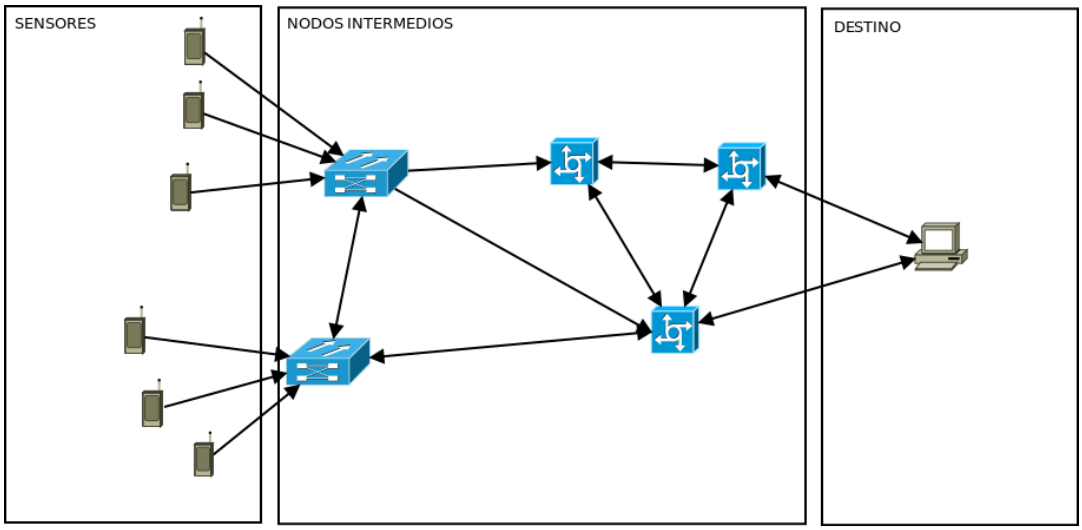

FIGURA 1.2: Ejemplo de escenario de pruebas

1.2.1. Objetivos específicos

Los objetivos detallados planteados serán los siguientes:

1. Identificar los principales problemas de seguridad en IoT.
2. Analizar las soluciones existentes en el mercado.
3. Establecer las diferencias teóricas entre el uso y la ausencia del modelo empleado.
4. Instalar y configurar el nodo que recoge la información de los sensores
5. Instalar y configurar los nodos que proporcionarán volumen a la red.
6. Instalar y configurar el nodo que recibe la información y la muestra por pantalla.
7. Analizar de forma práctica las ventajas y desventajas del uso de la tecnología empleada con respecto a otras ya existentes.

1.3. Metodología

Para la realización de este trabajo, primeramente se realiza un estudio teórico del estado actual de IoT con objeto de cumplir los objetivos 1, 2 y 3 planteados en el punto anterior de forma que se contextualice el ámbito de trabajo en el que se quiere intervenir. Para la consecución de los objetivos específicos 4, 5 y 6 se ha seguido la metodología Scrum [8] enfocada en la posibilidad de segmentar problemas complejos en tareas de menor dificultad con objeto de maximizar la productividad. Esta metodología está enfocada para grupos de trabajo principalmente, pero no por ello deja de ser una herramienta valiosa en el momento que se pretende abordar un proyecto amplio por un solo individuo, ya que permite regular las expectativas sobre el tiempo invertido en los objetivos a cumplir, generar resultados visibles desde las fases iniciales del proyecto, aumentar la productividad y proporcionar flexibilidad ante modificaciones técnicas o de requisitos. En concreto para la elaboración de este trabajo se han seguido los siguientes pasos:

- Planteamiento del objetivo del trabajo.
- División del objetivo final en tareas de corta duración.
- Establecer semanalmente las tareas que han de ser realizadas.
- Al inicio de cada semana identificar que es necesario para cumplir las tareas.
- Al finalizar la semana revisar los objetivos que han sido cumplidos, cuales no y el motivo por los que estas tareas no han podido ser realizadas.

Por último, para concluir el trabajo se pone en valor la solución planteada con respecto a las existentes, analizando las posibles ventajas y desventajas que pudiera presentar para resolver los desafíos previamente planteados.

1.4. Estructura de la memoria

El presente documento se ordena en torno a una serie de capítulos que siguen una estructura marcada por las distintas fases del proyecto.

Para el **Capítulo 2** se ha realizado una investigación sobre el Estado del Arte en este sector, ya que han sido muchos los documentos consultados encontrando gran cantidad de información, algunos de ellos con datos desactualizados, que en cierta medida demuestra como en un corto periodo de tiempo estás tecnologías evolucionan de manera significativa.

A continuación, en el **Capítulo 3** se hace un repaso de algunas de las soluciones existentes indicando sus ventajas y desventajas sobre sus competidores para después, comparar el impacto de la solución propuesta en el ámbito empleado.

El **Capítulo 4** detalla el funcionamiento y la estructura del software que aplica los principios estudiados en los capítulos anteriores.

El **Capítulo 5** tiene como objeto el análisis del funcionamiento de la solución propuesta en una serie de escenarios que reproducen algunos ejemplos de redes IoT. Se realizarán pruebas de estrés a la integridad de los datos generados por los sensores durante el intercambio entre dispositivos y su almacenamiento.

Por último, en el **Capítulo 6** se extraen conclusiones sobre la repercusión en la integridad de los datos en la red usando el sistema planteado, así como una serie de líneas futuras abiertas tras este proyecto.

Capítulo 2

Estado del Arte

A lo largo de este capítulo se recopila el estado en el que se encuentra IoT en la actualidad profundizando en los desafíos a los que se enfrentará en un futuro cercano. Además se analizarán tecnologías enfocadas en la seguridad e integridad de la información con objeto de conocer sus características de forma previa a su uso en posteriores capítulos.

2.1. Internet of Things

La gran cantidad de información que abarcan los conceptos tratados en este estudio, es obligado separar la información en apartados en busca de una mayor claridad. Para ello se estructurará en cuatro bloques:

- En primer lugar una **introducción** sobre las tecnologías que se van a estudiar y su aplicación práctica en la actualidad.
- En segundo lugar, se habla del estado actual de la tecnología **blockchain** y su aplicaciones, fundamentalmente en las criptomonedas como Bitcoin.
- En el tercer apartado se revisará la **criptografía asimétrica** y las aportaciones desarrolladas para IoT que posee.
- Por último se analizará las soluciones existentes en el mercado con sus ventajas y sus inconvenientes, así como una pequeña comparativa entre ellas.

2.1.1. Introducción

El término *Internet of Things*[1] [9] (IoT) fue acuñado por Kevin Ashton en una presentación de 1998 para referirse a una arquitectura emergente, basada en la interconexión global de dispositivos a través de Internet. IoT tiene el objetivo de proveer una infraestructura que facilite el intercambio de información entre "objetos" de forma transparente y así sobreponerse a la distancia entre los elementos del mundo físico y su representación en los sistemas informáticos. El ciclo de vida de IoT generalmente sigue los siguientes pasos: etapa de **recolección** donde los sensores recolectan información del mundo físico a su alrededor que servirá para una toma de decisiones del sistema. **Comunicación**, en esta etapa se envía la información recolectada hasta un lugar deseado. Posteriormente esta información se **almacena** para su posterior uso. La información recolectada se **analiza** con la finalidad de generar patrones de actuación. Y por último se **actúa** en base a lo anterior con la finalidad de solventar una situación o mejorar las condicciones existentes

[1] Internet de las Cosas.

2.1.2. Características de IoT

IoT tiene un gran ámbito de aplicación, siendo difícil definir unas características comunes para todos los casos. Por ello, atendiendo al libro de Keyur K. Patel y Sunil M. Patel sobre IoT[10], se establecen como características más comunes y destacables las siguientes:

Inteligencia: Iot es una combinación de *hardware* y *software* que junto a algoritmos de computación otorga la capacidad de una respuesta "inteligente" permitiendo actuar acorde a las distintas situaciones que se puedan dar en el entorno

Conectividad: Uno de los puntos fuertes de la *Internet de las Cosas* es su capacidad para interconectar distintos dispositivos, otorgando la capacidad de crear una red de objetos inteligentes que compartan información.

Heterogeneidad: Los elementos que forman las redes IoT son muy diferentes al estar implementadas sobre una gran variedad de plataformas *hardware* permitiendo la interacción con otros sistemas y servicios a través de las diferentes redes.

Dinamismo: El estados de los dispositivos cambia muy activamente: conectado y desconectado, movimiento y estacionamiento. Todo dependiendo del contexto en el que se encuentre, así como el número de dispositivos existentes que también puede cambiar dinámicamente.

Escalabilidad: La cantidad de aparatos que potencialmente serán añadidos a una implementación debe considerarse muy superior a todos aquellos que ya se encuentran conectados.

Seguridad: A la vez que se ganan beneficios con la información que se recolecta y se trata, crece la necesidad de conservar los datos seguros, siendo necesario un algoritmo que permita garantizar la seguridad a la vez que la red escala.

2.1.3. Aplicaciones de IoT

Actualmente el ámbito de aplicación de IoT es muy diverso, pues su uso en la sensorización y como controlador de equipos, propicia que su expansión por la industria manufacturera, el control medioambiental, la agricultura o la automatización en los hogares[11] sea muy acentuada dado las ventajas que supone frente a la competencia.

En la **industria manufacturera**, una infraestructura IoT puede ser usada para monitorizar cualquier evento o cambio en el proceso de control, dinamizando la detección de errores y optimizando la producción[12].

Para la **agricultura** contribuye significativamente el poder identificar el estado de la tierra, la climatología y otros parámetros que permiten optimizar la calidad de la cosecha, así como la posibilidad de estimar la producción. Esto ha desembocado en el desarrollo de nuevas técnicas de cultivo más idóneas[12].

El **control ambiental** permite monitorizar el estado de los componentes de un ecosistema, ya sea controlando la calidad del agua, la polución o los movimientos de la fauna[12]. Además se ha demostrado que con diversos sensores estratégicamente

colocados es posible detectar con más antelación que con los métodos tradicionales fenómenos peligrosos para el hombre como terremotos y tsunamis[13].

En el ámbito doméstico la inclusión de **domótica** donde se hacd uso de dispositivos IoT cada vez es más común, sobre todo en las nuevas edificaciones debido a las ventajas que ello conlleva. Sus objetivos son tres fundamentalmente[14]:

- Eficiencia energética del edificio.
- Monitorizar hábitos de los ocupantes optimizando el consumo de electricidad.
- Integración de dispositivos inteligentes para el desarrollo de nuevas aplicaciones.

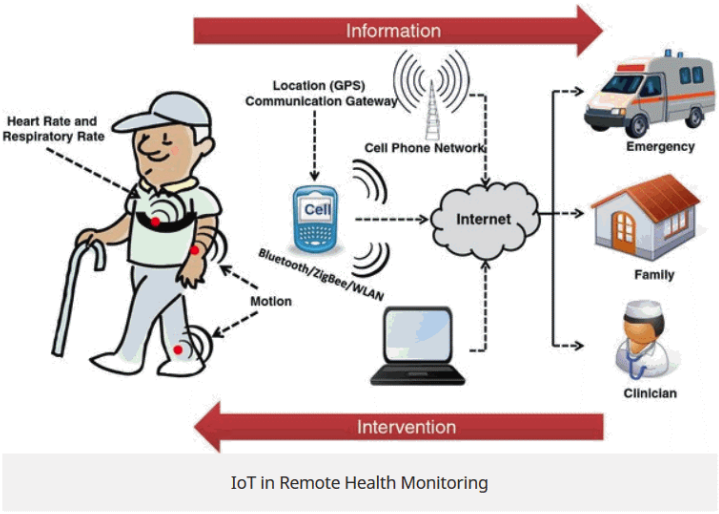

FIGURA 2.1: Monitorización de personas con problemas de salud

En **medicina** es usada para monitorizar patrones de salud y sistemas de notificación a los servicios sanitarios en caso de personas con salud delicada. Aunque la tendencia es que más allá de problemas de salud, toda persona posea algún tipo de sensor corporal. Estos reciben el nombre de "wereables" y permiten monitorizar otros parámetros como la higiene del sueño o el ejercicio realizado diariamente[15]. La figura 2.1 identifica un flujo de acción para personas vulnerables sin que sea necesaria la intervención del usuario.

Estas aplicaciones son un ejemplo de entornos en los que la corriente IoT se está haciendo fuerte, pero cada vez son más los campos que abarcan, pues ya forman parte de otros como "smartcities"[16], logística, conducción autónoma, etc.

2.1.4. Organizaciones de estandarización

Con lo visto hasta ahora es fácilmente imaginable que con la gran variedad de formas que puede adoptar uno de estos dispositivos, unificarlos todos bajo las mismas reglas se presenta complicado. Para ello, y pensando fundamentalmente en la interconectividad, varias entidades de estandarización se ocupan de crear una reglas comunes con la finalidad de que la mayor cantidad posible de dispositivos puedan compartir información. Las organizaciones más conocidas son:

- IEEE propone el estándar IEEE 802.15.4[17] de capa física y acceso. Este protocolo sirve de base para dos de los más conocidos como son ZigBee y 6LoWPAN.

- IETF aboga por el uso de TCP/IP en las capas de red y transporte.
- OCF tiene diseñado un protocolo de capa de servicio llamado CoAP[18] especialmente orientado a sensores por su bajo consumo energético y de memoria.
- OMA proporciona dos estándar llamados OMA DM y OMA LWM2M[19] orientados principalmente al intercambio de datos en redes M2M[2].

Los protocolos mencionados son todos de código abierto, pero también existen muchos protocolos propietarios, principalmente de fabricantes que buscan desmotivar a sus clientes a migrar su hardware a otras compañías una vez adquiridos sus productos.

2.1.5. Fragmentación

Como se viene mencionando, en la plataforma IoT se da la existencia de una gran variedad de dispositivos que aportan información de su entorno, y también de las formas de comunicación que tienen entre ellos. Como consecuencia, en los sistemas heterogéneos se combinan multitud de tecnologías y pasarelas entre ellas, generando una complejidad muy marcada, que repercute directamente en la seguridad de forma negativa.

2.1.6. Arquitectura IoT

Desde que se empezó a hablar de la comunicación entre dispositivos ha habido una gran cantidad de propuestas para dar solución a esta nueva necesidad. Actualmente la arquitectura IoT consiste en diferentes capas de tecnologías relacionándose entre ellas. A modo de ejemplo, la Figura 2.2 permite ver como se relacionan las tecnologías entre ellas. Las funcionalidades de cada una de ellas se describe a continuación [10]:

Capa de sensor. La capa más baja está compuesta de objetos inteligentes integrados con sensores. Estos sensores habilitan la interconexión del mundo físico y digital proporcionando información en tiempo real para ser almacenada y procesada. Hay varios tipos de sensores con diferentes propósitos. El sensor tiene la capacidad de tomar medidas como la temperatura, calidad del aire, velocidad, humedad, presión ambiental, flujo de aire, movimiento, corriente, etc. En muchos casos, estos sensores pueden tener una pequeña memoria permitiendo almacenar un número limitado de medidas. Un sensor puede medir propiedades físicas y convertirlas en una señal que sea entendible por un instrumento de medición. Los sensores son agrupados de acuerdo a sus objetivos: ambientales, corporales, sensores para el hogar, telemetría para vehículos, etc. La mayoría de estos sensores requiere de conectividad con un *gateway* de la misma forma que una red LAN con Ethernet o Wi-Fi. Para los sensores que no necesitan estar conectados con un concentrador de red, su conectividad con el servidor o aplicación puede darse directamente en una red amplia como GSM, GPRS o LTE. Los sensores de baja potencia y de corto alcance, normalmente forman redes denominadas WSN[3]. Estas redes están ganando popularidad al permitir ampliar redes de sensores a la vez que conserva un consumo bajo de batería y cubre grandes áreas

[2]Machine to Machine.
[3]Wireless Sensor Networks.

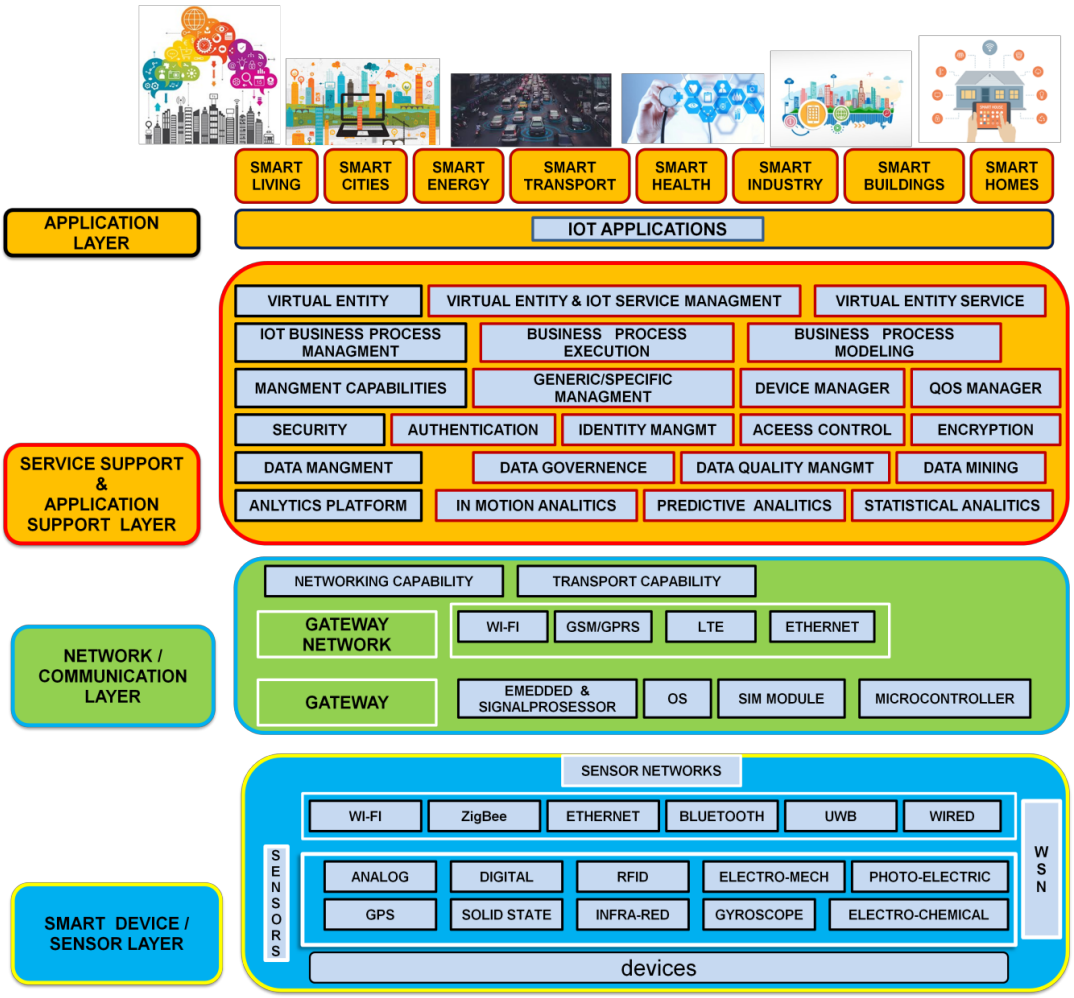

FIGURA 2.2: Arquitectura IoT

Redes y Gateways. La gran cantidad de datos que producen los sensores requiere una infraestructura robusta como medio de transporte, más allá de la que han utilizado las redes hasta la fecha. Fundamentalmente, se requiere para dar servicio a una amplia variedad de aplicaciones que demandan alta velocidad, además de soportar la gran variedad de tecnologías y protocolos que conviven en una red tan heterogénea como son las de IoT. Estas redes pueden ser de ámbito privado, publicas o mixtas, y son construidas para soportar los requerimientos de la comunicación: latencia, ancho de banda o seguridad.

Capa de Administración de Servicios. El servicio de gestión representa el procesamiento de información posible a través de análisis, controles de seguridad, modelado de procesos y gestión de dispositivos. Una de las características más importantes de la administración de servicios es el conjunto de reglas que permitan su gestión adecuada. IoT proporciona interacción entre objetos y sistema proporcionando información en forma de eventos o de telemetría como temperatura o localización. Algunos de estos eventos requieren ser filtrados o reenviados para ser procesados por sistemas de sensorización, mientras que otros requieren respuesta a situaciones inmediatas, como puede ser reaccionar a emergencias relacionadas con la salud de pacientes. Este conjunto de reglas debe soportar formulas para tomar decisiones

además de disparadores que inician respuestas automáticas para crear un sistema más eficaz.

Capa de Aplicación. Las aplicaciones IoT proporcionan entornos "inteligentes" que optimizan los recursos del ámbito de aplicación como transportes, construcción, agricultura, salud, etc.

