Máster universitario en Seguridad de las Tecnologías de la Información y de las Comunicaciones (MISTIC)

TFM - Redes de anonimización y cibermercados negros

Alumno: Xavier Salvadó Martí
Directora: Angela María García Valdés
Universitat Oberta de Catalunya (UOC)
Fecha de realización: Diciembre 2017

Resumen

TOR es una red de comunicaciones distribuida de baja latencia y superpuesta sobre Internet, en la que el encaminamiento de los mensajes intercambiados entre los usuarios no revela su identidad y permite mantener la integridad y la confidencialidad de la información que viaja por ella.

Bitcoin es una criptomoneda diseñada para funcionar como un medio de intercambio que utiliza la criptografía para controlar su creación y gestión en lugar de confiar en autoridades centrales.

El presente Trabajo Final de Máster engloba un estudio sobre los principios de la red TOR y Bitcoin, una investigación del uso de estas redes en los cibermercados negros y una investigación sobre los métodos de desanonimización de usuarios de la red TOR.

Abstract

TOR is a low-latency distributed communications network and overlay over Internet, in which the routing of the exchanged messages between users does not reveal their identity and it allows to maintain the integrity and the confidentiality of the information that travels in the network.

Bitcoin is a cryptocurrency designed to work as a medium of exchange which uses the cryptography in order to control its creation and management instead of trust in central authorities.

This Final Master's Project includes a study about the principles of TOR network and Bitcoin, an investigation of the use of these networks in the black cybermarkets and a research about users deanonymization methods of the TOR network.

Índice de contenido

Índice de Figuras

Índice de Tablas

1. Introducción

Cada uno de los mensajes que se envían por Internet tiene información de enrutamiento que puede ser usada para identificar el remitente y el destinatario del mensaje. Para muchos usuarios de Internet como por ejemplo activistas, denunciantes anónimos o defensores de derechos humanos, esto supone un problema porque pueden querer ser anónimos para evadir la posible represión de gobiernos o corporaciones. Hasta hoy en día, muchos sistemas de anonimato han sido desarrollados con el objetivo de facilitar la comunicación anónima online.

Una red muy popular de anonimato de baja latencia es TOR que funciona enrutando la conexión de un usuario a través de 3 routers onion (*onion routers*, ORs), los cuales forman un circuito y actúan como una cadena de proxies. Los mensajes enviados a través de la conexión se cifran por capas (usando una técnica llamada *onion encryption*) donde cada OR conoce solo su origen y destino inmediato. Los routers onion son puestos por voluntarios alrededor del mundo y son coordinados y catalogados por un pequeño conjunto de servidores de directorios que proveen información acerca de la red TOR y de los routers disponibles para los clientes (quienes son llamados por lo general *onion proxies* o OPs).

Esta red también permite mantener el anonimato de los servicios ubicados dentro de ella. A estos servicios se les llama Servicios Ocultos (*Hidden Services*). Mediante un servicio oculto, es posible ofrecer un servidor en la red TOR sin una IP que lo identifique. Por lo tanto, debido a las posibilidades de anonimato que ofrece, en ella existen diversos mercados negros para comprar cualquier tipo de bien o servicio ilegal.

El éxito de los cibermercados negros se debe al uso de las criptomonedas, principalmente Bitcoin, que ofrecen la posibilidad de efectuar transacciones de manera anónima. Bitcoin nació con la idea de descentralizar los pagos entre usuarios, eliminando la necesidad de la presencia de instituciones financieras en las transacciones. Aunque no exenta de polémica, esta solución ha demostrado en la práctica que es funcional y válida para la realización de transacciones y su adopción está creciendo en todo el mundo. Para hacer consistente su funcionamiento, teniendo en cuenta los problemas derivados de una gestión descentralizada, Bitcoin se basada en redes entre pares (peer-to-peer), manteniendo registros de transacciones que no pueden ser alterados sin tener que realizar complicados cálculos matemáticos para recomponer todo el sistema.

Se han publicado distintos ataques en la red TOR que permiten comprometer el anonimato de sus usuarios. Debido a que los datos que viajan dentro de la red TOR están cifrados, estos ataques principalmente utilizan técnicas de *Fingerprinting* de sitios web y técnicas de correlación extremo-a-extremo.

1.1 Objetivos

A continuación se listan los objetivos que se pretenden alcanzar en este TFM:

1. Estudiar el funcionamiento y las características de la red TOR.
2. Estudiar el funcionamiento y la tipología de la red P2P de Bitcoin.
3. Investigar el uso de las redes TOR y Bitcoin en los cibermercados negros.
4. Investigar los principales métodos de desanonimización en la red TOR.

1.2 Organización del trabajo de recerca

En el Capítulo 2 se explica el funcionamiento de la red TOR y en el Capítulo 3 sobre la criptomoneda Bitcoin. En el Capítulo 4 se investigan los cibermercados negros en la red TOR. En el Capítulo 5 se explican los principales métodos de desanonimización de usuarios en la red TOR. Finalmente, el Capítulo 6 contiene las conclusiones del proyecto.

2. TOR

En este capítulo se proporciona una introducción a la red TOR, sus componentes, los protocolos que utiliza y una explicación detallada de su funcionamiento.

2.1 Introducción

TOR (abreviatura de *The Onion Router* en inglés) es una red de comunicaciones distribuida de baja latencia y superpuesta sobre Internet, en la que el encaminamiento de los mensajes intercambiados entre los usuarios no revela su identidad, es decir, su dirección IP (anonimato a nivel de red) y que, además, mantiene la integridad y la confidencialidad de la información que viaja por ella.

Los orígenes de Tor se remontan a 1995 cuando se inician los trabajos de investigación sobre enrutamiento por capas o de cebolla (*Onion Routing*) en la Oficina de Investigación Naval de la Marina de los Estados Unidos. La motivación principal fue la de proteger las comunicaciones del gobierno. A esta versión se le conoce como generación cero de *Onion Routing* y fue desarrollada por Michael Reed, Paul Syverson y David Goldschlag.

Actualmente el proyecto TOR está en manos de *The Tor Project* [1] una organización sin ánimo de lucro fundada en 2006 por Roger Dingledine y Nick Mathewson pero conformada por cientos de voluntarios. Esta organización, orientada a la investigación y la educación, ha sido financiada por distintas organizaciones [2] y se encarga principalmente del desarrollo del sistema de la red TOR.

2.2 Componentes

La red TOR está formada por una serie de nodos que se comunican entre sí mediante el protocolo de seguridad en la capa de transporte (TLS, *Transport Layer Security*) [3], que es el predecesor del protocolo de capa segura (SSL, *Secure Sockets Layer*) [4]; ambos operan sobre el protocolo de control de transmisión (TCP, *Transmission Control Protocol*) [5]. El protocolo TLS permite mantener la confidencialidad y la integridad de la información transmitida entre nodos.

En TOR existen dos componentes principales, los nodos OR y OP.

- Nodos OR (*Onion Router*): son los nodos que funcionan como encaminadores en la red TOR, dependiendo del lugar que ocupen pueden ser la puerta de entrada, de salida o intermedios y en algunos casos como servidores de directorio (ver apartado 1.3.1 Servicio de Directorio). Los nodos OR se comunican entre ellos mediante conexiones TLS, estas conexiones no son nunca cerradas deliberadamente salvo cuando pasa cierto tiempo de inactividad.
- Nodos OP (*Onion Proxy*): son los clientes (usuarios) que solicitan una conexión a TOR. Estos nodos tienen la finalidad de obtener información del servicio de directorio, para conocer la dirección de los nodos OR y establecer circuitos aleatorios a través de ellos, es decir, los OP aceptan flujos TCP de aplicaciones de usuarios y las multiplexan a través de la red de ORs.

Las conexiones OP-OR no son permanentes, un OP debería cerrar una conexión a un OR si no hay circuitos ejecutándose sobre la conexión y ha vencido cierto tiempo.

2.2.1 Servicio de Directorio

Un servicio de directorio (DS, *Directory Service)* es una aplicación o un conjunto de aplicaciones que almacena y organiza la información sobre los usuarios de una red. En el caso de TOR, el servicio de directorio es una base de datos distribuida que almacena las direcciones de todos los nodos OR.

Cuando un OR inicia, recolecta un conjunto de datos que lo describen a él y a sus funcionalidades. Ejemplos de este tipo de atributos son la dirección IP, nombre de dominio, versión del software TOR, sistema operativo, clave pública, etc. Toda esta información se publica a través del servicio de directorio que es accesible por todos los nodos ORs y OPs, los cuales la usan para tener un conocimiento de la red. A efectos prácticos los servidores de directorios se comportan como grupos establecidos de ORs confiables.

2.3 Enrutamiento cebolla

El enrutamiento cebolla (*Onion Routing*) [6] fue introducido por David M. Goldshlag, Michael Reed y Paul Syverson [7] aplicando las ideas de las redes de mezclado de David Chaum a los sistemas de encaminamiento, para conseguir redes que preserven la privacidad (tanto del mensaje en si como de los interlocutores), de forma transparente a las entidades que se comunican. De esta forma podemos tener infraestructuras para comunicaciones privadas sobre una red pública.

El concepto de *Onion Routing* viene dado por la representación de las capas de una cebolla formada por un núcleo y sus capas externas. El núcleo de la cebolla es el paquete TCP, y las capas externas son envoltorios cifrados con clave simétrica.

Para proteger el paquete TCP cuándo se realiza una petición de conexión a un servidor web a través de TOR, el cliente (nodo OP) genera *n* capas, siendo *n* el número de servidores intermedios (nodos OR). Donde cada capa ha sido creada con una clave simétrica (o privada) negociada entre cada servidor intermedio y el cliente a través de un algoritmo de clave asimétrica (o pública). Una vez que el cliente ha generado la “cebolla”, se envía a través de la conexión TOR, y empieza a recorrer los diferentes servidores intermedios. Cada una de las capas de la cebolla es “pelada” por el servidor intermedio correspondiente a medida que la cebolla va pasando por los servidores intermedios, de manera que el nodo de salida “pela” la última capa obteniendo el paquete TCP original y se lo envía al servidor que aloja la página web.

En la Figura 1 se puede ver una representación gráfica de la transmisión de un mensaje con enrutamiento cebolla con tres capas.

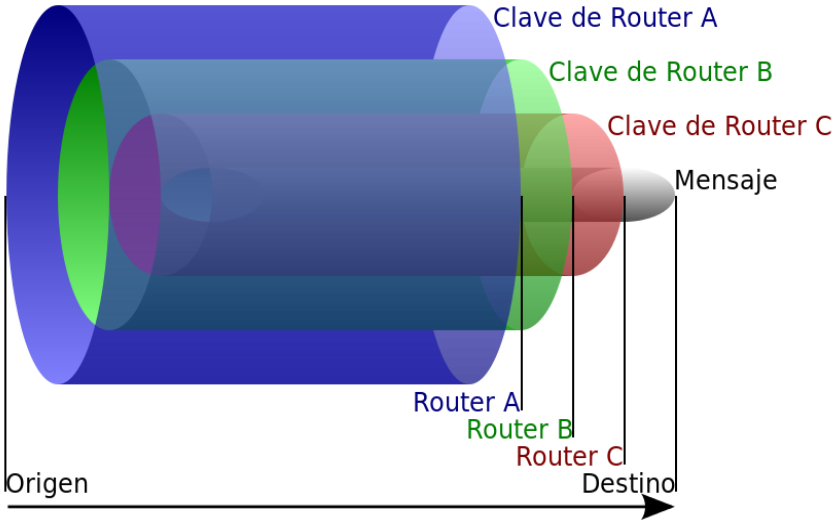

Figura 1. Estructura de la transmisión de los datos con encaminamiento cebolla.

La principal ventaja del enrutamiento por capas, es que no es necesario confiar en cada nodo OR. Si algún nodo OR está comprometido, la comunicación anónima aún es posible debido a que cada nodo OR dentro de la red TOR acepta un mensaje, descifra una capa y lo transmite al siguiente nodo OR. Es decir, si el atacante controla todos los nodo puede rastrear el camino de un mensaje dentro de la red, pero no puede determinarlo si solo controla un nodo OR dentro de la red.

2.3.1 Circuitos telescópicos

Los circuitos telescópicos son el núcleo de la segunda generación del Enrutamiento de cebollas: TOR reemplaza la creación de circuitos en un sólo paso de su antecesor por la creación en etapas, extendiendo el circuito un salto a la vez y negociando con cada uno una clave de sesión temporal, a través del túnel creado con su anterior. Este esquema supera varias desventajas del anterior, como la posibilidad de que el cliente sepa cuando un nodo que ha seleccionado no responde, o la vulnerabilidad a ataques de predecesor.

El proceso de creación de circuitos es el siguiente:

1. El cliente TOR elige un nodo de entrada a la red (e*ntry guards*) y crea una conexión TLS a la misma. Una vez realizada la conexión, ambos determinan un secreto compartido usando el algoritmo de clave pública Diffie-Hellman [8]. Este secreto es la clave de sesión efímera usada por ambos para cifrar y descifrar las celdas transmitidas de uno a otro, formando un túnel.
2. Ahora el cliente TOR tiene un túnel que llega al nodo de entrada. A través del mismo, realiza una conexión TLS con el siguiente nodo del circuito (nodo intermedio o m*iddle relay*), y nuevamente negocia un secreto compartido que servirá de clave de sesión mientras dure el

circuito. Ahora el cliente TOR tiene un túnel al segundo nodo que atraviesa el túnel ya creado al primer nodo.

3. La sucesiva extensión del circuito y creación de túneles continúa mientras el cliente lo desee. En la actual implementación de TOR, la extensión de los circuitos está fijada en 3 saltos.
4. Una vez creados todos los túneles, el cliente de TOR envía un mensaje al último nodo del circuito (nodo de salida o *exit relay*) pidiéndole que acceda al destino (servicio donde el cliente quiere conectarse). Como cada nodo sólo sabe de su antecesor y su posterior, este nodo no conoce la identidad del cliente.
5. Ahora el cliente de TOR puede empezar a enviar datos, cifrados con cada clave de sesión en el orden inverso al que van a ser recorridos los nodos, a través de estos mismos. Al llegar a cada nodo, la capa de cifrado correspondiente es eliminada, tal como ocurría con las "cebollas", hasta el nodo de salida, que envía el flujo hacia el destino.

A continuación en la figura 2 se puede ver una representación gráfica del circuito de TOR.

