UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

Análisis y diseño de la tecnología utilizada en las monedas virtuales

TESIS

Que para obtener el título de

Ingeniero en Computación

P R E S E N T A

Diego Alberto Castillo Castillo

DIRECTOR DE TESIS

Ing. Carlos Alberto Román Zamitiz

Ciudad Universitaria, Cd. Mx., 2017

Análisis y diseño de la tecnología utilizada en las monedas virtuales.

Índice

Índice de imágenes

Índice de tablas

Introducción

Hoy en día la tecnología ha aportado una gran cantidad de soluciones para nuestra vida cotidiana, es por ello que con todas las innovaciones, debemos tener conocer todas de tal modo que nuestro día a día se haga mucho más fácil para así poder sacar el mayor provecho.

Un aporte que suena cada día más en el ámbito tecnológico son las monedas virtuales, pero al tratarse de dinero, siempre habrá cierta desconfianza de todos los usuarios. Es por eso que este trabajo de tesis tratará de explicar de manera clara cómo se crean las monedas virtuales, cómo funcionan y cómo podemos utilizar dicha tecnología para nuestro beneficio.

De tal manera que en el primer capítulo planteare el problema más común que se genera al utilizar las monedas virtuales en nuestra sociedad, así como ir mencionando algunos de los términos que iremos utilizando a lo largo del trabajo.

Por otro lado, el segundo capítulo iré describiendo poco a poco conceptos que nos ayudaran a comprender como es que podemos utilizar las monedas virtuales en cualquier sito del mundo. Además, si nuestros intereses viajaran de un lugar a otro, hablaré de cómo se realiza una transacción con la seguridad de que nuestro dinero se encuentra a salvo de personas malintencionadas.

En el tercer capítulo iremos directamente al tema de interés, daré una breve historia de los inicios de las monedas virtuales para que podamos ver que el tema no es nada nuevo, sin embargo, para muchas personas podrá ser un concepto actual. También, hablaré de los tipos y las características que tienen las monedas virtuales, así como poder ir relacionando los temas del capítulo anterior para poder mostrar de modo más sencillo su funcionamiento. Veremos cómo la criptografía nos ayudará para poder hacer que las monedas sean muy seguras con respecto a otro tipo de dinero en el mundo. Explicaré un poco el proceso para generar nuevas monedas y con todo ello,

podremos ver cómo es que trabajan y algo muy importante, en dónde es que podemos almacenar nuestras monedas virtuales.

Pero aunque ya tengamos todo los conceptos claros, también mencionaré cómo es que podemos adquirir monedas virtuales y ver un poco más a fondo que existen diferentes alternativas, unas ya más adoptadas por los usuarios, y para finalizar poder mostrar que son más los beneficios que las desventajas que tienen las monedas.

Y con ayuda de todo lo anterior, espero despertar el interés de alguien para incursionar en el mundo de las monedas virtuales con la confianza de que sus intereses estarán a salvo, así como, poder tener la certeza de que existe un método con el cual podemos realizar una compra sin la necesidad de traer efectivo con nosotros. Pero lo más importante, que cada uno de ustedes pueda aprender algo nuevo y así estar prevenidos para cuando tengamos la oportunidad de tener interacción con este tipo de tecnología.

1. Planteamiento del problema: La falta de credibilidad sobre la creación, uso y/o manejo de monedas virtuales

Monedas virtuales… Simplemente el escuchar el nombre causa cierta desconfianza para personas que no tienen conocimientos acerca de que son o como se manejan. Sin embargo, son más seguras de lo que pensamos e inclusive mejor que ir cargando un objeto que pueda llamar la atención de alguna persona.

Hoy en día la gran mayoría de personas del mundo tiene un teléfono inteligente en su bolsillo, y de ese gran grupo, me atrevo a decir que todos tienen acceso a internet de alguno u otro modo. Gracias a estas herramientas modernas, el uso de monedas virtuales se vuelve un simple hecho como revisar nuestras redes sociales o simplemente realizar una llamada.

La tecnología ha cambiado nuestra forma de ver la vida que a cada instante la hace más fácil y sencilla, descubriendo nuevas maneras de revolucionar nuestro mundo y así permitirnos pertenecer a una generación de grandiosos descubrimientos.

Un muy claro ejemplo de un grandioso invento, es el internet, una red (en la actualidad) enorme conformada por miles de millones de nodos, con la cual es posible realizar múltiples actividades que van desde las comunicaciones, básicas y modernas, hasta lo que actualmente se conoce como *Cloud Computing*, que es un servicio de "préstamo" de hardware y software de una empresa a un cliente a través de dicha red.

Pero vivimos en un mundo donde todo gira alrededor de una sola cosa, dinero. Es muy triste decirlo de este modo pero lamentablemente la sociedad hoy en día es lo que más le importa, y gracias a esto es que la tecnología se ha tenido que emplear para facilitar los diversos servicios ofrecidos a través de la red.

La revolucionaria idea por la que se está optando poco a poco en diversos lugares del planeta, es utilizar una manera de intercambiar un servicio por algo que se le pueda asignar un valor, en este caso un valor monetario, con lo cual surge el concepto de moneda virtual, criptomoneda o coin; aunque no se trata de desplazar las monedas que conocemos, sino es más bien una nueva manera de realizar pagos de forma sencilla.

Desde la antigüedad, para poder realizar lo que hoy conocemos como una compra, se llevaba a cabo una acción conocida como trueque, intercambiar una cosa por otra siempre y cuando ambas partes tuvieran el mismo valor. Pero el problema llegó cuando el objeto de intercambio ya no era indispensable para alguna de las dos partes y con ello surgió la necesidad de realizar esos trueques de manera más sencilla y fue cuando emplearon un objeto el cual fuese aceptado por todo mundo para llevar a cabo los procesos de intercambio, por ejemplo, uno de los primeros objetos que se tiene reconocido como primera moneda fue el "cauri", una pequeña concha que no hace mucho se seguía utilizando en algunas regiones de África (para realizar el comercio de esclavos), América y principalmente en China; inclusive mucho antes de que se

utilizarán los metales como monedas, empleaban distintos tipos de objetos como es el caso del cacao, el trigo e incluso animales.

Es así como podemos darnos cuenta que la sociedad tiene la capacidad para poder utilizar diversos objetos para realizar "compras", pero para poder lograr hacerlo debieron primero llegar a un común acuerdo y es entonces cuando surge la moneda, pequeños metales para facilitar su transporte y su almacenamiento; con ello poco a poco las países fueron adaptando una moneda y estos a su vez agregando un valor frente a las demás, como lo podemos ver actualmente.

Hasta que en los últimos años, con tanta innovación de la tecnología el mundo ha ido adaptando nuevas formas de realizar el comercio de una forma más sencilla y confiable, y es entonces cuando entra el concepto de monedas virtuales.

Todas las nuevas tecnologías, como siempre, cuando nadie las conoce es muy fácil desconfiar de ellas, y más cuando se trata de una inversión económica.

Muchas veces, la razón de la sociedad hacia esa desconfianza es por tener muy mala información hacia el tema.