2.1.7. Topología

A diferencia de las topologías tradicionales de red donde se distinguen entre cableadas e inalámbricas y cada una de ellas tienen múltiples formatos, en IoT predominan las redes inalámbricas en las que se distinguen dos formatos [20]: **estrella** y **malla**, representados en la figura 2.3.

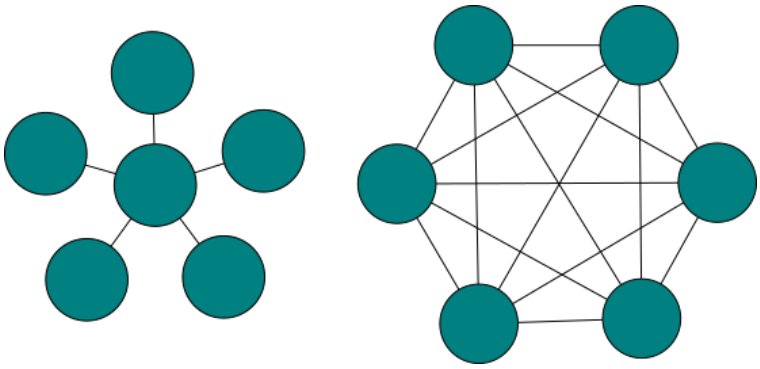

FIGURA 2.3: Topología en estrella (izquierda) y en malla (derecha).

Estrella En una topología de estrella, todos los nodos están conectadas a uno central, que normalmente ejerce de pasarela con exterior o al nivel superior de la red, este nodo central se le denomina AP[4] y el resto de los nodos de la red reciben el nombre de *estaciones*.

Malla En una red en Malla, cada nodo puede estar conectado a uno o varios de los otros. En este modelo, uno o varios de los nodos proporcionan conexión con el exterior. El beneficio de esta configuración principalmente es la resiliencia a fallos en la conexión entre nodos, permitiendo rutas alternativas a los datos para llegar al destino. La contrapartida de este sistema es la mayor dificultad en la configuración de la red, así como la mayor saturación de mensajes en la red a medida que el número de nodos es mayor.

2.1.8. Seguridad

Además de todas las ventajas mencionadas, IoT también supone nuevos riesgos y desafíos debido a que los dispositivos envían información recopilada tan sensible como la salud de un paciente, siendo en este caso, la privacidad y la integridad de la información piezas clave [21]. Estas necesidades ya se venían dando en las redes tradicionales, sin embargo en una red IoT algunas características requieren un nuevo enfoque:

[4] Access Point.

- El numero de dispositivos conectados ya es superior al números de personas en el planeta y se prevé que continuará creciendo como se muestra en la Figura 2.4 [22]. Y este hecho supone un desafío para el control de la privacidad en todos ellos.

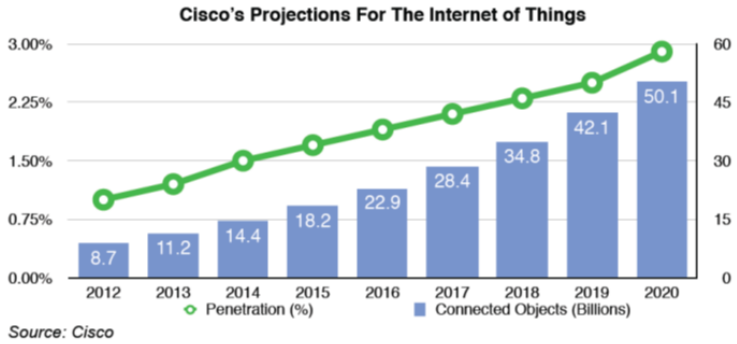

FIGURA 2.4: Previsión de dispositivos IoT para 2020

- Las redes que siempre han sido heterogéneas lo serían todavía más al integrar las diferentes soluciones que existen en el mercado.
- En contraposición a los servidores tradicionales que están protegidos en salas habilitadas al uso y con personal cerca, la configuración IoT fomenta que sensores puedan estar en cualquier parte estando expuestos a robo o alteración.
- Los dispositivos no tienen acceso a una fuente de alimentación en todas las ocasiones y se ven forzados a usar baterías que, con la finalidad de alargar su vida útil implementan una menor potencia de proceso y unas capacidades más limitadas en general.
- Con el incremento de dispositivos conectados, la cantidad de información generada aumenta y el coste de mantenimiento sube, pudiendo verse reducidas las medidas de redundancia para evitar su perdida.
- En los dispositivos actuadores es necesario un mayor control debido a que en caso de vulneración pueden poner en peligro a las personas, véase el caso de los coches autónomos.

2.2. Cifrado

En la actualidad, existe un gran número de procesos que incluyen el intercambio de información a través de redes que no tienen implementados mecanismos de seguridad, como por ejemplo Internet. No resulta posible asegurar que un mensaje no se pueda copiar y reenviar. Por este motivo, es necesario utilizar una aproximación diferente que garantice que el mensaje únicamente será leído por su legítimo destinatario. Para ello se utiliza la encriptación, que consiste en la codificación del mensaje según un código que solamente el emisor y el receptor conocen.
En esta sección se pretende contextualizar e introducir algunos conceptos asociados a la seguridad, que utilizan distintos mecanismos de criptografía, los cuales serán imprescindibles para comprender la solución propuesta a los desafíos de IoT. Sin embargo, al no ser objeto de estudio este tipo de herramientas y considerar que los protocolos que se explicarán a continuación están altamente probados y llevan largo

tiempo usándose en el ámbito de la seguridad informática, no se profundizará en su demostración a nivel matemático, dándose por supuestas las ventajas y desventajas que otorgan a aquellos sistemas en los que se implementan.

2.2.1. Modelo de cifrado

Para comprender mejor cómo funciona la encriptación y los elementos implicados, partiremos del modelo planteado en la figura 2.5 [23].

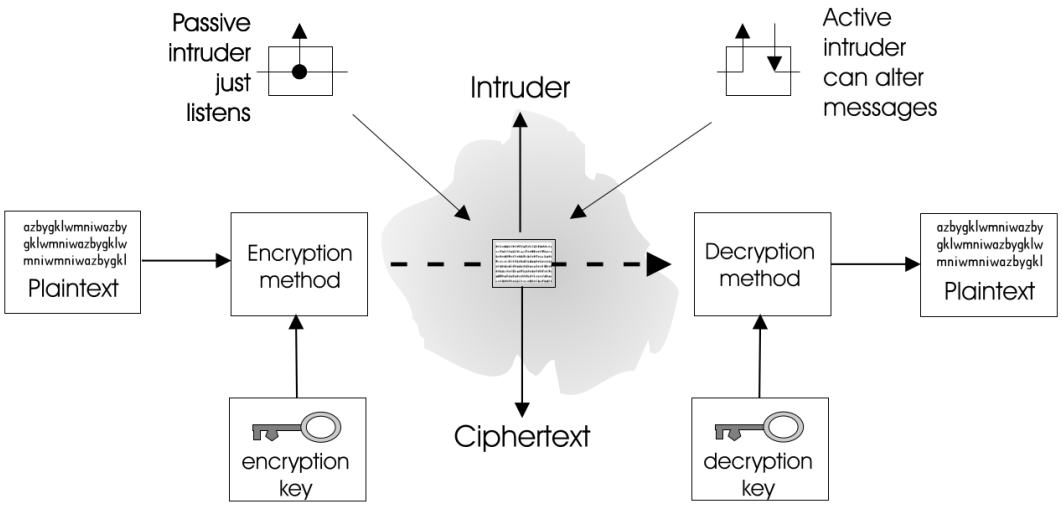

FIGURA 2.5: Modelo de encriptación [23]

En primer lugar, se pretende cifrar una información conocida como *texto plano* haciendo uso de un parámetro secreto denominado *clave* o *key*. Este proceso se conoce como encriptación. Su salida se denomina *texto cifrado*, y es lo que se transmite en un proceso de comunicación en el que se quiere mantener la información no legible al resto de usuarios no autorizados. En ocasiones, al usar canales de comunicación no seguros, como puede ser Internet, es posible que la información pueda ser escuchada por otros usuarios, denominados *intrusos pasivos* o interceptada de forma intencional, por los llamados *intrusos activos*. Y son en estos últimos casos en los que el uso de la encriptación evita que terceros puedan tener acceso a información confidencial.
La seguridad de la encriptación reside en la clave y su longitud, pues según aumenta ésta última, crecen los posibles valores que pueda tomar y, por lo tanto, también el esfuerzo que supone romper dicho cifrado.
En función de la clave que utilicen, existen tipos de encriptación: simétrica y asimétrica.

2.2.2. Cifrado simétrico

La encriptación simétrica se caracteriza por emplear la misma clave tanto durante el proceso de encriptación del mensaje en el lado del emisor, como durante el desencriptado del mismo. Un ejemplo sería el observado en la figura 2.5.
No obstante, la encriptación simétrica presente un problema difícil de resolver: el **Intercambio de claves**. Esto se produce debido a que las partes o entidades que desean comunicarse han de intercambiar la clave compartida antes de establecer cualquier comunicación segura y, además, han de mantener dicha clave en secreto. Aunque

podría realizarse un intercambio directo, en ocasiones no es posible debido a factores de riesgo, inconveniencia o coste.

Por otro, existe el inconveniente de que sería necesaria una clave distinta por cada individuo o entidad con la que se quisieran intercambiar mensajes cifrados.
Toda esta problemática suscitó la creación de un nuevo mecanismo para cifrar los mensajes, la encriptación asimétrica. Este método de encriptación, del que se hablará a continuación, es empleado en gran parte de las comunicaciones a día de hoy.

2.2.3. Cifrado asimétrico

En la década de 1970, Martin Hellman, Whitfield Diffie e, independientemente, Ralph Merkle inventaron una forma de resolver el intercambio de claves y los problemas de confianza de la criptografía simétrica reemplazando la clave secreta, compartida y única, por un par de claves relacionadas matemáticamente. De éstas claves, una puede ser conocida públicamente, mientras que la otra debe mantenerse en secreto por el individuo que las generó. Las ventajas de este mecanismo son inmediatas. En primer lugar, no se requiere ningún acuerdo para intercambiar la clave secreta por adelantado, ya que la única clave que debe compartirse con la otra parte es una clave pública que se puede compartir de forma segura con cualquiera. En segundo lugar, mientras que la seguridad de un algoritmo simétrico depende de que dos partes mantengan con éxito una clave en secreto, un algoritmo asimétrico solo requiere que la parte que generó el *par de claves* mantenga una de ellas en secreto. Esto resulta menos problemático en las implementaciones. Tercero, el tema de confiar en la otra parte desaparece en muchos escenarios, ya que sin el conocimiento de la clave secreta, esa parte no puede realizar ciertas acciones ilícitas, como firmar digitalmente un documento con su clave privada o divulgar su clave secreta a otros. Cabe mencionar, no obstante, que la encriptación asimétrica no constituye un sustituto de la simétrica, sino que ambas se complementan. Por ejemplo, los algoritmos simétricos suelen ser más rápidos y seguros para un determinado tamaño de la clave [23].

2.2.4. Funcionamiento

Para usar una criptografía asimétrica, el usuario que desea enviar un mensaje utiliza un par de claves público/privado, donde la clave privada se genera de forma aleatoria y la pública a partir de la privada [23]. Este usuario permite que todos tengan acceso a la clave pública. Luego, cuando el segundo usuario tenga alguna información secreta que le gustaría enviar al primero, encriptará estos datos utilizando un algoritmo asimétrico apropiado y la clave pública generada por el primero. El texto cifrado resultado de esta operación será lo que finalmente se envíe. Cuando el mensaje llegue al primer usuario, éste hará uso de su clave privada para desencriptarlo. Por este motivo, cualquiera que desconozca la clave privada no podrá recuperar el texto original desde el texto cifrado. Solo quien tenga la clave privada correspondiente a la clave pública con la que se ha cifrado puede descubrir el texto original. La Figura 2.6 muestra cómo se usa la criptografía asimétrica.

Estableciendo una analogía, el cifrado simétrico es análogo al uso de una caja fuerte, donde una combinación numérica sirve tanto para abrir la caja como para cerrarla. Mientras que con el cifrado asimétrico existen dos códigos, uno para abrir y otro para cerrar. Manteniendo una de las dos combinaciones secretas y haciendo publica la otra, se puede controlar quien deja o extrae información de la caja fuerte.

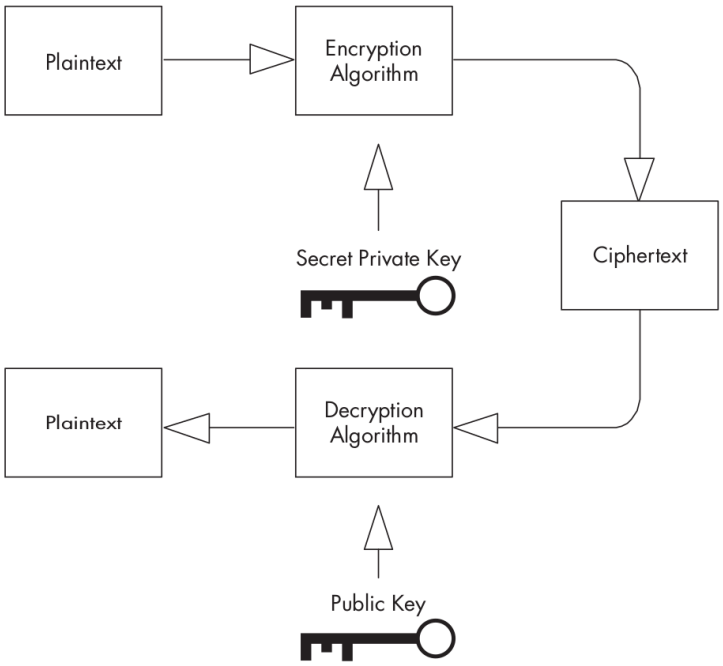

FIGURA 2.6: Esquema de funcionamiento de la criptografía asimétrica [23].

Este método aporta varias propiedades fundamentales en lo referente a seguridad: confidencialidad, integridad, autenticación y no repudio.

Confidencialidad. En el intercambio de mensajes explicado anteriormente, aun conociendo solo la clave pública pero no la privada, se tiene la capacidad de cifrar datos sin que nadie que carezca de dicha clave privada pueda recuperar la información original. En este caso de uso, se está garantizando la confidencialidad de la información sin tener que transmitir la clave secreta de cifrado.

Integridad. Dado que en un intercambio de mensajes, éstos estarán encriptados de manera que solo puedan ser leídos por las personas autorizadas, si se produjese cualquier alteración el contenido de los mensajes carecería de sentido una vez desencriptados.

Autenticación. Además de salvaguardar el contenido de mensajes confidenciales, existe otro caso de uso para la encriptación asimétrica. Supongamos un segundo escenario, en el que solo una persona posee la clave privada de un par público/privado. Con esta clave es posible codificar ciertos datos, de manera que únicamente puedan ser decodificados mediante la clave pública. Este proceso se conoce como *firmar*, pues es posible verificar que la información enviada solo ha podido ser generada por el poseedor de la clave privada y **ningún otro**, es decir, su autenticidad. Aunque, una tercera entidad pretendiera suplantar al emisor, quedaría descubierto al no coincidir su clave privada con la del emisor original.

No repudio. De la propiedad anterior se deriva la garantía de no repudio, pues solo el poseedor de la firma privada puede generar un mensaje que pueda descifrarse con la clave pública.

2.2.5. Funciones matemáticas para la encriptación asimétrica

A la hora, tanto de cifrar mensajes como de firmarlos, se hace uso de una función parametrizada por una clave. El tipo de función empleada ha de cumplir básicamente dos requisitos:

- Ha de ser una función *trampa*. Estas funciones reciben este nombre debido a que resulta muy sencillo resolverlas en su forma directa, mientras que son mucho más complejas en su forma inversa, si no se conoce un parámetro adicional.
- Han de ser irreversibles. Aunque por definición una función irreversible es aquella para la cual resulta imposible calcular su inversa, únicamente lo es a efectos prácticos pues en la teoría sí sería reversible (ha de serlo para permitir el descifrado).

Si se juntan ambas propiedades, el resultado es una función irreversible con trampa, es decir, una función fácil de calcular pero cuya inversa resulta inviable en términos de computación, a excepción de que se disponga de cierta información que, en este caso, sería la clave. No existen muchas funciones que cumplan estos requisitos y que den lugar a algoritmos prácticos y eficientes, que puedan considerarse seguros. En el siguiente apartado, se verán, a modo de ejemplo, algunos aspectos del algoritmo RSA.

2.2.6. El algoritmo RSA

Se trata de uno de los algoritmos más populares a día de hoy. Sus siglas hacen mención a sus inventores: Rives, Shamir y Adelman. Este algoritmo fue desarrollado en 1978 en respuesta a los conceptos de clave pública y privada introducidos por Hellman, Diffie y Merkel. La idea fundamental de este algoritmo reside en el problema de factorizar números enteros grandes en factores primos. A continuación se comentan algunos detalles más de este algoritmo:

- **Seguridad**: se basa en el problema de la factorización de números primos y en el problema RSA, que hace referencia a la dificultad de realizar operaciones de clave privada, conociendo solo la pública. La seguridad del algoritmo depende también de la longitud de la clave, por lo que para un número lo suficientemente grande resulta seguro. La clave más larga que se ha conseguido romper era de 1024 bits, aunque la operación duró 100 horas. Hoy en día, las claves de 2048 bits se consideran seguras debido a que se requeriría varios cientos de años para vulnerarla.
- **Velocidad**: RSA al ser un algoritmo de cifrado asimétrico, por definición, su tiempo de procesado es superior a los cifrados simétricos.
- **Distribución de claves**: como para cualquier método de cifrado ha de ser segura, para evitar ataques de *replay*. En estos casos el atacante envía una clave pública a la víctima, de manera que ésta cree que procede de otra persona o entidad. De esta manera, la víctima creerá que se comunica con esa otra persona y no con el atacante.

2.3. Blockchain

Blockchain, en esencia, es una base de datos distribuida que registra todas las transacciones digitales que han ocurrido y han sido compartidas en un grupo de

usuario a través de redes de pares[5][24]. Blockchain tiene dos características fundamentales:

- En primer lugar, el registro es **público**. Cualquier persona puede revisar todo el registro generado debido a que no está mantenido por un organismo central, si no por la comunidad que genera y guarda el registro.
- La segunda principal característica es la encriptación. Mediante clave privada y pública se puede garantizar el origen de la información de forma inequívoca. La figura 2.7 ejemplifica de forma introductoria como trabaja la tecnología, en este caso referida a Bitcoin. Las transacciones se agrupan en bloques que poseen una marca de tiempo donde se refleja el instante del tiempo en que se crea. Estos bloques quedan enlazados unos a otros a modo de cadena, de forma lineal en orden cronológico, donde cada bloque almacena el *hash* del anterior bloque.

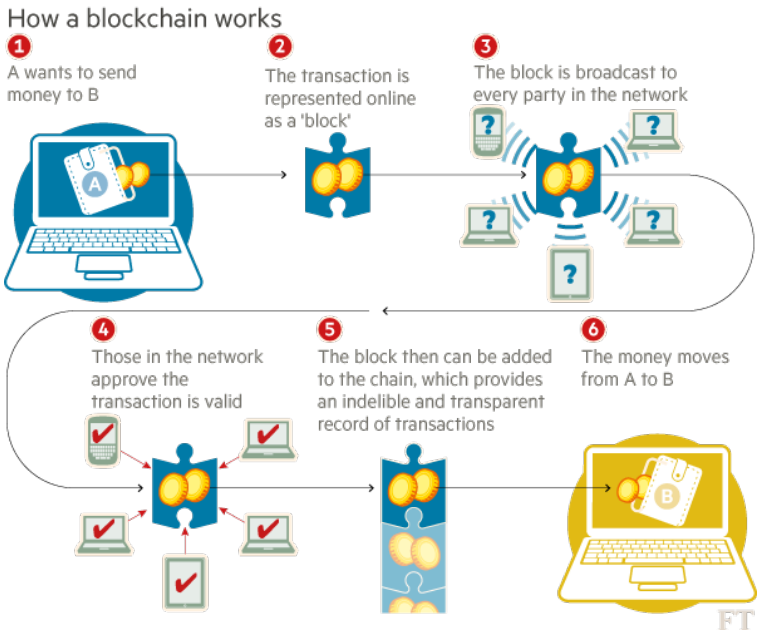

FIGURA 2.7: Transacción mediante Blockchain [25].

2.3.1. Conceptos Previos

Los conceptos técnicos fundamentales para entender la arquitectura de Blockchain son los siguientes [24]:

- **Nodo**. Es un sistema informático con el software necesario para mantener una cadena blockchain. Todos los nodos están conectados a la red blockchain para que puedan recibir y enviar transacciones.
- **Red**. Es el resultado de la cooperación de todos los nodos que ejecutan el software para comunicarse entre ellos.

[5]peer-to-peer.

