UNIVERSITAT POLITÈCNICA DE VALÈNCIA

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Gandia

Sistema de registro de contenido multimedia basado en redes descentralizadas

Trabajo Fin de Grado

Grado en Tecnologías Interactivas

AUTOR/A: Egea Escriba, Javier

Tutor/a: Sendra Compte, Sandra

Cotutor/a: Lloret Mauri, Jaime

CURSO ACADÉMICO: 2023/2024

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITÈCNICA SUPERIOR DE GANDIA

Grado en Tecnologías Interactivas

Sistema de Registro de Contenido Multimedia Basado en Redes Descentralizadas

TRABAJO FINAL DE GRADO

Autor/a:

Javier Egea Escribá

Tutor/a:

Sandra Sendra Compte

Jaime Lloret Mauri

ÍNDICE

ÍNDICE DE ILUSTRACIONES

Resumen

En este documento se expone y detalla un sistema innovador que ha sido desarrollado específicamente para verificar la autenticidad de los contenidos multimedia con los que interactuamos diariamente en diversas plataformas digitales, como redes sociales, sitios web, e incluso en sistemas de mensajería instantánea.

Como seres humanos, es fundamental poder discernir si el contenido visual o auditivo que estamos observando o escuchando en un momento dado ha sido fielmente capturado de la realidad o si ha sido manipulado o fabricado. Esta capacidad de distinguir nos permite evaluar si la información que adquirimos a través de estas redes es verídica o no. Es crucial saber si los eventos y datos que encontramos en línea realmente han tenido lugar o si son simplemente fabricaciones.

Primero se expondrá la motivación que provocó la creación de este sistema, así como los objetivos y problemas que resuelve el sistema. Puesto que resulta totalmente necesario para comprender la seguridad que aporta el **Sistema de Registro de Contenido Multimedia** (SRCM), será explicada la metodología empleada para la realización del proyecto y problemas u obstáculos encontrados durante el desarrollo.

Se realizará un recorrido por cada uno de los elementos que componen el sistema explicándolos detalladamente, ya que se utilizan tecnologías como Blockchain, redes P2P, cifrados Hash, función que toma una cadena de texto como entrada y la comprime en otra cadena de texto como salida, tipo Keccak256 y 512; incluso algunos de los elementos han sido especialmente creados para la lógica y el funcionamiento del propio sistema.

1- Introducción

El uso cotidiano de las tecnologías siempre ha implicado la aparición de muchas nuevas variables en nuestro entorno. Un ejemplo significativo es el de las inteligencias artificiales (IA), una tecnología que ha transformado múltiples aspectos de nuestra vida diaria. Sin embargo, esta revolución tecnológica también ha traído consigo desafíos importantes, especialmente en relación con la veracidad de los contenidos multimedia que consumimos en Internet.

El fenómeno de la creación de contenido falso no es nuevo, pero la capacidad de las IA para generar vídeos, audios e imágenes hiperrealistas sin basarse en datos reales ha llevado esta problemática a un nivel sin precedentes. En 2023, el Washington Post informó sobre un aumento significativo en la diseminación de noticias falsas y desinformación generadas por IA, destacando el impacto perjudicial que estas tecnologías pueden tener en la percepción pública y la confianza en la información (Verma, 2023).

De acuerdo con una investigación de Ipsos, el 63% de los adultos a nivel mundial han sido expuestos a noticias falsas en línea, y este porcentaje solo sigue creciendo en la era de la IA (Dunne, 2023). Además, un estudio de Sumsub reveló que los incidentes globales de deepfakes aumentaron diez veces de 2022 a 2023, abarcando desde la suplantación de identidad hasta la creación de contenido pornográfico no consentido y la manipulación de información sensible (Sumsub, 2023) .

El problema se agrava cuando las personas empiezan a cuestionar la autenticidad de prácticamente todo lo que ven en línea: "¿Esto será real?". Las implicaciones de la capacidad de una IA para crear contenidos que incluyen pornografía infantil, abusos, violencia hacia la mujer, suplantación de identidad y terrorismo son profundamente preocupantes. El nivel de realismo alcanzado por estas creaciones puede hacer que sea casi imposible diferenciar entre lo real y lo falso.

Aunque podemos intentar ser optimistas sobre el futuro, es probable que estas preguntas sobre la autenticidad del contenido digital se vuelvan cada vez más frecuentes. Podríamos llegar a un punto donde sea necesario utilizar IA para distinguir creaciones de otras IA, una situación que podría describirse coloquialmente como "apagar fuego con fuego". Aunque plausible, esta no sería la solución más efectiva.

Este proyecto trata de crear una solución preventiva para la situación planteada. Propone un estándar que registre el contenido multimedia en una cadena de bloques de datos descentralizada, permitiendo la verificación de cada contenido registrado sin depender de las IA.

La verificación funcionaría de la siguiente manera: si un determinado contenido ha sido extraído de la realidad, quedará registrado en el sistema y, por tanto, podrá verificarse que ha sido extraído de la realidad, sin posible alteración alguna de los datos.

Cuando surgiese cualquier duda sobre la veracidad de cualquier contenido, simplemente habría que comprobar si está registrado en el sistema.

1.1- Objetivos

1.1.1- Objetivo Principal

A partir de este punto nos referiremos al sistema en el que se centra el proyecto como el SRCM (Sistema de Registro de Contenido Multimedia). El SRCM tiene como objetivo **convertirse en un estándar con el que se consiga confirmar la veracidad** de cualquier tipo **de contenido multimedia** que haya sido **creado extrayendo los datos directamente de la realidad**, a partir del momento en que el SRCM fuese implementado en los dispositivos. Este sistema posee la propiedad de ser más eficiente cuanto mayor es el número de dispositivos que lo tienen incorporado.

1.1.2- Objetivos Secundarios

Una vez definido el objetivo principal del proyecto, se destacan a continuación los objetivos secundarios:

- Es importante que, para la **imparcialidad en las decisiones de autenticidad** de cualquier contenido multimedia, no haya ningún intermediario humano en el proceso de registro o consulta de veracidad.

- Observando el hecho de que los **mecanismos de consenso** actuales *[procedimiento en el que los pares (o nodos) de una red blockchain llegan a un acuerdo sobre el estado actual de los datos en la red* (Crypto.com, 2022). *Este acuerdo suele llevarse a cabo mediante la recompensa o castigo a dichos nodos por medio de criptomonedas. A través de esto, los algoritmos de consenso establecen fiabilidad y confianza en la red blockchain]*, presentes en las Blockchain o Cadenas de Bloques, consisten en métodos de confianza por medio de recompensa o castigo a sus nodos utilizando criptomonedas, se decide **crear un nuevo mecanismo de consenso que no implique el uso de criptomonedas** para su funcionamiento. Fue creado combinando una prueba de "Cero Conocimiento" con un mecanismo de consenso del tipo "Tiempo Transcurrido". El nombre de este mecanismo es "PoZET" o Proof of Zero-Knowledge Elapsed Time.

1.2- Metodología

Tal y como se puede apreciar en la [Fig. 1] "Diagrama de Gantt del proceso llevado a cabo", durante la realización del proyecto se pueden diferenciar distintas etapas que marcaron el curso del desarrollo:

1. **Observación**: El primer paso para desarrollar el proyecto fue, evidentemente, identificar el problema que estaba surgiendo: Un nuevo tipo de inteligencia capaz de recrear nuestras facultades, incluso ser mejor que el propio cerebro para ciertas funciones (Sadin, 2019) incrustándose cada vez más y más rápido en nuestro día a día. Gracias a esto se pudo identificar la necesidad que debía ser satisfecha cuanto antes: "¿Es esto real?"

2. **Investigación**: Seguidamente se realizó una investigación para conocer cómo funcionaban las Cadenas de Bloques que no fuesen privadas y contasen con un gran número de nodos. Se tomó de ejemplo la Blockchain de Ethereum (Wood, 2014). Además, se comprobó el comportamiento de los mecanismos de consenso existentes (Wackerow, 2024) (Crypto.com, 2022). La red P2P fue basada en una topología en árbol siguiendo una jerarquía de construcción de caminos o "Path Construction Hierarchy" (Korzum & Gurtov, 2012). Se investigó el lenguaje de programación Kotlin (Kotlin Foundation, 2024) para su posterior uso en el entorno de desarrollo Android Studio.

3. **Diseño**: La estructura del sistema fue diseñada e integrada totalmente de forma original. Las tecnologías como las Blockchain, redes P2P, encriptación Keccak512 y formato de datos CBOR (Bormann & Hoffman, 2020) ya son existentes. Sin embargo, fue necesario un cambio en la lógica de las Blockchain para poder implementar mecanismos como el PoZET anteriormente mencionados. Incluso fueron diseñados los paquetes de datos o bloques que almacenarían la información de cada contenido extraído. Se realizó un diseño completo el cual posteriormente fue digitalizado e implementado.

4. **Implementación**: Para comenzar a implementar el sistema al completo, se comenzó una segunda (aunque mucho más escueta) tanda de investigación un poco más técnica. La implementación de los servidores socket entre los nodos, así como la propia blockchain, fue basada en un proyecto realizado con anterioridad en la Universidad Politécnica de Valencia (Blasco, 2018).
   Este proyecto ha sido desarrollado en el entorno de programación Android Studio y está orientado a todos los dispositivos móviles que dispongan de un sistema operativo Android o similar a partir de la versión SDK 29 o Android 10.
   Para la realización de ciertas funciones básicas ya existentes en leguaje de programación Kotlin se ha utilizado la inteligencia artificial ChatGPT (OpenAI, 2023), únicamente buscando automatizar procesos como creación de bucles, funciones básicas, algunos algoritmos y sintaxis también comunes (implementación de callbacks, cómo se crean clases, objetos, tipos de variables, etc.); incluso traducción del lenguaje de programación Java a Kotlin.

5. **Prototipado:** Se ha realizado un prototipo incluyendo tres dispositivos móviles creando redes P2P entre ellos y analizando el comportamiento de los métodos implementados en los dispositivos. Se fueron implementando una a una las funcionalidades del sistema hasta completar la tarea de poder registrar imágenes en la cadena y pudieran ser corroboradas por los mismos dispositivos que disponen de la aplicación.