Las monedas virtuales no son la excepción, el concepto suena siempre muy interesante por todos los lados, además de que para todo mundo es muy práctico salir a gastar "dinero" de una manera más segura, teniendo una perspectiva de la seguridad física, además también llega a ser una muy buena fuente de inversión, a pesar de sus altibajos.

El problema radica más que nada en la mala imagen que se genera a partir de sucesos malintencionados que dan pie a la discordia entre que si son buenas o no las criptomonedas. Pero independientemente de esto, en este escrito trataré de dar una idea de lo bueno y lo malo, además de dar una buena explicación acerca del funcionamiento de la tecnología empleada en las monedas, tanto lo que se conoce como monederos virtuales, la creación de una moneda y su uso.

Las monedas virtuales, en simples palabras, son una serie de caracteres que tienen un cierto valor dentro de una red de computadoras, dicho valor monetario cambia dependiendo de la popularidad de las monedas (el número de personas que las utilizan), por ejemplo ver imagen 1:

Imagen 1 Ejemplo de bitcoin

La dirección anterior pasa de un comprador a un vendedor, ya que tiene cierto valor para realizar una compra.

Sin duda alguna puedo afirmar que las monedas virtuales son un gran salto para poder crear un mundo libre, confiable, honesto y democrático. Claro que siempre habrá personas que los utilicen con otros fines, pero debemos tener la confianza que sin duda cambiarán la forma de ver el dinero.

Las monedas virtuales ya llevan un breve tiempo en circulación, incluso han llegado a tener un valor muy grande en el mercado, como el caso de los bitcoins, es el más popular entre las monedas virtuales, en el 2013 su precio llegó a ser de hasta 1,200 dólares por unidad, siendo este su punto más alto, con lo cual podemos ver que la idea de pagar las cosas con una moneda virtual está siendo aceptado y cada vez se vuelve más popular entre las nuevas generaciones que utilizan la tecnología continuamente, y gracias a esto es que el valor monetario aumenta en los bitcoins. El único problema que han tenido fue que en febrero del 2014, una importante empresa de intercambio de bitcoins sufrió la desaparición de 850 mil bitcoins (750 mil pertenecientes a clientes y el resto eran propiedad de la propia empresa), lo cual provocó que disminuyera la confianza en la moneda y con ello la disminución de su valor.

Pero es así como las pequeñas ideas se convierten en un grandioso invento que tiene la posibilidad de cambiar nuestro mundo, sufriendo altibajos, y no es la primera y ni la última vez que esto pasa en proyectos tal impacto hacia la sociedad, pero con esfuerzo y dedicación puede llegar muy lejos.

Y así como bitcoin, existen otras monedas virtuales que a pesar de no ser tan conocidas, grandes empresas han estado optando por aceptar como medio de pago para realizar transacciones de compra y venta. Por lo cual solo es cuestión de tiempo para poder usarlas tan comúnmente como una tarjeta de crédito o inclusive a través de un dispositivo móvil.

2. Marco teórico

Hoy en día, el mundo gira alrededor de la red más grande que existe… internet.

Es indispensable para nuestras actividades diarias, tanto para el hogar como para el trabajo. Y con ello podemos hacer diversas actividades que hace no mucho pensábamos era imposible realizar. Es por ello que es necesario mencionar su funcionamiento con conceptos sencillos y a partir de ahí hablar sobre nuestro punto principal, BITCOINS.

Sin embargo, no solo es necesario tener una comunicación al tratarse de nuestros intereses económicos, es necesario protegerlos de personas malintencionadas que quieran hurtar nuestro dinero, es por ello que al mismo tiempo debemos hacer hincapié sobre las medidas que se utilizan para proteger nuestros bitcoins.

2.1 Topologías de red

Primero explicaré el orden y funcionamiento de una red, que es la base de las monedas virtuales.

A grandes rasgos podemos definir tres tipos de redes, centralizadas, descentralizadas y distribuidas.

Las centralizadas, todos los nodos o computadoras van conectadas a un solo nodo, lo que genera un problema de que si dicho nodo se desconecta se pierde total comunicación.

Las descentralizadas, este tipo de red se compone de muchas redes centralizadas que a su vez están conectadas entre sí, y en caso de que un nodo "central" se desconectará, se pierde en cierta parte la conexión pero no es totalmente.

Y por último tenemos las distribuidas, en este caso todos los nodos van conectados entre sí, con esto evitamos el problema de que si un nodo se desconecta los demás siguen trabajando sin problema.

En la imagen 2 vemos un ejemplo del comportamiento de lo antes explicado. De izquierda a derecha, una red centralizada, descentralizada y distribuida.

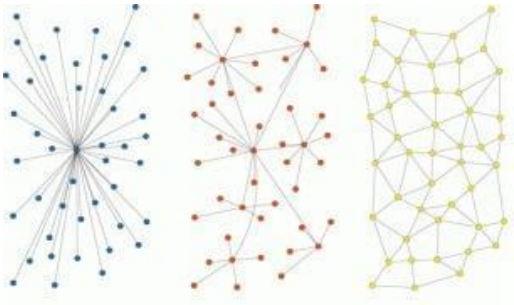

Imagen 2 Red centralizada, descentralizada y distribuida

Una vez que queda claro lo anterior, puedo definir el tipo de red que se utiliza para poner tener la conexión que mantiene viva el concepto de monedas virtuales.

Red Peer-to-Peer o P2P, en palabras sencillas, se trata de un tipo de conexión entre varias computadoras (red distribuida). Es comúnmente usada para realizar transferencias de archivos de una manera más rápida y anónima. Dentro de una red de este tipo, cada computadora se conoce como nodo o peer, y estas a su vez están conectadas de cierto modo que tengan comunicación con todas las demás pertenecientes a la red.

A diferencia de una red que conocemos, en donde solo interviene un solo servidor y los demás nodos son conectados hacia él, dentro de una red P2P cada nodo actúa como un servidor por lo cual todos los nodos se conectan entre sí. Lo cual hace que cada nodo conectado haga un aporte de recursos permitiendo que el tráfico sea a velocidades muy grandes. Por lo que podemos ver que cada nodo o Peer tiene derechos y obligaciones, contribuyen con sus recursos y obtienen privilegios como la gran velocidad y acceso a distinto contenido.

Todo lo anterior era la definición básica de cómo se lleva a cabo la conexión de computadoras a un nivel muy general, con el paso del tiempo se fueron definiendo de una forma más clara y precisa, con la cual se volvería más fácil diferenciar de qué tipo se trata.

2.1.1 Topología de bus

Vamos a comenzar con una topología muy básica de nombre bus, en dicha topología, todos los nodos tienen comunicación entre sí, pues la conexión se realiza a través de un solo cable, en donde un equipo transmite y todos los demás que estén conectados a la red están al pendiente de la información, para verificar que el destinatario, así como, si la información corresponde al equipo se la queda, pero en caso contrario la descarta.