- **Smart Contracts**. Estos son acuerdos contraidos por dos o más partes convertidos en código para ser incluidos en la cadena Blockchain. Estos acuerdos establecen las condiciones necesarias para realizar una operación, en el caso de las criptomonedas, una (o más) transacciones.
- **Transacción**. Cuando los usuarios generan nuevos datos para las transacciones, se envían a los nodos de la red en un proceso de *broadcasting*.
- **Validación de transacción**. Todas las transacciones son validadas criptográficamente por los nodos en la red Blockchain. Las transacciones inválidas son ignoradas.
- **Bloque**. Es un grupo de transacciones recogidas por nodos en un paquete. Para que los bloques sean válidos deben formarse de acuerdo con un conjunto predeterminado de reglas: no deben exceder un tamaño máximo en bytes, contener más de un número máximo de transacciones y deben hacer referencia al bloque válido más reciente.
- **Ledger**. Es una cadena de bloques organizada por el siguiente sistema: cada bloque nuevo se adjunta al bloque válido más reciente.
- **Consenso**. Es un acuerdo de todos los nodos en Blockchain. Para habilitar el funcionamiento del sistema distribuido, varios procesos cooperan entre sí. Las fallas en tales sistemas pueden ocurrir en cualquier lugar, es por eso que usan protocolos de consenso.
- **Función hash**. Es una función unidireccional que refleja una entrada de tamaño variable a una salida de tamaño fijo llamada hash. Propiedades de una función hash criptográfica:
  - Generación sencilla del hash dada la entrada.
  - Imposibilidad de generar la entrada original dado el hash.
  - Estadísticamente imposible que se de el caso en el que dos entradas similares tengan la misma salida, en lo que se denomina "colisión". SHA256 es un ejemplo de función hash criptográfica.

El esquema de la figura 2.8 puede facilitar la visualización de como alguno de los conceptos anteriores se relacionan entre sí.

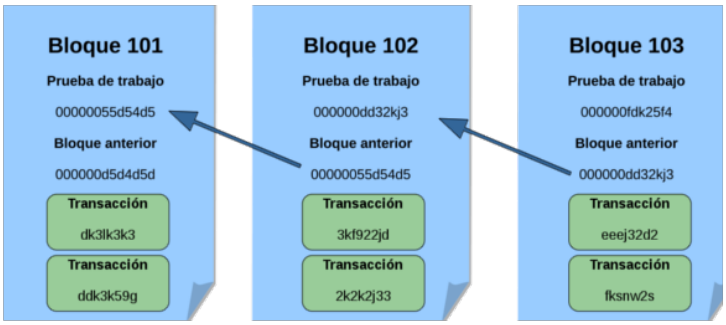

FIGURA 2.8: Formación de bloques

Actualmente, cualquier persona puede usar los ledgers distribuidos que admiten, por ejemplo Bitcoin, para poder formar parte de su red y poder interactuar con

su moneda. Además, todo usuario pueden leer o escribir en dichos ledgers, lo que los hace atractivos para muchas aplicaciones.

Sin embargo, hay aplicaciones en las que las partes implicadas pueden desear mantener su información privada, ya sean transacciones financieras, el intercambio de registros médicos o el envío de bienes. Para estos casos, el uso de Blockchains privados permite determinar el nivel de acceso de cualquier participante en la red para mantener su información bajo control. Para la participación de nuevos miembros en las redes privadas existen varias formas de invitación, por ejemplo, acuerdo unánime, invitación de usuario único o aceptación en base al cumplimiento de un conjunto predeterminado de requisitos.

Blockchain actualmente sigue bajo desarrollo, apareciendo constantemente nuevas tecnologías para su uso en bases de datos. Estas tecnologías son aplicables para muchas industrias diferentes y, como tal, requieren una serie de especificaciones cuyo objetivo principal es optimizar la creación de las cadenas de bloques, resolver la escalabilidad y mejorar la capacidad de producción de las mismas garantizando su seguridad, rendimiento y robustez. Estas áreas están siendo cubiertas por varios tipos de tecnologías de contabilidad distribuida con diversos grados de descentralización como se puede ver en la Figura 2.9.

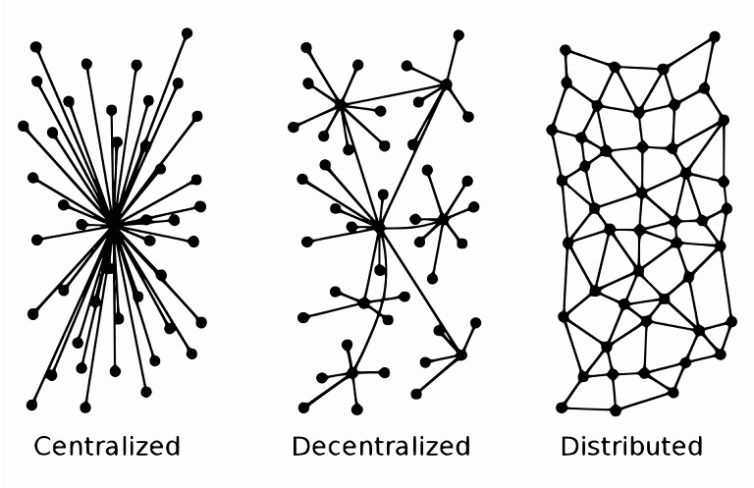

FIGURA 2.9: Grados de descentralización

El estado actual y el pasado de toda la red se almacenan en un nodo de cadena de bloques. A continuación se muestran las métricas cualitativas y cuantitativas que pueden evaluar el rendimiento de una arquitectura Blockchain.

- **Rendimiento de envío:** Es el número máximo de envío de transacciones por segundo posible/permitido por cada nodo y por toda la red.
- **Rendimiento de validación máximo/promedio:** es el parámetro que determina la velocidad de procesamiento de transacción máxima/promedio de la red.
- **Latencia promedio de validación de transacción:** es el período promedio de tiempo necesario para validar la transacción desde el momento de su envío. Esta métrica determina el período de espera de los usuarios para que se valide su transacción y se pueda incluir en un bloque. Cabe destacar que la confirmación del bloque y la noción de validación podrían ser diferentes en cada cadena de bloques.

- **Volatilidad en la latencia:** es la medida de la posible variación del tiempo de procesamiento por transacción.
- **Seguridad:** la evaluación del sistema de seguridad requiere un modelo de amenaza que sea capaz de definir el tipo y el alcance de los ataques al sistema. Tales modelos de amenazas podrían ser diferentes y varían según la aplicación Blockchain. Para la evaluación de seguridad se requieren los siguientes análisis:
  - Transacción e inmutabilidad de bloque.
  - Resistencia a la censura de transacción.
  - Resistencia de denegación de servicio.
  - Requisito de confianza de los usuarios.
  - Confidencialidad de la transacción y anonimato del usuario.
- **Confidencialidad:** es la capacidad de los nodos para ocultar los contenidos de la transacción o incluso la identidad como si los propietarios de los nodos hubieran participado en esa transacción desde otros nodos distintos.
- **Tarifas de transacción:** es el precio que los usuarios deben pagar a la red para procesar transacciones o ejecutar contratos inteligentes. Estas tarifas cubren los costos de mantenimiento de la cadena de bloques y brindan la protección necesaria ante el uso del sistema para tareas computacionales maliciosas.
- **Requisitos de hardware:**
  - **Memoria/almacenamiento:** es la capacidad total que se requiere por nodo.
  - **Procesador:** cantidad de recursos de procesamiento que se requieren para validar transacciones y bloques.
  - **Uso de la red en el tiempo**, incluidos los requisitos de rendimiento y latencia.
  - Los **requisitos** de hardware cambiarán a medida que la red se amplíe.
- **Escalabilidad:**
  - **Número de nodos:** el aumento del número de nodos conduce al cambio del rendimiento del sistema.
  - **Número de transacciones:** el aumento en el número de envíos de transacciones por segundo genera una disminución en el rendimiento del sistema.
  - **Número de usuarios:** el aumento en el número de usuarios activos que envían transacciones conduce a un menor desempeño del sistema.
- **Proceso de validación:** es un factor importante, es necesario para determinar el rendimiento de la red.
- **Complejidad:** es una medida de la dificultad del desarrollo, mantenimiento y operación de la infraestructura de Blockchain.
- **Limitaciones de contrato inteligente:** las principales limitaciones que pueden influir en la capacidad del código implementado en la cadena de bloques son el lenguaje de scripting de contrato inteligente y los protocolos de consenso subyacentes.

2.3.2. Principios

Más allá de la economía digital, Blockchain establece unos principios para la creación de software y servicios [26] detallados a continuación:

- **Integridad de la Red.** El sistema busca el consenso a través de la red de forma algorítmica almacenándola encriptada en la cadena de bloques. Los participantes pueden intercambiar información con la seguridad de que el resto de participantes otorgarán integridad a la transacción. A su vez, ningún participante puede ocultar una transacción, siendo todas totalmente rastreables de principio a fin.
- **Distribuido.** El sistema utiliza una red *peer-to-peer* sin ningún punto de control centralizado que pueda cerrar la red, otorgar prioridad a algunos miembros o tomar posesión de la red en alguna de sus formas.
- **Incentivos.** El modelo de Blockchain incentiva a los llamados "mineros[6]" a crear nuevos bloques y enlazarlo entre ellos.
- **Seguridad.** Todo individuo que participe en la red está obligado a utilizar criptografía. En función de la implementación se usa un sistema u otro. A modo de ejemplo, para bitcoin se utiliza una infraestructura de clave pública (PKI).
- **Privacidad.** Los participantes tienen el control de sus propios datos, pudiendo elegir el anonimato o proporcionar datos personales. La identificación de los miembros va separa de las transacciones.
- **Preservación de derechos.** Los derechos de propiedad son preservados en un registro público llamado *Proof of Existence* (PoE) donde se crean identidades.
- **Inclusión.** La forma en que Blockchain está diseñado hace que pueda ser utilizado en cualquier dispositivo incluso en algunos casos sin necesidad de conexión a Internet.

2.3.3. Tipos de Blockchain

Privado. Para un Blockchain completamente privado, los permisos de escritura son guardados por una organización. Los permisos de lectura pueden ser públicos o privados de manera arbitraria. Dado que las aplicaciones incluyen gestión de bases de datos internos de una sola compañía, la posibilidad de exponer públicamente esta información puede no ser necesaria, o en caso de una posible auditoría si podría serlo.

Público. Este tipo de Blockchain se mantiene principalmente por cualquiera que quiera acceder a sus datos. Esto incluye un proceso de consenso para poder escribir en la cadena. Un Blockchain público es de código abierto que es securizado mediante criptografía y otorgando incentivos a aquellos individuos que aporten verificación a las transacciones mediante algoritmos conocidos como **Proof of Work** (PoW) y **Proof of Stake** (PoS). Un ejemplo de Blockchain público es Bitcoin.

[6]Usuarios dedicados a validar bloques.

Consorcio. Blockchain compuesto por una parte privada y otra pública. El proceso de consensuado se lleva a cabo por unos nodos preseleccionados. Este tipo se utiliza fundamentalmente entre instituciones financieras privadas que para validar una transacción es necesario que sea verificada por las demás. en este caso la posibilidad de lectura puede ser publica o restringida a algunos participantes[27].

Características	Público	Privado/Mixto
Acceso	Lectura/Escritura Pública	Escritura privada - Lectura varía
Velocidad operación	Lenta	Rápida
Seguridad	PoW/PoS	Consensuada
Identidad	Anónima	Conocida

TABLA 2.1: Comparativa general de Blockchain. Público y privado/mixto.

2.3.4. Arquitectura

En esta sección se describe una arquitectura básica de Blockchain con el ledger[7] distribuido. Sin embargo, los elementos de la arquitectura pueden variar dependiendo del tipo de Blockchain y el despliegue que se quiera realizar.

2.3.4.1. Bloque

Blockchain permite una lista de transacciones altamente distribuida. La información es recordada en archivos llamados "bloques". Un bloque es un registro de algunas de las transacciones más recientes que han tenido lugar. El ledger, formado por los bloques que contienen la información y que a su vez están enlazados entre sí recibe el nombre de "Cadena de bloques" o **blockchain**[28]. Un bloque está formado por la cabecera y por el cuerpo[29]. La cabecera consiste en tres metadatos:

- En primer lugar se almacena el hash del bloque anterior, de forma que queda relacionado con el bloque anterior de la cadena de bloques (Figura 2.10).
- El segundo guarda la "dificultad", el timestamp y el nonce. Esta información es necesaria para la validación del bloque.
- El último metadato es una estructura de los datos para poder indexar todas las transacciones del bloque de forma eficiente. Esta estructura recibe el nombre de *árbol de Merkle* [30].

El cuerpo del bloque está formado por todas las transacciones indicadas de forma individual que se entrelazan entré sí como se indica en la figura 2.8.

2.3.4.2. Firma digital

Crear una transacción para la cadena de bloques requiere una firma digital que permita autentificar la transacción. El proceso de firma habitual tiene dos partes: la parte de firmado y la de verificado. Durante la etapa de firmado, primeramente se genera un hash de la transacción, se encripta mediante la clave privada del emisor y se envía la transacción y el hash encriptado. En la segunda etapa, el receptor realiza el proceso inverso, desencripta el hash con la clave pública del emisor y genera

[7] Libro de registros.

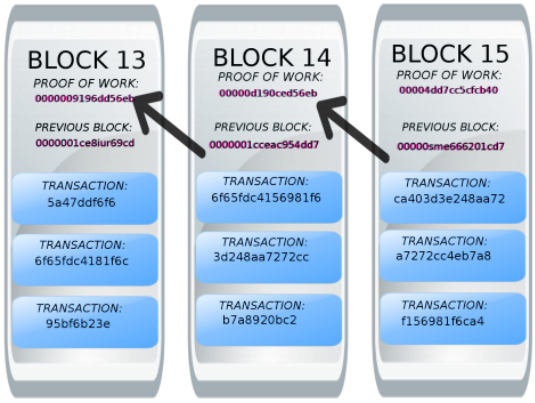

FIGURA 2.10: Enlazado entre bloques.

un hash de la transacción. Si ambos hashes coinciden, la transacción se considera legítima, o fraudulenta en el caso contrario[29].

2.3.4.3. Red descentralizada

La interacción entre usuarios de Blockchain es a través de una red descentralizada en la que cada usuario representa a un nodo. Cuando un usuario realiza una transacción con otro o cuando un nodo recibe información de otros nodos, esta es verificada. Con este mecanismo, los datos validados se propagan por toda la red. El beneficio que produce este funcionamiento es evitar la centralización y por tanto un único punto de fallo.

2.3.4.4. Consenso de red

Viendo el funcionamiento de Blockchain, se puede ver como es necesario la aceptación y verificación por todos los usuarios en la red. Este proceso se conoce como *consenso*. Aunque se puede dar el caso que cada nodo tiene una perspectiva distinta del estado de la red en función de los bloques que se han recibido, siendo necesario un mecanismo que permita a todos tener y verificar el mismo libro de registros. Para ello, fundamentalmente hay cuatro algoritmos de consenso aplicables[29].

Proof of Work (PoW) Hasta la fecha, el sistema de consenso PoW es el más usado. Introducido por Bitcoin, asume que todos los participantes votan con su "poder computacional", resolviendo problemas de cálculo únicos para cada bloque. En el caso de Bitcoin se basa en el cálculo de un hash buscando un número denominado *nonce* que verifique que el hash sea menor que un objetivo marcado[31].

Proof of Stake (PoS) En este algoritmo se utiliza fundamentalmente en las criptomonedas más recientes, y en lugar de resolver mediante una prueba de esfuerzo, el nodo genera un bloque que proporciona la prueba en la que demuestra que posee un numero de monedas antes de ser aceptado en la red[32]. Este método requiere que los usuarios prueben ser propietarios de sus monedas debido a que cuantas más se posean se asume que menor interés se tendrá en vulnerar la red[29]. Como consecuencia de este sistema, solo quien pueda proporcionar PoS puede formar parte del proceso de mantenimiento de la cadena. Su ventaja radica en la eficiencia energética comparada con el consumo de PoW.

Practical Byzantine Fault Tolerance (PBFT) Este algoritmo fue desarrollado para tolerar las faltas bizantinas[33], como por ejemplo, el comportamiento de un nodo que se conecta y desconecta de la red arbitrariamente. Este algoritmo replica una maquina de estados con un solo mensaje para ejecutar operaciones de lectura y dos para ejecutar operaciones de lectura y escritura. Además, usa un esquema de autenticación eficiente basado en mensajes durante el funcionamiento normal. En este caso a criptografía de clave pública solo se usa cuando hay fallos[34].

Delegated Proof of Stake (DPoS) La mayor diferencia entre PoS y DPoS es que la primera es un proceso es directamente democrático, mientras que la segunda es una democracia representativa, los poseedores de "stake" (aquellos con confianza otorgada al poseer criptomonedas en el caso de la economía digital) delegan quien genera y valida un bloque. De esta forma son necesarios menos nodos para validar el bloque, y por tanto puede ser más rápido[29].

2.3.5. Aplicaciones

El ámbito de aplicación de Blockchain es aquel en el que no haya confianza entre las partes que lo componen. Es fácilmente ejemplificable con transacciones financieras donde no hay confianza previa entre vendedor y comprado. Por este motivo es por lo que el ámbito financiero lo ha adoptado tan rápidamente. A continuación se identificarán aplicaciones de Blockchain financieras y no financieras.

2.3.5.1. Aplicaciones financieras

Sistema de pagos digitales. Esta es la función principal de Bitcoin como moneda digital. El nacimiento de esta tecnología alteró la evolución convencional de los sistemas de pagos controlados por bancos y otras organizaciones financieras. Las monedas digitales se basan en un libro de transacciones publico que es compartido a través de Internet y mantenido por los usuarios que son los responsables de validar las transacciones[35].

Smart Contract. Un Smart Contract o **Contrato inteligente** es una aplicación que ejecuta automáticamente acuerdos comerciales. Su objetivo es forzar a cumplir con sus obligaciones a todas las partes de un acuerdo sin la necesidad de intermediarios[36]. Su funcionamiento de cara a los implicados en los acuerdos consiste en crear un contenedor donde se guardan todos los recursos implicados en el contrato, se definen las condiciones de cumplimiento y llegados al momento de su resolución, los recursos se liberan en función de las condiciones pactadas[26].

Seguros. Cualquier activo o propiedad valiosa que sea difícil de replicar o destruir puede registrarse en Blockchain, se puede verificar la propiedad y rastrear el historial de transacciones. A modo de ejemplo, Everledger es una compañía que crea un libro permanente de certificaciones de diamantes. Las características que identifican al diamante de manera única, como altura, ancho, peso, profundidad, color, etc. son hasheados y registrados en el libro de registros[37].

Crowfunding. Actualmente, debido al incremento de startups y la falta de financiación, una forma de evitar las entidades financieras y las plataformas de *crowfunding* como Kickstarter o Indiegogo, que reciben una parte de las inversiones de los

proyectos, es habilitando una plataforma de crowfunding con Blockchain en la que venden "participaciones" de la empresa a los inversores a través de criptomonedas generadas por la empresa[38].

2.3.5.2. Aplicaciones no financieras

Servicios gubernamentales distribuidos El uso más común de Blockchain en servicios gubernamentales es del de notario público. Aplicando las características de Blockchain, se puede garantizar la privacidad de los documentos a la vez que se puede certificar su existencia y la imposibilidad de haber sido modificado una vez que se integra en la cadena, además de asegurar el momento en que se tramitó al incluir el timestamp[37]. Estonia fue el primer país en incluir entre sus servicios digitales la tecnología de Blockchain para notaría[39], así como el sistema de voto. El registro histórico no puede ser cambiado debido a que los votantes no pueden ver votos distintos a los suyos, además de no poder votar de forma ilegitima por otros ciudadanos[40].

Almacenamiento distribuido Este concepto ha sido implementado en el área de la salud y la música. En las aplicaciones de salud, la cadena de bloques proporciona una estructura para almacenar historiales médicos que permanecen privados pero pueden ser analizados cuando sea necesario, mientras los usuarios que contribuyen al mantenimiento de la cadena son compensados económicamente[38]. La identidad del paciente se mantiene privada haciendo uso la firma digital como seudónimo. La industria musical utiliza Blockchain como forma de mantener una base de datos de los derechos de autoría de forma pública. También incluyen clausulas de rescisión como smart contract para resolverlas automáticamente[37].

IoT descentralizado El uso en *Internet of Things* presenta grandes desafíos. Uno de ellos es el paradigma cliente/servidor que predomina en el mundo. Si bien este modelo ha conectado dispositivos informáticos genéricos durante décadas y continuará respaldando las redes de IoT en pequeña escala tal como hasta hoy, no podrá responder a las crecientes necesidades de los enormes sistemas de IoT del futuro. Las soluciones de IoT existentes son costosas debido a la gran infraestructura y los costos de mantenimiento asociados con nubes centralizadas, grandes granjas de servidores y equipos de red[41]. Al usar un modelo de comunicación peer-to-peer estandarizado para procesar el número de transacciones entre dispositivos, se reducirá significativamente los costos asociados con la instalación y mantenimiento de grandes centros de datos, distribuyendo las necesidades de computación y almacenamiento a través de la gran cantidad de dispositivos que forman redes IoT . En asociación con Samsung, IBM ha desarrollado ADEPT[8], una plataforma que utiliza elementos del diseño de Bitcoin para construir una red distribuida de dispositivos IoT[37].