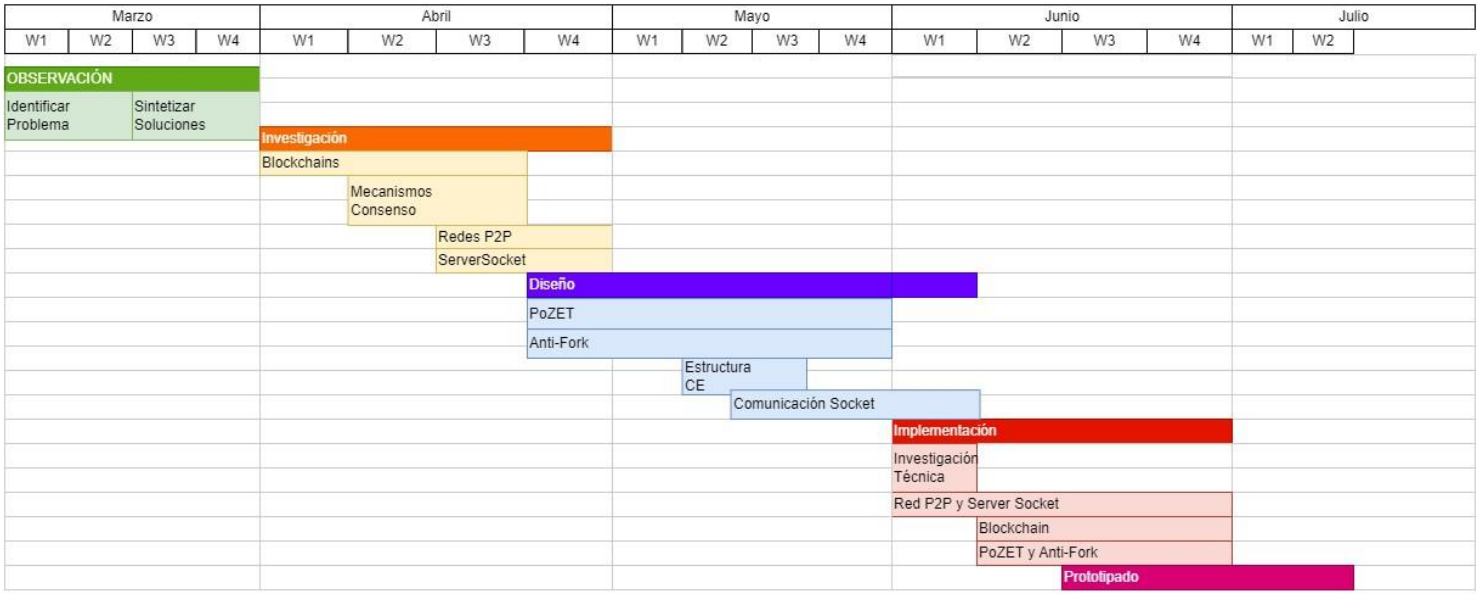

Fig. 1 Diagrama de Gantt del proceso llevado a cabo.

1.3- Etapas

El proyecto integra diversas tecnologías avanzadas, entre las que se incluyen Blockchain, una estructura de datos CBOR, hash de tipo Keccak256, un Mecanismo de Consenso diseñado específicamente para este proyecto denominado PoZET (Sec. 2.3), y un Algoritmo Anti-Bifurcaciones en Redes Blockchain (Sec. 2.4), cuyo objetivo es evitar la existencia de ramas divergentes de datos.

El Sistema de Registro de Contenido Multimedia (SRCM) opera en varias etapas o capas, cada una de las cuales desempeña un papel crucial en la verificación y validación del contenido multimedia:

- El nivel más bajo corresponde al trabajo realizado en cada dispositivo cuando se crea un contenido multimedia extraído directamente de la realidad. Este contenido, a partir de este momento, se denominará Contenido Extraído (CE). La función principal del nodo en esta etapa es transmitir los datos extraídos a los nodos correspondientes de la red, incluso antes de haber sido almacenados en la memoria del dispositivo. Esto asegura que el contenido sea registrado y verificado de inmediato, reduciendo al mínimo las posibilidades de manipulación o alteración.
- El segundo nivel, o capa, es la capa de topología de la red. En esta capa, el SRCM organiza los dispositivos conectados a la red según su geolocalización y su posición estratégica dentro de la estructura topológica del árbol. Es fundamental que el sistema administre esta etapa, ya que gran parte de la validez y seguridad del sistema depende del funcionamiento eficiente de esta capa. La correcta organización topológica garantiza que los datos se transmitan de manera óptima y segura entre los nodos de la red, minimizando los riesgos de pérdida de datos o accesos no autorizados.
- La tercera y última etapa es la capa de datos. Esta capa es una red virtual organizada mediante servidores y clientes socket, que se crean entre pares de nodos. En esta capa, el sistema distribuye los datos de manera ecuánime entre los dispositivos conectados a la red. Esta distribución se basa en algoritmos diseñados para optimizar la redundancia y la accesibilidad de los datos, asegurando que cada nodo pueda acceder a la información necesaria de manera eficiente. Además, la red virtual permite una escalabilidad flexible, adaptándose al crecimiento de la red y a la incorporación de nuevos nodos sin comprometer la integridad o la seguridad de los datos.

El proyecto también implementa mecanismos de encriptación y autenticación robustos para proteger la integridad de los datos transmitidos y almacenados. El uso de la estructura de datos CBOR permite una codificación eficiente y compacta de los datos, lo que es crucial para la transmisión rápida y el almacenamiento eficiente.

1.4- Problemas

Para la exitosa realización de este proyecto han sido necesarios ciertos cambios respecto del diseño original para poder mantener la línea de objetivos de la que dispone el sistema:

- Los sistemas Android cambiaron la normativa de seguridad respecto a cómo algunos servicios en segundo plano pueden recibir alertas internas del dispositivo (Android, 2024). Dichas alertas se transmitían por todo el dispositivo cuando éste se encendía, se capturaba una imagen, vídeo o audio, cuando cambiaba el estado de red, etc. Debido al cambio, fue necesario cambiar el diseño del SRCM por una aplicación, y no un servicio en segundo plano que pudiese detectar anuncios de otras aplicaciones.

  El objetivo era hacer el registro de un dispositivo, así como de sus contenidos multimedia creados a posteriori, independientemente de la aplicación que se utilizase para ello. Se ha conseguido mantener los objetivos de la solución, sin embargo, se abrió una brecha en la eficiencia del proyecto, pues es voluntaria la descarga y participación en el sistema. Por tanto, es voluntaria la legitimación de los contenidos extraídos de la realidad.

- Para que el mecanismo de consenso diseñado para este proyecto surtiese efecto, era necesario diseñar un protocolo de comunicación cliente-servidor en cada enlace P2P entre nodos de la red, para poder verificar los datos que intentaba insertar cualquier nodo en la misma. Esto fue un obstáculo cuanto menos problemático, puesto que debía hacerse una adaptación para que cada nodo que fuese servidor de distintos clientes realizase el mismo protocolo con cada cliente y luego se realizase un consenso teniendo en cuenta la decisión común de los nodos a los que les había sido transmitida la información por medio de dicho protocolo.
  El protocolo en sí no trata de un nuevo tipo de comunicación, sino de los pasos y "conversaciones" que se entablan entre nodos. Véase la sección "2.6- Protocolo de comunicación SRCM"

2- Diseño y Desarrollo completo del SRCM

En esta sección se explican de forma detallada cada uno de los componentes que forman el SRCM (Sistema de Registro de Contenido Multimedia). Cada uno de estos componentes desempeña un papel esencial en la funcionalidad y eficiencia del sistema, asegurando que los contenidos multimedia sean registrados y verificados de manera precisa y segura. Para ofrecer una visión general, a continuación, se presenta una lista de los componentes del SRCM:

Aplicación (Interfaz de Usuario)

La aplicación es el punto de acceso principal para los usuarios del sistema. A través de una interfaz de cámara intuitiva y fácil de usar, los usuarios pueden interactuar con el SRCM para hacer fotos, registrarlas, verificar otras y consultar el contenido multimedia. La aplicación permite a los usuarios capturar contenido directamente desde sus dispositivos, registrarlo en la red y verificar su autenticidad. La interfaz está diseñada para ser accesible a usuarios de todos los niveles técnicos, proporcionando una navegación simplificada. Además, la aplicación incluye características avanzadas de seguridad, como encriptación de datos, para proteger la privacidad y la integridad del contenido registrado.

Contenido Extraído (CE)

El Contenido Extraído (CE) es el término utilizado para referirse al contenido multimedia que ha sido capturado directamente de la realidad y registrado en el sistema. Una vez que el contenido ha sido capturado, se procesa para generar un hash único utilizando el algoritmo Keccak256. Este hash se utiliza como un identificador único para el contenido, asegurando que cualquier alteración en el contenido original pueda ser detectada fácilmente. El CE es esencial para el funcionamiento del SRCM, ya que proporciona la base o estructura de datos sobre la cual se verifica la autenticidad del contenido.

Mecanismo de Consenso PoZET

El Mecanismo de Consenso PoZET (Proof of Zero-Knowledge Elapsed Time) es una innovación clave del SRCM. Este mecanismo asegura que todos los nodos de la red estén de acuerdo sobre el estado actual de la cadena de bloques. A diferencia de los mecanismos de consenso tradicionales, PoZET se basa en la transferencia de claves por prueba de “Cero conocimiento”, lo que reduce significativamente el peligro ante ataques Man-In-The-Middle (Conti, Dragoni, & Lesyk, 2016). PoZET es crucial para mantener la integridad y la coherencia de la cadena de bloques, asegurando que todos los registros de contenido sean precisos y verificables.

Sistema Anti-Fork

El Sistema Anti-Fork es un componente esencial del SRCM que previene la creación de ramas divergentes en la cadena de bloques. Este sistema utiliza un Algoritmo Anti-Bifurcaciones en Redes Blockchain para asegurar que todas las transacciones y registros se mantengan en una única cadena continua y verificable.