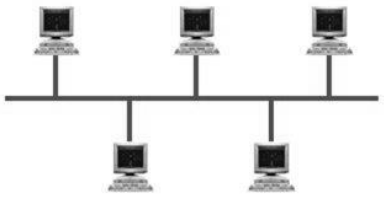

Imagen 3 Topología Bus

Las primeras conexiones eran realizadas por medio de un cable coaxial y un conector BNC (años después se reemplazó por un cable coaxial más delgado llamado "thinnet") con un par de resistencias de carga, una en cada extremo, esto con la finalidad de tener pérdidas de datos. Para los inicios era muy bueno tener una conexión entre computadoras para tener comunicación, pero poco a poco los problemas siempre tienen que salir para poder realizar mejoras.

Como podemos ver en la imagen 3, todas las conexiones comparten un medio de transmisión, el problema surge cuando uno de los nodos era desconectado, en este caso la comunicación era cortada y por tanto la red no funcionaba total o parcialmente, ya que la comunicación se realiza bidireccionalmente hasta encontrar el destinatario; lo mismo pasaba cuando un cable era cortado.

Generalmente esta conexión es muy poco utilizada, debido a que a pesar de ser implementada en una red privada (una red interna o local) puede ser muy insegura debido a que cualquier modo conectado puede interceptar información, independientemente de que sea valiosa o no.

2.1.2 Topología de anillo

La topología anillo, como su nombre lo indica, consiste en realizar una conexión linealmente entre sí entre todos los nodos, conectando el último con el primero para obtener una forma de un anillo. De manera que para que fluya la información, tiene que pasar por uno o varios nodos hasta llegar al destinatario.

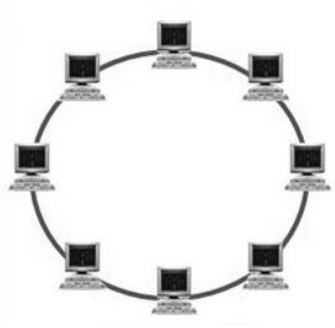

Imagen 4 Topología Anillo

En esta topología, el flujo se realiza unidireccional, es decir, la información viaja en un solo sentido. Del mismo modo, cada equipo solo tiene una comunicación punto a punto, o en otras palabras, cada nodo sólo tiene comunicación con otros dos, uno por cada lado, por lo cual si un nodo llega a fallar la comunicación puede fallar total o parcialmente, ver imagen 4.

Además de eso, otra desventaja que tenemos aquí es que para un número pequeño de nodos conectados funciona muy bien, pero cuando el número de nodos es muy grande el fijo de información se verá afectado y por ende la red se volverá lenta.

Pero no todo es malo, esta topología es muy fácil realizar y de administrar, ya que solo cuenta con dos conexiones por equipo y como lo dije anteriormente, el flujo es muy bueno cuando se trata de pocos nodos.

2.1.3 Topología de estrella

Esta topología está conformada por un nodo central, y varios nodos conectados a su alrededor. Lo que hace que el flujo de información vaya desde algún extremo hacia un punto o nodo central y de allí pase hacia su destino.

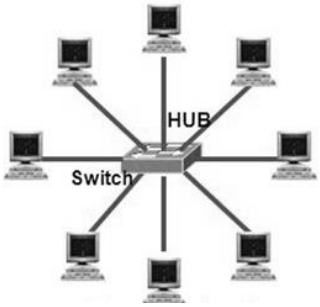

Imagen 5 Topología Estrella

Es utilizada este tipo de topología generalmente para redes locales (LAN- Local Area Network o Red de Área Local) pequeñas, ya que, para tener comunicación entre dos dispositivos, la información debe viajar primero al nodo central, y esto hace que ese flujo sea más poco lento por el tiempo que tarda en hacer el recorrido, ver imagen 5.

Si bien vemos que se asemeja mucho a una red centralizada, y por efectos de la misma es mucho más fácil agregar equipos nuevos, realizar una configuración más rápidamente, y la prevención y corrección de daños y/o conflictos se vuelve mucho más sencilla puesto que si un nodo deja de funcionar el funcionamiento continua. El problema surge cuando el nodo central comienza a fallar toda la red deja de transmitir.

2.1.4 Topología de árbol o jerárquica

Es una topología de red muy similar a la de estrella, pero aquí no tenemos un nodo central; como su nombre lo indica, los nodos se conectan en forma de árbol, tenemos un nodo principal en la parte superior y a partir de ahí se conectan los nodos hacia abajo, y así sucesivamente se realiza una conexión con los nodos consecuentes, viéndolo de un modo más sencillo cada nodo puede depender de otro y a su vez uno o varios nodos pueden depender de él, ver imagen 6.

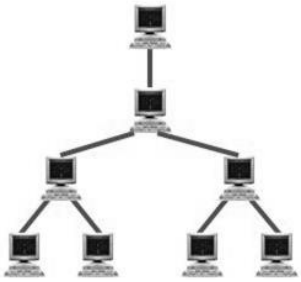

Imagen 6 Topología Árbol

La más grande desventaja que se puede presentar en este tipo de topología, es cuando se desconecta un nodo del cual dependen otros más, en este caso puede llegar a perderse la comunicación de una o varias máquinas, dependiendo del nivel en donde se encuentra el nodo desconectado.

Pero como una gran ventaja tenemos que gracias a esta jerarquía que nos brinda el acomodo de los nodos, podemos obtener prioridad para ciertos lugares de trabajo y con lo cual podemos administrar de un mejor modo toda nuestra red.

2.1.5 Topología de malla

Para este caso, se trata de una topología muy compleja, la diferencia entre las demás topologías, es que todos los nodos de la red están conectados “todos con todos”. Lo cual permite que si un medio o algún nodo falla, la comunicación pasará a otro nodo que se hará cargo del tráfico de la red y por tanto es más difícil perder la comunicación entre dos o más nodos y se vuelve muy confiable, ver imagen 7.

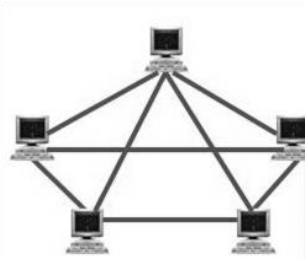

Imagen 7 Topología Malla

Esta red difícilmente podría tener desventajas, la más grande que tenemos es que los costos se elevan mucho, debido a que se utiliza una gran cantidad de cable para poder realizar tantas conexiones necesarias.

Pero como podemos ver, tiene muchísimas más ventajas, en esta topología no necesitamos un nodo central que administre la comunicación ya que todos están comunicados entre sí, por ello si se desconecta un nodo los demás se encargan de la comunicación y por tanto tenemos que la comunicación será menos probable que se interrumpa.

2.2 Protocolos

Anteriormente pudimos ver un poco la manera en la que se realizan distintos tipos de comunicación, pero no sólo basta con conectar uno con otro para poder tener una comunicación, tanto para enviar información como para recibirla, para poder tener una comunicación, independientemente del dispositivo o el sistema operativo, es necesario llevar a cabo una serie de reglas con las cuales los dispositivos obtendrán dicha comunicación.

Para ello es donde se emplean los protocolos, que son en pocas palabras, las reglas para llevar a cabo cuando se quiere establecer una comunicación entre dos nodos.

En sus inicios, cada compañía que se dedicaba a distribuir equipo de cómputo en la antigüedad, tenía su manera de realizar la comunicación. Pero esto tenía un gran problema, debías comprar dos máquinas de la misma marca para poderlas conectar, lo cual no era una opción rentable.