2.3.6. Desafíos en Blockchain

Hasta ahora se han mencionado el ámbito de aplicación de blockchain donde la privacidad, la base de datos distribuida y la inmutabilidad del registro son claras ventajas en el momento de hacer un nuevo desarrollo donde estás ventajas sean críticas. Sin embargo tiene varios inconvenientes que conviene tener en cuenta antes de implementarlo en una solución a una necesidad.

[8] Autonomous Decentralized Peer To Peer Telemetry.

Rendimiento Cuando se procesa una transacción, una cadena de bloques tiene que realizar las mismas tareas que una base de datos normal, pero también tres procesos adicionales adicionales.

- **Verificación de firma.** Cada transacción de Blockchain debe estar firmada digitalmente usando un esquema de claves público-privado. La generación y verificación de estas firmas es computacionalmente compleja. Por el contrario, en las bases de datos centralizadas, una vez que se ha establecido una conexión, no hay necesidad de verificar individualmente cada solicitud que se realiza.
- **Mecanismos de consenso.** En una base de datos distribuida, como una cadena de bloques, se debe hacer un esfuerzo para garantizar que los nodos de la red estén sincronizados. Dependiendo del mecanismo de consenso que se utilice, esto podría implicar una carga en la red mucho mayor. Si bien es cierto que las bases de datos centralizadas también deben se capaces de gestionar transacciones con posibles errores, es mucho menos probable que las transacciones se pongan en cola si ocurren en una sola ubicación.
- **Redundancia.** Además del rendimiento de un nodo individual, hay que tener en cuenta la cantidad total de cálculos que requiere una cadena de bloques. Mientras que las bases de datos centralizadas procesan las transacciones una vez (o dos); en una cadena de bloques, cada nodo de la red debe procesarlas de manera independiente, lo que se traduce en que se realiza mucho más trabajo para lograr el mismo resultado final.

Se puede suponer que los problemas de rendimiento en Blockchain son el resultado del paso de una arquitectura centralizada, a una descentralizada. Este cambio introduce una mayor complejidad en el procesamiento de la información y como resultado, el tiempo de computo puede ser más mayor que en una base de datos centralizada convencional.

Escalabilidad: En Blockchain con la arquitectura pública, la escalabilidad es un problema importante que los desarrolladores deben resolver o minimizar. A menudo se plantea en las discusiones técnicas del protocolo bitcoin debido a que bitcoin es un sistema autorregulable que funciona mediante el descubrimiento de bloques a intervalos aproximados. Su rendimiento de transacciones está limitado al tamaño máximo de bloques dividido por su intervalo[42]. Sin embargo, el principal obstáculo para la escalabilidad de Blockchain es una tendencia hacia la centralización con una cadena de bloques creciente: cuanto mayor sea la cadena, mayores serán los requisitos de almacenamiento, ancho de banda y potencia computacional de los nodos para que la red siga funcionando, tendiendo a una mayor centralización si la cadena de bloques se hace tan grande que solo unos pocos nodos pueden procesar un bloque[43].

Privacidad: Blockchain puede mantener una cierta privacidad a través de su par de claves pública-privada (una por persona o entidad), sin embargo es posible que Blockchain no pueda garantizar intercambios privados debido a que los valores de todas las transacciones y los balances por cada clave pública son visibles para todos los usuarios[44]. La alternativa es realizar pagos desde múltiples cuentas a través de herramientas existentes con este propósito[45].

Consumo de Energía: La creación de bloques mediante PoW requiere mucha potencia computacional, que está directamente relacionado con el consumo de electricidad. El poder computacional se usa solo para validar las transacciones, y los resultados no tienen ningún otro beneficio que no sea por el Blockchain[46].

2.3.7. Conclusiones

Este capítulo se estableció con el objetivo de evaluar las características de Blockchain y su funcionamiento. La teoría que sustenta a Blockchain es un libro público (ledger) con todas las transacciones ejecutadas. Utiliza un principio de descentralización y cifrado que implica claves públicas y privadas. Funciona colocando transacciones en grupos llamados bloques y vinculando estos bloques a través de lo que se llama cadena de bloques. Desde una perspectiva técnica, hay tres tipos de Blockchain: público, consorcio y privado. Los elementos básicos son los bloques donde se almacenan los datos, las firmas digitales para autenticar las transacciones, una red descentralizada para la interacción del usuario y un consenso de red para verificar la transacción. El uso de estos elementos básicos puede depender mucho del tipo de Blockchain que se use. En primer lugar, la tecnología Blockchain se considera comúnmente como la principal innovación tecnológica de bitcoin. Hoy en día, esta tecnología tiene implementaciones prácticas más avanzadas. Estas van desde la aplicación financiera, incluidos los sistemas de pago digital, contratos inteligentes, seguros y crowdfunding, hasta aplicaciones no financieras, como servicios gubernamentales, almacenamiento descentralizado y IoT descentralizada. Debido a los desafíos expuestos, se debe elegir cuidadosamente la implementación que se diseñará para IoT, cuando un dispositivo está conectado a una batería es necesario reducir el consumo energético en la medida de lo posible. También se intenta ajustar el rendimiento de los dispositivos para realizar despliegues más económicos pudiendo afectar al rendimiento de Blockchain.

Capítulo 3

Soluciones existentes

En este capítulo se abordarán las soluciones existentes en la actualidad que se enfrentan a las vulnerabilidades de IoT, las cuales se han venido mencionando en el presente documento. Algunas de ellas son presentadas como aproximaciones de la tecnología blockchain a la seguridad IoT visto desde distintos ángulos, y otras buscan la fortificación IoT por otros medios completamente distintos.

3.1. The Hyperledger project

Hyperledger[47] es un proyecto open-source de carácter colaborativo que pretende impulsar las tecnologías blockchain entre industrias. Surge en el año 2016 bajo la dirección de varias organizaciones, aunque actualmente forma parte de The Linux Foundation e incluye la participación de diferentes sectores: financiero, manufacturero, Internet of Things o de tecnología entre otros.
Hyperledger se crea con el propósito de crear nuevas aplicaciones transaccionales que se fundamentan en la confianza, la transparencia y la responsabilidad, mientras coordinen distintos procesos de negocio de acuerdo con las restricciones legales. Un posible enfoque es el de un sistema operativo para mercados, redes de compartición de datos, micro-monedas y comunidades digitales descentralizadas que permita reducir el coste y complejidad de las operaciones [47].
En definitiva, el proyecto Hyperledger pretende proporcionar frameworks y plataformas blockchain que garanticen la transparencia, durabilidad, interoperatividad y soporte requeridos para que sean adoptados en el ámbito comercial.

3.1.1. Objetivos de Hyperledger

El proyecto Hyperledger surge con una serie de objetivos concretos que se describen a continuación [47]:

- **Soporte de transacciones de negocio**: mediante la creación de frameworks y librerías open-sorce distribuidos y enfocados a la empresa.
- **Infraestructura neutral, abierta y gestionada por la comunidad**: esta infraestructura es soportada por una administración técnica y de negocio.
- **Desarrollo de comunidades técnicas**: éstas se encargan de desarrollar ledgers de blockchain compartidos, explorar casos de uso y nuevos campos de aplicación.
- **Educación**: pretende dar a conocer las posibilidades que ofrece Blockchain de cara al mercado.

- **Enfoque de *toolkit***: a través de las comunidades de usuarios pretende ofrecer una serie de herramientas, como distintas plataformas y frameworks.

3.1.2. Diseño de Hyperledger

Los requisitos de Blockchain aplicado al negocio pueden variar mucho. En algunos casos resulta fundamental que se alcance el consenso rápidamente, mientras que en otros puede que esta velocidad no sea crítica. Lo mismo ocurre con otro tipo de requisitos, como escalabilidad, confidencialidad, compatibilidad, complejidad o seguridad. Por este motivo, teniendo todo esto en cuenta, Hyperledger proporciona una serie de diferentes tecnologías de Blockchain, que abarcan ledgers distribuidos, motores para smartcontracts, librerías de cliente, interfaces gráficas u otras utilidades. Con lo que denomina una estrategia de paraguas, tal y como puede verse en la figura 3.1, Hyperledger fomenta la reutilización de bloques de componentes a través de una arquitectura de framework modular. Esto aporta flexibilidad, extensibilidad o la capacidad de alterar alguna parte del sistema sin afectar al resto.

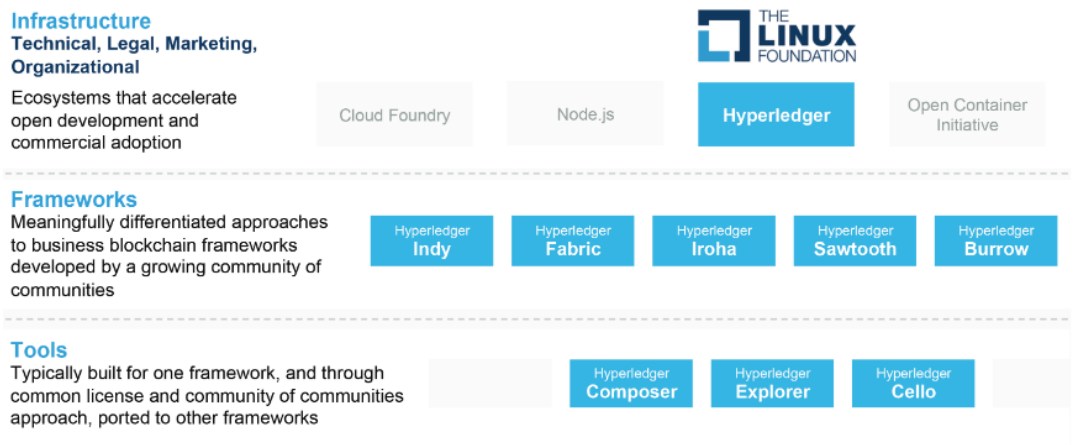

FIGURA 3.1: Enfoque de paraguas de Hyperledger

3.1.3. Arquitectura de Hyperledger

Dentro de su diseño modular, Hyperledger aboga por soluciones muy seguras y una API[1] extensa y de fácil uso. En su arquitectura se distinguen los siguientes componentes [47]:

- **Capa de consenso**: se encarga de generar el acuerdo y comprobar que las transacciones que forman un bloque son correctas.
- **Capa de smart contract**: procesa las peticiones para las transacciones y determina si éstas son válidas aplicando la lógica de negocio.
- **Capa de comunicación**: es responsable de la comunicación peer-to-peer entre los nodos que participan en una instancia compartida de ledger.
- **Abstracción del almacenamiento de datos**: permite que los datos guardados puedan ser usados por varios módulos.

[1] Application Programming Interface

- **Abstracción de la encriptación**: permite modificar el algoritmo de encriptación o el intercambio de algún módulo sin afectar al resto.
- **Servicios de identificación**: permite establecer una autoridad de confianza al configurar una instancia blockchain, así como el registro y unión de nuevas identidades. También gestiona cambios y proporciona autorización y autenticación.
- **Políticas de servicio**: gestiona varias políticas, como de consenso, gestión de grupo o aprobación. Requiere de otros módulos para aplicar estas políticas.
- **APIs**: interfaces entre las cadenas de bloques y clientes y aplicaciones.
- **Interoperabilidad**: soporta la interoperabilidad entre distintas instancias de blockchain.

3.1.4. Hyperledger Fabric

Bajo el proyecto Hyperledger se encuentran, a su vez otra serie de proyectos, entre ellos Hyperledger Fabric, que se describe en este apartado a modo de ejemplo. Al igual que otro sistemas que hacen uso de Blockchain, dispone de un ledger, utiliza smart contracts y son los participantes de las transacciones los encargados de gestiornarlas.
Las particularidades de esta plataforma residen en su caracter privado y con permisos. Mientras que otros sistemas permiten que identidades desconocidas se unan a la red con protocolos como *proof of work*, en este caso la participación solo es posible a través de un MSP (*Membership Service Provider*) de confianza [48].
Por otro lado, Hyperledger Fabric también ofrece diferentes formatos para el almacenamiento de los datos, distintos mecanismo de consenso intercambiables y soporta varios MPS. A mayores, también introduce el concepto de canal, de manera que un grupo de participantes puede crear un ledger separado para sus transacciones. Esta última faceta puede resultar muy importante en operaciones cuyos participantes puedan ser competidores.

Modelo Hyperledger Fabric. Los elementos claves que han guiado el diseño de esta plataforma son los siguientes [49]:

- **Bienes**: la definición de bienes permite el intercambio a través de la red de casi cualquier tipo de bien que posea valor monetario.
- **Cadena de código**: la ejecución de código se particiona limitando los niveles de confianza y verificación a través de los tipos de nodo, de forma que optimicen la escalabilidad y rendimiento de la red.
- **Características del ledger**: el ledger inmutable y compartido codifica todo el historial de la transacción para cada canal.
- **Privacidad en los canales**: los canales permiten transacciones laterales con la privacidad y confidencialidad necesarias que requiere cualquier operación de negocio.
- **Seguridad y servicios de afiliación**: la afiliación a un MSP proporciona confianza a la red de blockchain, de manera que los participantes saben que las transacciones pueden ser supervisadas por reguladores autorizados.

- **Consenso**: resulta esencial en la verificación completa de las transacciones de un bloque. Abarca todo el flujo de la transacción: petición, aprobación, solicitud y entrega al ledger.

3.2. IBM Blockchain Platform

IBM dispone de una plataforma de IoT denominada *Watson Platform*, que permite el empleo de la tecnología de Blockchain sobre algunos datos de estos dispositivos. La *IBM Blockchain Platform* tiene como objeto construir una cadena de bloques de tipo privado donde los datos son compartidos solamente por las entidades privadas que formen parte de las transacciones. En palabras de IBM: "IBM Blockchain proporciona una infraestructura de blockchain privado que replica los datos y valida las transacciones a través de contratos seguros"[50].

IBM Blockchain Platform proporciona una solución completa de Blockchain como servicio[2] (BaaS), a través de su *cloud* IBM, ofreciendo la posibilidad de gestionar una red con el rendimiento y seguridad requiere la regulación de la industria en muchos casos. Esta solución se basa en los principios de finalidad, confianza y privacidad, tal y como se explica a continuación:

- **Finalidad de los datos**: una vez las transacciones son guardadas en el *ledger*, no pueden ser modificadas o eliminadas. Para que estas transacciones se consideren finalizadas han de estar firmadas por las partes adecuadas, según un acuerdo conocido como política de aprobación.
- **Confianza**: la confianza se consigue mediante la aprobación de los miembros. Los participantes en las transacciones han de ser conocidos en la red, impidiéndose el anonimato.
- **Privacidad de la red**: las operaciones de negocio requieren que tanto los datos como las propias transacciones sean confidenciales. La plataforma de IBM proporciona las herramientas necesarias para gestionar una red descentralizada de forma rápida, simple y efectiva, en término de costes, entre varias organizaciones.

3.2.1. Arquitectura

La plataforma IBM Blockchain se apoya sobre la plataforma *open-source* Hyperledger Fabric (véase la sección anterior) que proporcionan la infraestructura necesaria para el desarrollo, administración y operación de soluciones de negocio. La figura 3.2 muestra la arquitectura *end-to-end* de esta plataforma. De acuerdo con IBM: "Se trata de la única solución lista para negocio y *end-to-end* que permite a las organizaciones desplegar una red de blockchain en un tiempo record"[51]. A continuación, en los siguientes apartados, se ofrecen algunos detalles más de los aspectos que engloba la arquitectura de la plataforma:

Desarrollo. Es el *framework*[52] para desarrollar aplicaciones basadas en blockchain. Incluye herramientas y lenguajes que ofrecen a los desarrolladores la posibilidad de modelar, desarrollar, probar y desplegar sus aplicaciones en una red de negocio distribuida. También consta de una serie de librerías, modelos de datos y un entorno de desarrollo web. Todo ello contribuye a aumentar la velocidad y la eficiencia durante el proceso de desarrollo, a la vez que reduce el riesgo.

[2] Blockchain As A Service

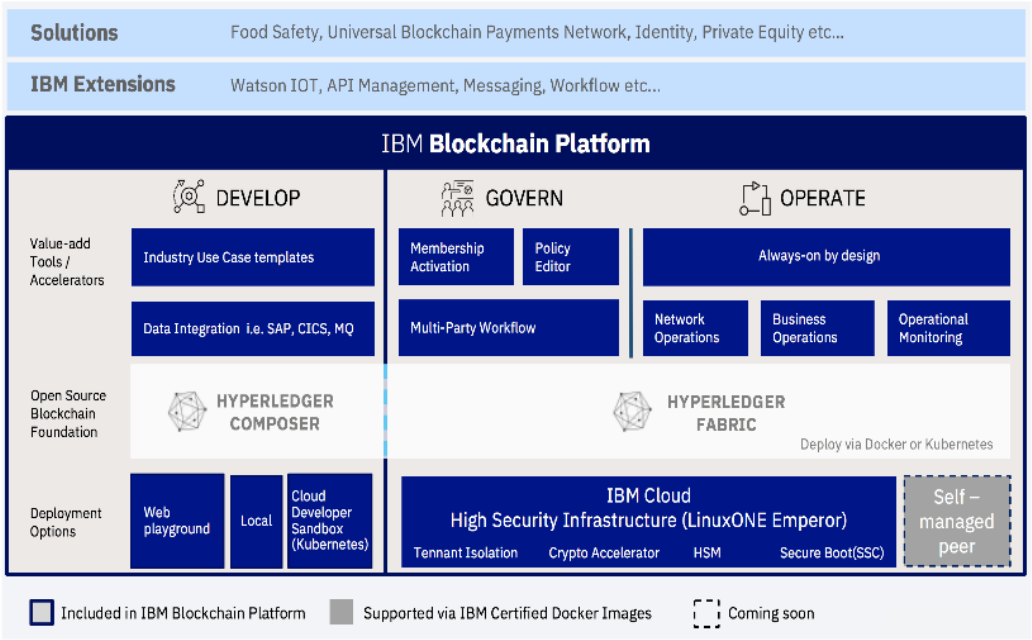

FIGURA 3.2: Arquitectura de la plataforma IBM Blockchain [51]

Administración. En lo que respecta a la administración de una red, resulta fundamental disponer de herramientas, definiciones y modelos claros y efectivos. La plataforma IBM Blockchain crea redes, garantizando un modelo bien definido y permitiendo su actualización sin reiniciar toda la red. Una administración adecuada elimina la incertidumbre y el riesgo de las operaciones de negocio, además de garantizar la privacidad y confidencialidad de las distintas transacciones.
Según IBM, las ventajas del uso de esta plataforma incluyen: herramientas de gestión democráticas, que permiten la gestión colectiva de la red por parte de todos sus miembros; entorno de gestión dinámico, que facilita la ampliación de la red con nuevos miembros; y las herramientas de serie que permiten, por ejemplo, personalizar y activar la red.

Operación. Cuando se trata de redes descentralizadas sobre las que se ejecutan aplicaciones y transacciones críticas, resulta necesario que soporten operaciones y actualizaciones de forma segura y escalable, estando siempre disponible la plataforma.
La plataforma *IMB Blockchain* tiene como núcleo el *Hyperledger Fabric* de Linux que es, a su vez, un sistema modular que permite diferentes implementaciones de distintos aspectos de la red, como encriptación, identificación, algoritmos de consenso, lenguajes para los *smartcontracts*, etc. Partiendo de esta base, se elimina la necesidad de añadir *parches* o soluciones dispares.

3.2.2. Funcionamiento

La *Watson IoT Platform* permite que los dispositivos participen en transacciones blockchain, comunicándose con los ledgers privados y ejecutando las acciones indicadas en los smartcontract. El smartcontract define muchos de los términos acordados por las partes e, incluso, algunas acciones fuera de la cadena de bloques, como la solicitud de un servicio.
Además, por otro lado, la plataforma también proporciona la conversión necesaria

entre el formato de los datos de los dispositivos y el requerido por el contrato.
En definitiva, el empleo de esta plataforma permite que grupos de participantes utilicen información de dispositivos de IoT, como puede ser la geolocalización, en contratos inteligentes que se ejecutan en las redes de IBM Blockchain, eliminando la necesidad de tener implementada una estructura blockchain. La diferencia de IBM con otras implementaciones que emplean Blockchain reside en que está diseñada exclusivamente para empresas [53].

3.2.3. Casos de uso

IBM define una serie de casos de uso y escenarios en los que pueden aplicarse tanto IoT, como Blockchain. Dichos casos se enumeran a continuación [53]:

- **Rutas comerciales y cadenas de suministros**: el seguimiento de mercancías promueve y garantiza su envío seguro. También afianza las líneas de crédito y acelera los pagos.