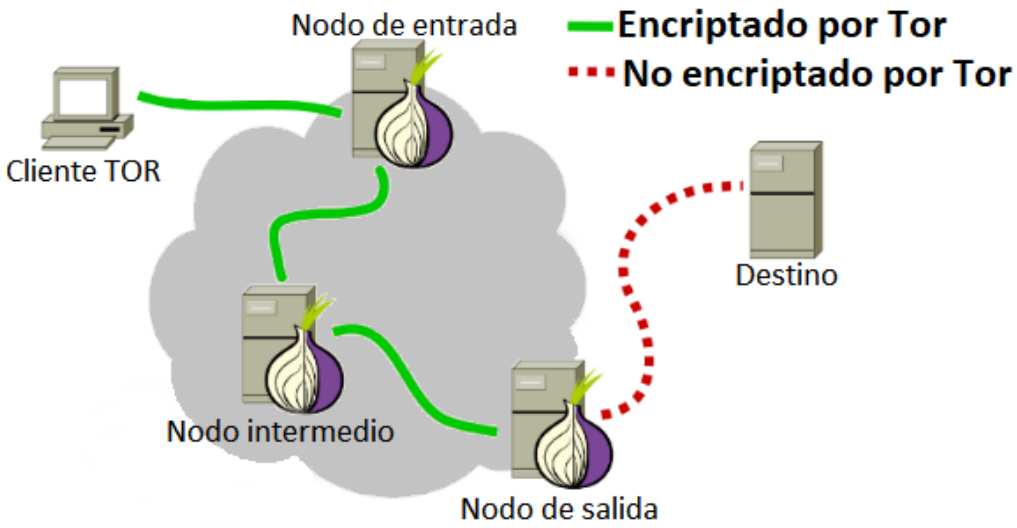

Figura 2. Representación gráfica del circuito de TOR.

2.4 Funcionamiento de la red TOR

Una vez explicado el concepto de enrutamiento cebolla, en este apartado se explican los pasos de cómo se consigue el anonimato de un cliente (nodo OP) a través de la red TOR:

1. A partir de la información obtenida de su configuración y del servicio de directorio, el OP decide un circuito por el que van a circular los paquetes. Por defecto el circuito tiene 3 nodos OR.
2. El OP negocia las claves de cifrado necesarias con cada OR del circuito con el fin de proteger sus datos en todo el camino, y antes de iniciar el proceso de transmisión. La obtención de las claves simétricas (una para cada sentido de comunicación: *forward key* (Kf), *backward key* (Kb), se realiza a partir del protocolo de establecimiento de claves Diffie-Hellman.

3. A continuación el OP cifra el paquete con la clave del último OR del circuito.
4. Al paquete se añadirá una capa de cifrado por cada punto de paso del circuito.
5. Tras finalizar el cifrado, el OP envía el paquete resultante al primer nodo del circuito.
6. El primer OR quita "su capa de la cebolla" y envía el paquete al siguiente nodo.
7. Al pasar por cada nodo intermedio, va perdiendo sus capas de cifrado. Esto permite que ningún OR puede hacerse con la imagen completa del circuito, ya que sólo conoce los OR/OP anterior y posterior.
8. Al alcanzar el nodo de salida, el mensaje queda descifrado por completo. No se trata del mensaje original, el nodo salida solo dispone de la información del servidor al que quiere acceder el cliente pero no conoce quien realiza la petición.

2.4.1 Interfaz de entrada

TOR sólo permite anonimizar tráfico TCP. Las aplicaciones acceden a la red TOR a través del interfaz SOCKS [9] lo cual significa que toda aplicación con soporte SOCKS puede usar TOR para realizar comunicaciones anónimas sin necesidad de modificaciones adicionales. El cliente TOR recibe tráfico SOCKS desde nuestras aplicaciones y luego, de forma transparente, se encarga de comunicarse con los routers de la red TOR para enviar las peticiones y posteriormente devolvernos los resultados.

SOCKS es un protocolo que facilita el enrutamiento de paquetes que se envían entre un cliente y un servidor a través de un servidor proxy. Según la pila de protocolos OSI está en el nivel 5 (sesión). Según la pila de protocolos IP está en la capa de aplicación.

TOR usa SOCKS para soportar la mayoría de programas basados en TCP sin hacer ninguna modificación.

2.5 Servicios ocultos

En el apartado anterior hemos visto cómo se consigue el anonimato de un cliente usando la red TOR. Pero también, TOR permite mantener el anonimato de los servicios ubicados dentro de la red. A esto servicios se les llama Servicios Ocultos (*Hidden Services*). Mediante un servicio oculto, es posible ofrecer un servidor en la red TOR sin una IP que lo identifique, con la desventaja de que es necesario accederlo a través de TOR.

El proceso paso a paso de creación de un servidor web anónimo es el siguiente:

1. Bob configura el servidor web para que sea accesible dentro de la red TOR, donde se generan un par de claves pública/privada que identifican al servicio. Con la clave pública se genera un *digest* de 16 caracteres en Base32 usando una función hash. El *digest* forma parte de la dirección del servicio (e.g. l73fuoioj5hzznxc), junto con el pseudodominio de nivel superior (pseudo TDL) .onion, dando como resultado la dirección con la que se accede al servicio: http://l73fuoioj5hzznxc.onion.

2. Para lograr la alcanzabilidad del servicio en la red TOR, el primer paso es elegir, aleatoriamente, un grupo de Routers de Cebollas (Onion Routers) para que sirvan de Puntos de Introducción (*Introduction Points*, IP) y generar circuitos TOR (o sea, de 3 saltos, cifrados) hacia cada uno de ellos.
3. El segundo paso es crear un Descriptor del servicio, que incluye: la dirección .onion, el/los puerto/s por los que se accede al servicio, una descripción textual opcional y la dirección de los Puntos de Introducción, y firmado con la clave privada. Este Descriptor se publica a una tabla hash distribuida, una base de datos de donde cualquier nodo o cliente puede obtener el Descriptor.
4. Ahora un cliente (Alice) quiere conectarse al servicio. Se asume que ya conoce la dirección, tal vez porque la vió en un índice de servicios ocultos o Bob se la dijo. Como cuarto paso, Alice descarga el Descriptor del servicio de la base de datos distribuida y obtiene la dirección de los Puntos de Introducción.
5. Para el quinto paso Alice ha creado un circuito TOR hasta un nodo cualquiera, pidiéndole que actúe como Punto de Encuentro (Rendezvous Point, RP). Para esto, le comunica un secreto de un sólo uso, generado para este encuentro.
6. Ahora Alice crea un mensaje de Introducción, cifrado con la clave pública del servicio oculto, que incluye la dirección del Punto de Encuentro y el secreto de un sólo uso, y le pide a alguno de los Puntos de Introducción que se lo envíe al servicio oculto.
7. En el séptimo paso el servicio oculto descifra el mensaje y obtiene la dirección del Punto de Encuentro y el secreto. Crea un circuito TOR hacia dicho Punto de Encuentro y le envía el secreto.
8. En el octavo y último paso, el Punto de Encuentro notifica a Alice de la conexión establecida, y ambos pueden empezar a comunicarse a través de sus circuitos TOR. El Punto de Encuentro simplemente reenvía los paquetes que le llegan.

Es importante aclarar que los Puntos de Introducción no son usados en la comunicación final para evitar que un sólo retransmisor sea completamente responsable del servicio oculto.

2.6 Herramientas para acceder en la red TOR

En este apartado se explican las herramientas principales para acceder en la red TOR proporcionadas por la organización *Tor Project*.

2.6.1 TOR Browser

Tor Browser [10] es un navegador web que integra el software necesario para conectarse a la red TOR. Este navegador web es una versión modificada de Mozilla Firefox ESR que incluye las extensiones TorButton, TorLauncher, NoScript y HTTPS Everywhere Firefox y el proxy Tor para

aumentar la seguridad de los usuarios [11]. Esta herramienta está disponible para los sistemas operativos Microsoft Windows, MacOS y Linux. También se puede ejecutar desde medios extraíbles.

Tor Browser inicia automáticamente los procesos en segundo plano de TOR y enruta el tráfico a través de la red TOR. Al finalizar una sesión, el navegador borra los datos sensibles a la privacidad, como las cookies HTTP y el historial de navegación.

2.6.2 Orbot

Orbot [12] es una aplicación para dispositivos Android que ofrece un proxy local que permite conectarse la red de TOR y encapsular todo el tráfico a través de ella. Esta aplicación ha sido desarrollada con la colaboración de *Guardian Project* [13] y está disponible en su página web y en la plataforma Google Play Store.

Orbot no incluye ningún navegador web integrado pero recomienda el uso del navegador Orfox [14]. Este navegador está construido a partir del mismo código fuente de TOR Browser con algunas modificaciones menores para hacerlo compatible con el Firefox de Android y el sistema operativo Android. Esta aplicación también ha sido desarrollada con la colaboración de *Guardian Project* y está disponible en su página web.

2.6.3 Tails

Tails [15] es una distribución Linux basada en Debian GNU/Linux que encapsula todo el tráfico a través de la red TOR para preservar la privacidad y el anonimato. Este sistema operativo es la evolución de la distribución Incognito. El sistema está diseñado para ser arrancado como un Live CD o USB y ejecutarse en la memoria RAM para no dejar ningún rastro en el almacenamiento local a menos que se indique explícitamente.

Este SO además de estar apoyado por *Tor Project*, también recibe el apoyo del *Debian Project* [16], Mozilla y *Freedom of the Press Foundation* [17].

3. Bitcoin

En este capítulo se explica la criptomoneda Bitcoin, su arquitectura, estructuras de datos y una explicación detallada de su funcionamiento.

3.1 Introducción

Bitcoin es una criptomoneda [18], todo y que este término se aplica también al protocolo y a la red P2P que lo sustenta y de forma común se denomina como una moneda digital. Generalmente se usa "Bitcoin" para referirse a la red o al protocolo y "bitcoin" para referirse a las unidades monetarias. Al contrario que la mayoría de las monedas, Bitcoin no depende de la confianza en ningún emisor central, sino que recurre a una base de datos distribuida en varios nodos de una red P2P para registrar las transacciones y utiliza la criptografía para proveer funciones de seguridad básicas, tales como garantizar que las bitcoins solo puedan ser gastadas por su dueño, y nunca más de una vez.

El diseño de Bitcoin permite poseer y transferir valor de forma potencialmente anónima. Las monedas pueden ser guardadas en cualquier ordenador en la forma de un archivo "de bolsillo", o con un tercero que ofrezca el servicio de almacenar ese archivo. En cualquiera de los casos, los bitcoins pueden ser enviados por medio de Internet a cualquiera que tenga una "dirección Bitcoin". La estructura P2P de Bitcoin y la falta de administración central hace imposible para cualquier autoridad, gubernamental u otra, la manipulación del valor de los bitcoins o la creación de inflación produciendo más de ellos, porque la propia red es la que gestiona las transacciones y la emisión de bitcoins, que se generan a través de la llamada minería, de forma controlada y descentralizada.

La primera aparición pública de Bitcoin se produjo en la lista de correo *cryptography* [19], donde un usuario con el pseudónimo "Satoshi Nakamoto" anunció, el 1 de noviembre de 2008, que había estado trabajando en un nuevo sistema de dinero electrónico, resumiendo sus propiedades. El contenido del artículo original que describe su trabajo se encuentra disponible en el portal de Bitcoin: *http://www.bitcoin.org.*

El 11 de febrero de 2009, un perfil creado en el *portal P2P foundation*, también con el nombre de "Satoshi Nakamoto", publicó un mensaje: "Bitcoin open source implementation of P2P currency" [20]. En el texto, "Satoshi" daba a conocer el portal oficial de Bitcoin, las características fundamentales de éste, donde se describía el diseño e, incluso, el cliente inicial con el que comenzar a participar en la red.

3.2 Conceptos generales

Para empezar con una idea básica de cómo funciona el sistema, se describen a continuación los conceptos básicos en los que se basa el sistema.

Direcciones Bitcoin

Dirección virtual de un usuario que contiene monedas Bitcoin y se utiliza para pagar y recibir pagos, similar a una cuenta de banco. Un mismo usuario puede tener tantas direcciones Bitcoin como necesite y se identifican con una clave pública.

Una dirección Bitcoin es, básicamente, una transcripción de una clave pública. La clave privada asociada sirve para firmar las transacciones y la clave pública sirve para identificar la dirección y validar las firmas.

Monederos

Espacio virtual, equivalente a un monedero físico, donde se almacenan y gestionan direcciones Bitcoin de un usuario y los pagos que se realizan con ellas.

Transacciones

Una transacción es una transferencia de dinero de una dirección Bitcoin "A" hacia otra dirección "B". Para componer una transacción, el propietario de la dirección "A" firma una transcripción de la dirección "B" (entre otros datos) con la clave privada asociada a la dirección "A", de forma que la red sabrá que el nuevo propietario legítimo es el dueño de la dirección "B".

Bloques

Es una estructura que agrupa transacciones. Las transacciones pendientes de confirmar se agrupan en un bloque sobre el que se realiza el denominado proceso de minería.

Cadena de bloques

Registro público de las transacciones de Bitcoins validadas en orden cronológico. Cuando un bloque ha sido confirmado, a través de la minería, éste pasa a formar parte de la cadena de bloques.

Minería

Proceso de realización de cálculos matemáticos para confirmar transacciones en la red Bitcoin. A través de la minería se pueden crear nuevas bitcoins al mismo tiempo que se confirman transacciones.

3.3 Roles

En cuanto a los actores que intervienen en el sistema, se pueden distinguir dos tipos de participantes, que componen dos conjuntos no necesariamente disjuntos:

- **Usuarios normales:** son usuarios del sistema Bitcoin. Compran y pagan bienes y servicios utilizando bitcoins, produciendo transacciones en el sistema.
- **Mineros:** son usuarios especiales que dedican potencia de cómputo a validar nuevas transacciones, creando lo que se conoce como bloques de transacciones. Los cálculos que tienen que realizar son muy costosos por lo que se ven recompensados por ellos.

Adicionalmente, hay un tercer rol que normalmente se ignora: los desarrolladores. El medio principal de Bitcoin es, en definitiva, un software. Como tal, necesita un desarrollo y mantenimiento activos para lo cual es imprescindible un equipo de desarrolladores. No obstante, al ser un protocolo libre, cualquiera puede crear un cliente de Bitcoin. De hecho, actualmente existen varios [21].