Por otra parte, un proyecto desarrollado en la milicia de nombre ARPANET fue la solución a todos los problemas de compatibilidad para llevar a cabo la comunicación entre dispositivos.

Todo comienza aproximadamente en 1960, durante la guerra de Vietnam, las computadoras digitales eran muy jóvenes y las investigaciones eran muchas. Pero dos

grandes mentes, Paul Baran, en Rand Corp y Donald Davies en el Laboratorio Nacional de Física de Inglaterra, inventaron la conmutación de paquetes, que consistía en romper mensajes en bloques discretos que podían ser enviados por separado a través de diversos canales de una red.

Después de unos años, en 1969, fueron patrocinados por la Agencia de Proyectos de Investigación Avanzados de Defensa (DARPA-Defense Advanced Research Projects Agency), creando la primera red de conmutación de paquetes, ARPANET.

El proyecto consistía en eso mismo, tener una comunicación entre diferentes equipos sin importar la marca de fabricante. Primero comenzando por dos dispositivos, pero poco a poco esa red comenzó a crecer hasta volverse de un uso hacia las personas dejando a un lado la parte militar.

El gran descubrimiento de la conmutación de paquetes tuvo tanto éxito, que grandes compañías como IBM trataron de realizar su propia conmutación para realizar la comunicación entre sus dispositivos.

Y durante 1972, se inició un primer intento por realizar un estándar para que todos tuviéramos comunicación, se le llamó a este proyecto INWG (InterNetworking Working Group), el propósito de este proyecto era generar protocolos generales para todos los dispositivos por medio de datagramas.

Y después de varios años de debate y problemas por aceptar dichos estándares, el proyecto desapareció, aunque el presidente del grupo, Vint Cerf, renunció a su cargo y salió de la Universidad donde estudiaba para unirse al grupo de trabajo de ARPA. Una vez en el nuevo grupo de trabajo y con todo el conocimiento sobre conmutación, comienzan el proyecto “Transmission Control Program” que más adelante se convertiría en la base de internet recibiendo el nombre de TCP / IP (Transmission Control Protocol / Internet Protocol).

Por otro lado, algunos de los compañeros de Cerf durante su participación en INWG, se reagruparon para realizar su propio modelo que lleva el nombre de OSI y con ello

podría decirse que se convirtieron en la competencia y a su vez en la base de todo lo que conocemos dentro de la comunicación entre dispositivos, formando parte de los pilares de internet.

Durante 1977, un grupo británico dedicado a la computación, realizan la proposición de crear un comité dedicado a las normas especializadas en la conmutación de paquetes a la Organización Internacional de Estándares (ISO). Y después de un tiempo, ISO aprobó la solicitud para estandarizar la conmutación nombrando como presidente del proyecto a Charles Bachman, experto en bases de datos proveniente de Estados Unidos.

Bachman desarrolló un modelo por capas, en donde la primera capa contendría todos los medios físicos (tales como cables de cobre); protocolos de transporte para mover, ajuste de los datos en la capa 4; y aplicaciones (como el correo electrónico y transferencia de archivos) pertenecen a la capa 7. Una vez que se estableció una arquitectura en capas, protocolos específicos entonces se desarrollarían.

Con ayuda de esa arquitectura por capas, Bachman logró lo que hasta ese entonces era muy complicado, comunicar computadoras entre sí. Esto fue realmente muy bueno para empresas de grandes nombres como General Motors, que en la década de 1980 fue un fiel defensor del modelo OSI, esto debido a que por la gran cantidad de proveedores que tenía la combinación de hardware y software era en gran medida incompatible. Y gracias a que GM seguía siempre el modelo OSI pudieron eliminar dicha incompatibilidad.

Para poder realizar normas, era necesario realizar varias sesiones plenarias y de comisiones, la primera sesión plenaria del OSI duró tres días, del 28 de febrero al 2 de marzo de 1978. Decenas de delegados de 10 países participaron, así como observadores de cuatro organizaciones internacionales. Inclusive, se encontraban muchas personas que alguna vez habían pertenecido a INGW, y todos ellos se encontraban con grandes ánimos de saber que por fin el monopolio de las

comunicaciones podría ser arrebatado a las grandes empresas que lo controlaban en esos días.

Pero no debemos olvidarnos del proyecto de ARPA, en 1980 junto con la Agencia de Comunicaciones de Defensa, tuvieron que acelerar la adopción de internet, para realizar esto tuvieron que implementar protocolos de internet dentro de los sistemas operativos más populares. Hasta que el 1 de enero 1983 ARPA dejó de apoyar el protocolo ARPANET con lo cual todo mundo tuvo que adoptar TCP/IP si querían mantenerse en contacto; en esta fecha es cuando se conoce el nacimiento de internet.

Y entonces después de sufrir toda clase de problemas tenían dos modelos, TCP/IP y OSI, por un lado tenían algo que era gratuito y simplemente habría que descargarlo y por el otro una arquitectura mucho más completa y elaborada, lo cual hacía que fuese muy difícil tomar la decisión de cual elegir.

¿Qué es un protocolo y un modelo?

Partiendo de lo que es un modelo, se define como el conjunto de recomendaciones o procedimientos ideal para llevar a cabo la comunicación entre dos diferentes dispositivos.

Mientras que un protocolo es muy similar a un modelo, son reglas que se deben de seguir para hacer funcionar un modelo y tener una comunicación buena y confiable entre dos o más dispositivos.

2.2.1 Modelo OSI

Como ya hemos visto, el modelo OSI fue creado durante los 80’s por la Organización Internacional para la Estandarización (ISO) que no solamente se encarga de realizar estándares sobre cuestiones de computación o comunicaciones, sino que va más allá, incluso podríamos decir que no tiene una especialidad o algún funcionamiento en específico para realizar estándares.

Además debemos nuevamente de remarcar el funcionamiento, la información que se preparara para ser enviada se tiene que dividir, para poder ser enviada, o lo que se conoce como conmutación de paquetes. Para realizar dicha división, los datos que serán enviados deben de pasar por siete capas que contempla el modelo OSI.

Las capas del modelo OSI son las siguientes: física, enlace de datos, red, transporte, sesión, presentación y aplicación. Cabe recalcar que la comunicación que se lleva a cabo dentro del modelo únicamente será de forma vertical y exclusivamente se comunicaran una y solo una capa arriba o abajo, dependiendo el sentido de la comunicación, ver imagen 8.

Imagen 8 Modelo OSI

Vamos a comenzar la explicación desde la capa siete hasta la capa uno.

Capa de aplicación, es la capa número siete dentro del modelo de comunicación OSI, en esta capa es donde se lleva a cabo la interacción con el usuario final, www, e-mail, etc. En esta capa encontramos además que se manejan protocolos como lo son FTP, SMTP, TELNET, SSH, etc.

Como antes se mencionó, esta capa interactúa con el usuario y recolecta los datos que serán enviados, una vez que tiene los datos listos, toda la información es pasada a la capa de presentación, que es la siguiente dentro del modelo.