El Sistema Anti-Fork es vital para mantener la unicidad y la coherencia de la cadena de bloques, previniendo problemas de duplicación de datos y garantizando que todas las verificaciones de contenido se realicen sobre una única fuente de verdad.

Protocolo de Comunicación Socket

El Protocolo de Comunicación Socket es la estructura u orden de mensajes transmitidos en la comunicación Cliente-Servidor Socket, el medio a través del cual los nodos de la red se comunican entre sí. Este protocolo permite la transmisión segura y eficiente de datos entre los nodos, asegurando que toda la información relevante se comparta de manera rápida y fiable.

El protocolo está diseñado para manejar múltiples conexiones simultáneas, proporcionando una infraestructura robusta para la red P2P del SRCM. Además, incluye mecanismos de encriptación para proteger la integridad y la privacidad de los datos transmitidos.

Red de nodos P2P

La Red de nodos P2P (Peer-to-Peer) es la estructura subyacente que soporta el SRCM. Esta red está compuesta por múltiples nodos distribuidos geográficamente, cada uno de los cuales desempeña un papel en la captura, registro y verificación del contenido multimedia.

La red P2P es altamente escalable y resistente a fallos, lo que la hace ideal para el SRCM. Cada nodo en la red tiene la capacidad de almacenar y procesar datos, lo que permite una distribución equitativa de la carga de trabajo y asegura que el sistema pueda manejar grandes volúmenes de contenido. La red P2P también facilita la redundancia de datos, asegurando que el contenido esté disponible incluso si uno o varios nodos fallan.

En la siguiente figura, [Fig. 2] “Relación entre componentes SRCM”, se muestra una síntesis de cómo los distintos componentes que emplea el SRCM interactúan entre sí, mostrando una básica funcionalidad de cada uno de ellos.

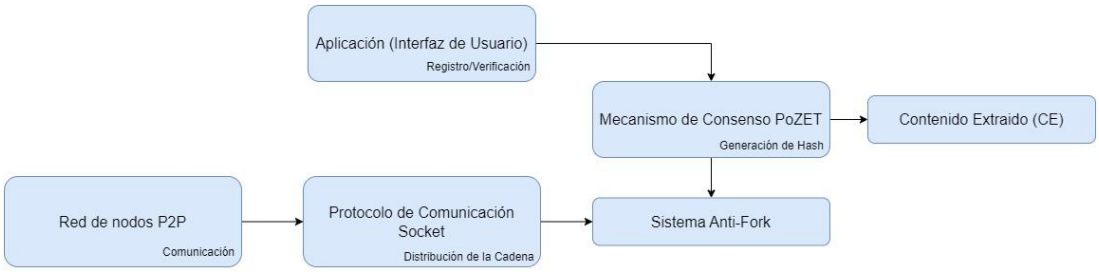

Fig. 2 Relación entre componentes SRCM

En resumen, cada uno de estos componentes trabaja en conjunto para proporcionar un sistema robusto y eficiente para la verificación de contenidos multimedia.

El SRCM no solo permite a los usuarios registrar y verificar contenido de manera segura, sino que también ofrece una solución escalable y resistente para la gestión de datos en una red distribuida.

2.1- Aplicación (Interfaz de Usuario)

El punto de acceso de los usuarios al SRCM (Sistema de Registro de Contenido Multimedia) es una aplicación de cámara normal y corriente, como si de cualquier otra se tratase, con una interfaz clara y minimalista para evitar cualquier tipo de complicación de usabilidad. Para el usuario no existe ninguna diferencia entre esta aplicación y cualquiera que ya haya utilizado con anterioridad, a excepción de un apartado en el que se puede aportar una imagen para verificar si fue registrada en el sistema en el momento de su creación o no.

El hecho de que no haya diferencia a simple vista aumenta la comodidad del usuario al utilizar el sistema, puesto que el cambio en su rutina de captura de imágenes no será prácticamente significativo.

Al iniciar la aplicación por primera vez, se requiere la atención del usuario para que tenga la posibilidad de aceptar los permisos requeridos por la aplicación, como en cualquier otra, además de leer y comprender los términos a los que se somete al utilizar el sistema.
Sin embargo, es importante reiterar en el hecho de que toda la información almacenada en la blockchain queda encriptada por medio de funciones Hash, las cuales sólo permiten la encriptación y nunca una desencriptación de la propia clave. Esto mantiene siempre la veracidad de los datos, así como el anonimato o seguridad sobre los mismos.
Sólo con el contenido original, se podría llegar a la misma clave para así compararlas.

2.2- Contenido Extraído (CE)

Como hemos comentado en la introducción, denominaremos **CE (Contenido Extraído)** a la estructura de datos diseñada para los contenidos multimedia creados por los dispositivos con periféricos (integrados o no) aptos para la extracción de datos de la realidad, sin importar el tipo de dato extraído, como vídeos, imágenes o audios.

El proceso de creación de un CE comienza en el momento en el que el dispositivo o nodo "U" enciende la aplicación de cámara "SrCam". En este momento, se inicia un servicio en segundo plano, el SRCM, como si de un servicio de ubicación se tratase. El servicio realiza un barrido en el dispositivo detectando todos los periféricos de los que dispone dicho dispositivo.

Si el servicio fuese apagado o deshabilitado de alguna forma obviamente maliciosa, las aplicaciones de cámara y audio no funcionarán, solicitando arrancar el servicio SRCM como en cualquier otra aplicación que solicita iniciar ciertos servicios en segundo plano.
Otra opción es clasificar cualquier contenido multimedia creado por un dispositivo con el servicio apagado como "no verificado".

Cuando se detecta un evento de captura en la cámara o micrófono de "U", el servicio realiza una copia de los datos capturados, antes incluso de ser guardados en la propia memoria interna del dispositivo, y los traduce a la estructura de datos CE (Sec. 2.2.1) para guardarlos en un archivo en la memoria interna del dispositivo.

La estructura de datos CE se forma a partir de un simple proceso en el que todos los metadatos de un contenido multimedia, así como cada valor de las muestras (píxeles o muestras digitales de audio) son codificados en CBOR una estructura de datos binaria similar a JSON, sin embargo, es más ligera. Acto seguido, tomados por separado, se combinan con un Nonce de 2048 bits de longitud obtenido mediante el mecanismo de consenso PoZET (Sec. 2.3) y se realiza un hash de tipo Keccak256 con claves de salida con una longitud de 2048 bits, a cada una de las dos cadenas de datos.

2.2.1- Paquete de datos de un CE

Todos estos datos del contenido multimedia serán guardados en un paquete de datos [Fig. 3] "Estructura de un Bloque CE", una cadena de 1.545 bytes, que tendrá la siguiente estructura:

- 64 bits para el **TimeStamp** o Marca de Tiempo en la que el bloque fue creado. Su valor es recogido automáticamente del valor de tiempo actual del sistema en el dispositivo que crea el bloque. Este valor se refiere al número de milisegundo que han transcurrido desde el 1 de enero de 1970 hasta el momento en que es capturada la marca de tiempo.

- 2048 bits para la clave resultante del **Header o Cabecera del bloque Padre PH_b**. Se trata de la clave "Block Hash" del bloque inmediatamente anterior en la cadena.

- 2048 bits para la **clave digital de identificación IK_u** de la que dispondrá cada dispositivo creador de CEs. La clave será suministrada por el propio SRCM en el momento en el que el dispositivo es añadido a la red.

- 2048 bits para la clave resultante del **hash de los datos del bloque BH_b** al completo. Es la combinación de cada una de las claves encriptadas para cada dato del CE (o SRCM_BLOCK). Esto permite mantener cierto nivel de seguridad sobre un mismo bloque, sin depender de ningún otro bloque, para que no pueda ser alterado ninguno de sus datos.

- 2048 bits para la clave resultante de los **metadatos al completo MR_b**.

- 2048 bits para la clave resultante de los **datos en crudo extraídos RD_b**.

- 4 bits para el **tipo de dato DT_b** que es (Imagen [0001], Nodo [0100], CE corrupto [0000], etc.). Este concepto se analiza más detalladamente en el apartado "2.2.2- Tipos de bloques CE"

- 4 bits para el **estado V_b en el que se subió** el bloque: "verificado" (0001), "no verificado" (0000)

- 2048 bits para **la clave de un Nonce** (número pseudoaleatorio de un solo uso).

Estructura de un bloque CE								
TimeStamp	**ParentHeader PHb**	**IdentificationKey IKu**	**BlockHash BHb**	**MetadataResult MRb**	**RawData RDb**	**DataType DTb**	**VerificationState Vb**	**Nonce**
64 bits	2048 bits	2048 bits	2048 bits	2048 bits	2048 bits	4 bits	4 bits	2048 bits

Fig. 3 Estructura de un Bloque CE.

2.2.2- Tipos de bloque CE

Los bloques utilizados en la blockchain del SRCM tienen distintas funcionalidades dependiendo del "DataType" que contengan en sus datos:

Tipo de Dato 01: Este tipo de dato se refiere a las Imágenes. El funcionamiento del bloque es el mismo que el anteriormente explicado.

Cabe destacar que **MR_b** y **RD_b**, contienen los metadatos de la imagen (datos que se pueden leer en cualquier dispositivo accediendo a la información de la propia imagen) y los bytes de cada uno de los píxeles que la conforman, respectivamente.

Tipo de Dato 04: Se trata de un bloque que se refiere a un nodo. El bloque ha sido incrustado como un nuevo eslabón en la cadena debido a un evento de "Iniciación" donde un nuevo dispositivo ha iniciado por primera vez el SRCM. El sistema registra, por medio de un nodo ajeno, al nodo "iniciado" en la cadena, otorgándole una clave de identificación única.

En este caso, la **IK_u** se trata de la clave creada para el nuevo nodo. Los **MR_b** contiene la clave del dispositivo que ya estaba integrado en el SRCM y ha "iniciado" al nuevo nodo. Es el nodo responsable de la validación de dicho ingreso en el sistema.