- **Garantía entre partes**: es posible mantener un historial que incluya los eventos críticos en una cadena de suministros, como pueden ser los mantenimientos programados. Por ejemplo, un sensor integrado puede proporcionar un historial inmutable de piezas desde su fabricación y montaje a lo largo de la cadena de suministro, incluyendo potenciales eventos críticos que afecten a la vida o el mantenimiento programado. Esta información se puede compartir con los involucrados en la cadena de suministro, fabricante y reguladores de manera segura. Es un importante desafío asegurar la calidad de un producto final sin un registro estándar que detalle de dónde proviene cada pieza que lo compone, y cómo se usa. El no hacerlo, puede producir una perdida económica ante un mantenimiento innecesario o atrasado. Un ledger IoT y blockchain compartido mantiene un registro para el uso, mantenimiento, trabajo de garantía y las piezas de repuesto. En el caso de una situación de retirada de productos, se puede identificar lotes específicos de piezas que pueden ser defectuosas en lugar de requerir una retirada preventiva más amplia.

- **Computación descentralizada**: resulta posible realizar cálculos computacionales, como ánalisis, en dispositivos extremos, propiedad de terceros.

- **Interconexión de dispositivos**: es posible permitir a dispositivos distribuidos solicitar servicios o pagar por ellos mediante el empleo de micropagos y gestión de roles.

- **Cumplimiento de la regulación**: es posible compartir un historial de seguimiento de algunos productos con agencias reguladoras o seguros. Un dispositivo de IoT podría usarse de muchas maneras diferentes como parte de una cadena de suministro. Por ejemplo, un sensor de temperatura podría integrarse en un paquete para rastrear la temperatura durante todo el proceso de envío, almacenando los datos localmente y enviándolos a la cadena de bloques privada a través de la Plataforma Watson IoT en los puntos de referencia (fábrica, depósito, supermercado) en el momento que se disponga de conectividad.

3.3. Ethereum

Ethereum ha sido y sigue siendo el principal competidor de Bitcoin en el mundo de las criptodivisas y aunque está claramente orientado al sector económico, su diseño permite que pueda implementarse en el mundo de IoT. Actualmente hace uso del protocolo de verificación que utiliza Bitcoin (Proof of Work[31]) pero su futuro pasa por otra implementación más eficiente denominada Proof of Stake[32]. Con Proof of Work, los ordenadores que forman parte de la red compiten por solucionar fórmulas matemáticas y ganar el derecho de confirmar transacciones de datos para el blockchain, haciendo la red segura. Sus dos mayores desventajas son la posibilidad de verse sobrepasado si el 51 % de los nodos son vulnerados, por un lado, y la enorme cantidad de energía consumida para mantener la seguridad de la red, por otro. Para ello Noam proponer sustituir el Proof of Work por algo que llama Proof of Stake.
Otro concepto importante al hablar de Ethereum es el de token [54]. En este caso, un token representa por lo general un valor financiero o un activo similar[3]. Los token pueden utilizarse con diversos fines, desde medios de pago hasta para administrar una red de forma descentralizada. Ante las distintas propiedades y funciones de los token, existe el Token ERC20[55], una interfaz estándar que garantiza la interoperabilidad entre tokens.
Partiendo de la plataforma Ethereum, en las próximas páginas se identifican y se explican algunas de sus ramificaciones que pretenden solucionar alguno de los problemas más importantes que se han identificado en la red para adaptarla al caso de IoT.

3.3.1. The Raiden Network

The Raiden Network [56] es un proyecto de código abierto ofrece una solución escalable para realizar transferencias de tokens, compatibles con el estándar ERC20, sobre el blockchain de Ethereum. Utiliza una tecnología denominada *payment channel*, que habilita la posibilidad de realizar transacciones fuera de la cadena con token generados dentro de la misma. Permite la transferencia de tokens entre dos participantes sin la necesidad de un consenso global. Para ello, se hace uso de las denominadas pruebas de equilibrio o *balance proofs*, que se firman digitalmente y sellan con un hash, y se deposita al inicio en el blockchain. Se establece así un canal que permite transferencias practicamente ilimitadas y bidireccionales, en las que el blockchain por sí mismo no está involucrado. Esta transferencia es, as su vez, inmune a la duplicación y, debido que solo los dos participantes tienen acceso a los tokens, es tan seguro como una transacción en una cadena. Este funcionamiento puede verse en la figura 3.3. Lo que resulta, no obstante, importante en este sistema es la conexión entre los pares. Dado que la creación de un canal requiere transacciones en el blockchain resulta inviable crear canales entre todos los nodos. por este motivo, y como los canales no requieren conexión directa entre las partes, los canales se crean sobre varios nodos. Y, de hecho, es esta red y los protocolos de enrutamiento y creación de los canales lo que constituye realmente la Raiden Network.
En definitiva, The Raiden Network promete ser una herramienta poderosa en el entorno machine to machine. Sin embargo, a día de hoy sigue en su etapa de desarrollo y no ha sido implementado en un entorno real, estando lejos de ser lanzada una versión final.

[3]Sería equiparable a las fichas de un casino que representan dinero y pueden ser utilizadas en las máquinas y juegos de su interior.

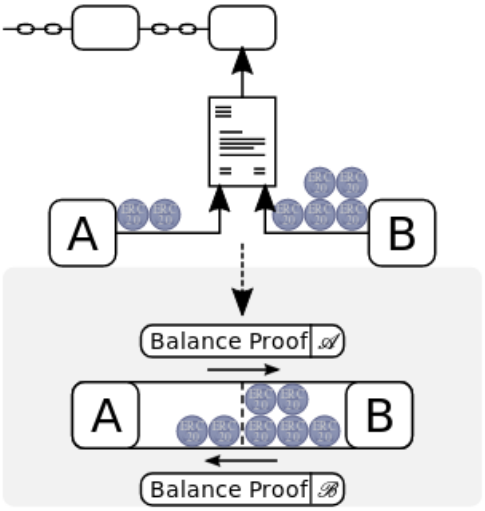

FIGURA 3.3: *Payment channel* bidireccional

3.3.2. Sharding

Sharding es otro proyecto que pretende abordar la escalabilidad en Ethereum. A diferencia de otras propuestas, que trabajan en la capa dos de la torre de protocolos, Sharding utiliza un enfoque distinto basado en modificar el protocolo en sí para conseguir cierta paralelización [57].
Tal y como esta diseñado Ethereum actualmente, todos y cada uno de los nodos han de procesar cada transacción que tiene lugar en la red. Pese a que este sistema proporciona mucha seguridad, también significa que la máxima velocidad de la cadena es la de cualquiera de sus nodos. La idea de *blockchain sharding* es dividir el estado de la red en su totalidad en diferentes particiones denominadas *shards*, cada una con su estado e historial de transacciones independientes. De esta manera cada nodo solo tendría que procesar las transacciones correspondientes a un shard, lo que supondría un aumento de la carga de transacciones totales procesadas.
Para ejemplificar su funcionamiento, Vitalik Buterin (creador de Ethereum) explica que Sharding es similar a tener cientos de universos similares, aunque únicos. Estos "mundos" solo interactúan entre sí compartiendo un consenso y un poder de autorización. Puede observarse un esquema de sharding en la figura 3.4[57].
De forma gráfica podemos imaginar que tres nodos: A, B y C tienen que verificar el dato T. De la forma tradicional, los nodos A, B y C deberían verificar por separado la misma información. Para evitar esto Sharding separa T en tres "shards": T1, T2 y T3 y cada uno de los nodos verifica cada "shard" de forma simultánea, reduciendo la latencia.

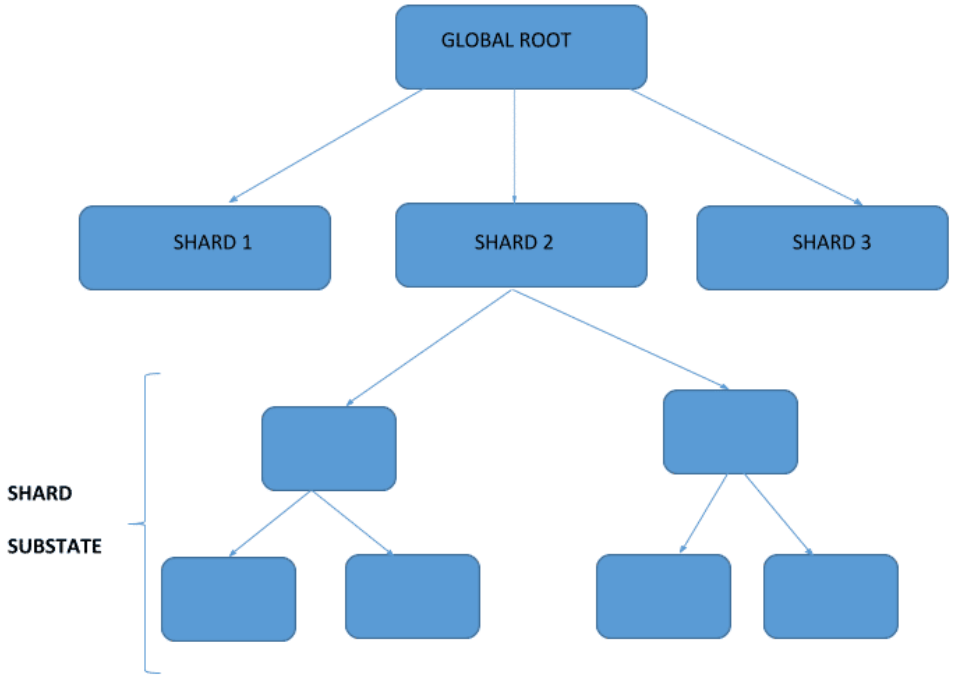

FIGURA 3.4: Organización Sharding.

3.3.3. Plasma

Plasma funciona de una forma similar a Raiden network, con la diferencia de que facilita los Smart contract por encima de las transacciones de datos con objeto de reducir la cantidad de trafico almacenado y verificado en la cadena principal [58]. Para ello Plasma se plantea dos propósitos: el primero es replantear la red de blockchain para simplificarla, mientras que el segundo es habilitar un método para utilizar tokens empleados en "Proof of Stake" (tecnología derivada de Proof of Work). El uso de este último sistema podría fomentar la retención de bloques, lo que produciría que no publicaran los resultados y, a su vez, generar un fraude al no estarse aportando validez a los datos e impedírselo a otros nodos.
Para evitar este segundo problema Plasma crea "cadenas hijas" por encima de la cadena principal que permiten transmitir información a la cadena principal, aunque haya bloques retenidos, evitando así el secuestro de información. Su funcionamiento puede verse en la figura 3.5[58].

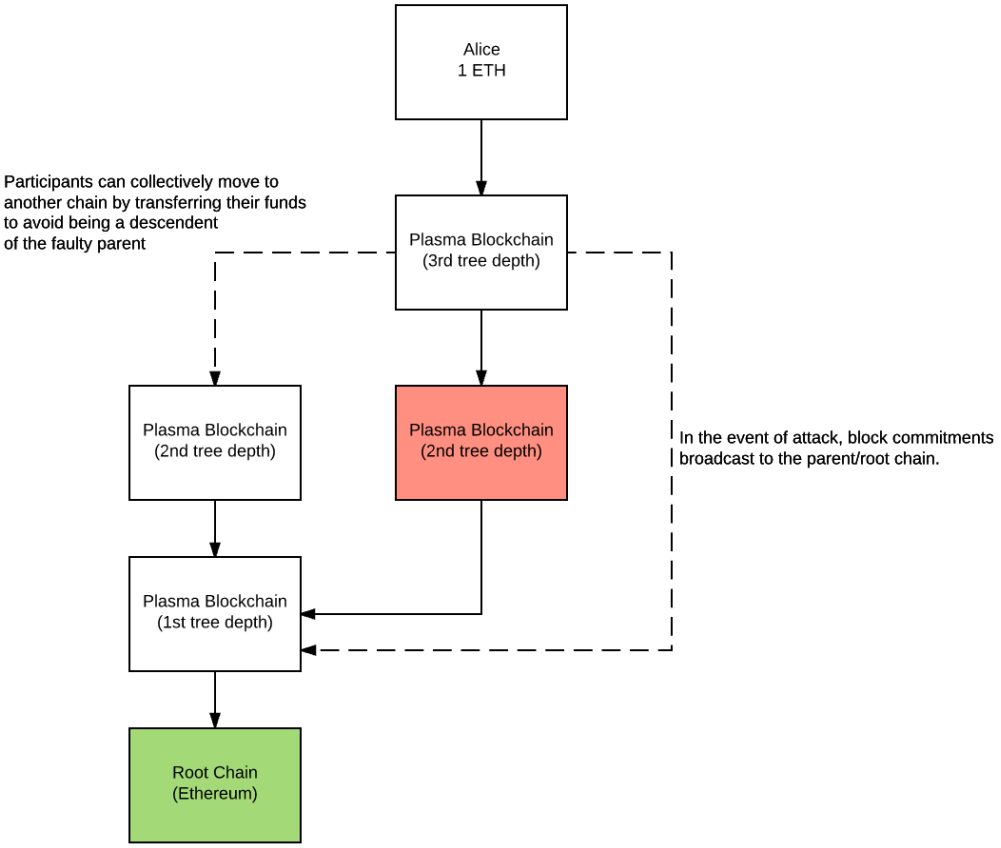

FIGURA 3.5: Diagrama de funcionamiento Plasma

3.4. Conclusiones sobre las tecnologías existentes

En este capítulo se han descrito algunas de las soluciones basadas en blockchain que podrían emplearse en un entorno de IoT. Sin embargo, después de una comprobación más en detalle se pueden extraer algunas conclusiones que dificulten su implementación:

- The Hyperledger Project es un nuevo desarrollo open-source amparado por la Linux Foundation con unas bases solidas, donde se describen perfectamente las capas que componen su arquitectura. Y es precisamente la idea de superponer un software complejo sobre un sistema operativo de un dispositivo IoT la que puede suponer un sobreesfuerzo en aquellos sistemas que no dispongan de muchos recursos pudiendo provocar bloqueos o bajo rendimiento.
- IBM Blockchain Platform es una solución propietaria de pago donde se describe de forma somera la arquitectura y su funcionamiento, siendo imposible acceder a la documentación técnica. Sin embargo se describe como una solución basada en The Hyperledger Project, lo que deja translucir, al igual que en el punto anterior, unos requisitos técnicos elevados.
- Ethereum es un proyecto open source que proporciona funcionalidades avanzadas en el ámbito de la economía distribuida por encima de otra criptomonedas como Bitcoin. Sin embargo, Ethereum y sus derivados están fuertemente orientado al intercambio de dinero sobre una informática con alta capacidad computacional, lo que hace que los dispositivos con una potencia más discreta no puedan ejecutar apropiadamente este software.

Es precisamente porque ninguna de estas soluciones están optimizadas para su uso en aparatos con arquitectura de componentes hardware heterogeneas en despliegues amplios, que se opta por el desarrollo de un software nuevo, que haciendo uso de blockchain, se enfoque estrictamente en el ámbito de **Intenet of Things**.

Capítulo 4

Solución propuesta

En este capítulo se muestra como funciona el software creado para resolver los problemas de seguridad ya mencionados en IoT. Su objetivo es proporcionar un método de comunicación seguro, transparente y escalable en dispositivos de bajas capacidades. Para ello se describirá el flujo de trabajo general para luego profundizar en su funcionamiento paso a paso, así como su correspondiente motivación desde su ejecución por primera vez hasta finalizar la primera iteración del ciclo del programa.

4.1. Conceptos Previos

Es necesario antes de empezar, establecer una serie de conceptos que serán recurrentes a lo largo de todo el capítulo. Esta nomenclatura hace referencia a dispositivos agrupados por sus funcionalidades:

- Nodo Activo(NA). Este dispositivo no forma parte de la red en sí mismo, sino que se trata más bien de todo aquel dispositivo capturador de información (sensor de temperatura, humedad, etc) o actuador (interruptor electrónico) que está conectado directamente con el nodo maestro de la forma apropiada, ya sea cableada o inalámbrica.

- Nodo Maestro(NM). Dispositivo con la lógica necesaria para recuperar datos de sensores, generando bloques donde se agrupa esta información, para posteriormente ser enviada a los nodos esclavos. Se encuentra conectado con los sensores, y con uno o más nodos esclavos.

- Nodo Esclavo(NE). Dispositivo que recoge la información enviada por nodos maestros con el fin de realizar las operaciones de validación, además de recoger la información validada de otros nodos esclavos para comprobar la validación y agregar los datos a la cadena de bloques.

- Nodo Visualizado(NV). Es aquel sistema al que deben llegar los datos para poder ser almacenados y representados. Puede estar conectado a nodos maestros, esclavos o puede estar temporalmente sin contacto con ninguno, para luego actualizar toda la información que haya en la cadena en el momento que vuelva a tener conexión con alguno de los nodos.

Un ejemplo visual de la estructura de funcionamiento que puede tener, es el mostrado en la figura 4.1. En el cual, dos nodos maestros tienen conectados directamente tres nodos activos, que a su vez se interconectan entre ellos y con los nodos esclavos para finalmente alcanzar un nodo visualizador.

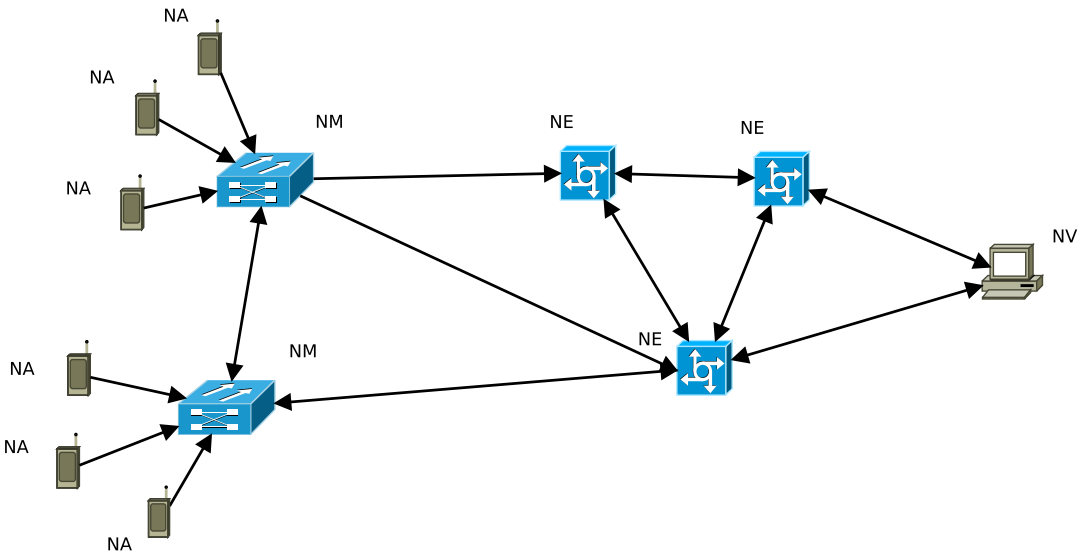

FIGURA 4.1: Ejemplo de escenario de pruebas

4.2. Funcionamiento General

En el presente apartado se hará una descripción a alto nivel del funcionamiento describiendo la configuración, los primeros pasos de la primera ejecución y el funcionamiento cíclico a partir de ese momento.

4.2.1. Configuración

En primer lugar, antes de iniciar el software es necesario realizar algunos ajustes en la configuración. Para ello se edita el archivo "mbiot.conf" (ver Anexo A) incluido en la raíz del software, que sigue el formato habitual de los ficheros de configuración de los sistema basados en GNU/Linux [59]. En este fichero se pueden editar algunos parámetros que alteran el funcionamiento del dispositivo que se configura. Son los siguientes:

- NODE. Este parámetro selecciona que comportamiento tendrá el dispositivo pudiendo ser master, slave o viewer en función de la posición que ocupa en la red.
- LOOP. Este parámetro solo tiene utilidad en caso de que se haya seleccionado "maestro" como tipo de nodo. En esta situación, el valor de la variable indica el número de datos por NA que se han de recoger antes de pasar a generar un bloque y proceder a la siguiente fase. Cuantos más datos se recojan antes de enviar la información, hay menos envíos y se satura menos la red.
- PROB. Este valor es aplicable en caso de ser un nodo "esclavo", e indica inversamente la probabilidad de que un nodo esclavo intente validar la información que recibe (un 3, hace referencia a $\frac{1}{3}$, un 2 a $\frac{1}{2}$, etc.). Esto es particularmente útil en aquellos casos que haya muchos nodos maestro enviando información, pues evita que todos los nodos esclavo intenten verificar toda la información que les llega.
- POW. En este proyecto se ha optado por una verificación de Proof of Work ya descrita en el capítulo 3, donde el objetivo matemático es encontrar un hash

que comience por el número de ceros indicados en este parámetro. A mayor número de ceros, mayor difícultad y mayor tiempo se tardará en verificar pero a su vez, mayor será la dificultad de suplantar el dato verificado.