Capa de presentación, ocupa la posición número 6 dentro del modelo OSI, esta capa en simples palabras se puede comprender como un traductor. Se encarga de convertir los datos en información que después pasarán a la capa de sesión, o en su defecto recibe la información proveniente de la capa de sesión y la traduce a la capa de aplicación para que el usuario final pueda entenderla de una forma más agradable.

En esta capa podemos encontrar HTML, MP3, MP4, JPG, Doc, etc., seguramente estamos familiarizados con este tipo de formatos, pues los vemos muy seguido, pero, aunque no lo creamos, estos formatos son protocolos. Por ejemplo, un MP3 puede ser reproducido en una computadora o un teléfono celular y en ambos obtendremos el mismo resultado, una buena canción o un audio cualquiera; con ello podemos corroborar que cuando todos los dispositivos manejan un protocolo es más fácil poder hacer tareas en ellos, independientemente de la plataforma en la que trabajen o el tipo de marca que tengan.

Capa de sesión, es la capa con el número 5. Esta capa es la encargada, como su nombre lo indica, de mantener la sesión entre las dos anteriores capas y la capa de transporte, podemos decir que tiene tres funciones básicas. Se encarga de mantener un control en el diálogo, en un sentido o ambos sentidos (full-duplex o half-duplex), tener un agrupamiento, o en pocas palabras marcar el flujo de los datos, y recuperación de la sesión, si ocurre algún problema de comunicación la capa de sesión es la encargada de volverla a iniciar dependiendo de las marcas que va a ir generando en cada paquete.

Los protocolos que podemos encontrar dentro de esta capa son: RCP, que permite a algún programa en ejecución no preocuparse por la comunicación, SCP que es muy similar al protocolo anterior, la única diferencia que existe es que los datos son más seguros ya que la información tiene que ser cifrada para mantener mayor seguridad, ASP, que es un protocolo asociado a APPLETALK, que básicamente hace lo mismo que los anteriores pero este se utiliza para dispositivos de la compañía Apple.

Capa de transporte, ocupa el lugar número 4 dentro del modelo OSI, esta capa es la que se dedica a realizar el transporte de los datos, todo lo que viene de las capas superiores, divide todo en paquetes para poder hacer la transferencia más efectiva y confiable, dependiendo del caso.

En esta capa podemos encontrar tres de los protocolos más importantes, TCP, UDP e ICMP, los cuales dependiendo del servicio requerido puede utilizarse uno u otro, dependiendo el caso.

TCP

Para realizar el encapsulado de un paquete, se deben de agregar ciertos valores independientes de la información que va a viajar en la red, es decir una vez que la información a viajar es dividida en pequeños pedazos, se le coloca un identificador. En el caso de la capa de transporte se coloca un header o encabezado, en el caso de TCP, se coloca para poder indicar el protocolo por el que viaja y tener un mejor control de la información. Ya que toda la información que viaja es dividida en pequeñas partes, es necesario tener una serie de parámetros que indiquen toda la información correspondiente para que una vez que llega a su destinatario pueda ser nuevamente ensamblada de la misma forma en como salió.

Primeramente, debemos de comentar que, cada equipo cuenta con 65536 (del 0 al 65535) de los cuales, los primeros 1024 (0-1023) se conocen como puertos reservados y son puertos bien conocidos; a partir del 1024 al 49151 son puertos registrados y pueden ser utilizados por cualquier aplicación, y por último tenemos los que van del

49152 al 65535 y se les conoce como puertos privados o dinámicos, generalmente estos puertos son asignados a los equipos cuando se requiere iniciar una comunicación y les toca tomar el papel de clientes.

La imagen 9 es un ejemplo de una cabecera TCP, que indica cómo se divide la información para ser enviada y comprendida de una manera más ordenada:

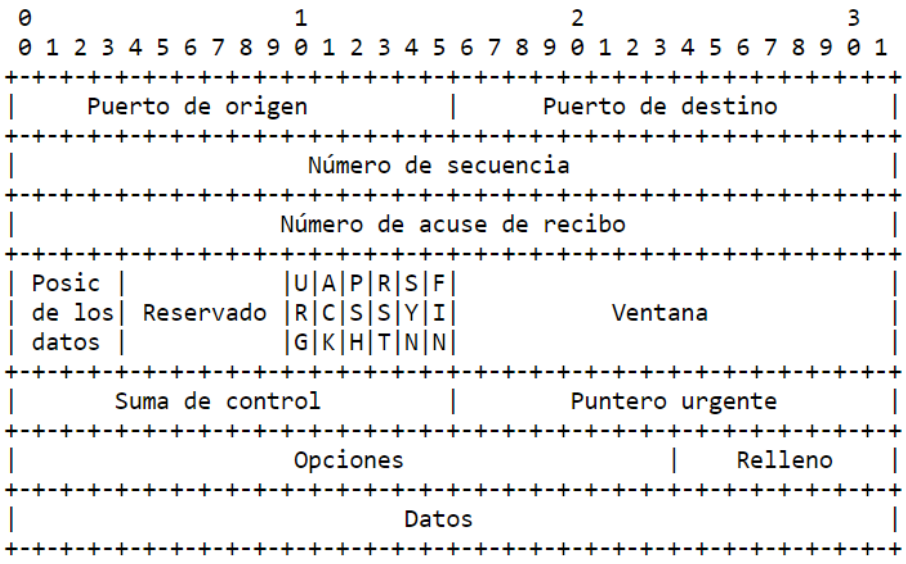

Imagen 9 Cabecera TCP

Una vez que definimos lo que son los puertos, podemos ver que dentro de la cabecera TCP podemos encontrar primeramente el puerto origen (El puerto asignado al cliente) y el puerto destino (Puerto asignado al servidor), cada uno se representa en 16 bits. Después de los puertos tenemos un campo que indica el número de secuencia que indica la posición que ocupa el paquete.

El siguiente valor que encontramos es un número de reconocimiento el cual indica el siguiente valor de secuencia que deberá tener el siguiente paquete que encuentre. Enseguida podemos ver que vienen 4 bits que indican en donde comienzan los datos. Después, siguen seis bits que son reservados para operaciones futuras, en este caso

siempre serán ceros. Después continuamos con las banderas, son seis bits que tienen un significado por separado, el significado de cada una es el siguiente:

URG: Indica si el paquete es urgente
ACK: Indica la confirmación de recibido
PSH: Funciona para insertar paquetes
RST: Reinicia la conexión
SYN: Realiza la sincronización con la secuencia de números
FIN: Finaliza la comunicación

Los siguientes dieciséis bits siguientes son los que indican el tamaño de la ventana, es decir, el número de bytes de datos que el remitente de este segmento está dispuesto a aceptar. Siguiendo el orden tenemos el Checksum que es un identificador de cada paquete para evitar duplicados.

Los siguientes dieciséis bits se utilizan para indicar si el paquete será urgente y atendido con prioridad o tendrá que ser encolado como todos. Por último tenemos las opciones que queramos agregar, y el relleno que debe ser hacerse con ceros para que en caso de no completar los bits de la longitud, se realice con la cantidad de ceros necesarios.