El **RD_b** contiene el número de puerto asignado para la conexión cliente-servidor formada entre los dos dispositivos.

Tipo de Dato 00: Este tipo de bloque se trata de un bloque que ha sido introducido de alguna forma inconveniente. Si ha ocurrido cualquier anomalía con el bloque quedará marcado como 00. Aunque dicho bloque se tratase de una falsificación, son los demás nodos los que verifican la inmediatez de dicha información.

2.3- Proof of Zero-knowledge Elapsed Time

Una vez copiados los datos extraídos para el CE, antes de ser guardados en la memoria interna, el dispositivo o nodo “U” realiza una petición para generar un bloque nuevo en multicast a los N_u vecinos de la red más cercanos, enviándoles los datos para el CE creado.

Se define N_u vecinos como los vecinos que tiene guardados cada nodo en una tabla de conexiones P2P, siendo $N_u = N_{Total}/D_{Red}+1$.

En este momento comienza el mecanismo de consenso **Proof of Zero-knowledge and Elapsed Time (PoZET).**

Los vecinos que reciben la petición generan un número pseudoaleatorio que indica el tiempo de espera o *sleep* de cada nodo.

El nodo “V” que despierte primero solicitará una cadena de bytes arbitraria al nodo “U”, enviándole una cadena de 2048 bits pertenecientes a un Nonce creado por “V” para el nuevo bloque. El nodo “U” deberá aportar al nodo “V” los bytes que se le piden.

Es evidente que tanto el nodo “U” como el nodo “V”, deberán utilizar el Nonce para combinarlo con el Ce y realizar un Hash. De este Hash se utilizarán las cadenas de bytes para la comprobación.

El proceso se repite 5-10 ciclos cambiando los bytes que se solicitan en cada ciclo. Si todas las comprobaciones concuerdan, el CE es legítimo y será incluido en un bloque en la blockchain con estado $V_b = 0001$.

2.4- Sistema Anti-Fork

El sistema **Anti-Bifurcaciones** o **Anti-Fork,** [Fig. 4] “Diagrama de flujo del proceso del Sistema Anti-Fork”, trata de evitar la existencia de múltiples cadenas en la red del SRCM, pues la mejor forma de mantener la legitimidad de los bloques es contenerlos en una sola cadena que actúe como único libro de cuentas en la red, sin ninguna versión que difiera de la “troncal”.

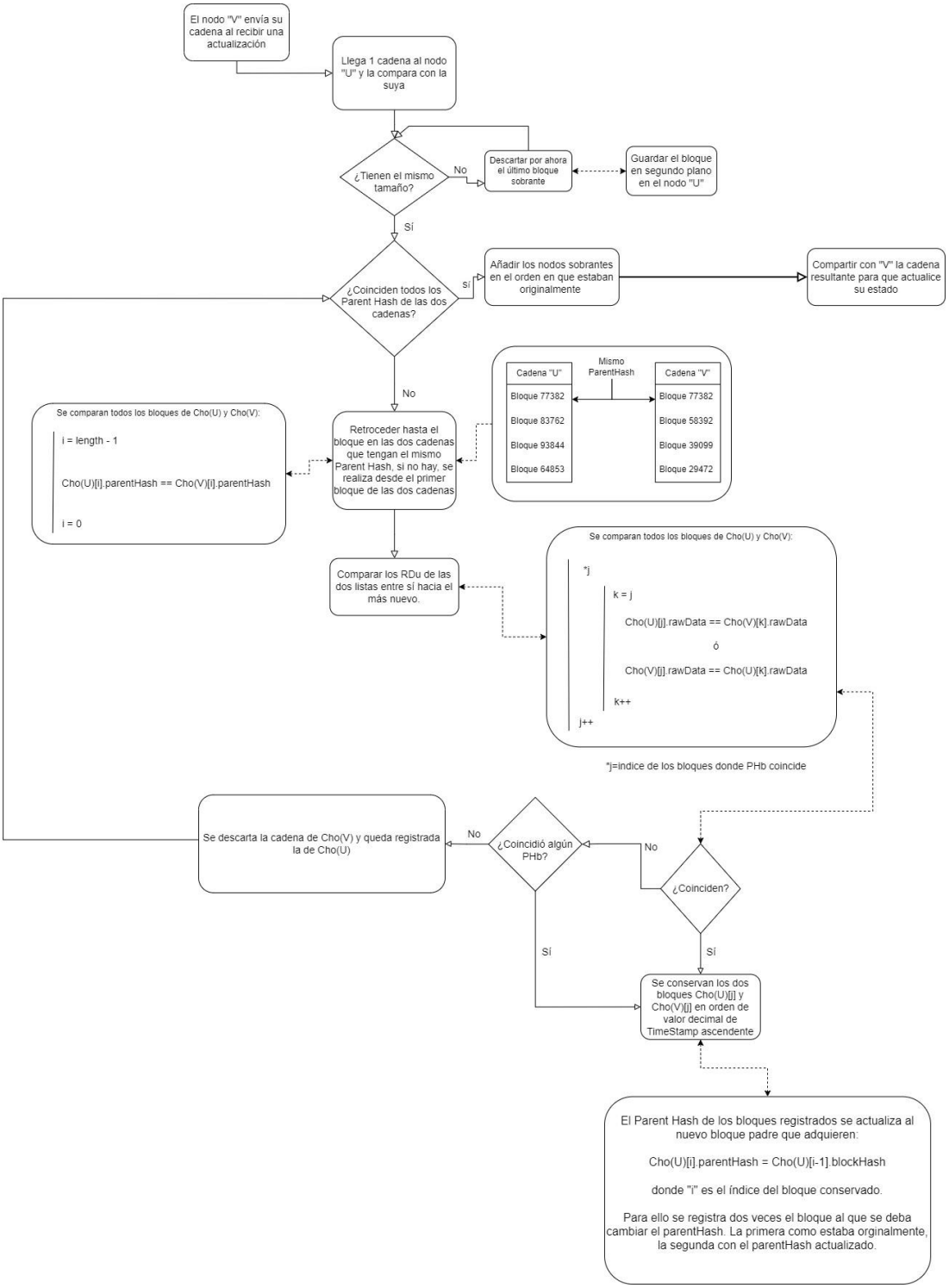

Fig. 4 Diagrama de flujo del proceso del Sistema Anti-Fork

2.5- Distribución y Actualización de la Cadena

Cada uno de los nodos estará conectado a N vecinos los cuales, a su vez, estarán conectados a sus respectivos N vecinos, siendo $N = N_{Total} / D_{Red}+1$.
Dichos vecinos son seleccionados según la proximidad en las claves IK_n de cada nodo respecto al valor de sus cadenas de 256 bytes. El nodo "U" se conectará a sus N_u vecinos que tengan IK_n más próximos.

Cuando un nodo reciba una actualización, como desconexión de la red, inclusión de un nuevo bloque, búsqueda de un bloque en la cadena, etc.; comunicará a la totalidad de sus vecinos dicha actualización, independientemente de si la procedencia es de otro nodo o de sí mismo.

Lo mostrado en la figura 5, [Fig. 5] "Estructura de Vecindad de la Red Overlay del SRCM", con la estructura de vecindad de la red, aplica para cada nodo "U" en la red, identificándose "U" como cada nodo por sí mismo. Así mismo, la siguiente figura, [Fig. 6] "Emparejamiento Cíclico en la Red Overlay del SRCM", muestra cómo existe emparejamiento cíclico en una red P2P de este tipo.

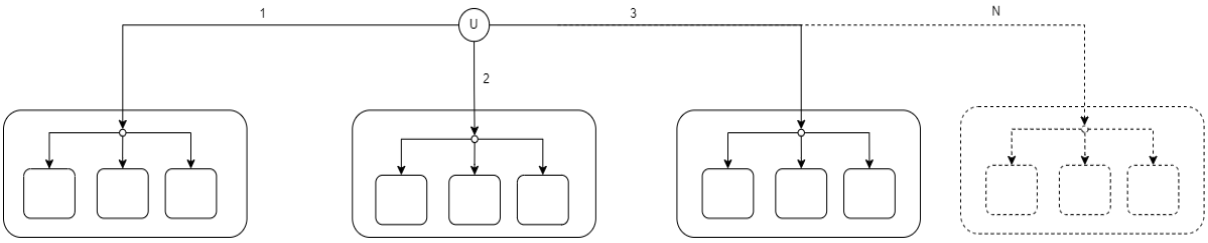

Fig. 5 Estructura de Vecindad de la Red Overlay del SRCM

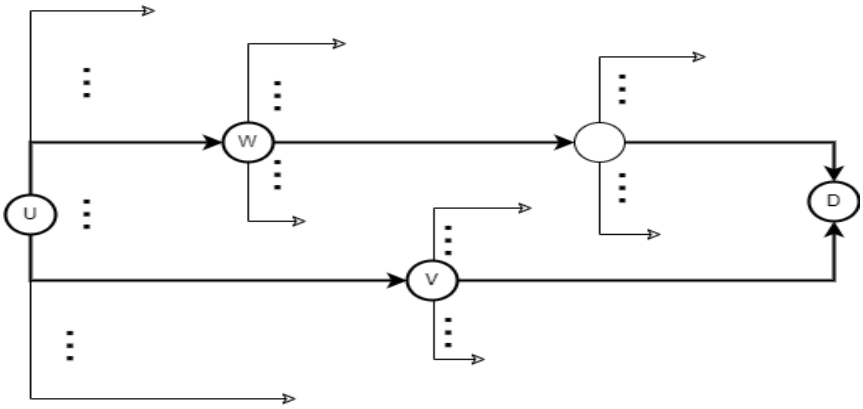

Fig. 6 Emparejamiento Cíclico en la Red Overlay del SRCM

Lo que hemos podido ver en dichas figuras es cómo un mismo nodo tiene conexiones con distintos nodos. Como en el sistema no existe ninguna regla de cómo un nodo puede conectarse a cuáles vecinos, es claro que se pueden encontrar distintos nodos “u” y “v” que conecten a un mismo nodo “d”, independientemente de la relación que guarden con dicho nodo.
Existe una propiedad en este sistema y es que, un nodo “u” puede ser el servidor de otro nodo “v” y al mismo tiempo puede existir dicha conexión a la inversa. Es importante recordar que cada nodo actúa, al mismo tiempo, como cliente y servidor.