- BUFFER. Establece el número de bloques validados que debe haber antes de verificar su integridad. Cuanto mayor sea, más difícil es que se produzca una inconsistencia en la cadena, pero los datos incluidos serán más antiguos.

4.2.2. Primera ejecución nodo Maestro

El nodo maestro tiene asignadas unas tareas previas al inicio de su ejecución, donde se comprueba si es la primera vez que ese está ejecutando en el nodo. Si no fuera así pasaría directamente al siguiente paso donde se recolecta la información de los sensores. En caso de que fuese su primera ejecución, realizaría las siguientes tareas:

- Crear los directorios necesarios para almacenar el flujo de datos: Datos, datos donde se almacenarán agrupados y firmados por el propio nodo, y un directorio de bloques donde se almacena lo que será el ledger (datos firmados y validados).
- Posteriormente se genera el identificador del nodo maestro y los de nodos activos conectados, en sus directorios particulares identificados por "NAx"donde la x representa el número de nodo conectado.
- Por último se generan claves publico-privada para el nodo. Esto habilita la capacidad de firmar la información recogida de los nodos asegurando su integridad.

Se puede ver el proceso de forma gráfica en la figura 4.2.

4.3. Nodo Maestro

Profundizando un poco más en las funcionalidades por nodo, el proceso realizado por el nodo maestro, comienza recolectando los datos de los sensores que estén conectados al nodo. Debido a la heterogeneidad del proceso de conexión, la diversidad de datos y la amplitud de formatos con el que es posible encontrarse en IoT, solamente se establece la generalidad de dejar los datos en la carpeta etiquetada con NA y número de sensor (ejemplo: ”NA1” para el primer sensor.) en el formato que decida el usuario, pues ese mismo formato tendrán los datos al final del proceso.

Los datos se recopilan en un fichero nombrado con el número de sensor en la carpeta *data*. Dentro de este fichero se incluyen:

- **Dato del sensor:** la información recogida (temperatura, humedad, luminosidad, etc.).
- **Timestamp**[1]**:** valor númerico con los milisegundos pasados desde el 1 de enero de 1970. En este valor indica el momento de la toma de datos por parte del sensor.

[1]Marca temporal.

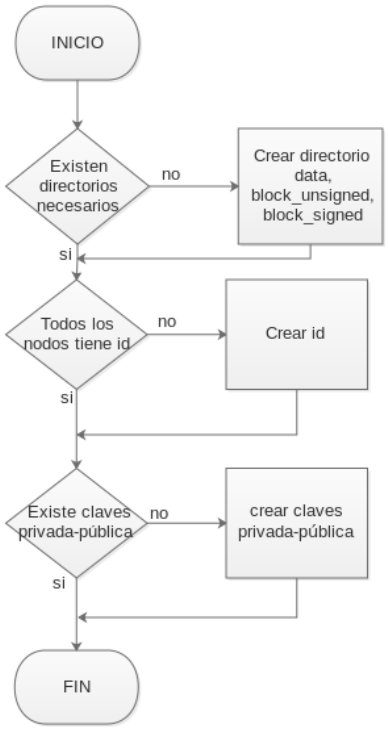

FIGURA 4.2: Diagrama de flujo de la primera ejecución

- **ID sensor:** identificador asignado al sensor por el nodo. Este valor tiene doble objetivo: diferencia a un sensor de otro y añadir complejidad en caso de intento de suplantación. Técnicamente se trata de un valor hexadecimal de 32 dígitos, que a semejanza del protocolo IPv6[60], genera 6.7×10^{-17} identificadores, suficientes para la mayoría de las implementaciones.

Posteriormente, con los datos generados anteriormente se genera un fichero llamado *BLOCK_"timestamp"* en una nueva carpeta denominada *BLOCK_UNSIGNED* que contiene la siguiente información:

- **Información de sensores:** en este primer campo se concatena la información generada por todos los sensores en el paso anterior quedando agrupada bajo un mismo fichero.
- **Firma digital:** el nodo firma digitalmente mediante el uso del software *OpenSSL*[61] los datos anteriores con dos propósitos: Asegurar que sea fácilmente identificable si los valores del punto anterior se modifican y por otro lado, identificar de forma inequívoca al nodo como remitente de la información.

Después de este paso, el fichero generado se comparte por la red a todos los nodos que estén escuchando, sean esclavos o maestros, pasando a la etapa de realización de la prueba de esfuerzo (*Proof of Work*).

4.4. Nodo Esclavo

El nodo esclavo tiene como objetivo validar los datos transmitidos mediante la prueba de esfuerzo con objeto de dificultar la inclusión de datos ilícitos a los posibles

atacantes. Por otro lado, tiene la tarea de comprobar que la validación se ha realizado correctamente antes de incluir la información al *ledger* y enviar las novedades por la red. Cabe destacar que excepto el nodo maestro origen de los datos, puede validar cualquier otro nodo, pudiendo comportarse un nodo maestro como esclavo a ojos de aquel que genera datos. Esta es una opción posible recomendable en el caso de nodos maestros con poca carga de trabajo, pero queda supeditado a las necesidades de la red, determinado por el administrador.

4.4.1. Proceso de validación

En el momento que el nodo recibe la información del nodo maestro comienza el primero de los pasos, consistente en dar validez a la información recibida.

Este proceso se basa en generar un *hash* mediante el sistema SHA-512[23], elegido por ser un algoritmo equilibrado entre rendimiento y dificultad[23]. El hash a calcular debe cumplir que comience por un número determinado de ceros definido en el fichero de configuración. Cuanto mayor sea el número de ceros por el que tenga que empezar el hash, mayor será la dificultad y por tanto se necesitará mas tiempo para resolver el problema matemático. Debido a que en el cálculo de hashes, siempre que haya la misma entrada, el resultado es el mismo, se necesita incluir un valor que varíe para cumplir las exigencias de dificultad, a este valor por el paralelismo del sistema con Bitcoin se le denomina [nonce].

Una vez que se ha definido el tipo de hash y como funciona, se crea uno con respecto a la información recibida del nodo maestro (datos, nodo y timestamp) junto a la que se le añade el hash del bloque anterior, de forma que si se quisiera modificar maliciosamente alguno de los datos, además de modificar el bloque actual, tendría que modificarse el anterior para que la relación entre información y hashes coincida, tal y como se ve en la figura 4.3. Una vez que se halle el nonce que proporcione un hash con los requisitos de dificultad, se puede considerar a la información validada. En este momento se crea el bloque en *BLOCK_SIGNED* y se difunde a todos los nodos de la red con objeto de ser verificada. El bloque quedaría compuesto por la siguiente información:

- **Datos del sensor:** Son los datos recopilados por los sensores con el identificador de sensor y un timestamp que indica el momento de la recogida.
- **Firma del nodo maestro:** Firma electrónica realizada por el nodo maestro sobre los datos anteriores.
- **Hash del bloque anterior:** Almacenando el hash anterior se conforma una cadena de bloques que otorga robusted a la integridad de la información.
- **Nonce:** Numero que proporciona la posibilidad al hash de cumplir requisitos de la configuración.
- **Hash:** Cadena alfanumérica de longitud fija generada a partir de la información anterior.

4.4.2. Proceso de verificación

Una vez recibida la nueva información se procede a verificarla. Este proceso consiste en el inverso a la validación, se comprueba que el hash cumple los requisitos de dificultad y que los datos, junto al nonce, generan el hash incluido en el fichero.

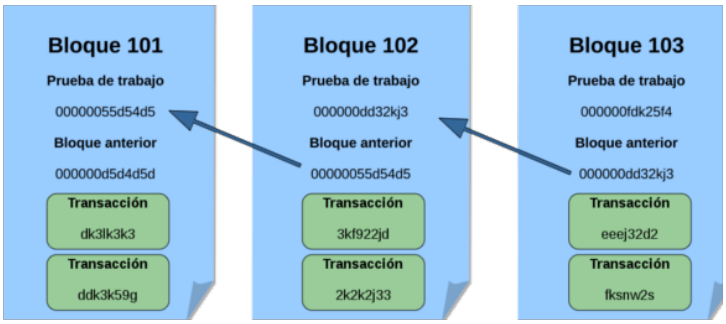

FIGURA 4.3: Encadenamiento de bloques mediante hashes

Este es el momento donde se considera la información como correcta y se incluye en el ledger.

Si se diese el caso en el que no coincidiese alguna de las combinaciones anteriores, se considera que la información ha sido comprometida siendo desechada.

4.5. Nodo Viewer

El nodo denominado *Viewer* o visualizador es el paso final de la cadena de transporte y tiene en común con el nodo esclavo la función de verificar la información recibida, asegurando que esta sea correcta. Su principal función radica en presentar los datos almacenados en el ledger ordenándolos cronológicamente para poder ser utilizados por el usuario receptor.

Capítulo 5

Análisis de la Implementación

En este capítulo se aborda una serie de escenarios de pruebas donde se comprueba la eficacia de aplicar el sistema descrito en el capítulo 4 de forma práctica, comprobando de forma general el comportamiento de blockchain en el transporte de datos en un sistema de sensorización.

5.1. Escenario de pruebas

El planteamiento inicial que se propone, es el transporte de información desde seis sensores de temperatura y humedad hasta un dispositivo final al que se considera objetivo del envío. La distribución de los nodos y los sensores tiene la forma indicada en el la figura 5.1.

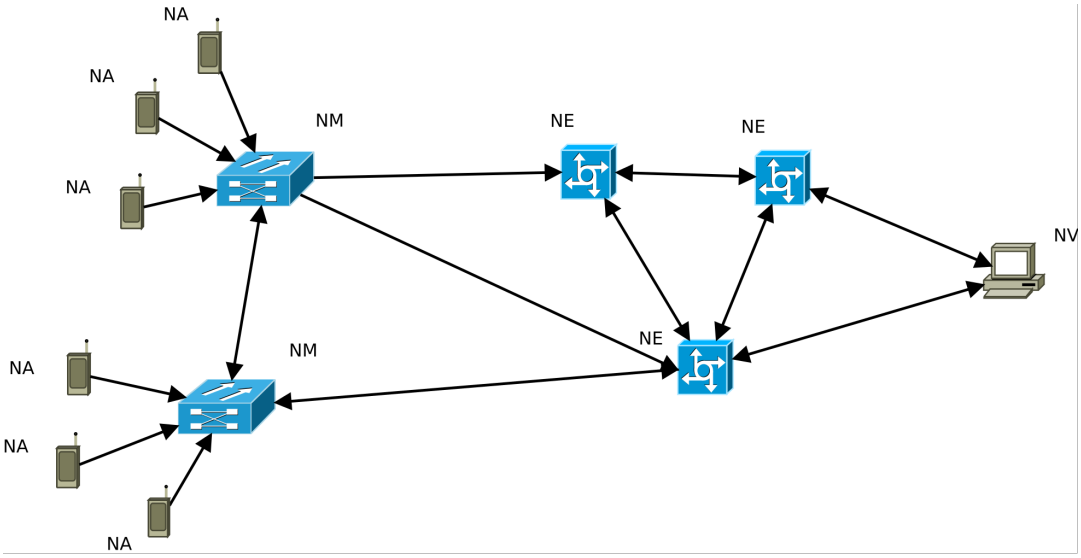

FIGURA 5.1: Escenario planteado para el estudio del comportamiento de la solución descrita en el capítulo 4.

Para la realización de las pruebas de integridad de los datos, este escenario sufrirá modificaciones de cara a la representación de diferentes ataques, ya sea añadiendo nuevos componentes a la red o sustituyendo alguno de los existentes por otro con el fin de simular un acceso ilícito a la red.

5.1.1. Dispositivos

Para realizar las pruebas se han utilizado distintos modelos de placas SBC[1] con diferentes configuraciones de hardware, indicadas en la tabla 5.1, con objeto representar en la medida de lo posible la heterogeneidad de IoT. Estas placas tienen en común un bajo coste y unas capacidades de procesamiento limitadas, características habituales en dispositivos IoT como se menciona el capítulo 2.

Dispositivo	Rasberry Pi 1 model B	Rasberry Pi 2	OrangePi Zero
SoC	BCM2835	BCM2836	AllWinner H2
CPU	ARM11 700MHz	ARM Cortex A7 900MHz	H2 Cortex A7 600MHz
RAM	512MB	1GB	512MB
Consumo	3,5W	4W	N/A
S.O.	Raspbian	Raspbian	ARMbian

TABLA 5.1: Tabla comparativa de dispositivos empleados.

Además, para la recepción de los datos a modo de Nodo Visualizador y como equipo encargado de realizar los ataques que se describen en apartados posteriores, se utiliza un ordenador portátil con las características descritas en la tabla 5.2:

Dispositivo	Asus UX410U
CPU	Intel i5 7200U
RAM	8GB
Gráfica	Intel HD 520
S.O.	Debian 9

TABLA 5.2: Especificaciones del equipo de pruebas.

La correspondencia de los dispositivos mencionados en la tabla 5.1 y 5.2 con los dispositivos de la figura 5.1 queda de la siguiente manera:

- **Nodos Maestro:** Raspberry Pi 1 model B y Raspberry Pi 2.
- **Nodos Esclavo:** OrangePi Zero.
- **Nodo Visualizador:** Asus UX410U.

De esta manera los dispositivos encargados de la recolección de datos son las placas Raspberry Pi y los que validan la información como nodos esclavos son las Orange Pi. Para la recepción y visualización de datos se hará cargo el ordenador portátil Asus.

5.2. Ataques

Con los elementos disponibles se han seleccionado los ataques más representativos que puede recibir una red IoT con el objetivo de perturbar el correcto funcionamiento de la red.

[1]Single Board Computer.

5.2.1. Ataque man in the middle

Uno de los ataques que se realizan con más frecuencia cuando se trata de la interconexión de dispositivos es el denominado *Ataque Man in the Middle* [62]. La base de su funcionamiento es suplantar al destino de la información, recibiendo los datos para examinarlos, alterarlos o eliminarlos previamente a enviarlos a su destino original. Una posible representación de esta situación es la dada en la figura 5.2 [62].

FIGURA 5.2: Representación ataque man in the middle

Al darse esta situación en el escenario de pruebas planteado, una posible opción donde el atacante se situaría queda reflejada en la figura 5.3.

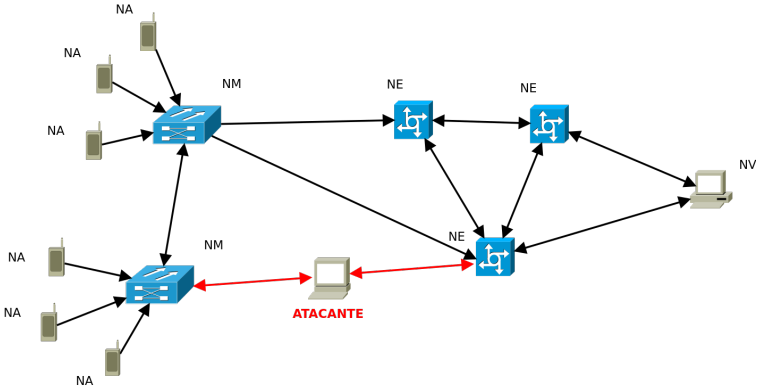

FIGURA 5.3: Intromisión de equipo ilícito en la red de pruebas.

Cómo se puede ver en la figura 5.3, un atacante interrumpe la conexión entre uno de los nodos maestros y un nodo esclavo. En este momento, el curso de acción por el intruso con respecto a la información, fundamentalmente puede seguir dos caminos: bloquear el intercambio de datos o alterarlos:

Bloqueo de la comunicación. La posibilidad de interrumpir el flujo de información entre dos nodos de la red es muy real, por eso mismo el almacenamiento de la información sigue una topología distribuida por la red y es recomendable que cada

nodo esté conectado al menos a otros dos, de forma que si se diese el caso de la figura 5.3 donde el atacante interrumpe la conexión, el flujo de información puede fluir libremente por otra ruta alternativa como se muestra en la figura 5.4.

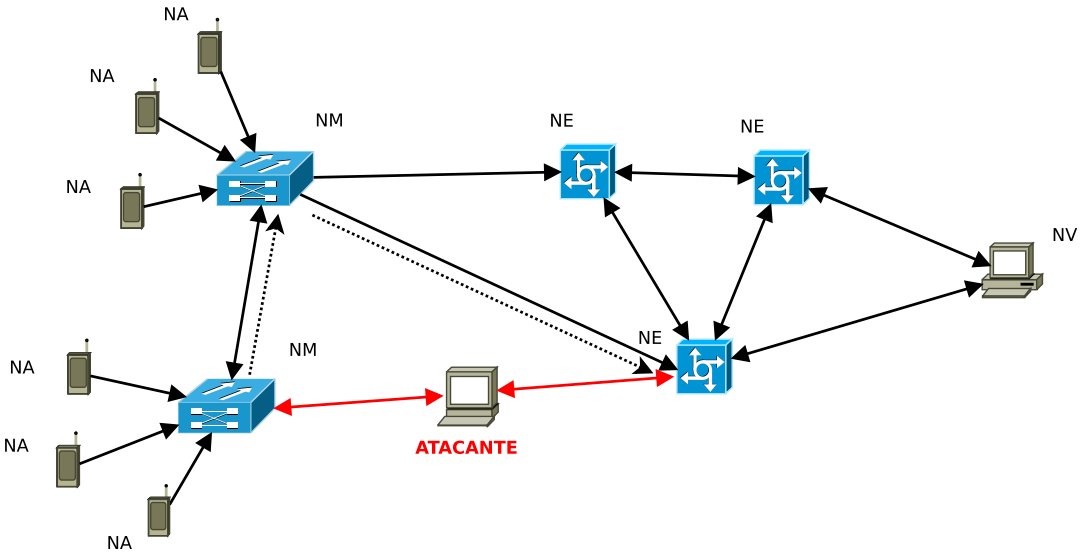

FIGURA 5.4: Nueva ruta de información

Alteración de la información. Otra de las posibilidades de un atacante que logra con éxito realizar un ataque *man in the middle* es alterar los datos que se transmiten entre nodos. Ante esta situación el software presenta varios frentes de defensa. El primero y más importante se trata de detectar que hay un cambio en la información tránsmitida, esta detección se encuentra debido a que al modificar alguno de los datos enviados, la firma digital realizada por el nodo maestro no encajará con la información. En la figura 5.5 se aprecia el contenido de un fichero con con los datos transmitidos y su correspondiente firma.

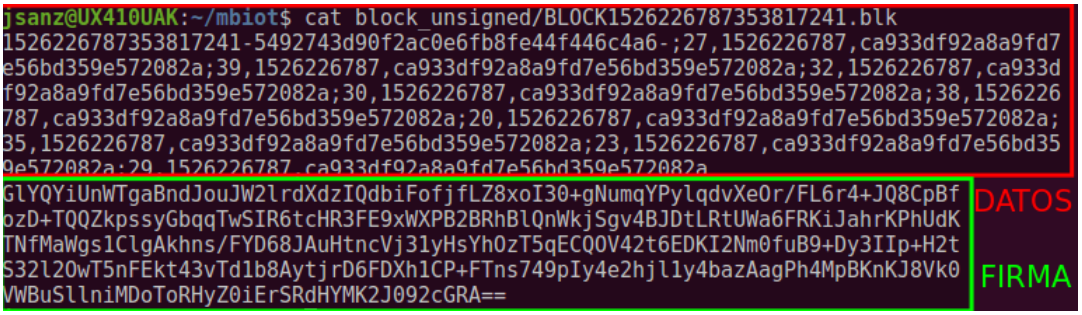

FIGURA 5.5: Información firmada digitalmente.

Pero si se modifica uno de los datos como se aprecia en la figura 5.6, en el proceso de comprobación donde se genera una nueva firma a partir de los datos, el resultado de la comprobación de la integridad de los datos indicará que la verificación ha sido fallida (figura 5.7). En este caso los datos se descartan.

```
jsanz@UX410UAK:~/mbiot$ cat block_unsigned/BLOCK15262267873538172 41.blk
15262267873538172 41-5492743d90f2ac0e6fb8fe44f446c4a6-;28,1526226787,ca933df92a8a9fd7e56bd35
9e572082a;39,1526226787,ca933df92a8a9fd7e56bd359e572082a;32,1526226787,ca933df92a8a9fd7e56b
d359e572082a;30,1526226787,ca933df92a8a9fd7e56bd359e572082a;38,1526226787,ca933df92a8a9fd7e
56bd359e572082a;20,1526226787,ca933df92a8a9fd7e56bd359e572082a;35,1526226787,ca933df92a8a9f
d7e56bd359e572082a;23,1526226787,ca933df92a8a9fd7e56bd359e572082a;29,1526226787,ca933df92a8
a9fd7e56bd359e572082a
GlYQYiUnWTgaBndJouJW2lrdXdzIQdbiFofjfLZ8xoI30+gNumqYPylqdvXeOr/FL6r4+JQ8CpBf
ozD+TQQZkpssyGbqqTwSIR6tcHR3FE9xWXPB2BRhBlQnWkjSgv4BJDtLRtUWa6FRKiJahrKPhUdK
TNfMaWgs1ClgAkhns/FYD68JAuHtncVj31yHsYhOzT5qECQOV42t6EDKI2Nm0fuB9+Dy3IIp+H2t
S32l2OwT5nFEkt43vTd1b8AytjrD6FDXh1CP+FTns749pIy4e2hjl1y4bazAagPh4MpBKnKJ8Vk0
VWBuSllniMDoToRHyZ0iErSRdHYMK2J092cGRA==
```

FIGURA 5.6: Información modificada.

```
Verification Failure
```

FIGURA 5.7: Proceso de verificación fallido.