3.4 Conceptos criptográficos

Los sistemas con una complejidad como la de Bitcoin están siempre respaldados por un conjunto de primitivas avanzadas. Sin conocer las primitivas, comprender cómo se consiguen muchas de las propiedades anunciadas por el sistema no es posible. Por ello, una vez adquirido un conocimiento general del sistema y cómo funciona, en esta sección se verá en detalle los conceptos criptográficos fundamentales del sistema.

Firmas digitales

Bitcoin utiliza el algoritmo ECDSA [22] (*Elliptic Curve Digital Signature Algorithm*, Algoritmo de Firma Digital de Curva Elíptica) para firmar las transacciones, utilizando los parámetros recomendados por el *Standards for Efficient Cryptography Group* (SECG), secp256k1 [23]. Las firmas utilizan la codificación DER [24] para empaquetar sus componentes en un único flujo de bytes.

ECDSA ofrece ventajas frente a otros esquemas de firma que lo hacen ideal para su utilización en un protocolo distribuido en Internet, como son:

- Longitudes de clave y de firma muy cortas.
- Generación y verificación de firmas muy rápidas.

Hashes criptográficos

En los cálculos de hashes realizados en Bitcoin se utilizan los estándares SHA-256 [25] y, cuando se requiere que el hash sea más corto, RIPEMD-160 [26]. Normalmente el cálculo de hashes se realiza en dos fases: la primera con SHA-256 y la segunda, dependiendo de las necesidades de longitud del resultado, con SHA-256 o RIPEMD-160.

Números aleatorios y nonces

Los números aleatorios y su generación son pilares fundamentales de la criptografía. Los *nonces* son números aleatorios "especiales" que, en principio, sólo se utilizan una vez (de ahí su nombre, que en inglés viene de *number used only once*), aunque a veces los dos términos se utilizan de forma indistinguible.

En Bitcoin, los números aleatorios y nonces se utilizan de forma directa para la generación de bloques. Como se verá a continuación, para obtener un nuevo bloque es necesario encontrar un número aleatorio que satisfaga ciertos requisitos. También se utilizan en Bitcoin, aunque de manera indirecta, como parte del algoritmo de firmas digitales (ECDSA).

Pruebas de trabajo

Las pruebas de trabajo (*proofs of work*, en inglés) son el principal componente de Bitcoin responsable de garantizar que la red mantiene un comportamiento legítimo. Brevemente, esta idea hace que validar/calcular nuevos bloques de transacciones conlleve un coste computacional muy elevado, de forma que, para hacerse con el control de la red (y por tanto de qué se valida y qué no), un atacante necesitaría una potencia de cómputo extremadamente difícil de conseguir. El principal precursor de esta idea es el método Hashcash [27] ideado en 1997 para evitar el envío de spam.

En concreto, en Bitcoin este control de complejidad en los cálculos para los nuevos bloques se realiza obligando a que el hash de cada nuevo bloque deba comenzar con un número determinado de ceros. Como se verá más adelante, para el cálculo de este hash se combinan datos de bloques anteriores y un nonce. Dado que las funciones hash criptográficas no son invertibles, para encontrar un bloque válido la única alternativa será ir obteniendo diferentes nonce hasta encontrar uno que cumpla el requisito preestablecido.

3.5 Arquitectura del sistema

Los nodos que integran Bitcoin componen un sistema de comunicaciones P2P o *peer-to-peer*. Como ya se ha mencionado, la filosofía aquí es evitar roles de servidor que pudieran evolucionar en, o ser utilizados por, autoridades centrales, gobiernos, etc.

Como todo sistema P2P, Bitcoin dispone de una serie de mecanismos para descubrir nuevos nodos [28] en la red, y mantener una lista actualizada de los mismos. Además, distintos clientes de Bitcoin pueden también ofrecer mecanismos adicionales, como el cliente Satoshi [29]. Entre las principales opciones, destacan los mensajes de tipo “addr” y “getaddr”, mediante los cuales un cliente envía (o solicita) a otro un listado de clientes actualmente conectados a la red. También, en el código de los clientes se suele incluir un listado de nodos semilla, que se utilizarán para iniciar el proceso de conexión a la red en caso de que el resto de mecanismos fallen.

Además de los mecanismos para descubrir otros nodos en la red, hay otros tipos de mensajes de uso frecuente en Bitcoin. Por ejemplo, los mensajes “tx” y “block”, utilizados para enviar datos de transacciones y bloques, respectivamente, de manera que los nodos de la red puedan mantener la sincronía requerida por el protocolo. O los mensajes de tipo “inv”, que se utilizan para anunciar (y retransmitir) nuevas transacciones.

En la Wiki de Bitcoin [30] hay un listado de los tipos de mensaje, su definición y explicación.

3.6 Estructuras de datos

En esta sección se verán cómo se construyen los distintos conceptos del sistema a partir de las primitivas criptográficas enumeradas anteriormente.

3.6.1 Direcciones y monederos

Una dirección Bitcoin se compone de un par de claves pública y privada ECDSA (*Elliptic Curve Digital Signature Algorithm*). La dirección se identifica con el hash de la clave pública, al que se añade una suma de verificación. Dicho resumen se codifica en una versión modificada de base 58 [31], que básicamente mantiene los ceros a la izquierda cuando realiza la codificación. Así, una dirección se identifica de la siguiente forma:

```
$Version = 1 byte de ceros
$KeyHash = $Version + RIPEMD-160(SHA-256($PublicKey)
$Checksum = SHA-256(SHA-256($KeyHash))[0-3]
$BitcoinAddress = Base58Encode($KeyHash + $Checksum)
```

Al estar identificada por la clave pública ECDSA, todas las operaciones que se realizan con esa dirección deben estar apoyadas por la utilización de la clave privada asociada.

Por lo tanto, los monederos son una agrupación de pares de claves públicas y privadas. En cualquier caso, esto no supone ninguna limitación a que aplicaciones de monederos puedan incluir alguna funcionalidad adicional para realizar otras tareas, como por ejemplo efectuar transacciones.

3.6.2 Transacciones

Las transacciones en Bitcoin son registros firmados digitalmente que cambian el propietario de fondos en bitcoins asignándolos a otra dirección.

Para componer una transacción se necesitan los siguientes componentes, formando la estructura que se muestra en la Figura 3:

- Entradas: registros que referencian los fondos de transacciones previas
- Salidas: registros que determinan el nuevo propietario de las bitcoins transferidas.

Las salidas se utilizarán nuevamente como entradas de transacciones futuras. Además, siempre se utilizan todos los bitcoins que hay en las direcciones de entrada, es decir, la suma de todas las entradas no se puede dividir aunque sea mayor que la cantidad a pagar. En este caso, entre las direcciones de salida se incluirá una dirección de devolución, a través de la cual el pagador recibirá "la vuelta".

En concreto, una transacción en Bitcoin está compuesta por los campos mostrados en la Tabla 1 [32].

Campo	Descripción
Version no	Actualmente 1
In-counter	Número de entradas de la transacción
List of inputs	Lista de entradas de la transacción
Out-counter	Número de salidas de la transacción
List of outputs	Lista de salidas de la transacción
Lock_time	El número de bloque o marca temporal hasta la cual esta transacción está bloqueada.

Tabla 1. Campos de una transacción.

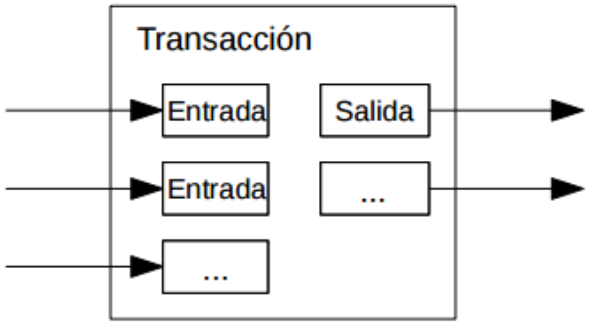

Figura 3. Transacción.

Cada entrada de una transacción es firmada digitalmente por el pagador, lo que desbloquea los fondos contenidos en la dirección asociada a la clave privada utilizada para firmar. Así solamente el usuario que posee la clave privada correspondiente es capaz de crear una firma válida lo que asegura que solamente el propietario del saldo puede utilizarlo. Este proceso se muestra gráficamente en la Figura 4.

En las salidas se especifica, entre otros datos, la dirección del cliente que recibirá los bitcoins y, en caso necesario, una dirección en propiedad del pagador para recibir la vuelta.

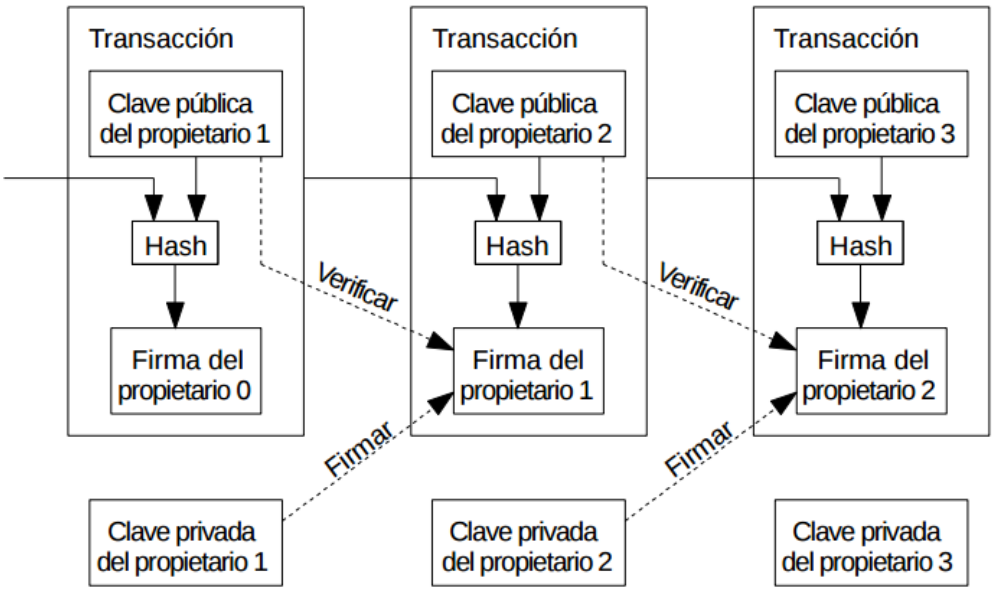

Figura 4. Firma de transacciones.

En una transacción, la suma de sus entradas debe ser igual o mayor que la suma de las salidas. En el caso de que la cantidad de bitcoins de la entrada sea mayor que la de la salida, la diferencia se considera una tasa de transacción, y quien incluya esa transacción en la cadena de bloques puede disponer de esa cantidad. Esta recompensa es una manera de motivar a los mineros, que obtienen beneficios por su trabajo en forma de bitcoins, siendo los pagadores los que suelen establecer la tasa a incluir en sus pagos (aunque muchos clientes de Bitcoin utilizan valores por defecto). Por ello, es frecuente que transacciones con tasas (o recompensas) mayores sean procesadas más rápido que transacciones con tasas menores.

También existen además transacciones especiales que suponen la creación de nuevas bitcoins que son generadas a través de la minería por lo que no tienen entradas.

3.6.3 Bloques

Un bloque es un registro que contiene confirmaciones de transacciones que se encontraban pendientes. Aproximadamente cada 10 minutos, en promedio, un nuevo bloque que incluye nuevas transacciones se anexa a la cadena de bloques a través de la minería.

Un bloque está compuesto por los campos mostrados en la Tabla 2 [33].

Campo	**Descripción**
Magic no	Valor establecido siempre a 0xD9B4BEF9
Blocksize	Número de bytes que siguen, hasta el final del bloque
Blockheader	Cabecera con metainformación sobre el bloque y la cadena
Transaction counter	Número de transacciones en la siguiente lista
Transactions	Lista de transacciones contenidas en el bloque

Tabla 2. Campos de un bloque.

De los campos anteriores, destaca en primer lugar la lista de transacciones que incluye las transacciones nuevas que el minero que ha calculado el bloque ha decidido incluir en el mismo. Qué transacciones se incluyen depende principalmente de su prioridad. Dentro de la cabecera se incluyen los campos mostrados en la Tabla 3 [34].

Campo	**Descripción**
Version	Versión de bloque
hashPrevBlock	Hash del bloque anterior
hashMerkleRoot	Hash de la raíz del árbol Merkle
Time	Marca de tiempo de creación del bloque
Bits	Especificiación de la complejidad del bloque
Nonce	Nonce que resuelve la prueba de trabajo

Tabla 3. Cabecera de bloque.

Los hashes incluidos en los campos segundo y tercero tienen como fin establecer la cadena de bloques cuyo concepto se desarrolla a continuación.

El campo Bits define cual era la complejidad requerida en el momento de generación del bloque para que dicho bloque fuera válido. Esta complejidad es variable en función de la capacidad de cómputo total, de forma que cada bloque se genere, en promedio, cada 10 minutos.

El valor Nonce es el número que resuelve la prueba de trabajo. Concretamente, la prueba de trabajo consiste en calcular el hash (SHA-256) de los seis valores de la cabecera. El hash resultante debe ser menor que el número codificado en Bits.

Para optimizar el espacio en disco necesario para almacenar la cadena de bloques, las transacciones que se incluyen en cada bloque se organizan ya en forma de árbol de Merkle (ver Figura 5). Dada la construcción de estos árboles, gran parte de las transacciones incluidas en el árbol pueden ser descartadas o podadas sin comprometer la integridad del bloque.