Gracias al emparejamiento cíclico existen distintas rutas alternativas por las que un nodo puede llegar a su destino, aunque en dichas rutas alternativas existan mayor número de saltos que en la ruta principal.

La ruta principal por la que un nodo “U” se comunicará para llegar a otro “D”, será la que en base a las claves $IK_{v,\,w}$ de cada nodo “V” o “W” quede más próxima a la clave IK_d del nodo destino “D”.

Cada uno de los nodos conectados a la red registrará los mensajes que han sido transmitidos por sí mismos, para evitar transmitirlos de nuevo por la red de forma constante, solventando el problema de generación de bucles en la red.

2.6- Protocolo de comunicación Cliente-Servidor SRCM

El protocolo de comunicación Cliente-Servidor Socket en el SRCM consta de distintos eventos que pueden ocurrir a lo largo del transcurso del tiempo de vida del sistema.

El primer evento, [Fig. 7] “Comunicación Socket: Evento 1”, se centra en la apertura de comunicación entre los dos nodos o pares P2P recién enlazados en la capa IP. Para que este evento se dé entre dos nodos, es necesario que uno de ellos no estuviese conectado a la red previamente. A este nodo se le denominará nodo “iniciado”. El nodo iniciado es verificado por los nodos hijos (servidores socket a los que está conectado) del mismo.

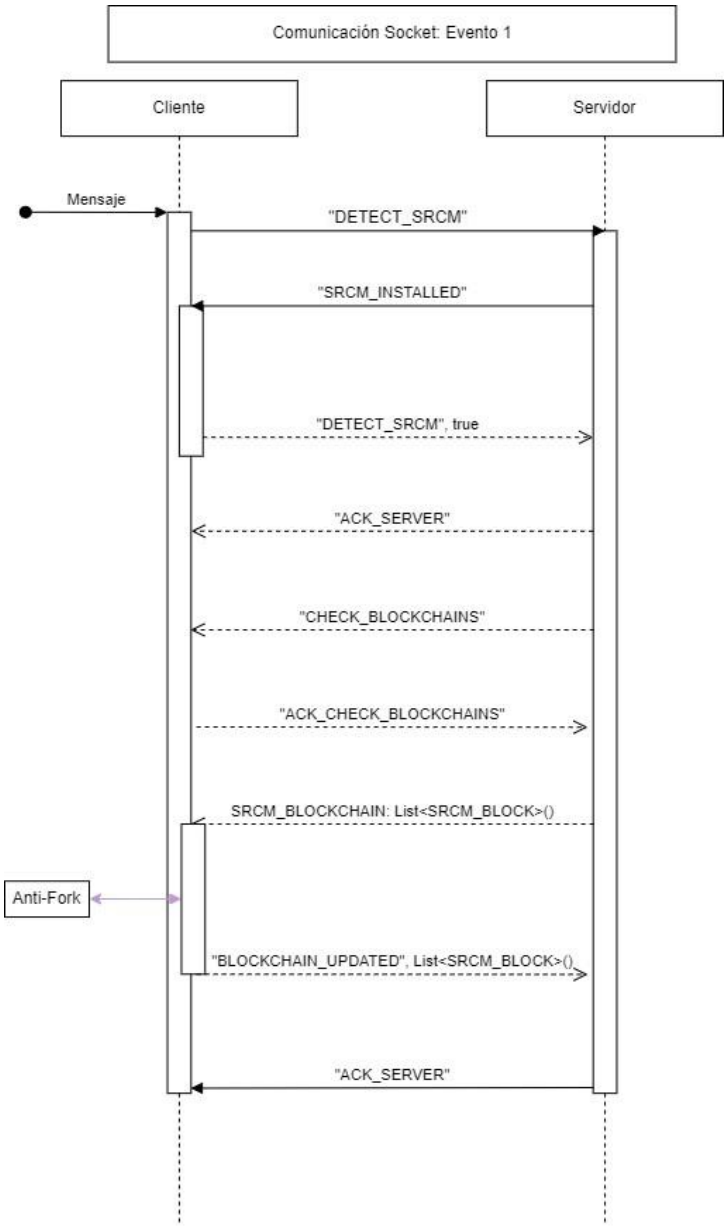

Fig. 7 Comunicación Socket: Evento 1

El segundo evento, [Fig. 8] “Comunicación Socket: Evento 2”, se centra en el momento en el que cualquier nodo ha comunicado a sus vecinos que se ha realizado una captura mediante la cámara del dispositivo. En este momento se realiza un intercambio de mensajes donde se verifica la información del nuevo bloque introducido en la cadena.

Es por esto por lo que se deducen dos fases en este evento:

La primera fase es cuando el nodo realiza la captura de contenido con la cámara y lo comunica con todos sus nodos vecinos y mediante el PoZET se crea un nuevo bloque y se incrusta como último “eslabón” en la cadena.

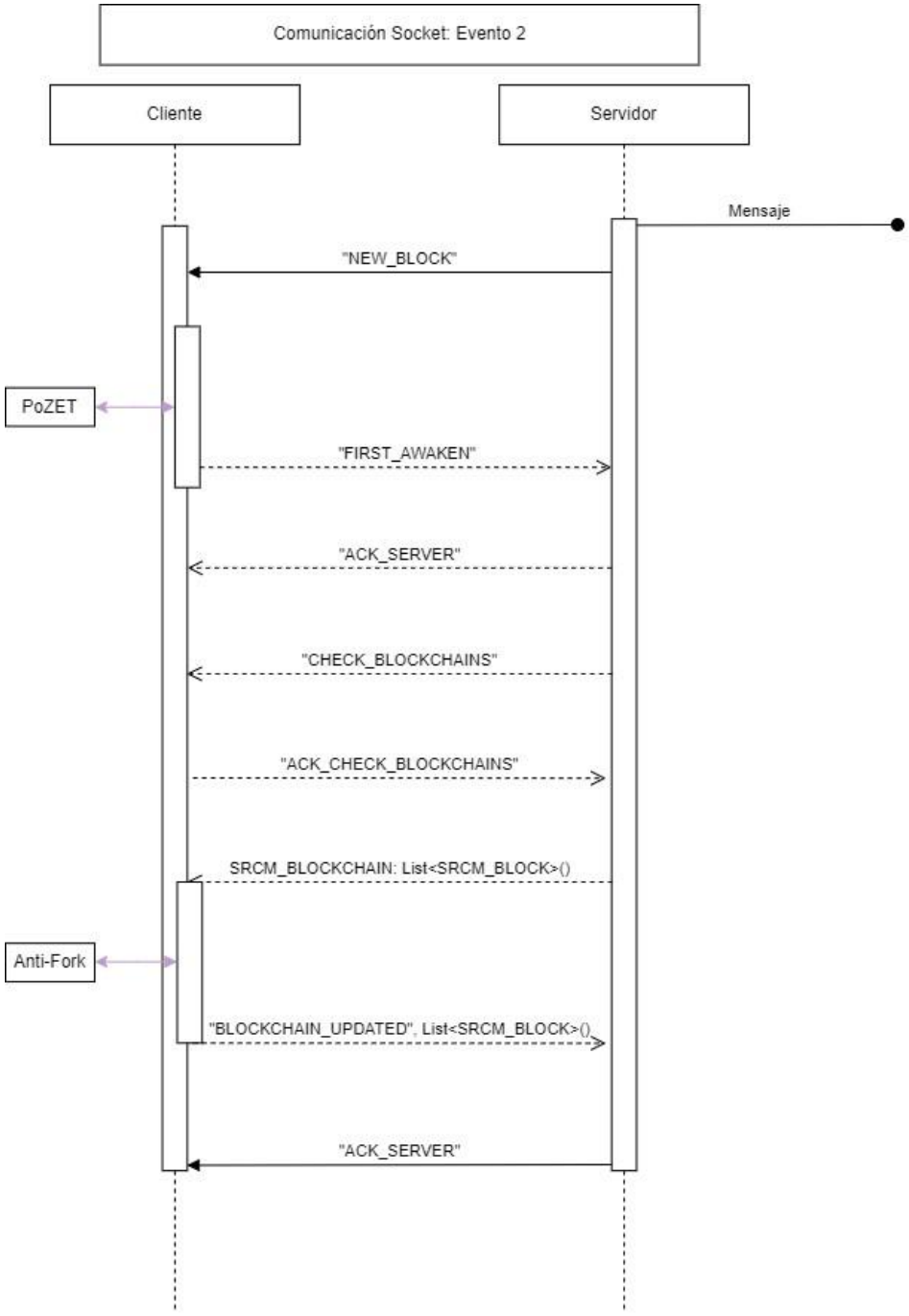

Fig. 8 Comunicación Socket: Evento 2

La segunda fase comienza una vez añadido el bloque a la cadena, momento en el cual, el nodo que ha incluido el bloque (el nodo verificador) lo comunica con todos sus vecinos para comprobar que se ha incluido de forma correcta según el protocolo de cualquier blockchain.

El punto más importante en esta fase es la verificación de que el nuevo bloque tiene una referencia al bloque inmediatamente anterior.

El proceso es muy simple, el servidor se comunica con todos sus clientes, sin importar el puerto al que estén conectados, y comunica su blockchain registrada. Automáticamente los clientes ejecutan el mecanismo anti-fork.

El tercer evento, [Fig. 9] "Comunicación Socket: Evento 3", se centra en el momento en el que cualquier nodo comunica a sus vecinos que va a desconectar de la red. Un nodo se mantendrá conectado a la red mientras esté con la aplicación de cámara del SRCM, SrCam, abierta.