El siguiente espacio es para indicar opciones, puede tener o no, además se ocupa para hacer un relleno y completar el paquete del tamaño requerido. Y al final de todo el paquete van los datos que serán enviados.

Generalmente el protocolo TCP se utiliza para garantizar que la comunicación será todo un éxito y no habrá fallas por cuestión de que se pierda un paquete. Es por eso que maneja las banderas antes mencionadas, si jamás llega un paquete de respuesta confirmando la llegada del paquete de un lado a otro, el protocolo por sí mismo se encarga de enviar el paquete hasta que se confirma la llegada. Antes de hacer el envío de información, la máquina que actúa como cliente y la que actúa como servidor, tienen que establecer un “saludo” para comenzar la conexión. Este saludo se conoce como Three Way Hand Shake, analizándolo de una manera sencilla, una máquina que actúa como cliente, manda un paquete con una bandera de SYN para iniciar la comunicación

además de indicar el número inicial de la secuencia. La máquina servidor responde con un SYN además de un ACK que indica la confirmación de que llegó en paquete y por último el cliente envía ahora los datos para iniciar la transferencia de información. Es de esa forma que se pide una confirmación para asegurar que el paquete no se va a perder, por lo que se vuelve una conexión más confiable.

UDP

El otro protocolo más importante manejado dentro de la capa de transporte es el UDP (User Datagram Protocol). Se trata de un protocolo muy similar al protocolo TCP, pero a diferencia de este, es un poco menos confiable con respecto a la entrega de información.

Dentro de los headers que se agregan en el protocolo UDP, no encontramos banderas porque es la esencia del protocolo. No es necesaria una confirmación, si llegan o no los paquetes, el protocolo se encarga de enviarlos una vez sin importar si llego a su destino.

En la imagen 10 podremos ver los campos de los headers que se utilizan para identificar un paquete UPD.

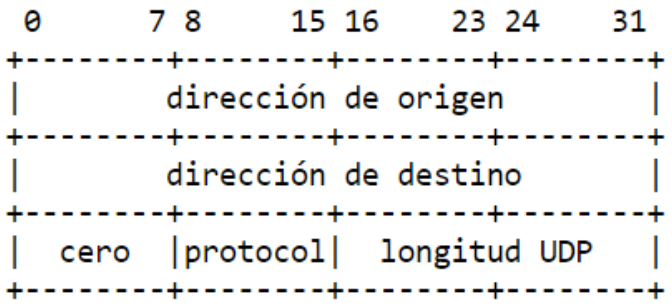

Imagen 10 Cabecera UDP

Las partes que lo componen son muy similares a TCP, lo primero 16 bits indican el puerto origen y los siguientes el puerto destino. Después encontraremos la longitud total del mensaje y el checksum indicando un número para identificar que los paquetes

no fueron alterados, ambos de 16 bits. Y por último tenemos la sección que ocuparán los datos a enviar.

Al tratarse de un protocolo en donde nunca se realiza una confirmación de parte del otro equipo, se considera un protocolo menos confiable, pero más veloz al no realizar tantos pasos al momento de envío y recepción. Es por ello que se utiliza para ofrecer servicios en los que la confirmación de llegada no sea necesaria. Un claro ejemplo de ello es una llamada a través de la red, los paquetes se envían por medio de UDP, esto es debido a que la velocidad con la que deben viajar debe de ser muy rápida y aquí si perdemos algún paquete durante el camino la llamada es interrumpida solo un momento y después para recuperar lo perdido puede pedirse a la otra persona que se encuentra del otro lado que repita lo que dijo y así el proceso de la llamada será mejor desempeñada.

ICMP

Un protocolo utilizado casi a la par de TCP o UDP es el ICMP (Internet Control Message Protocol). Como su nombre lo indica, es el encargado de enviar mensajes de error para llevar un control en la red. Dos muy grandes ejemplos de esos mensajes es el uso de las herramientas ping y traceroute, donde se envían mensajes a un host dentro de la red y las respuestas serán interpretadas, obteniendo resultados como host inalcanzable, el tiempo que toma un paquete en ir y regresar e inclusive el número de host por donde tiene que pasar antes de llegar al destino.

De tal modo, que los mensajes que se obtienen en dicho protocolo son de gran ayuda para el diagnóstico o control de paquetes.

Capa de red, esta capa es la encargada de principalmente dos cosas, lo relacionado con el ruteo, es decir, hasta aquí se selecciona la ruta por donde deberán viajar. Para poder identificar de quién proviene y para quién va dirigido, se hace uso del protocolo IP (Internet Protocol). Y por otro lado tenemos que esta capa es la encargada de empaquetar la información que viajará a través de la red.

Si hacemos una comparación con el servicio postal, la capa de red sería la parte del sobre en donde se coloca el remitente y destinatario. Lo cual corresponde a poner direcciones específicas de cierto sitio para que ahí sea entregada nuestra correspondencia. Así mismo funciona la capa de red, se encarga de poner una dirección de origen y de destino para recibir o enviar, según sea el caso, la información que se prepara para viajar.

Primeramente debemos tener presente que existen dos versiones de direcciones, llamadas IPv4 e IPv6, la primera versión, es la primera dirección que se implementó desde hace años, y la segunda es la versión moderna que en unos años más podremos implementar de manera completa en nuestros dispositivos.

Primero hablaremos de IPv4, fue utilizada por primera vez durante los años 70's. Dicha dirección consta de cuatro octetos separados por puntos para poder obtener una mejor administración de la red (XXX.XXX.XXX.XXX).

Siguiendo con lo anterior, el encapsulado trata de ir dejando una marca para que cada capa correspondiente pueda leer su información e interpretarla de acuerdo a su papel asignado. Y la capa de red maneja el protocolo IP como el más conocido.

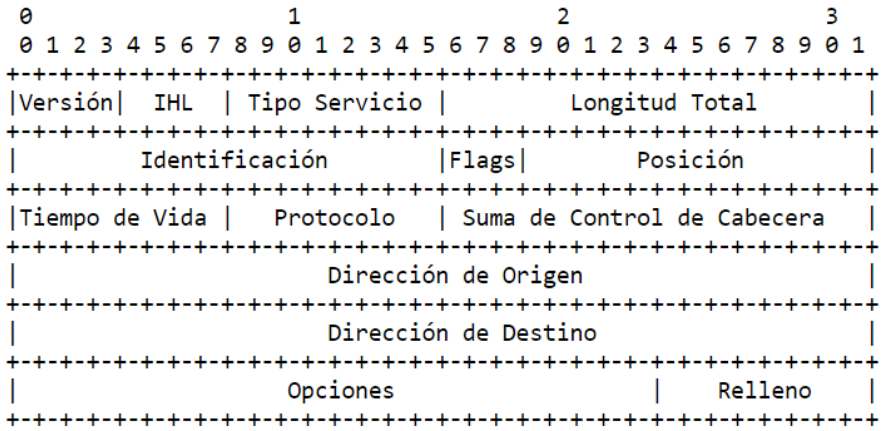

Imagen 11 Cabecera IPv4

Como podemos observar en la imagen 11, es muy similar al encabezado de la capa de transporte, pero aquí podemos encontrar aún más información muy relevante que harán que la información llegue a su destino.