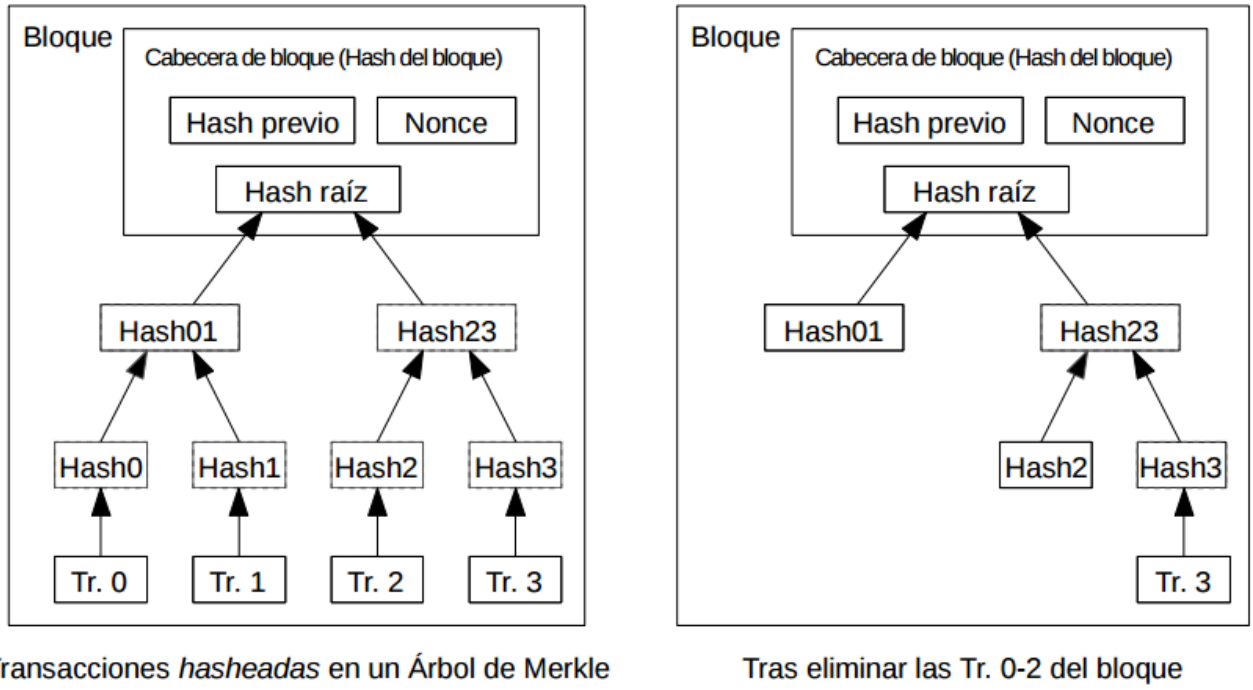

Figura 5. Bloques y poda de transacciones.

3.6.4 Cadena de bloques

La cadena de bloques de la red Bitcoin es una lista creada de forma colectiva con todas las transacciones que han sido confirmadas y validadas por la propia red mediante la inclusión de transacciones en bloques y de estos últimos en la cadena.

Cuando un nodo de la red consigue crear un nuevo bloque, lo transmite al resto de nodos. El resto de nodos verifican que el bloque es correcto, y en caso afirmativo, lo añaden a su cadena y lo difunden. Mediante la difusión del nuevo bloque, éste acabará añadiéndose siempre y cuando no se haya creado otra rama en la cadena de bloques en la que haya participado una cantidad de usuarios con más capacidad de cómputo.

Por la propia naturaleza de la cadena de bloques, se puede extraer el historial de posesión de todas las monedas, siguiendo la lista de transacciones. Así, un usuario no puede reutilizar monedas que ya usó, ya que la propia red rechazará la transacción. Por ejemplo, en Blockchain [35] se pueden ver casos recientes de reutilización de monedas que se han detectado y, convenientemente bloqueado.

Nótese, no obstante, que puede darse el caso de que haya bitcoins reutilizadas de manera no malintencionada. Por ejemplo, por fallos de comunicación masivos, como caídas de redes de comunicaciones, o cuando se crean ramas en la cadena, conteniendo cada una aproximadamente la mitad de la potencia de cálculo del sistema.

Por ello, es buena práctica esperar un tiempo determinado para confirmar una transacción y, por lo tanto, que el receptor de la bitcoin pueda considerar el pago como recibido. Por defecto, los clientes más extendidos incluyen un tiempo de espera de 6 bloques. Es decir, hasta que no se hayan validado 6 bloques desde el que incluyó la transacción, no se considera el pago como realmente efectuado. Dado que el tiempo medio de generación de bloques es de uno cada 10 minutos, esto supone que las transacciones tardan en confirmarse aproximadamente una hora.

Por último, la cadena de bloques, en lo que a estructura de datos se refiere, se establece mediante los campos “hashPrevBlock” y “hashMerkleRoot” de cada bloque vistos anteriormente.

3.7 El protocolo

En esta sección se explica cómo funciona el sistema Bitcoin utilizando el servicio de explorador de bloques de bitcoin “blockchain.info” [36]. Con este servicio online se puede visualizar la información de una transacción y/o un bloque concreto.

Primero se explica su funcionamiento a través de una transacción y un bloque concretos. A continuación se detalla cómo funcionan las recompensas en el proceso de minería y por último se explica cómo funciona la confirmación de transacciones.

3.7.1 Transacción

Para explicar el contenido de una transacción se ha utilizado como ejemplo una transacción concreta del bloque 450000 generado el 25 de enero de 2017. Esta información se ha extraído con el explorador ‘blockchain.info’.

En la Figura 6 se muestra la transacción de forma esquemática, donde se ha dividido en los siguientes tres apartados:

1. El identificador único de transacción, es decir, el resto de campos combinados y pasados por la función SHA-256, utilizada para identificar de forma exclusiva una transacción.
2. La lista de entradas, que incluye la dirección de entrada de la transacción, la cantidad de bitcoins existentes en esa dirección al momento de la transacción.
3. La lista de salidas, incluye las direcciones de salida de la transacción, la cantidad de bitcoins existentes en esas direcciones al momento de realizarse la transacción. También nos

muestra que en el momento de la observación ya se han gastado la cantidad de bitcoins depositados en las direcciones de salida.

Figura 6. Ejemplo transacción.

En la siguiente figura 7 se muestra la tabla 'Resumen' de una transacción que contiene los siguientes campos:

- Tamaño: tamaño en bytes que ocupa la transacción dentro de un bloque.
- Hora de recepción: fecha y hora en la que la red ha recibido la transacción.
- Incluídas en el bloque: altura del bloque en el que está añadida esta transacción.
- Confirmaciones: a partir de la emisión de una transacción en la red y de estar incluida en un bloque, cada nuevo bloque minado supone una confirmación. Es recomendable que siempre que realicemos un envío esperemos a que reciba más de 6 confirmaciones para poder decir que el pago se ha realizado correctamente.
- Retransmitido por la IP: ip del nodo bitcoin que transmitió esta transacción a la blockchain, esta dirección IP no necesariamente representa la dirección IP del origen de la transacción, es decir, si el usuario usa un "light wallet" lo que habrá sera la IP del nodo que usa el "light wallet".
- Visualizar: esta página nos mostrará un mapa de árbol con todo el procedimiento que ha realizado una transacción con las salidas utilizadas de forma gráfica.

Resumen	
tamaño	225 (bytes)
Hora de Recepción	2017-01-25 22:10:19
Incluídas en el Bloque	450000 (2017-01-25 22:11:29 + 1 minutos)
confirmaciones	11485 confirmaciones
Retransmitido por la IP	188.122.92.132 (whois)
Visualizar	Ver Gráfico de Árbol

Figura 7. Ejemplo resumen transacción.

A continuación, en la Figura 8, se muestra la tabla 'Entradas y Salidas' de una transacción que contiene los siguientes campos:

- Entrada total: cantidad de bitcoins seleccionados para poder realizar la transacción satisfactoriamente.
- Salida total: entrada total menos las comisiones de transacción.
- Comisiones: céntimos de bitcoin que varían según la prioridad con que el emisor quiere que la transacción se añada a un nuevo bloque.
- Tarifa por byte: es el resultado de dividir el valor de la comisión por el tamaño en bytes de la transacción
- Estimado de BTCs transaccionados: cantidad total de bitcoins transmitidos al receptor de la transacción.

Entradas y Salidas	
total de entrada	1.66083207 BTC
Salida Total	1.66057411 BTC
Comisiones	0.00025796 BTC
Tarifa por byte	114.649 sat/B
Estimado de BTCs transaccionados	0.39458544 BTC
Scripts	Ocultar scripts y Coinbase

Figura 8. Ejemplo entradas y salidas transacción.

3.7.2 Bloque

En el apartado anterior se ha analizado una transacción concreta del bloque 450000. En este apartado se analiza todo el bloque en conjunto con el mismo servicio online de exploración 'blockchain.info'.

En la Figura 9 se muestra la tabla 'Resumen' de un bloque que contiene los siguientes campos:

- Número de Transacciones: Indicador con la totalidad de transacciones procesadas en el bloque.
- Salida Total: Corresponde al "output" o salidas totales de las transacciones incluidas en el bloque.
- Volumen Estimado de la Transacción: Total en bitcoins transmitidos en el bloque.
- Comisiones de la Transacción: Las comisiones destinadas a los mineros de la red que han validado el bloque y sus respectivas transacciones.
- Altura: número de bloque en la blockchain.

- Fecha y Hora: Instante en el que se minó el bloque. Este campo nos facilita la fecha y la hora completa en la que se resolvió.
- Dificultad: Es un índice para medir la dificultad de minar un nuevo bloque.
- Bits: valor para determinar qué hashes serán considerados válidos. En concreto, este campo es una versión codificada del máximo valor que puede tomar el hash del bloque para ser considerado válido.
- Tamaño: tamaño total en KB que ocupa la información del bloque buscado dentro de la cadena de bloques.
- Versión: Número que identifica la versión del protocolo que está corriendo en el nodo encargado del minado de ese bloque.
- Mientras tanto: Número que corresponde al 'nonce' de un bloque. Este número está presente en la cabecera de cada bloque y es utilizado en el algoritmo de Proof of Work para resolver la prueba de trabajo.
- Recompensa del Bloque: Cantidad de bitcoins emitidos con el minado del bloque.

Bloques #450000

Resumen	
Número de Transacciones	2156
salida total	5,861.54686374 BTC
Volumen Estimado de la Transacción	918.55366074 BTC
Comisiones de la Transacción	0.84304038 BTC
Altura	450000 (cadena principal)
Fecha y Hora	2017-01-25 22:11:29
Hora de Recepción	2017-01-25 22:11:29
Resuelto por	F2Pool
Dificultad	392,963,262,344.37
Bits	402836551
tamaño	999.908 KB
Versión	0x20000000
Mientras tanto	2972550269
Recompensa del Bloque	12.5 BTC

Figura 9. Ejemplo resumen bloque.

En la siguiente figura 10 se muestra la tabla 'Hashes'. Concretamente, el Hash del bloque analizado, los Hashes de su bloque anterior y posterior y la Raíz de Merkle constituida a partir de todos los identificadores de transacción de un bloque determinado.

hashes	
Hash	00000000000000000014083723ed311a461c648068af8cef8a19dcd620c07a20b
Bloque Anterior	00000000000000000024c4a35f0485bab79ce341cdd5cc6b15186d9b5b57bf3da
Bloque(s) siguiente(s)	00000000000000000011cf0c4c2ecb0312aac4b321884ee25e46a61913466e443
Raíz de Merkle	ff508cf57d57bd086451493f100dd69b6ba7bdab2a0c14254053224d42521925

Figura 10. Ejemplo hashes bloque.

A continuación, en la Figura 11, se muestra una parte de la tabla llamada 'Actas'. Esta tabla contiene todas las transacciones incluidas en el bloque 450000 y se ha dividido la primera transacción en cinco apartados para poder explicar los campos que componen cada transacción. Los campos son los siguientes:

1. Identificador de la transacción.
2. Entradas de la transacción. En este caso no hay ninguna entrada.
3. Salidas de la transacción. En este caso hay una salida que corresponde a la recompensa por el bloque minado más las comisiones de la resta de transacciones del bloque.
4. Cantidad de bitcoins transmitida en la transacción.
5. Fecha y hora en la que se produjo la transacción en la red y su tamaño en bytes.

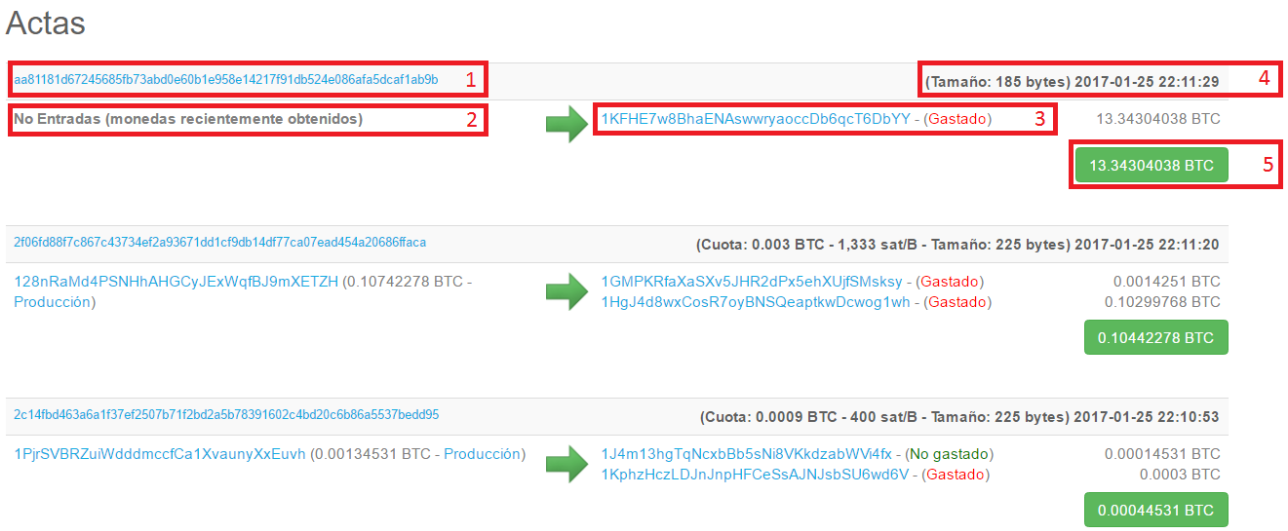

Figura 11. Ejemplo actas bloque.

3.7.3 Recompensas

Dado que calcular nuevos bloques es muy costoso, el minero o los mineros que encuentren nuevos bloques reciben una recompensa.

Esta recompensa puede recibirse por dos medios. Por un lado, Bitcoin tiene establecido un límite máximo de 21 millones de bitcoins y hasta que se llegue a ese límite, la generación de cada nuevo

bloque es recompensada con una cantidad predefinida de bitcoins nuevos. Por ejemplo, hasta noviembre de 2012, se recompensaba con 50 bitcoins a cada nuevo bloque. Después hasta el julio de 2016 la recompensa era de 25 bitcoins. Desde entonces, la recompensa es de 12.5, estimándose que se reducirá a la mitad hacia 2020.

Por otro lado, para mantener una motivación similar para los mineros pese al decrecimiento de las recompensas (que eventualmente llegará a cero), existen las comisiones, mediante las cuales, los usuarios "donan" parte del dinero implicado en una transacción al minero que la valide.