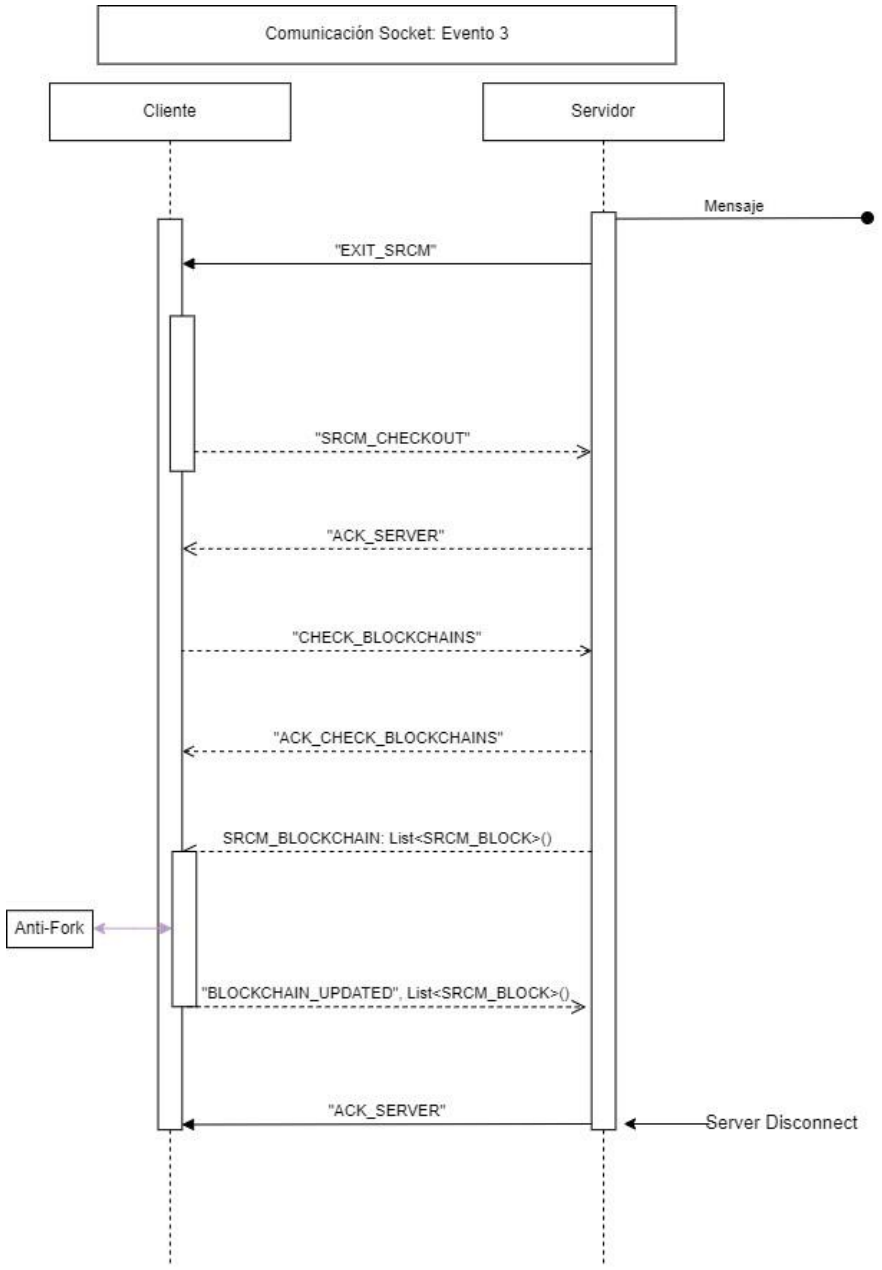

Fig. 9 Comunicación Socket: Evento 3

Al suspender la aplicación el dispositivo realiza una última función y es esta comunicación a todos sus vecinos de que su estado de conexión en la red va a cambiar a “desconectado”.

Realmente no existe tal estado de forma activa, sencillamente el dispositivo dejará de estar disponible para el resto de los nodos, hasta que se vuelva a abrir la aplicación, donde se ejecutaría el primer evento del protocolo.

Cabe destacar que la topología de la red overlay (la red lógica formada a partir de los clientes y servidores socket entre pares) está invertida a la topología de la capa IP en la red P2P.

Mientras que, en esta capa, cada nodo tiene varios nodos hijos conectados, en la capa de cliente-servidor socket cada nodo tiene varios nodos “padres”. Los servidor actuaría como nodo padre lógicos de cada uno de los nodos que estén conectados a dicho servidor. Para que distintos nodos se puedan conectar a un mismo dispositivo, realizando así una conexión cliente-servidor, es necesario que los servidores adopten o escuchen por distintos puertos para abarcar a todos sus clientes.

La asignación de puertos a los servidores se realiza de forma procedural, así como la asignación de puertos a los clientes. Esto ocurre durante el primer evento en el protocolo de comunicación del Sistema de Registro de Contenido Multimedia (SRCM).

2.7- Primer Prototipado y Test del SRCM

El primer prototipo del SRCM será nombrado como SRCM E-0.1 y se basará en una pequeña red P2P formada por un reducido grupo de dispositivos móviles (smartphones y tabletas) con Android como sistema operativo.

Para el desarrollo de las aplicaciones y servicios destinados a los dispositivos Android se hará uso del IDE “Android Studio” utilizando el lenguaje de programación JAVA. La primera red constará de tan solo 2 smartphones Android y una tableta también Android.

El primer prototipo consta de 4 objetivos clave:

- El primer objetivo es crear una cadena con un bloque “Génesis” que será sobre el cual comience la pila de bloques de la red. Este bloque inicial es fundamental ya que establece la base sobre la cual se añadirán todos los bloques subsecuentes, asegurando la integridad y la coherencia de la cadena de bloques desde su inicio.

- El segundo objetivo es conectar los 3 dispositivos en una red P2P manteniendo una red Overlay entre ellos organizada por el SRCM. La creación de esta red Overlay es esencial para asegurar que todos los dispositivos puedan comunicarse de manera eficiente y segura, estableciendo un canal de transmisión de datos robusto.
- El tercer objetivo es agregar en total 10 bloques nuevos a la blockchain desde cualquiera de los 3 dispositivos que la integran. Este paso es crucial para probar la capacidad del sistema de aceptar y validar nuevos bloques de datos, demostrando así la funcionalidad de adición de datos en un entorno distribuido.
- El cuarto y último objetivo es distribuir los datos de la Blockchain entre los 3 dispositivos. Esta distribución no solo asegura la redundancia y disponibilidad de los datos, sino que también prueba la capacidad del sistema para mantener la sincronización de la blockchain entre múltiples nodos.

Una vez finalizado el primer prototipo, se incluirán nuevos nodos en la red, manteniendo la cadena formada en el prototipo “SRCM E-0.1”. La incorporación de nuevos nodos permitirá probar la escalabilidad del sistema y su capacidad para integrar más dispositivos sin comprometer la seguridad y la eficiencia de la red.

Además, se evaluarán las capacidades de gestión de la topología de red del SRCM, asegurando que la red Overlay se ajuste automáticamente para optimizar la conectividad y el rendimiento a medida que se añaden más dispositivos.

Esto también permitirá observar cómo el sistema maneja la distribución de datos en una red más amplia y si la implementación del algoritmo anti-bifurcaciones es efectiva en escenarios más complejos. En esta fase, se podrán identificar y abordar posibles limitaciones o desafíos que no se presentaron en la red más pequeña del prototipo inicial.

3- Resultados

Al iniciar la aplicación por primera vez con un dispositivo (al que nombraremos nodo “U”) nunca incluido en el sistema, se han solicitado los permisos de acceso a geolocalización precisa, wifi, bluetooth, cámara y micrófono. Es importante la solicitud de estos permisos para el cumplimiento del consentimiento del usuario.

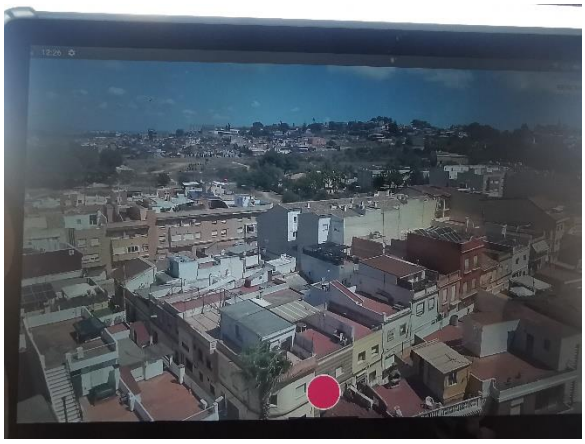

Fig. 10 Interfaz Cámara SRCM

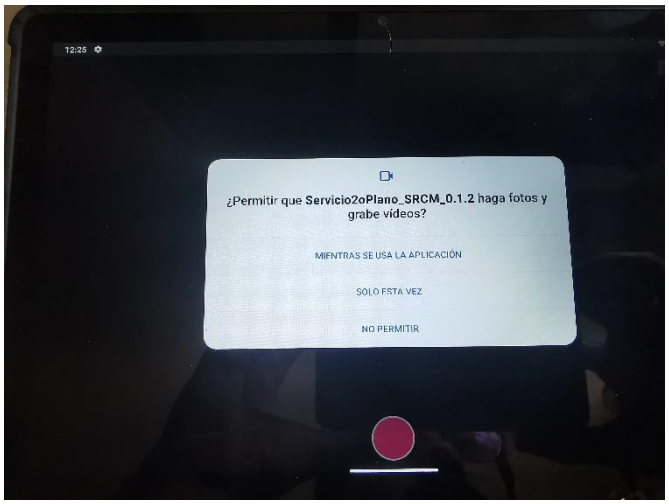

Fig. 11Solicitud Permisos Cámara

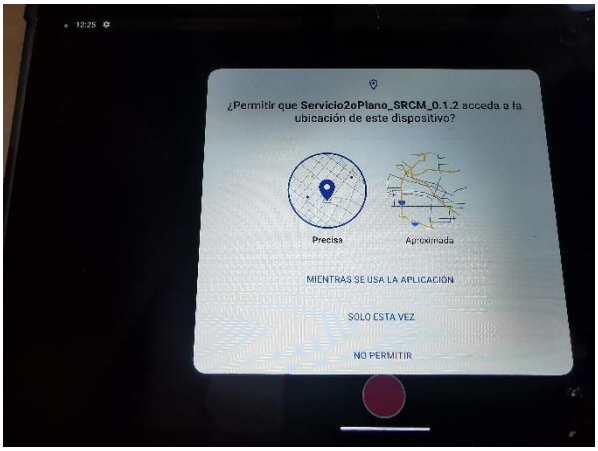

Fig. 12 Solicitud Permisos Ubicación

Una vez aceptados los permisos, el SRCM automáticamente ha conectado este nuevo nodo con otro de los nodos ya incluidos en el sistema (al que nombraremos nodo “V”), el cual ha introducido a nuestro nuevo nodo en la cadena mediante un bloque de tipo 04.