En la imagen 11 vemos que los primero cuatro bits nos indican la versión de protocolo utilizado, después los siguientes cuatro bits nos indican el tamaño de la cabecera. Después encontramos seis bits que indican el tipo de servicio, algunas redes ofrecen un servicio pero por prioridades. Seguimos con 16 bits que nos indican la longitud total del datagramas.

Después tenemos 3 bits que nos indican las banderas que tendrá nuestro paquete, las cuales indican si el paquete se puede fragmentar o es último fragmento. Cuando un paquete es fragmentado, tenemos 13 bits para poder indicar la posición que ocupan.

Tenemos 8 bits que nos indican el tiempo de vida que estará el paquete viajando, es decir, el número de dispositivos que tiene que atravesar para llegar al destino. Después tenemos 8 bits indicando el protocolo que va a ser utilizado (TCP, UDP, etc.). Siguiendo, tenemos 16 bits que representan una suma de control que se utiliza para verificar que el paquete viajó sin modificación, se calcula cada que algún campo tiene distinto valor.

El siguiente apartado de 32 bits nos indica la dirección de la máquina que generó el paquete y los siguientes 32 son para la dirección del equipo que tendrá que ser entregado el paquete. Y finalmente tenemos un apartado de relleno para completar el tamaño establecido (puede o no existir) y el espacio sobrante es para los datos que serán enviados por la red.

Capa de enlace de datos, como pudimos ver, la información viaja hacia una dirección pero para poder ubicar la máquina destino es necesario utilizar la capa de enlace de datos, sus dos funciones son, relacionar la dirección IP con una dirección MAC, que es una dirección muy similar a una IP pero esta es única para cualquier dispositivo, en otras palabras, es un identificador para distinguir los dispositivos. Y por otro lado tiene

la función de identificar tramas (los bits de intercambio con la capa física) para poder hacer la traducción de 1's y 0's y realizar el control del flujo.

Capa física, en esta última capa podremos ver los medios físicos por los cuales podrán ser enviados los bits, en ella se definen si las señales serán analógicas o digitales. En simples palabras se trata de los cables y la forma de mandar un pulso a través de ellos, en el caso de que el medio sea un cable, o por dónde viajarán las ondas en el caso de tratarse de una señal inalámbrica.

2.2.2 Modelo TCP/IP

Transmission Control Protocol/Internet Protocol o Protocolo de Control de Transmisión/Protocolo de Internet, es comúnmente conocido como el protocolo antecesor de lo que ahora conocemos como internet, como lo hemos visto antes, surgió con el proyecto ARPANET, desarrollado por el Departamento de Defensa de los Estados Unidos.

Simplemente es una guía que describe el uso de cada protocolo específico, utilizado para mantener una comunicación entre dos dispositivos en una red. Podemos encontrar paso a paso cómo se realiza la comunicación de un lado a otro. La manera en que son traducidos los datos, direccionados, transmitidos, enlazados y recibidos.

Para poder llevar a cabo todo lo anterior, es necesario dividir los trabajos tal como sucedía en el modelo OSI, crear capas con cierto valor jerárquico y hacer pasar los datos de una en una para poder hacer que lleguen de buena forma a un destinatario, ver imagen 12.

Imagen 12 Modelo TCP/IP

Cada capa tiene como función ir preparando la información tomándola de la capa precedente y pasarla a la capa subsecuente, dependiendo del sentido de la comunicación, pero antes de intercambiar información entre capas se realiza un procedimiento con dirigido única y exclusivamente para la capa del destino equivalente. De este modo tenemos que la información será ordenada y viajará de una mejor manera.

A comparación del modelo OSI, dentro del modelo TCP/IP solo tenemos cuatro capas que se encargan de realizar todo el trabajo de procesar datos para el envío o recepción de información. Pero a pesar de tener menos capas, el modelo TCP/IP es capaz de funcionar de igual modo que el modelo OSI.

Primero dejemos en claro que los protocolos específicos en cada capa no cambian, debido a que solo son eso, un protocolo que puede ser utilizado en cualquier dispositivo y no necesita sufrir algún tipo de edición para poder ser utilizado.

Podemos decir que el modelo TCP/IP es un resumen del modelo OSI, esto es debido a que el primero engloba distintas capas del segundo, pero en una misma capa, como lo veremos más adelante, lo que hace que el trabajo sea más rápido de procesar al tener que ahorrar paso entre más capas.

Comencemos con la capa número cuatro del modelo TCP/IP, lleva por nombre aplicación, y claro que cumple con la misma función que podemos observar en OSI, pero la diferencia radica en que esta capa además de ser equivalente a la capa de aplicación de OSI, también podemos encontrar las capas de presentación y sesión en una misma, para simplificar los modelos, dentro de TCP/IP se unieron esas tres capas que son más similares, las cuales se encargan de la presentación de la información al usuario, la codificación de la información y el control de diálogo. Con esto hay una disminución de información, ya que todo va en una sola capa. Del mismo modo, dentro de esta nueva capa de aplicación, podemos encontrar protocolos correspondientes a las capas de modelo OSI, como son HTTP, FTP, SMTP, MP3, DNS, EBCDIC, entre otros.

La siguiente capa, la número tres, lleva por nombre transporte. En esta capa podemos decir que es la misma capa que la del modelo OSI. Nada cambia, se mantienen los mismos protocolos de transporte. TCP, UDP e ICMP siguen siendo los principales protocolos utilizados para el transporte de información a través de una red. Y es la única capa es que se mantiene exactamente igual en ambos modelos.

Continuando con el orden, tenemos la segunda capa que tiene por nombre Internet. Aquí tenemos que su similitud es con la capa de red del modelo OSI, a decir verdad solo hace un cambio de nombre pero tenemos que lo demás se mantiene. Tiene las mismas funciones que la capa de red, asignación de direcciones de origen y destino dentro de cada paquete. De igual forma, el protocolo que se maneja es el mismo, como ya hemos visto, IPV4 e IPv6, y del mismo modo cumplen con las mismas características dentro de la información que se agrega en esta capa.

Por último tenemos la primera capa del modelo, de nombre acceso a la red, la cual es equivalente a las capas física y de enlace de datos del modelo OSI. En esta capa se especifica la forma en la cúal los paquetes se enrutan sin importar el tipo de red que se esté utilizando, es decir, la capacidad que existe para acceder a cualquier red con cualquier tipo de conexión (alámbrica o inalámbrica). Los protocolos manejados dentro de esta capa igual son los mismos que se utilizan en las dos capas del modelo OSI, ethernet (para cableados), 802.11 (para inalámbrica), RJ45 (para conectores), entre otras.

Ambos modelos se basan en una estructura de capas, dividir el trabajo, y por lo que hemos visto dichas capas son muy similares, pero a pesar de ello el protocolo más utilizado es el TCP/IP para la gran mayoría de comunicaciones que se realizan hoy en día, es por ello que se considera la base para la red conocida como internet.

2.3 Red Peer to Peer

Todo lo anterior, son las bases para poder realizar la comunicación entre dos dispositivos sin importar el lugar donde se encuentren, siempre y cuando estén conectados mediante algún medio, físico o inalámbrico.