Un detalle adicional, que permite aumentar la resistencia de Bitcoin frente a atacantes o ante situaciones atípicas, es que las recompensas (incluyendo nuevas monedas) obtenidas a través del minado no se pueden gastar hasta que se hayan añadido 100 bloques nuevos a la cadena. Esta política es útil por ejemplo para evitar casos en los que se genera un bloque (por tanto creándose nuevas monedas) y el minero que lo creó utiliza algunas de esas bitcoins, pero posteriormente el bloque es descartado por no pertenecer a la cadena más larga. El término utilizado para este concepto es "*100-block maturation time*" [37].

3.7.4 Confirmación de la transacción

Aunque una transacción nueva haya sido incluida en un bloque y dicho bloque en la cadena, inicialmente puede ser posible que esa modificación sea revertida. Esto podría pasar cuando se crean dos ramas inicialmente válidas, lo cual puede ocurrir por diversos motivos. Por ejemplo, si dos mineros reportan dos nuevos bloques válidos al mismo tiempo. Este escenario se muestra en la Figura 12.

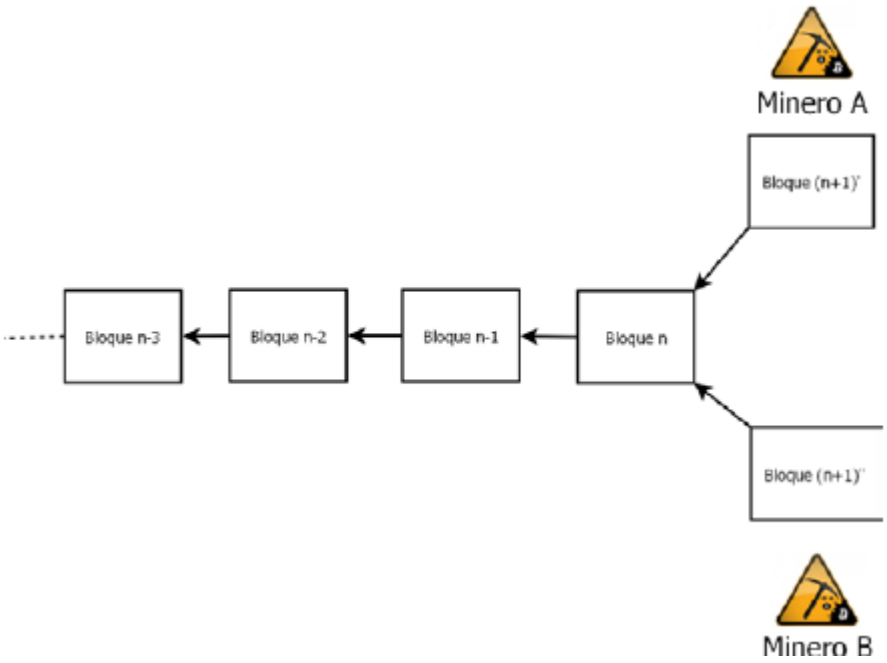

Figura 12. Creación de ramas alternativas en la cadena de bloques.

Al generarse dos ramas distintas, cada una de ellas será respaldada inicialmente por una cantidad determinada de mineros, que irán extendiéndola. Cuanto más similares sean las capacidades de cómputo de las ramas, más se tardará en resolver la ambigüedad, aunque eventualmente una de las ramas recibirá un nuevo bloque antes que la otra y prevalecerá sobre ella, como se muestra en la Figura 13.

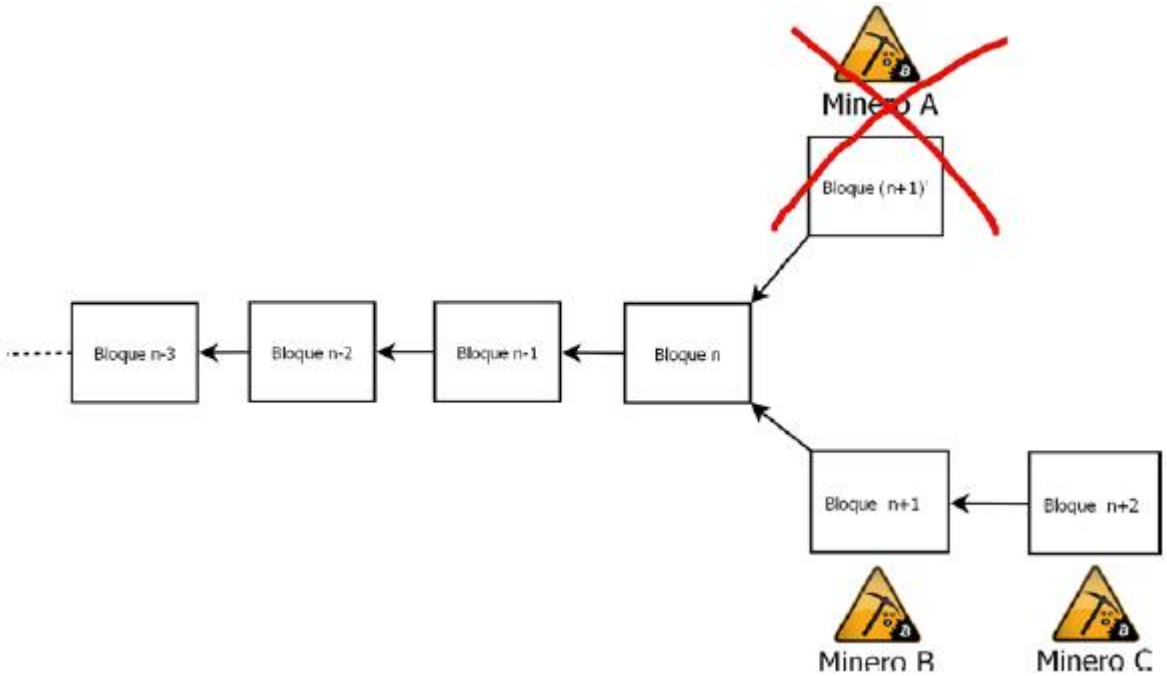

Figura 13. Resolución de ramas alternativas.

No obstante, esto es un caso posible y dar por válida una transacción no respaldada por nuevos bloques no es buena idea. Por ello, es aconsejable esperar un número determinado de bloques hasta considerar una transacción como confirmada. El número de bloques puede variar dependiendo de la cantidad involucrada en la transacción, y obviamente, en función de las consideraciones personales. Normalmente, se considera que tras 6 bloques nuevos, la transacción será difícilmente revertida y, por tanto, se puede considerar confirmada. Nótese que la probabilidad de revertir una transacción decrece exponencialmente por cada nuevo bloque que la respalda.

4. Cibermercados negros en la red TOR

En este capítulo se explica el concepto de Darknet y las características principales de los cibermercados negros. A continuación se muestra una comparativa de los cibermercados negros más populares en la red TOR. Finalmente, se explica a modo de ejemplo, el funcionamiento de uno de ellos.

4.1 Darknet

El término Darknet no tiene una definición universalmente aceptada. Sin embargo, se podría decir que la Darknet es una colección de redes y tecnologías usadas para compartir información y contenidos digitales que está "distribuida" entre los distintos nodos y tratan de preservar el anonimato de quienes intercambian dicha información, es decir, persiguen el anonimato del origen y el destino cuando se produce la transferencia de información. En la definición anterior, cuando se habla de redes, no se refiere a redes físicas separadas de las redes actuales sino a redes superpuestas que pueden usar protocolos y puertos "no estándar" sobre la red subyacente. Por eso se dice que estas redes operan aparte de las redes públicas sobre las que se montan y que sus contenidos se mantienen inalcanzables para el público general de la red subyacente. Para acceder a sus contenidos es necesaria cierta información adicional, la cual puede ser compartida por un grupo restringido de personas. Esa información suele incluir la necesidad de ejecución de un software específico y a veces es necesaria la conexión a algún tipo de servidor que no estará accesible vía los DNS tradicionales. Por esta dificultad de acceso los motores de búsqueda no suelen buscar en estas redes, permaneciendo sus contenidos invisibles.

Se tiende a confundir los términos Deep web (o internet profunda) y Darknet, pensando que son términos sinónimos e intercambiables entre sí. Así pues, la Deep web se trata de la parte de la web que no está indexada por los motores de búsqueda convencionales como Google, Bing o Yahoo. Un ejemplo, serían las páginas de acceso restringido (páginas protegidas con contraseña, contenido protegido por un Captcha, etc).

Principalmente hay dos tipos de Darknet, las peer-to-peer, por ejemplo las construidas con Freenet [38], i2p [39], GNUnet [40], Entropy [41], ANts P2P [42] y las que no son peer-to-peer por ejemplo TOR. La más popular de todas es TOR, una red de anonimización que tiene también su propia Darknet, es la que suele referirse todo el mundo cuando habla de ellas y es sobre la que vamos a tratar en este proyecto.

4.2 Cibermercados negros en las Darknet

En las Darknet existen diversos mercados negros para comprar cualquier tipo de bien o servicio ilegal. Este hecho se debe a las posibilidades de anonimato que ofrecen estas redes y por consiguiente a la impunidad que este medio proporciona. Hay que reseñar que no todo el que emplea algún medio de anonimización es, por si, alguien que está en disposición de cometer un

ilícito penal, sino que, en ocasiones, se tratan de usuarios de la red que tratan de preservar que se conozca su identidad, o bien puede ser empleado para saltar los filtros de acceso de los ISPs nacionales en países que tratan de someter censura a la red.

Los mercados negros en las darknet se sostienen principalmente sobre los siguientes cuatro pilares:

1. PGP

   Es un criptosistema cuya finalidad es proteger la información distribuida a través de Internet mediante el uso de criptografía de clave pública, así como facilitar la autenticación de documentos gracias a firmas digitales.

   Concretamente en los mercados negros se utiliza para cifrar/descifrar mensajes confidenciales enviados entre el vendedor y el comprador. Por ejemplo, si el comprador quiere enviar al vendedor la dirección de entrega del producto, el comprador cifrará el mensaje utilizando la clave pública del vendedor para que únicamente el vendedor tenga acceso a la dirección.

2. Criptomoneda

   Bitcoin es la criptomoneda actualmente más utilizada en los cibermercados negros. Las direcciones donde los usuarios reciben las transferencias son anónimas, aunque las transacciones son públicas. Por este motivo existen servicios como los mezcladores de bitcoins [43], por ejemplo BitBlender [44] que mezclan los fondos de distintos usuarios con el fin de dificultar la trazabilidad del flujo de las transacciones y en consecuencia la identidad de los usuarios.

3. Sistemas de reputación

   Al igual que en los mercados online populares como eBay, la reputación como vendedor es lo que da confianza en los cibermercados negros. Las valoraciones se pueden encontrar en las propias tiendas, en foros de la deep web e incluso en Reddit. Los usuarios advierten a otros de las estafas y dejan comentarios completos de los vendedores fiables y de sus productos.

4. Escrow y Multisig Escrow

   El Escrow sirve para proteger contra las estafas. La mayor parte de los cibermercados negros ofrecen este servicio. El dinero queda depositado en manos de los administradores de la tienda durante el proceso de compra, y no pasa al vendedor hasta que el producto se envía al comprador. Las tiendas cobran un porcentaje, alrededor de un 0,5%, por utilizarlo. Además muchas también ofrecen un sistema de disputas por si hay algún problema.

   El Multi-Signature Escrow (o Multisig) es una alternativa en la que el dinero del comprador está retenido en una dirección de Bitcoin firmada tanto por el comprador como por el vendedor. De esta forma, son los involucrados los que arbitran sobre el dinero.

4.3 Comparativa de cibermercados negros en TOR

Como se ha explicado en el apartado anterior, la red TOR es parte de la Darknet y en ella también existen diversos cibermercados negros.

En este apartado se muestra el estudio que se ha realizado sobre los diez cibermercados negros más populares de la red TOR con el objetivo de realizar una comparación de los aspectos más relevantes. Las características utilizadas en la comparación son las siguientes:

- Fecha de creación
- Registro abierto
- 2FA (Autenticación en dos factores)
- PGP Obligatorio del Vendedor
- Scrow
- Multisig
- Comisión
- Fianza del vendedor
- Moneda aceptada

A continuación, en la tabla 4 se muestran las características de los cibermercados analizados.

Nombre Cibermercado	Fecha creación	Registro abierto	2FA	PGP Obligatorio Vendedor	Scrow	Multisig	Comisión	Fianza vendedor	Moneda aceptada
AERO Market	01/10/2017	Si	Si	Si	Si	Si	2-5%	100-200	Bitcoin/Monero
Berlusconi Market	07/07/2017	Si	Si	No	Si	No	2%	Gratis – 250$	Bitcoin/Litecoin
CGMC	07/06/2016	Si	Si	Si	No	Si	2% - 3%	Gratis	Bitcoin
Dream Market	15/11/2013	Si	Si	No	Si	No	4%	300$	Bitcoin
Libertas Market	14/10/2017	Si	Si	Si	Si	Si	3%	Por invitación	Monero
RsClub Market	11/07/2016	Si	Si	Si	Si	No	4%	70$	Bitcoin
Sourcery Market	18/07/2017	Si	Si	Si	Si	Si	2%-4%	50$	Bitcoin
The Majestic Garden	-	Si	No	No	No	No	No	Gratis	Bitcoin/Otras
Tochka	30/01/2015	Si	Si	Si	Si	Si	2% - 10%	Gratis - 100$	Bitcoin/Ethereum
Wall Street Market	19/10/2016	Si	Si	Si	Si	Si	2% -5%	Gratis – 80$	Bitcoin/Monero

Tabla 4. Comparativa cibermercados negros en TOR.