Fig. 13 Nuevo Nodo Incluido en Blockchain

En cuanto ha acabado el proceso de registro del nodo “U” en la cadena, en tan solo unas pocas décimas de segundo, el nodo “V” ha transmitido y actualizado la cadena de bloques con el nuevo nodo introducido en el sistema. Esta actualización se ha llevado a cabo mediante el sistema Anti-Bifurcaciones anteriormente explicado.

Una vez actualizada la cadena, somos libres de realizar fotos a nuestro antojo, siendo todas ellas registradas mediante el mecanismo de consenso PoZET. No siempre nos acepta el bloque el mismo nodo. En este caso, el bloque no ha sido aceptado por el nodo “V”, sino por un tercer nodo “W”.

Para comprobar si las imágenes que hemos realizado estaban en la cadena de bloques simplemente ha sido necesario presionar el botón de verificación y seleccionar la imagen que queríamos comprobar. Automáticamente el sistema nos ha informado de si la imagen ha sido introducida o no en algún momento dentro de la propia blockchain.

Fig. 14 Botón Verificar Interfaz SRCM

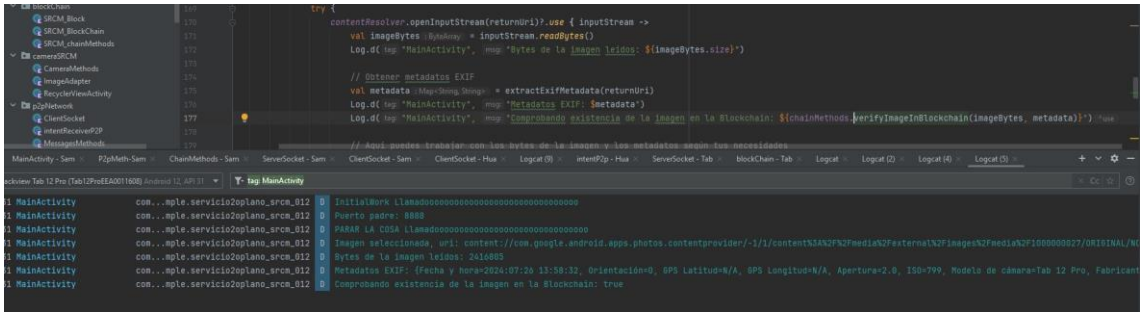

Fig. 15 Verificación Imagen en SRCM

El sistema ha sido creado para que sea lo más intuitivo y minimalista posible para los usuarios. Es necesario que el cambio respecto de una cámara corriente sea el mínimo posible, para que los usuarios se adapten fácilmente.

4- Conclusiones

En conclusión, este proyecto ha demostrado de manera efectiva cómo un sistema de registro de contenido multimedia (SRCM) basado en blockchain puede contribuir significativamente a la verificación de la autenticidad del contenido multimedia en un entorno digital cada vez más susceptible a la manipulación. Al implementar tecnologías avanzadas como la estructura de datos CBOR, el hash Keccak256, el mecanismo de consenso PoZET y un algoritmo anti-bifurcaciones, se ha logrado un sistema robusto y seguro que cumple con los objetivos iniciales del proyecto.

Los resultados obtenidos han mostrado que el SRCM es capaz de registrar y verificar contenido multimedia de manera eficaz. Las pruebas realizadas indicaron que el sistema puede detectar alteraciones en los datos registrados, confirmando su capacidad para mantener la integridad del contenido. Además, la distribución estratégica de los datos entre los nodos de la red garantiza que el sistema sea resistente a fallos y manipulaciones.

Estos resultados están en línea con los objetivos del proyecto, que buscaban crear una solución capaz de autenticar contenido multimedia sin la intervención de Inteligencias Artificiales. La capacidad del sistema para verificar contenido registrado sin alteraciones proporciona una base sólida para su aplicación en escenarios reales.

A pesar del éxito alcanzado, es totalmente necesario evaluar tanto las fortalezas como las debilidades del método seguido y de los resultados obtenidos. Una de las principales fortalezas del SRCM es su capacidad para distribuir datos de manera eficiente y segura entre los nodos de la red. Esta característica no solo mejora la seguridad del sistema, sino que también asegura su escalabilidad.

Sin embargo, una de las debilidades observadas es la dependencia del sistema en una red de nodos adecuadamente configurada y geolocalizada. Cualquier discrepancia en la topología de la red puede afectar la eficiencia del sistema. Además, la implementación del mecanismo de consenso PoZET, aunque innovadora, puede requerir ajustes y optimizaciones adicionales para funcionar de manera óptima en diferentes escenarios.

Durante la realización de este proyecto, se presentaron varias limitaciones que podrían ser abordadas en futuras investigaciones. Por ejemplo, la capacidad de procesamiento y almacenamiento de los dispositivos individuales podría limitar la eficiencia del sistema en redes de gran escala. Asimismo, la latencia de la red puede afectar la rapidez con la que se distribuyen y verifican los datos.

Una posible vía de investigación futura podría enfocarse en la optimización del algoritmo de consenso PoZET para mejorar su rendimiento en redes de nodos más grandes y diversificadas. Además, explorar la integración de tecnologías

emergentes como la computación cuántica podría ofrecer soluciones innovadoras para aumentar la capacidad de procesamiento y mejorar la seguridad del sistema.

En conclusión, mientras que el SRCM ha demostrado ser un sistema viable y eficaz para la verificación de contenido multimedia, su implementación práctica revelará áreas de mejora y proporcionará oportunidades para la innovación continua. La evolución constante de las tecnologías digitales y las amenazas asociadas demandarán una adaptación y perfeccionamiento continuo del sistema, asegurando su relevancia y efectividad en un entorno digital en constante cambio.

En un mundo donde la autenticidad de la información es cada vez más crucial, el SRCM representa un paso adelante hacia un entorno digital más seguro y confiable. Con su capacidad para manejar grandes volúmenes de datos y su resistencia a fallos, el SRCM está bien posicionado para convertirse en un estándar para la verificación de contenido multimedia en el futuro.

Opinión Sincera del Alumno

Personalmente me fascinó la idea de este proyecto. Me di cuenta de la necesidad que nos imperaba a todas las personas. Necesidad que, pese a su escasa visibilidad en la vida cotidiana de la gran mayoría de usuarios, nos acabará afectando a todos los niveles. Me resultó difícil aceptar mi instinto de temor cuando vislumbré un futuro donde nuestros sucesores generacionales no pudiesen discernir la realidad del contenido audiovisual del que estarían aprendiendo en las redes (puesto que todo lo que conoce, observa o escucha cualquier ser humano, lo guarda y aprende de forma involuntaria, sobre todo si dicha información o estimulo se repite constantemente).

Recuerdo cómo, en el presente, mi madre se acercó a mí para preguntarme si una imagen que estaba viendo en Instagram era real o estaba creada por una Inteligencia Artificial. Por suerte, hoy en día, aún tenemos capacidad humana para discernir. Sin embargo, pude comprobar como actualmente, ya no es absoluta e inequívoca dicha capacidad de distinción.

Al diseñar y crear este sistema, tenía en mente una sociedad en la que pudiese comprobar, sin necesidad de fiarme de un intermediario, si lo que estaba viendo por las redes era verdad. Una sociedad en la que, si tuviese que ir a un juicio, pudiera presentar imágenes como pruebas, sin dejar lugar a dudas acerca de la veracidad de dichas pruebas, puesto que ya habrían sido registradas en el SRCM.

Este sistema no sólo nos protege contra las Inteligencias Artificiales. Nos protege contra nosotros mismos. Nos da el derecho a la libre fundamentación de los hechos. A la objetivación de la realidad, sin intermediarios centralizados.

Referencias

Android. (25 de Abril de 2024). *Restricciones para la recepción de emisiones de imagen y video*. Obtenido de Android Developers: https://developer.android.com/develop/background-work/background-tasks/bg-work-restrictions?hl=es-419

Bertoni, G., Daemen, J., Peeters, M., & Van Assche, G. (2014). The making of Keccak. *Cryptologia Vol. 38 - Issue 1*, 26 - 60. Obtenido de Keccak Team: https://keccak.team/files/MakingOfKeccak.pdf

Blasco, P. (27 de Mayo de 2018). *GitHub - pepebndc*. Obtenido de GitHub: https://github.com/pepebndc/Blockchain-JAVA

Bormann, C., & Hoffman, P. (Diciembre de 2020). *RFC8949*. Obtenido de RFC Editor: https://www.rfc-editor.org/info/rfc8949

Conti, M., Dragoni, N., & Lesyk, V. (2016). A Survey of Man In The Middle Attacks. *EEE Communications Surveys & Tutorials, 18*(3), 2027-2051. Obtenido de https://ieeexplore.ieee.org/abstract/document/7442758

Crypto.com. (13 de Mayo de 2022). *Crypto.com*. Obtenido de Crypto.com: https://crypto.com/university/consensus-mechanisms-explained

Dunne, M. (1 de Noviembre de 2023). Obtenido de Ipsos: https://www.ipsos.com/en-us/data-dive-fake-news-age-ai

IBM. (20 de Mayo de 2024). *IBM: Blockchain*. Obtenido de IBM: https://www.ibm.com/es-es/topics/blockchain

Kalita, L. (2014). Socket programming. *International Journal of Computer Science and Information Technologies, 5*(3), 4802-4807.