5.2.2. Introducción de nodo ilícito

Este tipo de ataque se caracteriza por la inclusión en la red de un nodo no autorizado o ilícito. Este nodo podría interconectarse con el resto de nodos, tanto nodos esclavos como maestros, recibiendo los datos transmitidos en el sistema. La representación gráfica de esta situación puede verse en la figura 5.8.

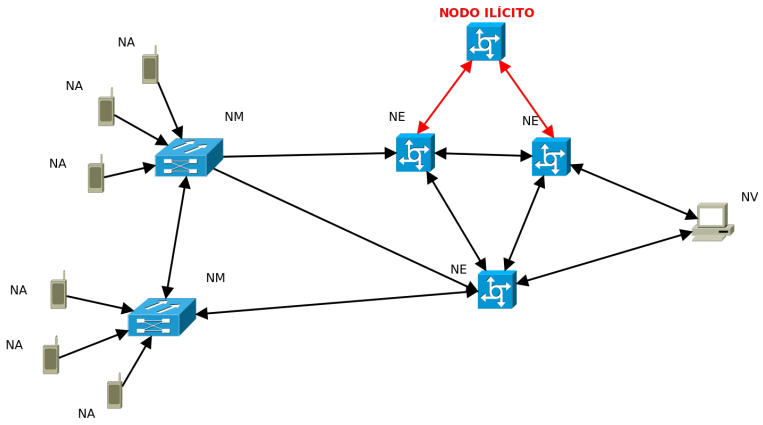

FIGURA 5.8: Introducción de un nodo ilícito en la red.

En esta situación, el atacante tiene la oportunidad de realizar una validación falsa de la información recibida desde los sensores con objeto de modificar los datos, además de realizar cambios sobre alguno de los datos anteriores para posteriormente propagarlos por los demás nodos.

Para evitar que el nodo ilícito perturbe la información legítima, cada vez que a un nodo le llega nueva información, comprueba la correspondencia del hash con el nonce, el hash del bloque anterior, la firma del nodo maestro y los datos. De tal forma que si no hay concordancia, se descarta el nuevo bloque.

En un ejemplo práctico se puede ver que en la figura 5.9 se encuentra un hash legítimo que valida toda la información del bloque.

Si se quiere modificar alguno de los datos contenidos en el bloque (cualquiera de ellos), en la etapa de comprobación el hash calculado no tendría ninguna similitud con el existente en el bloque (figura 5.10).

FIGURA 5.9: Captura de nodo con hash legítimo

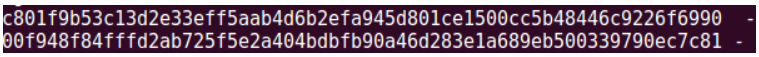

FIGURA 5.10: Primera fila: Hash generado con la información del bloque. Segunda fila: Hash incluido en el bloque.

5.2.3. Suplantación de nodo maestro

En este caso, también se introduce un nodo ilegal en la red, aunque en esta ocasión simula ser un nodo maestro. Al comportarse como un nodo maestro, este nodo podría generar bloques con información falsa para hacérselos llegar a los nodos esclavos. Tal y como se ve en la figura 5.11, uno de los nodos maestro ha sido suplantado, de manera que tanto los sensores como los nodos esclavos intercambian información con él.

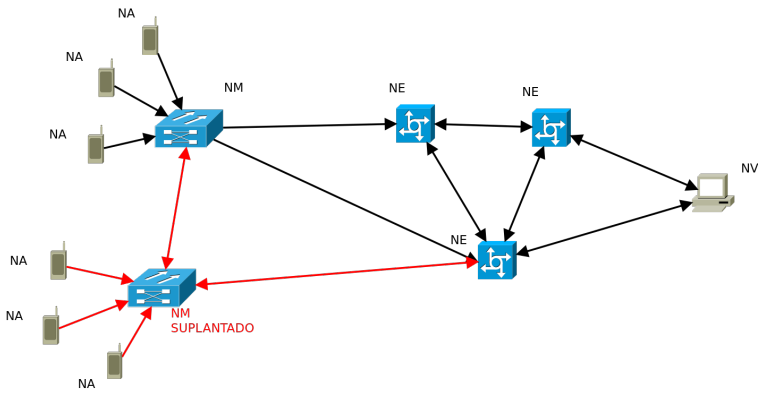

FIGURA 5.11: Suplantación de un nodo maestro.

Esta situación generaría un esfuerzo extra a la red debido a que los nodos esclavos no tienen obligación de tener las claves publicas de todos los nodos maestros. Los nodos esclavos seguirían validando toda la información que les llegase indistintamente. Sin embargo, el nodo visualizador si que debe tener las claves apropiadas que certifiquen quien es el origen de la información, de tal manera que en este supuesto, al no reconocer como válido el origen, se descarta todo la información que llegue.

5.2.4. Alteración de información ya existente en el ledger

Otro posible ataque consiste en modificar o alterar la información ya escrita en el ledger de una transacción pasada. Tal y como se muestra en la figura, cierto nodo malicioso trata de alterar datos validados por algún nodo legítimo, con objeto de aprovechar que solo se valida la última información recibida. La figura 5.12 representa este tipo de ataque.

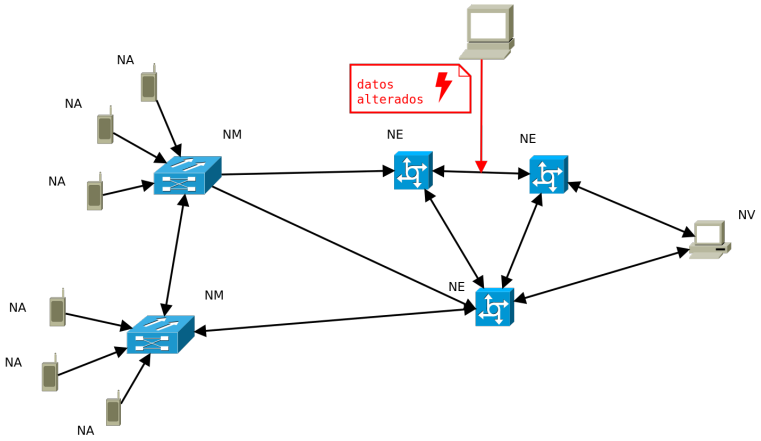

FIGURA 5.12: Alteración de la información ya existente en el ledger.

Sin embargo, en esta situación blockchain es donde saca su mayor potencial: en el capítulo 2 se definía la inmutabilidad del ledger como una de sus características más destacables, de tal forma que un nodo que reciba un bloque antiguo con que presente discrepancias con el que tiene almacenado, será descartado inmediatamente.

En el caso de un nodo que se sincroniza por primera vez con la red y recibe esta información alterada, verificará la concordancia entre hashes e información de una forma similar a la del apartado anterior mostrando un mensaje de error (figura 5.13) y desechando los datos recibidos

```
====================================
== Comprobaci?n integridad bloque ==
====================================
slave hasher bien:  ./block_signed/BLOCK1526226797837459541.blk
nonce signer:  5715
nonce ledger:  5715
====================================
== Comprobaci?n integridad bloque ==
====================================
slave hasher bien:  ./block_signed/BLOCK1526226803028362486.blk
nonce signer:  4109
nonce ledger:  4109
====================================
== Comprobaci?n integridad bloque ==
====================================
slave: hash no coincide con la información
slave:  ./block_signed/BLOCK1526226808206785700.blk
```

FIGURA 5.13: Detección de discordancias en los bloques recibidos.

5.2.5. Vulnerabilidad del 51 % en blockchain

Esta vulnerabilidad es de obligada inclusión por ser intrínseca a blockchain y se origina debido a su método de validación de los datos: consenso por mayoría. Cuando hay multiples nodos con información distinta, cada nodo emite un voto por la información que contiene, siendo los datos que más confianza tengan los que se identifican como válidos. Por esto mismo, si se diese el caso en que se realizase un

ataque en el que los nodos ilícitos superase a los lícitos (es decir, tuviesen al menos el 51 % de los votos) la información válida podría ser corrompida perdiendo la red su integridad [5]. Un ejemplo de esta situación referido al escenario que se ha estado usando para las pruebas la encontramos en el supuesto de la figura 5.14.

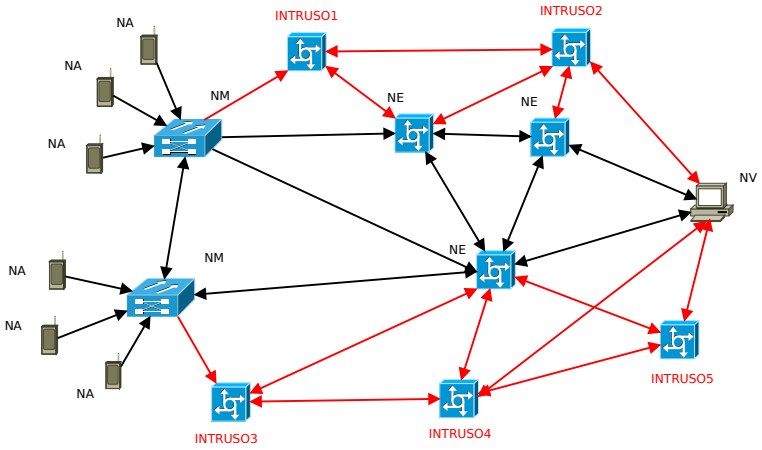

FIGURA 5.14: Vulnerabilidad del 51 % en blockchain.

Ante esta situación la red no tiene forma de garantizar que lleguen al nodo visualizador información legítima.

5.2.6. Conclusiones de la implementación

A lo largo de este capítulo se han expuesto una serie de situaciones que conformarían la forma de intento de vulneración más elemental y aquella en la que blockchain es vulnerable debido a su arquitectura de forma específica. En los aspectos positivos, se ha observado como el sistema se sobrepone a las situaciones en las que una máquina intenta alterar el correcto funcionamiento de la red, cumpliendo los objetivos para lo que fue concebido, haciendo especial incapié en la integridad y en la inmutabilidad de los datos. De forma mejorable, se observa que la información de un nodo maestro ilegítimo llega a formar parte de la red, a pesar de que finalmente estos datos serán desechados al alcanzar el destino. Esta situación, aunque aceptable por cumplir objetivos, no se entiende como óptima de cara a rendimiento.

En el aspecto negativo se puede distinguir dos tipos distintos: los problemas e ineficiencias intrínsecas a blockchain y los derivados del uso de blockchain en el ámbito IoT al utilizar una verificación mediante *Proof of Work*. Proof of Work es un proceso donde se validan transacciones mediante el cálculo de operaciones matemáticas que hacen uso de toda la capacidad del procesador, haciendo que el consumo energético alcance el máximo y el procesador eleve su temperatura. Esta situación presenta dos inconvenientes, por un lado, la posibilidad de alimentar los aparatos mediante baterías se presenta complicado al reducir su autonomía (ver Apendice B), y por otro, la reducción de la vida útil de los aparatos al usar sus componentes de forma intensiva a una temperatura mayor.

Blockchain hace un uso de una validación democrática, donde cada nodo es un voto y la información de la mayoría es la que se identifica como valida. Esto deja a un sistema en desventaja cuando hace uso de un número pequeño de nodos, pues un atacante puede hacer caer una red por número como se identifica en el caso presentado en la figura 5.14.

Capítulo 6

Conclusiones

Durante este capítulo se analizarán los resultados obtenidos de forma práctica, comprobando si se han cumplido los objetivos planteados en el capítulo introductorio mediante el uso del software desarrollado. Se estudiarán además, los beneficios y perjuicios si les hubiera en la inclusión de la tecnología Blockchain en la Internet de las Cosas. También se detallarán las dificultades encontradas en el desarrollo del presente documento y las posibles lineas futuras de trabajo con objeto de mejorar y ampliar los objetivos logrados.

6.1. Análisis de resultados

A continuación se detallarán las conclusiones a los objetivos específicos planteados en el capítulo 1:

Los Principales problemas de seguridad en IoT descritos durante el capítulo 2 se han identificado como los principales problemas que hay en IoT es la fragmentación de dispositivos y que por lo general la potencia de estos aparatos es reducida. Esta situación se ha podido identificar de forma práctica durante las pruebas realizadas en el capítulo 5 donde para la implementación del escenario de pruebas compuesto por seis dispositivos, se han empleado cuatro sistemas distintos. Esta situación no es casual, el auge de IoT ha desembocado en la aparición de muchos nuevos fabricantes con infinidad de dispositivos que montan diferentes configuraciones de CPU, RAM, almacenamiento y otros, con el objeto de tener sistemas que se adapten al mayor número de potenciales clientes. A mayores de esta situación, la incesante aparición de nuevas mejoras de los componentes se encarga de que los sistemas desplegados se queden obsoletos mucho antes de haber terminado su ciclo de vida.

Las soluciones existentes en el mercado cuyo objetivo sea reforzar la seguridad de las redes son muchas debido a que no hay un solo enfoque que permita hacerlo. Sin embargo, si se busca en aquellas que hagan uso de la cadena de bloques como núcleo central de su funcionamiento se puede ver que ha día de hoy, la variedad no es muy grande y presentan claras deficiencias: *Hyperledger* de la Linux Foundation propone lo que se puede considerar una nueva pila de protocolos encima del sistema operativo que, desde un punto de vista académico resulta muy completo, sin embargo, complica mucho la estructura del la red y los dispositivos requerirán una potencia de procesamiento no despreciable.

Por otro lado, IBM cuenta con una gran reputación y mucha experiencia en el mundo de la informática en su sentido más amplio, siendo en muchas ocasiones, garante de productos de calidad. No obstante, la información proporcionada por su página web índica que su negocio se trata de despliegues *Ad Hoc* basándose en Hyperledger. Este tipo de despliegues personalizados elimina toda posibilidad de ser una solución generalista que pueda integrarse en cualquier situación.

Por último Ethereum, con todas sus derivaciones, está claramente enfocado en su aspecto de economía distribuida siguiendo la estela de Bitcoin. Este comportamiento parece razonable al estar en una situación que reporta beneficios inmediatos a especuladores económicos y usuarios que ceden potencia de cálculo de sus ordenadores para mantener la red. Sin embargo, pese a este enfoque económico, es una de las opciones mejor posicionadas para integrarse en otros ámbitos al disponer de una API para su implementación en sistemas, además de los *smart contract* que permiten incluir nuevas funcionalidades de forma flexible.

El uso de la aplicación desarrollada en este trabajo tras haber sido analizada en el capítulo 5 permite una serie de conclusiones que reflejen los aspectos positivos, conclusiones donde su comportamiento es adecuado, y donde debe mejorar.

El comportamiento del software cabe destacar que es muy positivo de cara a la integridad, en aquellas ocasiones en las que un atacante pretenda interrumpir una conexión entre dos nodos, debido fundamentalmente a que la configuración de red distribuida habilita multiples rutas entre dispositivos. Además, el sistema presentado es muy resistente a todos los intentos de alteración de la información, ya sea en tiempo real o datos almacenados en el histórico debido a la redundancia y el encadenamiento entre bloques propios de blockchain. Estas ventajas mencionadas que aporta el sistema se acentúan cuantos más dispositivos compongan la red debido a que aumentan las copias de los datos y se generan más rutas entre nodos.

Debido a que las claves públicas de los nodos maestros solo las posee de forma obligatoria el nodo destino, la información generada por nodos maestros maliciosos sería validada, añadiendo carga innecesaria a la red. Esta situación no se presenta óptima debido a que aunque la red seguiría funcional en cualquier caso, se estaría gastando poder de computo en validar datos falsos que posteriormente serán descartados por el dispositivo encargado de procesar la información.

En el aspecto negativo, en la implementación del software se cumple una de las debilidades conocidas de blockchain que se da en el caso de que los nodos ilegítimos superen en número a los legítimos y den por válida información falsa, pues los nodos legítimos de la red no tendrían capacidad de identificar que los datos que reciben desde múltiples fuentes son los correctos. Además, en base a las pruebas realizadas, es posible ver como durante el proceso de verificación, los dispositivos usan en su totalidad la capacidad de la CPU para resolver las operaciones matemáticas, elevando su consumo energético sensiblemente. Esta situación, plantea muchas dificultades al uso de blockchain mediante la técnica de *Proof of Work* en dispositivos alimentados por baterías.

Sintetizando toda la información extraída, se puede concluir que la solución aportada permite establecer una barrera de seguridad a las ya existentes o aportar suficiente seguridad por sí misma a un entorno de funcionamiento donde haya un mínimo de dispositivos.

6.2. Dificultades encontradas

Durante la redacción de este documento se han encontrado algunas dificultades inesperadas que solucionar. En la recolección de información para realizar el estado del arte se ha vuelto complejo encontrar información técnica que indicase el funcionamiento en profundidad de protocolos basados en blockchain, la mayor parte de la información se centra en mencionar su potencial y posibles aplicaciones. De la misma forma, en la búsqueda de soluciones a la problemática de IoT mediante

blockchain, con la excepción de Hyperledger no ha sido posible establecer el método de funcionamiento de las soluciones, mencionando únicamente las ventajas de su uso.

6.3. Reflexión personal sobre Blockchain

Por último, me gustaría escribir unas lineas con un criterio puramente personal sobre blockchain con la experiencia adquirida después de realizar este trabajo. Blockchain, es ante todo una tecnología muy reciente, probada con éxito en el ámbito de las criptomonedas y con potencial para entrar en muchos nuevos ámbitos debido a su flexibilidad. Sin embargo, como tecnología nueva que es, todavía tiene muchos frentes abiertos en los que mejorar: el alto consumo energético, el desproporcionado poder de proceso usado en tareas de verificación o la necesidad de tener los nodos lícitos en mayoría, son algunos aspectos claros donde se necesita incidir para que blockchain pueda abordar finalmente aquellos campos donde se requiere un extra de seguridad.

6.4. Lineas de trabajo futuras

La linea de trabajo seguida hasta este punto puede servir de inspiración para continuar por otros caminos como los que se proponen a continuación:

Durante la realización de este trabajo se encontraron nuevas formas de validación que surgen con objeto de sustituir Proof of Work. La exploración de estos distintos sistemas podría proporcionar una mejora en aquellos puntos de debilidad estudiados en el presente documento.

Otra linea de investigación puede abrirse de cara a la automatización del software en el momento de la configuración, eligiendo la óptima en función de las características de la red.

Por último se propone establecer un nivel mayor de seguridad mediante un sistema que permita a los distintos nodos generar sus propios certificados y compartir sus claves públicas con objeto de interceptar en una fase más temprana los posibles intentos de vulneración de la red.

Apéndice A

Estructura de ficheros y Diagrama de funcionamiento

A.1. Estructura de ficheros

El proceso descrito en el capítulo 4 se implementa a través de una serie de ficheros y carpetas que componen el software, siguiendo la estructura identificada en la imagen A.1

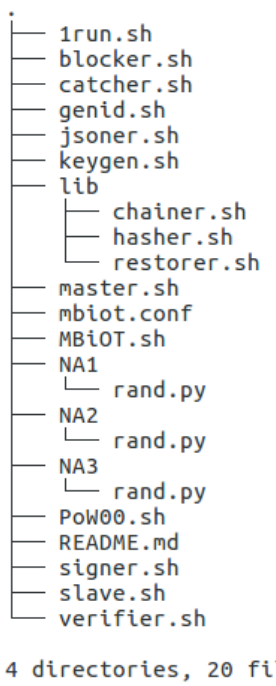

FIGURA A.1: Árbol de directorios de la solución propuesta.

A.2. Diagrama de funcionamiento

Para describir el hilo de ejecución del software, se incluyen a continuación una serie de diagramas de flujo por el que se puede seguir el funcionamiento del mismo:

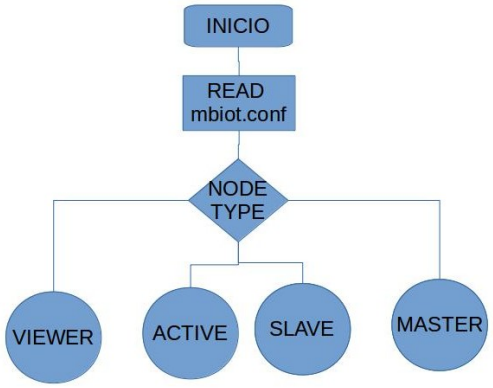

FIGURA A.2: Diagrama de flujo del ejecutable principal.