A continuación, se explica el resultado de la comparativa realizada para cada una de las características analizadas (ver Tabla 4):

- Fecha de creación: La mayoría de los mercados se han creado en los últimos dos años, exceptuando el mercado “Tochka” que se creó a principios de 2015 y el mercado “Dream Market” que se creó a finales de 2013. Respecto al mercado “The Majestic Garden” no se ha encontrado su fecha de creación.
- Registro abierto: Todos los mercados tienen el registro abierto a cualquier usuario. Aunque en el mercado “CGMC” es necesario esperarse 48 horas para obtener un código de invitación para realizar el registro.
- 2FA: El único mercado que no tiene autenticación de dos factores (2FA) es “The "Majestic Garden”. Ya que este mercado no se trata de un mercado como tal, sino que es un foro que su función principal es poner en contacto los vendedores con los compradores y no se realizan las ventas en esta plataforma, por este motivo no es tan necesario este requerimiento de seguridad.
- PGP obligatorio del vendedor: Como medida de seguridad adicional, la mayor parte de los mercados obligan a los vendedores a usar PGP para comunicarse con otros usuarios. Excepto los mercados “Berlusconi Market”, “The Majestic Garden” y “Dream Market” que su uso es opcional.
- Scrow: Todos los mercados ofrecen el servicio de Scrow excepto el mercado “The Majestic Garden” y “CGMC”. En el caso de “The Majestic Garden” el motivo por lo que no ofrece este servicio es el mismo que lo explicado en el requerimiento de seguridad 2FA, al no realizarse las ventas en esta plataforma, no tiene sentido ofrecer el servicio.
- Multisig: Seis de los diez mercados ofrecen el servicio de Multisig. De los cuatro mercados que no ofrecen este servicio uno de ellos, el mercado “The Majestic Garden” es debido a que no se realizan las ventas en esta plataforma.
- Comisiones: Las comisiones que se quedan los propietarios de los mercados sobre la venta de los productos oscilan entre el 2% y el 5%, exceptuando el mercado “Tochka” que su comisión pueden llegar hasta el 10% y el mercado “The Majestic Garden” que no tiene comisiones pero permite ofrecer donaciones. Estas comisiones pueden ser incluso variables dentro del mismo mercado. Este hecho se debe a que existen diferentes planes para los vendedores.
- Fianza: La fianza que tienen que pagar los nuevos vendedores para poder vender sus productos en el mercado puede llegar hasta los 300$ aunque en algunos mercados no hay fianza o los vendedores se la pueden ahorrar a través de invitaciones. En la mayoría de los mercados la fianza depende del plan de vendedor escogido. Por ejemplo, en el mercado “Tochka”, existen tres planes. El plan free, las comisiones que se queda la plataforma en cada venta es del 10% pero el vendedor no tiene que depositar ninguna fianza para poder empezar a vender en ella. El plan Premium, las comisiones son del 5% pero se necesita

depositar una fianza de 50$. Y en el plan Premium+, la fianza es de 100$ y las comisiones solo son del 2%.

- Moneda aceptada: Todos los cibermercados aceptan el pago con Bitcoin's a excepción del mercado "Libertas Market" que solo acepta la criptomoneda Monero. Además Monero es aceptada por los mercados "AERO Market" y "Wall Street Market". El mercado "Berlusconi Market" acepta también el pago con Litecoin's y el mercado "Tochka" con Ethereum's. El mercado "The Majestic Garden", como ya hemos comentado anteriormente, al no realizarse los pagos a través de la plataforma, el vendedor y el comprador pueden acordar el pago con cualquier criptomoneda. Como se puede ver en la comparativa, Bitcoin es la criptomoneda más utilizada en los cibermercados analizados y la segunda es Monero, debido a las características de anonimato que ofrece.

4.3.1 Cibermercado negro: Dream Market

Actualmente, uno de los principales cibermercados de la Darknet que opera en un servicio oculto de la red TOR y que cumple los cuatro pilares explicados en el apartado anterior es "Dream Market", fundado a finales de 2013. El mercado vende una gran variedad de contenido a través de la criptomoneda Bitcoin, que incluye drogas, medicamentos, datos robados, productos falsificados, etc. A continuación, se enumeran las direcciones disponibles para acceder a este cibermercado negro.

1. http://lchudifyeqm4ldjj.onion/?ai=1675
2. http://jd6yhuwcivehvdt4.onion/?ai=1675
3. http://t3e6ly3uoif4zcw2.onion/?ai=1675
4. http://7ep7acrkunzdcw3l.onion/?ai=1675
5. http://vilpaqbrnvizecjo.onion/?ai=1675
6. http://igyifrhnvxq33sy5.onion/?ai=1675

Para poder entrar al cibermercado es necesario registrarse. La información requerida en el formulario de registro es la siguiente: nombre de usuario, contraseña y PIN. El PIN sirve para poder recuperar la cuenta y parar retirar Bitcoins. Opcionalmente el PIN también sirve para activar una contraseña de seguridad adicional (*security password*) que se solicita al retirar bitcoins y al comprar o vender productos. Una vez realizado el registro, ya es posible acceder al mercado con la cuenta creada.

En la figura 14 se muestra una captura de la página principal de "Dream Market".

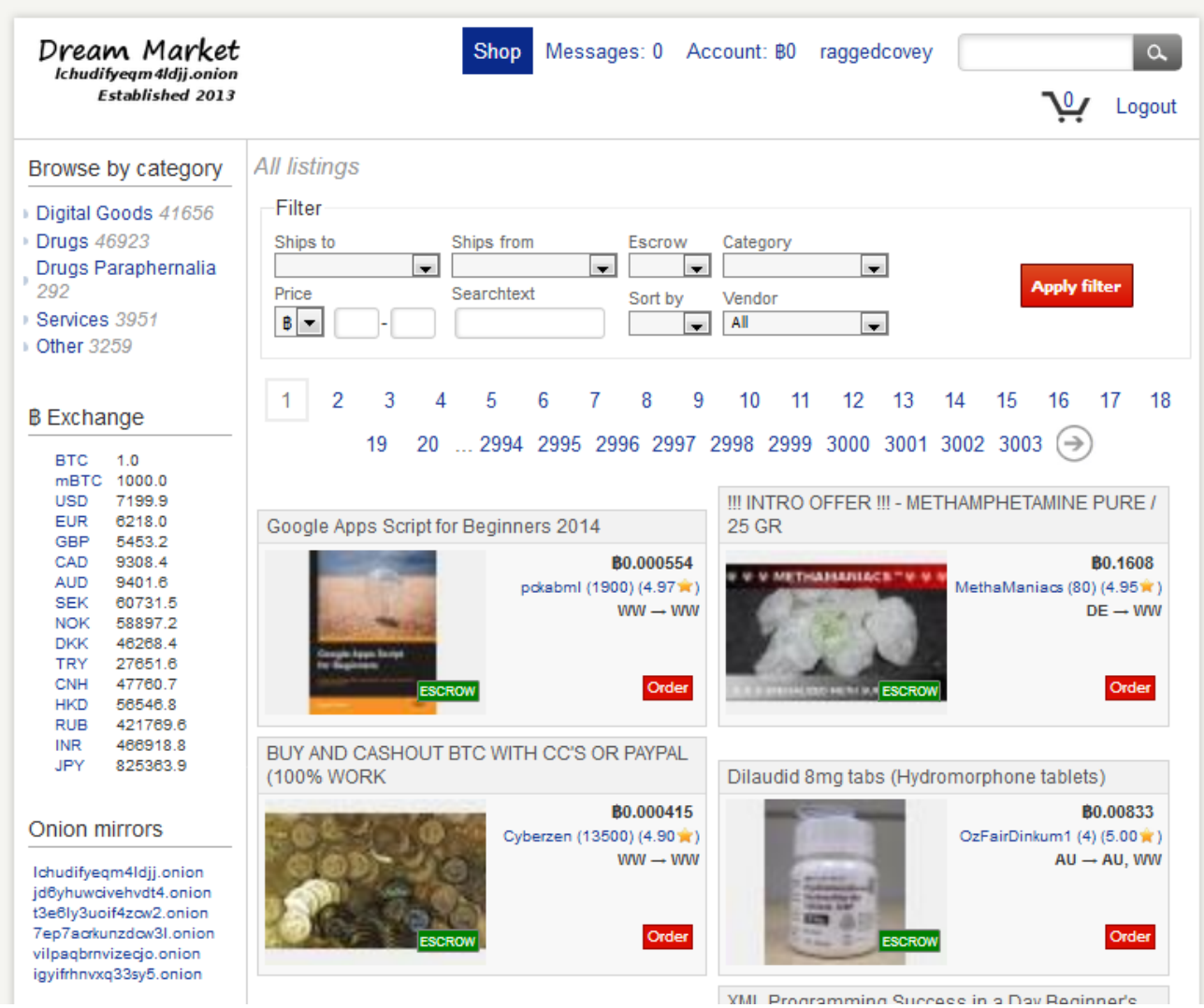

Figura 14. Página principal de "Dream Market"

Tal como se muestra en la captura, en la parte superior izquierda se puede navegar por las diferentes categorías y subcategorías de los productos que se ofrecen. Las categorías principales son las siguientes:

- Productos digitales.
- Drogas.
- Accesorios para droga.
- Servicios.
- Otros.

En la parte superior central se puede realizar diferentes filtros para precisar la búsqueda. Como por ejemplo, limitar el país de origen o destino de los productos, si estos tienen el servicio de scrow, etc.

En la parte central aparecen los diferentes productos. En cada producto se puede observar el nombre del producto, una foto, el precio (por defecto en Bitcoins), el pseudónimo del vendedor, su

reputación, el país de origen y destino del envío del producto y si se permite scrow o no. En la figura 15 se muestra como ejemplo un producto concreto.

Figura 15. Producto de cannabis de "Dream Market"

Donde se puede observar que se trata de un producto de cannabis con el nombre de "1g Amnesia Haze", se vende por 0.0021 Bitcoins, el pseudónimo del vendedor es "CannabisClub" y su reputación es de 4.94 sobre 5. El producto se envía desde Alemania (Deutschland, DE) hacia cualquier país de la Unión Europea (European Union, EU).

En cualquier producto de la lista, pulsando sobre el nombre del producto se abre una página con la descripción detallada del producto, los términos y condiciones del vendedor, la opción de poder comprar el producto y las valoraciones de los distintos compradores que ya lo han comprado anteriormente.

Por otro lado, pulsando en el pseudónimo de un vendedor se abre una página con la información de contacto del vendedor, los productos que vende y un formulario que permite enviarle un mensaje. También, pulsando en el valor de su reputación se obtiene el historial de valoraciones y comentarios de sus vendas anteriores.

5. Métodos de desanonimización en la red TOR

En este capítulo se explican métodos de desanonimización de usuarios en la red TOR basados en técnicas de *Fingerprinting* y de correlación.

5.1 Ataques de Fingerprinting

El objetivo principal de un ataque de *Website Fingerprinting* (WF) es identificar qué páginas está visitando un usuario. El ataque de WF generalmente se trata como un problema de clasificación, donde las categorías de clasificación son páginas web y las observaciones son trazas de tráfico.

El atacante primero recopila rastros de tráfico al visitar páginas web y entrena con un clasificador supervisado usando características tales como el tamaño, la dirección y los tiempos entre llegadas de paquetes de red. En el caso de ataques sobre la red TOR, cada vez que un usuario visita una página web sobre TOR, el atacante registra la traza de la red, ya sea interceptando el tráfico localmente (LAN), teniendo acceso a los enrutadores del ISP del usuario o al controlar un *Entry guard* de la red TOR. Finalmente, el atacante ejecuta el clasificador para intentar adivinar los sitios web que el usuario ha visitado.

Los primeros ataques de WF se desarrollaron para identificar páginas dentro de un único sitio web a través de conexiones SSL [45] [46]. En 2002, Sun et al. abordó el problema más difícil de identificar páginas individuales dentro de un conjunto de sitios web [47] que condujo al ataque de Hintz de un proxy web anónimo (SafeWeb) [48].

En 2009, Herrmann et al. [49] desplegó el primer ataque de WF en la red TOR con solo una tasa de éxito del 3% para un mundo de 775 páginas. Los ataques que siguieron, mejoraron significativamente la precisión: Shi y Matsuura obtuvieron una tasa de éxito del 50% para 20 páginas [50]; Panchenko et al. obtuvieron un 54.61% de precisión utilizando el conjunto de datos de Herrmann [51]; y, finalmente, Cai et al. y Wang y Goldberg informan tasas de éxito superiores al 90% utilizando clasificadores basados en distancia de edición en un mundo de 100 páginas [52] [53]. Aunque en estos trabajos se demuestra la efectividad de los ataques de WF en TOR a pesar del cifrado, el relleno y las defensas de nivel de aplicación, como la canalización aleatoria. Se hicieron suposiciones que simplificaron enormemente el problema y le otorgaron ventajas irreales al atacante al sobrestimar sus capacidades o al simplificar el mundo. Por ejemplo, la mayoría de los trabajos actuales suponen de manera implícita o explícita que el adversario y el usuario usan el mismo Tor Browser Bundle (TBB), visitan la misma versión de una página web dentro de un conjunto limitado de páginas casi al mismo tiempo (o con unos pocos días de diferencia) utilizando solo una pestaña de navegación. Sin embargo, violar al menos una de estas suposiciones puede reducir la eficacia de estos ataques significativamente.

5.1.1 El modelo del ataque

En un ataque de WF el atacante es pasivo y local, es decir, el adversario puede espiar el tráfico del usuario, pero no puede agregar, eliminar o modificar paquetes. También suponemos que el adversario no puede descifrar el contenido de los paquetes de red, ya que eso haría innecesario un ataque de WF.

Los ataques de WF se pueden clasificar en dos categorías dependiendo de la cantidad de usuarios a los que apunta el atacante:

- Dirigido: En este ataque, el atacante se dirige a una víctima específica para recuperar su actividad de navegación. Esto le permite al atacante entrenar a un clasificador en condiciones similares a las de la víctima, lo que posiblemente incremente el éxito del ataque. El atacante puede tener suficiente conocimiento previo sobre el usuario para reproducir su configuración, o podría detectarlo a partir de los datos de tráfico observados.
- No dirigido: en este caso, el atacante se dirige a un conjunto de usuarios en lugar de a uno. El atacante entrena al clasificador en una configuración específica y usa el mismo clasificador en todas las comunicaciones que observa. Por ejemplo, los ISP podrían desplegar un ataque no dirigido al observar el tráfico de red de muchos usuarios.

El coste de un atacante para mantener un sistema de WF típico requiere las siguientes cuatro tareas:

- Recopilación de datos: al principio, el atacante necesita recopilar datos de las páginas de entrenamiento. Para maximizar la precisión del clasificador, el atacante puede querer entrenar con diferentes versiones localizadas de la misma página web y recopilarlas bajo diferentes configuraciones, por ejemplo, diferentes versiones de TBB, configuraciones de usuario, configuraciones de guardias de entrada.
- Entrenamiento: en la fase de entrenamiento, el atacante necesita entrenar a su clasificador con los datos recopilados.
- Prueba: para probar un rastro, el adversario necesita recolectar los datos de prueba, extraer las características de los datos recolectados y probarlas usando el clasificador.
- Actualización: para poder mantener el rendimiento del clasificador, el adversario necesita actualizar el sistema a lo largo del tiempo.