Korzum, D., & Gurtov, A. (2012). *Structured peer-to-peer systems: fundamentals of hierarchical organization, routing, scaling, and security.* New York: Springer Science & Business Media. Obtenido de https://books.google.es/books?hl=es&lr=&id=VC4Rg82lxbEC&oi=fnd&pg=PR7&dq=P2P+Networks+-+KORZUN,+Dmitry%3B+GURTOV,+Andrei.+Structured+peer-to-peer+systems:+fundamentals+of+hierarchical+organization,+routing,+scaling,+and+security.+Springer+Science+%26+Busi

Kotlin Foundation. (16 de Abril de 2024). *Basic Syntax*. Obtenido de Kotlin Programming Language: https://kotlinlang.org/docs/basic-syntax.html

Maurer, W. D., & Lewis, T. G. (1975). Hash table methods. *ACM Computing Surveys (CSUR)*, 5-19.

OpenAI. (14 de Marzo de 2023). *ChatGPT*, GPT-4o y GPT 3.5. Recuperado el 1 de Mayo de 2024, de ChatGPT: https://chatgpt.com/

Sadin, É. (2019). La inteligencia artificial: el superyó del siglo xxi. *Nueva Sociedad*, 141-148.

Segura, J. (17 de Abril de 2023). *Bit2Me: ¿Qué es Nonce?* Obtenido de Bit2Me: https://academy.bit2me.com/que-es-nonce/

Sumsub. (8 de Noviembre de 2023). Obtenido de Sumsub.com: https://sumsub.com/newsroom/sumsub-research-global-deepfake-incidents-surge-tenfold-from-2022-to-2023/

Verma, P. (17 de Diciembre de 2023). *AI Fake News Misinformation*. Obtenido de The Washintong Post: https://www.washingtonpost.com/technology/2023/12/17/ai-fake-news-misinformation/

Wackerow, P. (Ed.). (29 de Marzo de 2024). *Ethereum.org*. Obtenido de Ethereum.org: https://ethereum.org/en/developers/docs/consensus-mechanisms/

Wood, G. (2014). *Ethereum: A secure decentralised generalised transaction ledger*. Ethereum project yellow paper. Obtenido de https://cryptodeep.ru/doc/paper.pdf

ANEXO I

Glosario

Bifurcaciones: En el contexto de las cadenas de bloques, o blockchain, una bifurcación es un evento en el cual los datos de la propia cadena divergieron en dos ramas distintas. Esto podría deberse a que dos nodos han introducido un bloque con la misma "cabecera del padre", o incluso que se ha introducido un bloque que no tiene ningún sentido en cuanto al orden de los datos, lo cual implicaría un plausible ataque en contra de la integridad de los datos. Existen distintas posibilidades como abandonar una de las ramas. En este proyecto la intención es fusionar las dos ramas, dejando constancia de los errores o cambios, si los hubiere.

Blockchain: Una Blockchain o Cadena de Bloques, es una especie de libro de cuentas, donde cada bloque guarda información única, e inmutable. Esta información está cifrada y depende del último bloque (el último movimiento apuntado en el libro de cuentas). Además, el libro de cuentas, la blockchain, se distribuye por todos los nodos que forman la red, creando así copias de seguridad de esta, haciéndola difícil de alterar de forma maliciosa (IBM, 2024).

Cabecera del Bloque Padre: La cabecera en un paquete de información, se refiere a los datos que preceden al cuerpo del mensaje. Dichos datos sirven para redirigir el mensaje, localizarlo, confirmar que los datos llegan correctamente, etc. En el contexto de este proyecto, cuando hablamos de la cabecera del bloque padre nos referimos a la clave resultante de la combinación de las funciones Hash que se realizan sobre cada parámetro del CE (o SRCM_BLOCK). A esta clave se le denomina "Block Hash" debido a que se trata de la clave encriptada que contiene todos los datos del bloque. Para un bloque nuevo, su "Parent Header" será el "Block Hash" del bloque inmediatamente anterior.

Capa IP: En términos de telecomunicaciones, la capa IP es una forma de referirse al tipo de organización y funciones que se llevan a cabo referente a las IP de las que dispone cada dispositivo conectable a cualquier tipo de red. En este caso, la capa IP identificaría los nodos entre sí, sin embargo, para organizar la información y nodos, es posible utilizar una capa superior como la de una red P2P.

CBOR: La representación concisa de objetos binarios (CBOR) es un formato de datos cuyos objetivos de diseño incluyen la posibilidad de un tamaño de código extremadamente pequeño, un tamaño de mensaje bastante pequeño y extensibilidad sin necesidad de negociación de versiones (Bormann & Hoffman, 2020).

Construcción de esponja: En el contexto de la criptografía, la construcción de esponja es un modo de operación, basado en una permutación (o transformación) de longitud fija y en una regla de relleno, que construye una función que asigna una entrada de longitud variable a una salida de longitud variable. Toma un elemento como entrada de longitud entera (no decimal) par, y devuelve un elemento de salida con longitud elegida también entera y par.

Hash: En ciencia de computación el término Hash (del inglés picadillo, plato resultante de la mezcla de ciertos ingredientes) se refiere a una función que toma una cadena de texto como entrada y la comprime en otra cadena de texto como salida. Esta función satisface ciertos criterios de seguridad, por lo que se utilizan para autenticidad, claves digitales, esteganografía (ocultar información dentro de otro mensaje para evitar su detección), generación de números pseudoaleatorios, etc. (Maurer & Lewis, 1975).

Keccak 256/512: Keccak es una función criptográfica versátil. Más conocida como función hash, también se puede utilizar para autenticación, cifrado (autenticado) y generación de números pseudoaleatorios. Su estructura es una construcción de esponja extremadamente simple e internamente utiliza la innovadora permutación criptográfica Keccak-f, donde f es el número de bits que la función toma como bloques para cada ciclo o permutación (Bertoni, Daemen, Peeters, & Van Assche, 2014). El valor 256 o 512 se refiere a la longitud de la clave de salida resultante, en bits.

Mecanismo de Consenso: El mecanismo de consenso para una blockchain es un procedimiento en el que los pares (o nodos) de una red blockchain llegan a un acuerdo sobre el estado actual de los datos en la red (Crypto.com, 2022). Este acuerdo suele llevarse a cabo mediante la recompensa o castigo a dichos nodos por medio de criptomonedas. A través de esto, los algoritmos de consenso establecen fiabilidad y confianza en la red blockchain.

M.C. de Tiempo Transcurrido: El PoET (Proof of Elapsed Time), es un mecanismo de consenso por prueba de tiempo transcurrido como su nombre indica. Esto quiere decir que la prueba de que los nodos sean imparciales en términos de verificar datos de la blockchain trata de la aleatoriedad del tiempo transcurrido. Cuando un nodo quiere subir un bloque a la red, se envía un mensaje con su intención a todos los nodos a los que está conectado y éstos se marcan un tiempo de espera pseudoaleatorio. El primer nodo que responda, una vez transcurrido su tiempo pseudoaleatorio, será encargado de la verificación de los datos a incrustar en la blockchain. En este método, se sigue con la mecánica de recompensar o castigar a dichos nodos mediante el uso de criptomonedas.

Nodo: En redes y telecomunicaciones, nodos son cada uno de los dispositivos que están conectados a la red y que pueden enviar o recibir datos. Algunos tipos de nodos son ordenadores, impresoras, dispositivos móviles como smartphones, un router, etc. Si se trata de una red P2P, a estos nodos también se les denomina "pares".

Nonce: Un nonce es un número arbitrario que se puede usar una única vez en una comunicación criptográfica. Suele tratarse de un número aleatorio o pseudoaleatorio emitido en un protocolo de autenticación para garantizar que las comunicaciones antiguas no se puedan reutilizar en ataques de playback. También pueden ser útiles como vectores de inicialización y en funciones hash criptográficas (Segura, 2023).

Prueba de Cero Conocimiento: Basada en una de las historias de las 1001 noches: Alibaba y los 40 ladrones. La prueba consiste en verificar que un objetivo dispone de cierta información, sin revelar cual es dicha información. En la historia se narra como existe una cueva con forma de "O" con una sola entrada que bifurca en dos caminos: "A" y "B". Para pasar por el arco interior de la cueva, es necesario abrir una puerta con una contraseña mágica. Para demostrar que el objetivo dispone de la contraseña sin revelarla, se le hace pasar un numero arbitrario de veces la siguiente prueba:
Se entra por uno de los caminos al azar, pero una vez entrado, se debe salir por el camino elegido por el examinador.

La probabilidad de que ocurra por pura casualidad, sin disponer de la clave que abre la puerta al final de la cueva, va disminuyendo de forma exponencial a medida que se realizan ciclos de esta prueba.

En este caso, el nodo examinador solicita un numero arbitrario de bits al nodo objetivo, durante un numero arbitrario de ciclos. Si el nodo objetivo responde bien, es que la información de la que dispone concuerda con la del nodo examinador.

Red P2P: “Una red P2P es un sistema distribuido por naturaleza, sin un control centralizado.” Como pudiera ser el caso de un servidor de una empresa, que centraliza los datos y funciones que se llevan a cabo. “Nodos con simetría en los roles de cliente y servidor forman una red superficial que se autoorganiza, por encima de la red de capa IP” (Korzum & Gurtov, 2012).

Servidor/Cliente Socket: “Este término, Socket (del inglés "enchufe") proviene de una metáfora de la toma de electricidad/teléfono en la que los enchufes actúan como interfaces que se conectan entre sí a través de una red” (Kalita, 2014). Esto se refiere a que un servidor socket actúa de la misma manera, activando distintos “enchufes” hacia el mismo destino, el servidor, a los que se puede conectar un cliente. Para cada enchufe o socket se asigna un número de puerto, el cual difiere del resto de posibles números de puerto para dicho servidor. Es por esto que, si un servidor abre y procede a escuchar por distintos sockets, cada uno con su puerto, tiene capacidad para aceptar y comunicarse con distintos clientes, manteniendo las mismas funciones y conjuntos de datos.