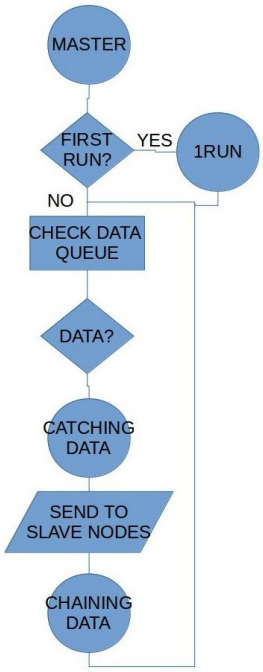

FIGURA A.3: Diagrama de flujo del nodo maestro.

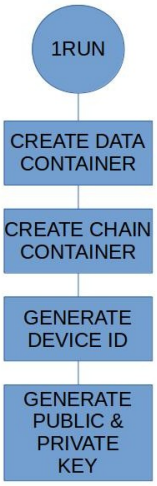

FIGURA A.4: Diagrama de flujo de la primera ejecución.

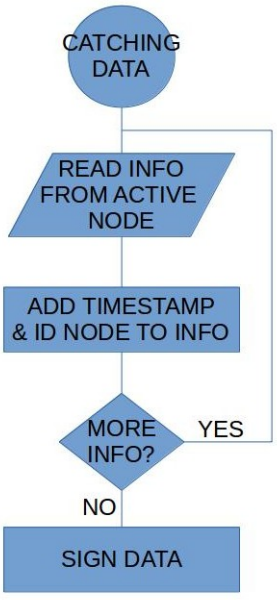

FIGURA A.5: Diagrama de flujo de çatcher.sh".

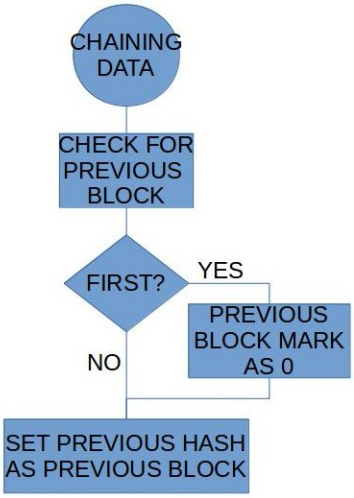

FIGURA A.6: Diagrama de flujo dela generación de un bloque.

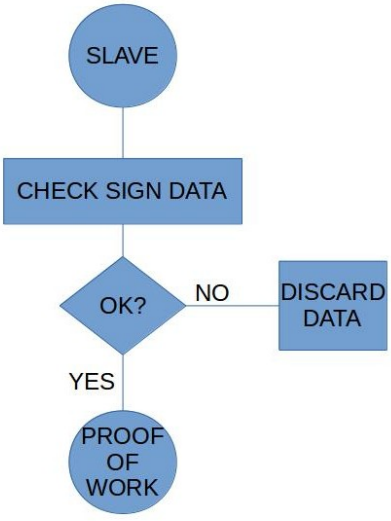

FIGURA A.7: Diagrama de flujo de nodo esclavo.

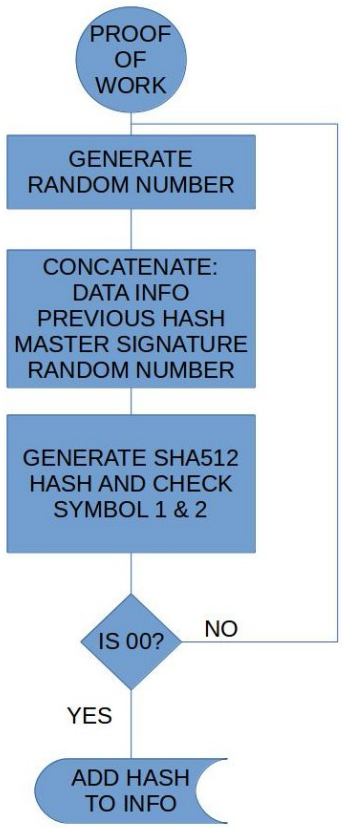

FIGURA A.8: Diagrama de flujo de *Proof of Work*.

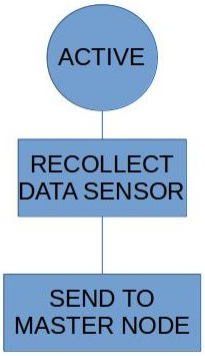

FIGURA A.9: Diagrama de flujo de nodo activo.

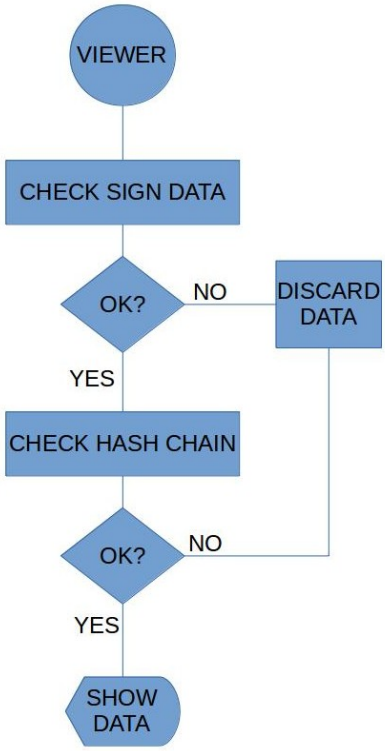

FIGURA A.10: Diagrama de flujo de visualizador.

Apéndice B

Consumo energético

En este apéndice se realiza una aproximación del impacto energético de la utilización del software en un nodo que realice la validación mediante *Proof of Work* dado que es el elemento de la red que más carga computacional debe soportar. Para ello se ha empleado la placa SBC Raspberry versión 1 modelo B (tabla B.1) con una instalación nueva del sistema operativo Raspbian, basado en el sistema operativo Debian y adaptado para procesadores ARM. Este conjunto se ha conectado a una placa de alimentación "Makerfocus Raspi UPS HAT Board" que dispone de una batería de litio con capacidad para 2500mAh y la posibilidad de brindar el estado de la batería mediante conexión spi.

Dispositivo	Rasberry Pi 1 model B
SoC	BCM2835
CPU	ARM11 700MHz
RAM	512MB
S.O.	Raspbian

TABLA B.1: Tabla comparativa de dispositivos empleados.

Las pruebas han consistido en arrancar el sistema y comprobar con frecuencia de un minuto el estado de la batería hasta su agotamiento. En un primer lugar, se ha realizado sin que el la solución planteada se encuentre en ejecución, y posteriormente, con la solución planteada activa.

En la gráfica B.1 se identifica el voltaje entregado por la batería al sistema cada minuto, alcanzando los 722 minutos (aproximadamente 12 horas) antes de apagarse.

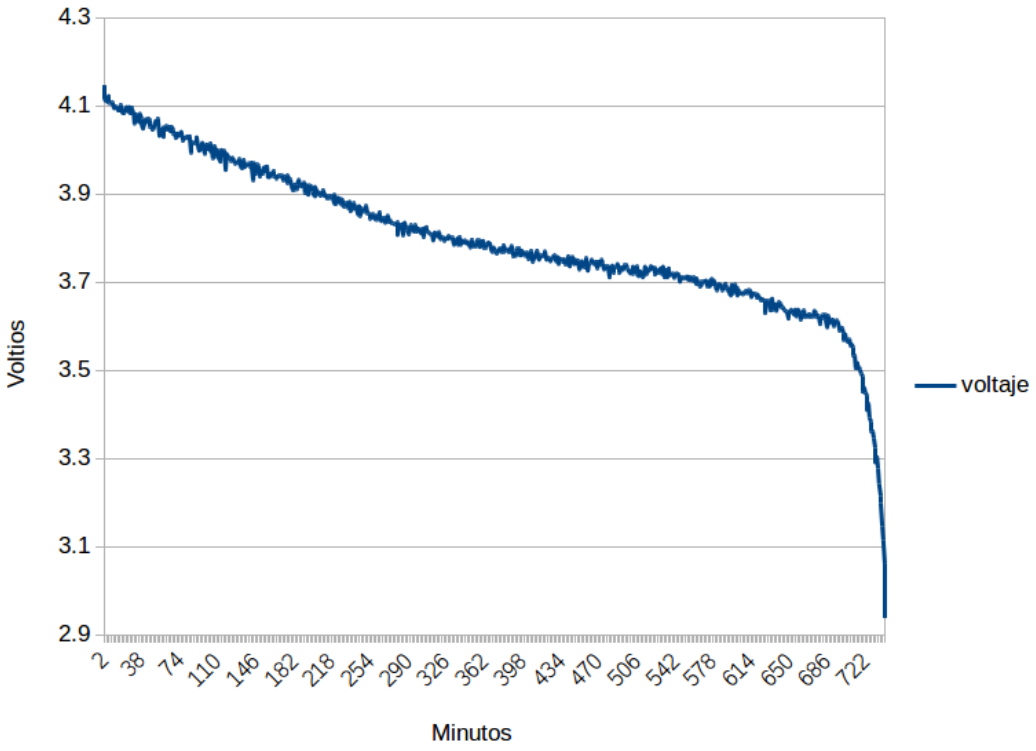

FIGURA B.1: Consumo del sistema en estado de espera.

En la gráfica B.2 se puede ver como el tiempo capaz de funcionar el sistema con una carga alta de trabajo se reduce drásticamente a los 301 minutos (aproximadamente 5 horas).

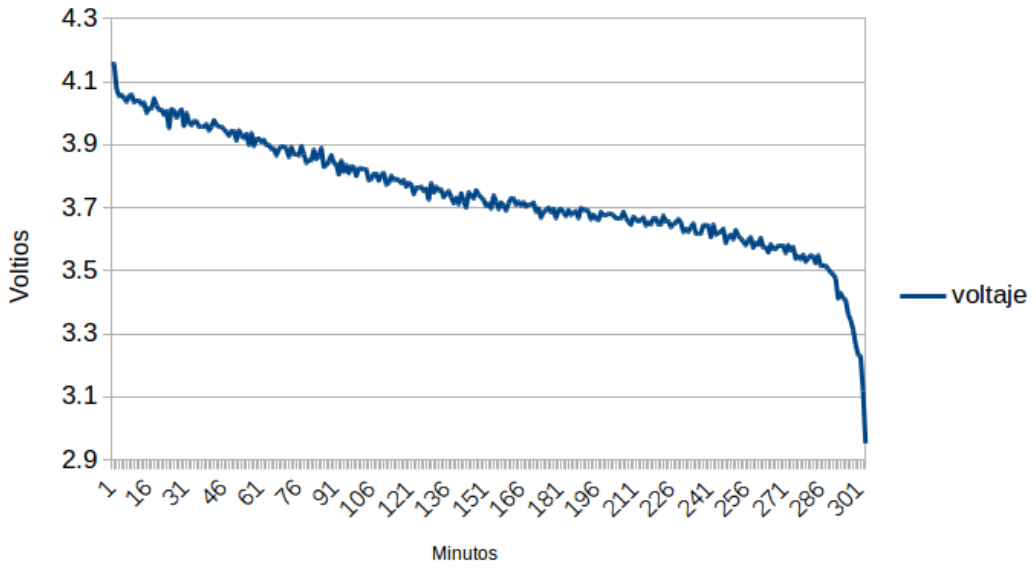

FIGURA B.2: Consumo de batería con la solución propuesta y alta carga.

Comparando los resultados anteriores se puede concluir en esta prueba de consumo, que hacer uso de *Proof of Work* influye de manera significativa en el consumo eléctrico.

Bibliografía

[1] *The Internet of Things*. 2013. URL: https://ec.europa.eu/digital-single-market/en/internet-of-things.

[2] *Gartner Says 8.4 Billion Connected "Things"Will Be in Use in 2017, Up 31 Percent From 2016*. 2017. URL: https://www.gartner.com/newsroom/id/3598917.

[3] Patrick Guillemin Harald Sundmaeker Markus Eisenhauer Klaus Moessner Marilyn Arndt Maurizio Spirito Paolo Medagliani Raffaele Giaffreda Sergio Gusmeroli Latif Ladid Martin Serrano Manfred Hauswirth Ovidiu Vermesan Peter Friess y Gianmarco Baldini. «Internet of Things strategic research and innovation agenda». En: *Internet of Things - From Research and Innovation to Market Deployment* (2014).

[4] Kim Zetter. «Countdown to Zero Day: Stuxnet and the Launch of the World's First Digital Weapon». En: *Crown Publishing Group, New York* (2014).

[5] Aaron Wright y Primavera De Filippi. *Decentralized Blockchain Technology ans the Rise of Lex Cryptographies.* 2017.

[6] *Tapping the Potential of Distributed Ledgers to Improve the Post-Trade Landscape*. 2017. URL: http://www.dtcc.com/.

[7] *Bitcoin, A Peer-to-Peer Electronic Cash System*. 2010.

[8] Ken Schwaber y Jeff Sutherland. *The Scrum Guide*. Ed. por scrum.org. 2017. URL: www.scrum.org.

[9] Rolf H. Weber y Romana Weber. *Internet of Things*. 2010.

[10] Sunil M Patel Keyur K Patel. «Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges». En: *IJESC* (2016).

[11] Institute of Electrical y Electronics Engineers (IEEE), eds. *Advanced Computer Theory and Engineering. International Conference. 3RD 2010. (ICACTE 2010)*. 2010.

[12] Schoenwaelder J. Sehgal A. Ersue M. Romascanu D. «Management of Networks with Constrained Devices: Use Cases». En: *IETF Internet Draft* (2014).

[13] Martinez Kirk Hart Jane K. «Toward an environmental Internet of Things». En: *Earth & Space Science* (2016).

[14] K. F. Haase J. Alahmad M. Nishi H. Ploennigs J. Tsang. «The IOT mediated built environment: A brief survey». En: *IEEE 14th International Conference on Industrial Informatics (INDIN)* (2016).

[15] Melanie Swan. «Sensor Mania! The Internet of Things, Wearable Computing, Objective Metrics, and the Quantified Self 2.0». En: *Sensor and Actuator Networks* (2012).

[16] T.; Rossi A.; White D.; Cooper J. Kyriazis D.; Varvarigou. «Sustainable smart city IoT applications: Heat and electricity management & Eco-conscious cruise control for public transportation». En: *IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (2013).

[17] *IEEE 802.15 WPAN Task Group 4*. 2018.

[18] *The Constrained Application Protocol (CoAP)*. 2014.

[19] *OMA-ORG Guidelines Creation Registration LwM2M Objects Resources*. 2018.

[20] Steven Hendriks. «The Internet of Things». Master Thesis. Utrecht University, 2016.

[21] Ioana Rodhe Farzad Kamrani Mikael Wedlin. «Internet of Things: Security and Privacy Issues». Master Thesis. FOI Swedish Defence Research Agency, 2016.

[22] Stuart Taylor. «Predictions for the Future of the Internet of Things». En: *Cisco Blogs* (2015).

[23] A.S. Tanenbaum. *Computer Networks*. 2016.

[24] Michael Crosby. «Blockchain technology: Beyond bitcoin». En: *Applied Innovation* (2016).

[25] World Economic Forum. *All you need to know about blockchain, explained simply*. 2016. URL: `https://www.weforum.org/agenda/2016/06/blockchain-explained-simply/`.

[26] Don Tapscott y Alex Tapscott. *Blockchain Revolution: How the Technology Behind Bitcoin Is Changing Money, Business, and the World*. 2016.

[27] *BlockchainHub.Types of Blockchain*. 2016. URL: `https://blockchainhub.net/blockchains-in-general/`.

[28] Sören Matthew Auer y John Domingue. «Block chain technologies & the semantic web: A framework for symbiotic development». En: *Computer Science Conference for University of Bonn Students, J. Lehmann, H. Thakkar, L. Halilaj, and R. Asmat* (2016).

[29] Zibin Zheng. *Blockchain Challenges and Opportunities: A Survey*. 2016.

[30] Georg Becker. *Merkle Signature Schemes, Merkle Trees and Their Cryptanalysis*. 2013.

[31] Arthur Gervais. «On the security and performance of proof of work blockchains». En: *Conference on Computer and Communica- tions Security* (2016).

[32] *Blackcoin's proof-of-stake protocol v2. 2014*. 2015. URL: `http://blackcoin.co/blackcoin-pos-protocol-v2-whitepaper.pdf`.

[33] Shostak R. Pease M. Lamport L. *The Byzantine Generals Problem*. 1982.

[34] David W Chadwick. «Federated identity management». En: *Foundations of security analysis and design V* (2009).

[35] *Digital currencies: call for information*. 2015. URL: `https://www.gov.uk/government/consultations/digital-currencies-call-for-information/digital-currencies-call-for-information.`.

[36] Clayton Christensen. «Disruptive Innovation». En: *http://www.claytonchristensen.com/key-concepts/* (2017).

[37] Michael Crosby. «Blockchain technology: Beyond bitcoin». En: *Applied Innovation 2* (2016).

[38] Melanie Swan. *Blockchain: Blueprint for a new economy*. 2015.

[39] *itnation starts offering blockchain public notary service to Estonian e-Residents*. 2015. URL: https://bravenewcoin.com/news/bitnation-starts-offering-blockchain-public-notary-service-to-estonian-e-residents/.

[40] Philip Boucher. «What if blockchain technology revolutionised voting?» En: *Scientific Foresight Unit (STOA)* (2016).

[41] *Decentralizing IoT networks through blockchain*. 2016. URL: https://techcrunch.com/2016/06/28/decentralizing-iot-networks-through-blockchain/.

[42] Kyle Croman. «On scaling decentralized blockchains». En: *nternational Conference on Financial Cryptography and Data Security* (2016).

[43] *Blockchain scalability : A look at the stumbling blocks to blockchain scalability and some high-level technical solutions*. 2015. URL: https://www.oreilly.com/ideas/blockchain-scalability.

[44] Ahmed Kosba. «Hawk: The blockchain model of cryptography and privacy preserving smart contracts». En: *Security and Privacy (SP), 2016 IEEE Symposium on (2016)* (2016).

[45] *Reclaiming Financial Privacy With HD Wallets*. 2013. URL: http://bitcoinism.blogspot.nl/2013/07/reclaiming-financial-privacy-with-hd.html.

[46] Christian Sprecher y Ulrich Gallersdörfer. «Challenges and Risks of Blockchain Technology». Master Thesis. Technical University of Munich, 2017.

[47] The Linux Foundation. *About Hyperledger*. 2018. URL: https://www.hyperledger.org/about.

[48] The Linux Foundation. *HyperLedger-Fabric. Introduction*. 2018. URL: http://hyperledger-fabric.readthedocs.io/en/master/blockchain.html.

[49] The Linux Foundation. *Hyperledger Fabric Model*. 2018. URL: http://hyperledger-fabric.readthedocs.io/en/master/fabric_model.html.

[50] *Implement your first IoT and blockchain project*. 2018. URL: https://www.ibm.com/internet-of-things/spotlight/blockchain.

[51] International Business Machines Corporation (IBM). *IBM Blockchain Platform. Technical Overview*. 2018. URL: https://www.ibm.com.

[52] *Framework*. 2018. URL: https://es.wikipedia.org/wiki/Framework.

[53] International Business Machines Corporation (IBM). *Trusting the transaction of things: IoT and blockchain intersect*. 2018. URL: https://www.ibm.com.

[54] José Surga. *¿Qué son los Tokens ERC20 de Ethereum y cómo funcionan?* 2017. URL: https://www.criptonoticias.com/colecciones/tokens-erc20-ethereum-como-funcionan/.

[55] *ERC20 Token Standard*. 2018. URL: https://theethereum.wiki/w/index.php/ERC20_Token_Standard.

[56] Raiden Network. *What is the Raiden Network?* 2018. URL: https://raiden.network/101.html.

[57] Raúl Jordan. *How to Scale Ethereum: Sharding Explained*. 2018. URL: https://medium.com/prysmatic-labs/how-to-scale-ethereum-sharding-explained-ba2e283b7fce.

[58] Joseph Poon y Vitalik Buterin. *Plasma: Scalable Autonomous Smart Contracts*. 2017. URL: https://plasma.io/.

[59] *¿Qué es GNU?* 2018. URL: www.gnu.org.

[60] *Internet Protocol, Version 6 (IPv6) Specification*. 2017. URL: https://tools.ietf.org/html/rfc8200.

[61] *OpenSSL. Cryptography and SSL/TLS Toolkit*. 2018. URL: https://www.openssl.org/.

[62] Franco Callegati & Walter Cerroni. *Man-in-the-Middle Attack to the HTTPS Protocol*. 2009.