5.1.2 Variables que afectan a la precisión

Juárez et al. [54] en 2014 criticó varios ataques de WF porque hicieron suposiciones que eran demasiado ventajosas para el adversario sobrevalorando la efectividad de los ataques. Debido a que el éxito de un ataque de WF depende de muchos factores, como la proximidad temporal de los rastros de entrenamiento y prueba, las versiones de TBB utilizadas para el entrenamiento y las pruebas, y los hábitos de navegación de los usuarios, que suelen simplificarse demasiado en los modelos de WF. Por lo tanto, en la mayoría de los casos, un ataque no dirigido no es factible.

En esta sección, mostramos estas suposiciones en forma de variables que afectan a la precisión en los ataques de WF dividiéndolas en tres grupos, dependiendo de si están en el lado del cliente, en él del atacante o en él de las páginas web.

Las variables que afectan a la precisión en el cliente son:

- Mundo cerrado/abierto: En un entorno de mundo cerrado, el atacante supone que los sitios web visitados por el cliente se encuentran entre una lista de k sitios web conocidos, y el objetivo del atacante es identificar cuál. La configuración de mundo abierto es más realista, en el sentido de que se asume que el cliente visitará un conjunto mayor de sitios web n, y el objetivo del atacante es identificar si el cliente está visitando un sitio web supervisado de una lista de k sitios web. La suposición de un mundo cerrado, es muy grande porque k es siempre muy pequeña en comparación con la cantidad real de páginas web existentes.
- Comportamiento de navegación: Se presupone que el comportamiento de navegación de los usuarios es de forma secuencial, accediendo una página tras otra y teniendo solo una sola pestaña abierta. Es seguro pensar que este no es el comportamiento habitual de navegación de los usuarios de TOR.

Las variables que afectan a la precisión en el atacante son:

- Análisis de carga de páginas: Se presupone que un atacante puede detectar el inicio y el final de diferentes cargas de páginas en una traza de tráfico. Aunque se ha demostrado que la reconstrucción de sesiones del tráfico de red en el mundo real es una tarea muy difícil [55].
- Tráfico de fondo: Se presupone que un atacante puede filtrar todo el tráfico de red de fondo producido por otras aplicaciones u otras conexiones que pasen por el mismo circuito de TOR. Por el contrario, TOR se utiliza cada vez más en entornos donde se envían múltiples aplicaciones o el tráfico completo del sistema operativo a través de la red TOR y en estos casos, separar el tráfico de navegación del tráfico de fondo no es trivial para el atacante.
- Replicabilidad: Se presupone que un atacante puede entrenar a su clasificador bajo las mismas condiciones que la víctima. Por ejemplo, se supone que el adversario puede replicar las configuraciones del lado del cliente, como el sistema operativo, la conexión de red o la versión de Tor Browser Bundle (TBB). Esta suposición permite a los investigadores entrenar y evaluar los datos recopilados utilizando la misma configuración. Dependiendo del tipo de ataque, esto puede ser imposible, ya que el atacante puede encontrar dificultades para detectar y replicar la configuración de los usuarios (especialmente en ataques no dirigidos).

La variable que afecta a la precisión en las páginas web es:

- Variabilidad de las páginas web: Las páginas web cambian constantemente su contenido. Este hecho afecta en la precisión del clasificador y plantea un desafío al atacante que deberá entrenar de forma regular el clasificador.

5.1.3 Ataque de Fingerprinting de circuitos

Albert Kwon et al. [56], en 2015 publicaron un ataque de *fingerprinting* de circuitos que está centrado en desanonimizar operadores de servicios ocultos (HS) y sus usuarios a través de un *Entry guard* controlado por el atacante.

Para conseguir detectar la presencia de actividad de HS en los clientes y operadores de servicios ocultos, el atacante recopila el tráfico de red que pasa por el *Entry guard* que controla e intenta identificar qué tipo de circuito está asociado a cada parte del tráfico. La identificación es posible debido a que durante la fase de construcción y comunicación de circuitos entre un cliente y un servicio oculto, TOR exhibe patrones de tráfico que permiten al adversario identificar y correlar de manera eficiente y precisa los circuitos involucrados en la comunicación con servicios ocultos.

Se pueden clasificar los tipos de circuitos de la red TOR en cinco clases diferentes, que son:

- Cliente-IP: circuito entre un cliente y el punto de entrada (*Introduction Point*, IP) de un HS.
- Cliente-RP: circuito entre un cliente y uno de sus puntos de encuentro (*Rendezvous Point*, RP).
- HS-IP: circuito entre un HS y su punto de entrada (*Introduction Point*, IP).
- HS-RP: circuito entre un HS y un punto de encuentro (*Rendezvous Point*, RP) de un cliente.
- Circuitos generales: circuito entre un cliente y un servidor no oculto (servicios ubicados fuera de la red TOR).

Para identificar estos circuitos se utilizan las siguientes características:

- Celdas entrantes y salientes.
- Duración de la actividad.
- Secuencias de construcción del circuito.

Las diferentes características son más indicativas en ciertos tipos de circuitos. Para aprovechar mejor esas características, se realiza la clasificación en dos pasos. Primero, el adversario busca circuitos Cliente-IP y HS-IP, ya que estos son los más fáciles de clasificar. Esto también le permite al adversario averiguar si está monitorizando HS o un cliente de un HS. En el segundo paso, el adversario examina los circuitos que no son IP para encontrar circuitos RP entre ellos.

En el ataque publicado se pudo identificar la presencia de HS con unas tasas de más del 98% de verdaderos positivos y menos de 0,1% de falsos positivos.

Una vez detectada la actividad de HS, en lugar de monitorizar cada circuito, que puede ser costoso, el primer paso en la estrategia del atacante es identificar los circuitos sospechosos para reducir el espacio del problema a solo los servicios ocultos. Finalmente, el atacante debe aplicar un ataque de WF para identificar a los clientes y servidores de HS.

En el ataque publicado se pudo desanonimizar correctamente en un mundo abierto de 50 servidores de HS monitorizados con una tasas de verdadero positivo del 88 % y de falsos positivos del 7,8%.

A continuación, se explica porque el mundo de los HS es un buen escenario para poder realizar ataques de WF debido a que no se ven afectados por la mayoría de las variables que afectan a la precisión en los ataques de WF de páginas web generales (ver apartado 4.1.2):

- Mundo cerrado/abierto: En el caso de los servicios ocultos, el tamaño del mundo es significativamente menor que todo Internet. Además, si bien es cierto que no todos los servicios ocultos existentes están a disposición del público, se ha demostrado que la enumeración de servicios ocultos es posible [57].
- Comportamiento de navegación: Este ataque usa características que se basan en interacciones de circuitos y son independientes de los hábitos de navegación o ubicaciones de los usuarios.
- Análisis de carga de páginas: La suposición previa de que el atacante puede distinguir diferentes cargas de páginas sigue siendo válida. Es probable que los "tiempos de reflexión" del usuario dominen la sesión de navegación y creen brechas de tiempo notables entre las celdas.
- Tráfico de fondo: Los ataques previos de WF suponían que el atacante podía eliminar el tráfico de fondo. En el mundo de los servicios ocultos, observamos que TOR usa circuitos separados para diferentes dominios .onion. Además, TOR no multiplexa flujos generales que acceden a servicios generales no ocultos con flujos que acceden a servicios ocultos en el mismo circuito. Desde la perspectiva del atacante, esta es una gran ventaja ya que simplifica el análisis de tráfico.
- Variabilidad de las páginas web: El contenido de la web general cambia muy rápidamente. Sin embargo, las páginas de servicios ocultos muestran cambios mínimos a lo largo del tiempo, al contrario que las páginas de servicios no ocultos. La naturaleza lentamente cambiante de los servicios ocultos reduce los falsos positivos y negativos del atacante y minimiza el coste del entrenamiento. Además, los servicios ocultos no sirven versiones localizadas de sus páginas.

La única variable que afecta a la precisión en los ataques de WF y que se conserva también en este ataque, es la replicabilidad.

- Replicabilidad: En este ataque se asume que se puede entrenar al clasificador con las mismas condiciones que la víctima. Debido a que es difícil detectar y replicar la configuración de los usuarios.

5.2 Ataques de correlación

El objetivo principal de un ataque de correlación en una red de anonimato es identificar los usuarios relacionando los flujos de tráfico entre la entrada y la salida de la red. En el caso de la red TOR debido a que el tráfico de entrada va cifrado, este no puede ser correlar directamente con el tráfico de salida de la red, por lo que es necesario realizar antes un *fingerprinting* para obtener información útil que se pueda utilizar en el ataque de correlación, como por ejemplo, las longitudes y direcciones de los paquetes.

El modelo de amenaza de TOR excluye a los atacantes globales [58], pero la amenaza práctica de estos atacantes es una cuestión importante que la comunidad investigadora ha dedicado un esfuerzo considerable para responder. En 2004, cuando la red TOR comprendía solo 33 nodos, Feamster y Dingledine investigaron la amenaza práctica que los adversarios a nivel de Sistemas Autónomos (Autonomous Systems, AS) representan para las redes de anonimato [59]. Los autores consideraron un ataque que controlaba un AS que atravesaba tanto el tráfico de entrada cómo de salida, lo que permitía al atacante correlar ambos flujos. Utilizando una predicción de rutas de AS [60], Feamster y Dingledine descubrieron que los poderosos ISPs de nivel 1 reducen la diversidad de ubicaciones de las redes de anonimato. En 2007, Murdoch y Zielinski consideraron los atacantes a nivel de punto de intercambio de Internet (Internet Exchange Point, IXP), una clase de atacantes no incluidos en el trabajo de Feamster y Dingledine [61]. Murdoch y Zielinski demostraron que los atacantes del IXP pueden correlar las trazas de tráfico, incluso en presencia de tasas de muestreo de paquetes tan bajas como 1 entre 2.000.

En 2013, Johnson et al. [62] presentó el primer estudio a gran escala sobre el riesgo de los usuarios de TOR que se enfrentan a atacantes a nivel de nodos y de AS. Los autores desarrollaron TorPS [63] que simula los circuitos de TOR para una serie de modelos de usuario. Al combinar TorPS con la predicción de las rutas de los AS, Johnson et al. respondieron preguntas tales como el tiempo promedio hasta que un circuito de usuario de TOR sea desanonimizado por un AS o un IXP. Más recientemente, en 2016, Nithyanand et al. [64] utilizó la predicción de las rutas de AS para evaluar la amenaza práctica a la que se enfrentan los usuarios en los diez países principales que usan TOR. En 2015, Juen et al. [65] examinó la precisión de los algoritmos de predicción de rutas que los trabajos anteriores utilizaron para estimar la amenaza de los ataques de correlación. Los autores compararon las predicciones de trayectorias de AS con millones de traceroutes, iniciados a partir del 25% de los nodos de TOR por ancho de banda en el nivel de AS, y descubrieron que solo el 20% de las trayectorias predichas coincidían con las trayectorias observadas en traceroutes. En 2015, Sun et al. [66] mostró que la naturaleza dinámica del enrutamiento de Internet hace que los atacantes del nivel de AS sean más fuertes de lo que se habían considerado en trabajos previos.

5.2.1 Técnicas de correlación

En los ataques de correlación, un atacante observa dos partes de flujos de paquetes de una red e intenta verificar si pertenecen al mismo flujo dentro de la red. Estos flujos de paquetes están cifrados, es decir, no pueden ser comparados directamente y el atacante debe intentar relacionarlos usando otra información disponible como:

- Contar paquetes: un atacante puede contar el número de paquetes que entran y salen del primer nodo para determinar el siguiente nodo en el circuito. El procedimiento es posteriormente repetido en otros nodos del circuito hasta que se determina el destinatario. Aunque esta manera de contar paquetes es relativamente sencilla, requiere que el atacante sea capaz de observar una gran cantidad de la red, y asume que nunca hay una variación en el número de paquetes entrantes y salientes de un nodo en un flujo concreto. Actualmente, el conteo de paquetes ha sido reemplazado por técnicas más sofisticadas de correlación de flujos basadas en la sincronización de paquetes.
- Análisis de sincronización de paquetes: un atacante puede utilizar los datos de la sincronización de paquetes para correlar flujos de red a través del retraso entre paquetes, es decir, el tiempo entre la llegada de paquetes adyacentes al flujo. Sin embargo, este enfoque puede tener problemas con los paquetes descartados.
- Correlación activa de sincronización: los ataques de correlación activa intentan hacer las correlaciones basadas en el tiempo mucho más fáciles y efectivas. Funcionan al tener un nodo atacante alterando la señal de retraso de un paquete de una conexión al soltar o retrasar paquetes en un flujo. Los ataques activos de sincronización son factibles y efectivos contra protocolos altamente interactivos como VoIP.

5.2.2 Ataques de correlación con el tráfico DNS (DefecTor)

Benjamin Greschbach et al. [67], en 2017 publicaron un conjunto de ataques de correlación de tráfico DNS (DefecTor) para desanonimizar usuarios de la red TOR. En este trabajo, consideraron atacantes que pueden observar el tráfico de entrada a la red TOR y algunas solicitudes de DNS que salen de la red. Con el objetivo de correlar los sitios web observados a partir de técnicas de *fingerprinting* en el tráfico de entrada a la red TOR con los sitios web identificados a partir del tráfico DNS.

En este estudio mostraron que un atacante que puede montar un ataque DefecTor a menudo puede determinar con una alta probabilidad el sitio web que un usuario TOR está visitando, particularmente para sitios web menos populares donde el conjunto de nombres DNS asociados con ese sitio web puede ser exclusivo del sitio.

Para observar el tráfico de entrada, el atacante puede operar a nivel de red (como un ISP malicioso o una agencia de inteligencia) u operar a nivel de retransmisor (controlando un nodo de entrada a la red TOR). En ambos casos, el atacante solo puede observar datos cifrados, por lo que es necesario realizar antes un *fingerprinting* para obtener información útil que se pueda utilizar en el ataque de correlación, como por ejemplo, las longitudes y direcciones de los paquetes.

En el caso del tráfico de salida, el atacante debe observar el tráfico DNS ya sea estando dentro de la ruta entre un nodo de salida de TOR y un servidor DNS o controlando un servidor DNS malicioso.

Por otra parte, en este trabajo desarrollaron también un método para identificar los resolvedores de DNS que utilizan los nodos de salida de la red TOR y encontraron que existen entidades globales que podrían montar ataques DefecTor. Cómo por ejemplo, encontraron que los servidores públicos de DNS de Google observan una tercera parte de todas las solicitudes de DNS que salen de la red TOR, llegando a máximos del 40%. En segundo lugar encontraron que los nodos de salida "Local" que ejecutan su propio sistema de resolución, tienen un promedio del 12%. También encontraron que las solicitudes DNS a menudo atraviesan ASes que las conexiones TCP correspondientes no transitan, lo que permite que ASes adicionales obtengan información sobre el tráfico de los usuarios de TOR.

Para conocer qué solicitudes de DNS puede ver un atacante, se debe considerar el almacenamiento en caché de las respuestas de DNS de los nodos de salida, ya que almacenan en caché las solicitudes de DNS y los clientes de TOR que usan el mismo nodo de salida comparten la caché. Si un cliente de TOR intenta resolver un dominio que un nodo de salida ha almacenado en caché, el atacante no podrá observar esta solicitud. Sin embargo, el atacante puede registrar todas las solicitudes de DNS observadas en los últimos x segundos, donde x es el valor máximo de TTL (es decir, mantener una ventana deslizante de longitud x). Si un cliente TOR está intentando resolver un nombre de dominio, la solicitud puede estar en caché o no. Si no está en caché, el atacante la verá como una nueva solicitud de DNS saliente desde el nodo de salida. Si está en caché, debe haber sido resuelta por el nodo de salida en los últimos x segundos, y por lo tanto estará en la ventana deslizante. La técnica de ventana deslizante permite al atacante capturar todas las solicitudes de DNS relevantes, independientemente de si están en caché o no.

En esta publicación los autores recomiendan a corto plazo evitar que los nodos de salida utilicen los resolvedores de DNS públicos, como Google y OpenDNS. En su lugar, deberían usar los resolvedores proporcionados por su ISP, o ejecutar los suyos propios, particularmente si el ISP del operador ya alberga muchos otros nodos de salida. Los resolvedores locales también pueden configurarse aún más para minimizar la fuga de información, al habilitar la minimización de QNAME [68].

Como soluciones a largo plazo, recomiendan utilizar T-DNS [69], que emplea varias optimizaciones TCP para transportar el protocolo DNS a través de TLS y TCP e implementar defensas contra los ataques de *fingerprinting* de sitios web en TOR.

6. Conclusiones

Los principales objetivos de este trabajo fueron determinar el funcionamiento y las características de los cibermercados negros en la red TOR e investigar los principales métodos de desanonimización en la red TOR.

Para ello primero se estudió el funcionamiento de la red TOR y la red Bitcoin. Una vez realizados estos estudios, se investigaron las características principales de los cibermercados negros de la red TOR y a continuación se utilizaron estas características para elaborar una comparativa de los cibermercados negros más populares. En esta comparativa se puede observar que existe una homogeneización en los servicios que ofrecen. La mayoría ofrecen los servicios de comunicación cliente-vendedor con PGP, los servicios de Scrow y Multisig para proteger de posibles estafas a los compradores, la moneda más aceptada es Bitcoin y sus comisiones son muy parecidas.

Finalmente, se investigaron métodos de desanonimización de usuarios en la red TOR. Concretamente, se investigó el estado actual de las técnicas de *Fingerprinting* de sitios web y las técnicas de correlación extremo-a-extremo sobre la red TOR y a continuación se analizaron los últimos ataques publicados que utilizan estas técnicas.

Los ataques de *fingerprinting* de circuitos TOR permiten determinar con alta probabilidad si un cliente de la red TOR es un usuario que accede o no a un servicio oculto de la red o si por el contrario es un operador de un servicio oculto. Por ejemplo, en el ataque analizado se pudo identificar la presencia de HS con unas tasas de más del 98% de verdaderos positivos y menos de 0,1% de falsos positivos. Este hecho provoca que gracias al análisis de los circuitos se pueda obtener información relevante sobre los tipos de clientes. Aunque si es cierto que las técnicas de *fingerprinting* no son muy precisas para determinar los sitios web que visitan los usuarios de TOR que no acceden a servicios ocultos debido a las variables comentadas en el apartado 5.1.2 Variables que afectan a la precisión. En el caso concreto de los servicios ocultos estas variables no repercuten sobre la precisión a excepción de la replicabilidad. Lo cual hace que los ataques de *fingerprinting* dirigidos a un usuario que accede a servicios ocultos puedan ser altamente efectivos, si el atacante puede entrenar un clasificador con las mismas condiciones que la víctima (replicabilidad).

Los ataques de correlación extremo-a-extremo permiten identificar los usuarios de TOR relacionando los flujos de tráfico entre la entrada y la salida de la red. Estos ataques son difíciles de realizar porque requieren que el atacante controle una gran cantidad de nodos de la red TOR, para poder correlar los flujos de tráfico entre los nodos de entrada y salida. Sin embargo, los ataques de correlación que utilizan el tráfico DNS (ataques DefecTor) son más factibles de llevar a cabo debido a que existen entidades globales que pueden observar el tráfico de entrada a la red TOR y algunas de las solicitudes de DNS que salen de ella, sin la necesidad de observar los nodos de salida de la red TOR. Por ejemplo, los servidores públicos de DNS de Google, actualmente observan una tercera parte de todas las solicitudes de DNS que salen de la red TOR, llegando a máximos del 40%. Por lo que un atacante que tenga acceso a dichas solicitudes, podría realizar un ataque de correlación igual de efectivo que observando los nodos de salida de la red TOR.

Bibliografía

[1] TOR project. [Online]. https://www.torproject.org

[2] TOR: Sponsors. [Online]. https://www.torproject.org/about/sponsors.html.en

[3] E. Rescorla T. Dierks, "The Transport Layer Security (TLS) Protocol," IETF RFC 5248, 2008.

[4] P. Karlton A. Freier, "The Secure Sockets Layer (SSL) Protocol Version 3.0," IETF RFC 6101, Agosto 2011.

[5] Information Sciences Institute University of Southern California, "Transmission Control Protocol (TCP)," IETF RFC 793, 1981.

[6] Wikipedia Onion routing. [Online]. http://en.wikipedia.org/wiki/Onion_routing

[7] M. Reed, P. Syverson D. Goldschlag, "Hiding Routing Informations" In proceedings of the First International Workshop on Information Hiding, vol. 1174 of LNCS, no. pág. 137-150, 1996.

[8] Wikipedia Diffie-Hellman. [Online]. https://es.wikipedia.org/wiki/Diffie-Hellman

[9] M. Ganis, Y. Lee, R. Kuris, D. Koblas, L. Jones M. Leech, "SOCKS Protocol Version 5," IETF RFC 1928, 1996.

[10] TOR Browser. [Online]. https://www.torproject.org/projects/torbrowser.html.en

[11] TOR Browser Design. [Online]. https://www.torproject.org/projects/torbrowser/design/

[12] Orbot. [Online]. https://guardianproject.info/apps/orbot/

[13] Guardian Project. [Online]. https://guardianproject.info

[14] Orfox. [Online]. https://guardianproject.info/apps/orfox/

[15] Tails. [Online]. https://tails.boum.org/

[16] Debian. [Online]. https://www.debian.org/

[17] Freedom of the Press Foundation. [Online]. https://freedom.press/

[18] Wikipedia Criptomoneda. [Online]. https://es.wikipedia.org/wiki/Criptomoneda

[19] (2008) Mail Archive: Bitcoin P2P e-cash paper. [Online]. https://www.mail-archive.com/cryptography@metzdowd.com/msg09959.html

[20] (2009) Bitcoin open source implementation of P2P currency. [Online]. http://p2pfoundation.ning.com/forum/topics/bitcoin-open-source

[21] Bitcoin clients. [Online]. https://en.bitcoin.it/wiki/Clients

[22] Wikipedia ECDSA. [Online]. http://es.wikipedia.org/wiki/ECDSA

[23] Daniel R. L. Brown, "SEC 2: Recommended Elliptic Curve Domain Parameters," v2, 2010.

[24] Wikipedia DER. [Online]. http://en.wikipedia.org/wiki/Distinguished_Encoding_Rules

[25] Wikipedia SHA. [Online]. https://es.wikipedia.org/wiki/Secure_Hash_Algorithm

[26] Wikipedia RIPEMD-160. [Online]. http://es.wikipedia.org/wiki/RIPEMD-160

[27] Wikipedia Hashcash. [Online]. https://en.wikipedia.org/wiki/Hashcash

[28] Bitcoin network. [Online]. https://en.bitcoin.it/wiki/Network

[29] Satoshi Client Node Discovery. [Online]. https://en.bitcoin.it/wiki/Satoshi_Client_Node_Discovery

[30] Bitcoin Protocol Specification: Message types. [Online]. https://en.bitcoin.it/wiki/Protocol_Specification#Message_types

[31] Codificación Base58Check. [Online]. https://es.bitcoin.it/wiki/Codificaci%C3%B3n_Base58Check

[32] Bitcoin transaction. [Online]. https://en.bitcoin.it/wiki/Transaction

[33] Bitcoin block. [Online]. https://en.bitcoin.it/wiki/Block

[34] Bitcoin Block hashing algorithm. [Online]. https://en.bitcoin.it/wiki/Block_hashing_algorithm

[35] Blockchain double spends. [Online]. https://blockchain.info/double-spends

[36] Blockchain. [Online]. https://blockchain.info/

[37] Bitcoin Block chain. [Online]. https://en.bitcoin.it/wiki/Block_chain

[38] Freenet. [Online]. https://freenetproject.org/

[39] The invisible internet project (I2P). [Online]. https://geti2p.net/

[40] GNUnet. [Online]. https://gnunet.org/

[41] The Entropy Project. [Online]. http://entropyproject.net/

[42] ANts P2P. [Online]. http://antsp2p.sourceforge.net/

[43] Wikipedia Cryptocurrency tumbler. [Online]. https://en.wikipedia.org/wiki/Cryptocurrency_tumbler

[44] Bitcoin Blender. [Online]. https://bitblender.io/

[45] R. Avnur H. Cheng, "Traffic Analysis of SSL Encrypted Web Browsing," University of Berkeley, 1998.

[46] B. Raman S. Mistry, "Quantifying Traffic Analysis of Encrypted Web-Browsing," University of Berkeley, 1998.

[47] D. R. Simon, Y. M. Wang Q. Sun, "Statistical Identification of Encrypted Web Browsing Traffic" In IEEE Symposium on Security and Privacy, no. pág. 19–30, 2002.

[48] A. Hintz, "Fingerprinting Websites Using Traffic Analysis" In Privacy Enhancing Technologies, no. pág. 171–178., 2003.

[49] R. Wendolsky, and H. Federrath. D. Herrmann, "Website Fingerprinting: Attacking Popular Privacy Enhancing Technologies with the Multinomial Naïve-Bayes Classifier," no. pág. 31–42, 2009.

[50] K. Matsuura Y. Shi, "Fingerprinting Attack on the Tor Anonymity System," no. pág. 425–438. , 2009.

[51] L. Niessen, A. Zinnen, T. Engel A. Panchenko, "Website Fingerprinting in Onion Routing Based Anonymization Networks," no. pág. 103–114, 2011.

[52] X. Zhang, B. Joshi, R. Johnson X. Cai, "Touching from a Distance: Website Fingerprinting Attacks and Defenses," no. pág. 605–616, 2012.

[53] I. Goldberg T. Wang, "Improved Website Fingerprinting on Tor," no. pág. 201–212, 2013.

[54] S. Afroz, G. Acar, C. Diaz, R. Greenstadt M. Juarez, "A Critical Evaluation of Website Fingerprinting Attacks" In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, no. pág. 263–274, 2014.

[55] M. P. Collins, C. V. Wright, F. Monrose, M. K. Reiter, et al. S. E. Coull, "On Web Browsing Privacy" In Anonymized NetFlows In USENIX Security Symposium, no. pág. 339–352, 2007.

[56] M. Al-Sabah, D. Lazar, S. Devadas, M Dacier A. Kwon, "Circuit Fingerprinting Attacks: Passive Deanonymization of Tor Hidden Services" In the Proceedings of the 24th USENIX Security Symposium, 2015.

[57] I. Pustogarov, R. Weinmann A. Biryukov, "Trawling for Tor Hidden Services: Detection, Measurement, Deanonymization" In Proceedings of the 2013 IEEE Symposium on Security and Privacy, no. pág. 80–94, 2013.

[58] N. Mathewson, P. Syverson R. Dingledine, "Tor: The second generation onion router" In Proceedings of the 13th conference on USENIX Security Symposium, vol. 13, 2004.

[59] R. Dingledine N. Feamster, "Location diversity in anonymity networks" In Proceedings of the 2004 ACM workshop on Privacy in the electronic society, no. pág. 3, 2004.

[60] L. Gao, "On inferring autonomous system relationships in the Internet," vol. 9, no. 6, 2001.

[61] P. Zielinski S. J. Murdoch, "Sampled traffic analysis by Internet-exchange-level adversaries" In International Workshop on Privacy Enhancing Technologies, vol. 4776, no. pág. 1-6, 2007.

[62] C. Wacek, R. Jansen, M. Sherr, and P. Syverson A. Johnson, "Users get routed: Traffic correlation on Tor by realistic adversaries In CCS," no. pág. 1-12, 2013.

[63] A. Johnson. The Tor path simulator. [Online]. https://github.com/torps/

[64] O. Starov, A. Zair, P. Gill, and M. Schapira R. Nithyanand, "Measuring and mitigating AS-level adversaries against Tor" In NDSS, no. pág. 3, 2016.

[65] A. Johnson, A. Das, N. Borisov, M. Caesar J. Juen, "Defending Tor from network adversaries: A case study of network path prediction," vol. 2015, no. 2, 2015.

[66] A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, P. Mittal Y. Sun, "RAPTOR: Routing attacks on privacy" In Proceedings of the 24th USENIX Conference on Security Symposium, no. pág. 3, 2015.

[67] T. Pulls, L. M. Roberts, P. Winter, N. Feamster B. Greschbach, "The Effect of DNS on Tor's Anonymity" In NDSS, 2017.

[68] S. Bortzmeyer. (2016) RFC 7816 – DNS query name minimisation to improve privacy. [Online]. https://tools.ietf.org/html/rfc7816

[69] Z. Hu, J. Heidemann, D. Wessels, A. Mankin, N. Somaiya L. Zhu, "Connection-oriented DNS to improve privacy and security" In Proceedings of the 2015 IEEE Symposium on Security and Privacy, no. pág. 14, 2015.