Ahora bien, para poder hacer esto a grandes escalas, se utiliza el modelo de red Peer to Peer, se trata nada más y nada menos de una tecnología utilizada hace más de diez años, utilizada muy comúnmente para realizar intercambios de archivos. En este tipo de redes no es necesario tener un servidor central que atienda a todos los clientes, cada equipo conectado a dicha red pueden tener ambos roles, ser servidores y clientes al mismo tiempo por lo cual también puede conocerse como red entre iguales.

Dado que el uso más común de las redes P2P es el intercambio de archivos sin un intermediario que regule ese tráfico, tienen la mala reputación de ser utilizada por algunos usuarios para compartir contenido sujeto a las leyes de copyright, por lo cual su uso siempre será en gran polémica entre si es bueno utilizar dicho tipo de red o no.

Sin embargo, mientras que esa parte de usuarios que dan un mal uso a las redes P2P, existe otro lado bueno que realmente sabe aprovechar el protocolo P2P compartiendo archivos realmente importantes o realizando actividades que realmente son licitas.

Pero para poder explicar mejor una red P2P, es necesario explicar un poco el funcionamiento de la comunicación Cliente-Servidor, que es la parte fundamental de este tipo de estructura.

2.4 Cliente-Servidor

Este tipo de comunicación es prácticamente unos de los pilares de las redes P2P, consiste básicamente en 2 equipos conectados entre sí sin importar el medio. Uno de esos equipos será el que ofrezca un servicio y el otro será un cliente que requiere dicho servicio.

Primeramente, antes de realizar la conexión entre los dos dispositivos, por parte del servidor, se hace una negociación con el sistema operativo para obtener un puerto por donde esperar las peticiones, comúnmente es un puerto del rango de los conocidos, es decir, los que tienen un servicio ya definido (1-1024), y el cliente hace lo mismo con su sistema operativo, solo que en este caso la negociación es respecto a los puertos mayores a 1024 (1025-65535).

Una vez teniendo el puerto seleccionado en cada dispositivo, simplemente se realiza la petición TCP o UDP, según sea el caso que se ocupe dentro de la estructura, para comenzar una petición del cliente hacia el servidor.

En el caso de TCP comenzamos, como ya lo hemos visto, con un Three Way Handshake para sincronizar los dispositivos y comenzar la comunicación, y en el caso de UDP simplemente se envían los paquetes con la información del archivo compartido, y claro, todo lo anterior con sus respectivos puertos de origen y destino que ocupará la aplicación.

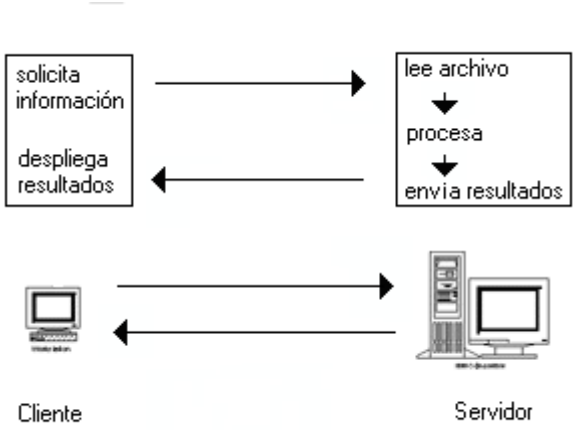

Imagen 13 Cliente-Servidor

La imagen 13 es un gran claro ejemplo del modo en el que el cliente hace la solicitud a un servidor, esté procesa la información y realiza una respuesta al cliente, y así continúan hasta que uno de los dos termina la conexión.

Una vez teniendo más claro el concepto de cliente servidor, podemos ver que así funcionan las redes Peer to Peer, pero en este caso no hay un servidor central que atienda todas las peticiones, sino que cada equipo conectado actúa como servidor o cliente, dependiendo el caso.

La primera aplicación en utilizar este tipo de estructura fue “Hotline Connect”, desarrollada en 1996 para el sistema operativo de Apple (Mac OS), por un joven australiano de nombre Adam Hinkley, la cual pretendía ser una plataforma destinada a la distribución de archivos para empresas y universidades, sin embargo, se comenzó a utilizar para el intercambio de contenido ilegal. Pero a pesar de esto también había contenidos completamente legales. Este sistema era una red descentralizada, es decir, no se utilizaban servidores centrales, sino más bien, los archivos se almacenaban en los dispositivos de cada usuario y ellos mismos tenían la función de actuar como servidor para realizar la distribución a otro usuario que solicitara el contenido. Y eso era una gran ventaja, en una red común, se tiene un servidor central distribuyendo y atendiendo peticiones, y en caso de que dicho servidor sufriera algún percance o se

cerrara, no podríamos continuar con nuestra descarga y perderíamos el archivo, obligando al usuario a comenzar una nueva descarga desde cero en otro servidor.

Pero por otro lado, con una red Peer to Peer no sucede esto, simplemente cuando perdemos comunicación con algún dispositivo en la red que actué como servidor, solo basta con buscar otro equipo o "servidor" para continuar la descarga del archivo.

Como vimos anteriormente, la famosa red Peer-to-Peer es un arreglo de dispositivos encargada de conectar nuestro protocolo Bitcoin, que veremos más adelante, para que pueda existir una comunicación entre dichos dispositivos. Y para ello debemos de tener muy en cuenta los protocolos que se manejan para poder completar nuestra comunicación.

Para ello nos centraremos en el modelo TCP/IP, que como ya vimos, es la base para que exista el internet. Nos concentraremos más en la parte de los protocolos manejados en a capa de transporte de dicho modelo. Vimos que los protocolos principales empleados son el TCP y el UDP, además también vimos que la diferencia entre esos protocolos es una confirmación que se utiliza en TCP y que UDP carece. Pues bien, hablando en comunicaciones muy generales, prácticamente el protocolo UDP es utilizado muy comúnmente para comunicaciones como **Streaming** o **VoIP**, es to es debido a que solamente se dedica a repartir esos paquetes de información sin la necesidad de requerir una confirmación de recibido. Por ejemplo, al realizar un una llamada telefónica por medio de VoIP, la información simplemente es enviada sin requerir una confirmación por parte del protocolo, sin embargo, cuando alguno de esos paquetes llega corrupto o incompleto, simplemente hace falta que la otra persona que se encuentra al otro lado pida que se repita la información que no llegó. Esto tiene una gran ventaja, pues el protocolo al no pedir que se envíe una confirmación de llegada antes de enviar el siguiente paquete, hace que la velocidad de transferencia sea ligeramente mayor a la de TCP. Este ejemplo es muy sencillo pero es un muy bueno para poder entender un poco mejor dicha diferencia.

Por otro lado tenemos que TCP no se queda atrás, la gran ventaja que tiene sobre UDP es la seguridad que ofrece al transferir la información sin importar el servicio ofrecido. Simplemente al momento de realizar una conexión a otro dispositivo tenemos lo que hemos visto como el Three Way Handshake, que es la parte del protocolo que realiza una sincronización entre ambos nodos.

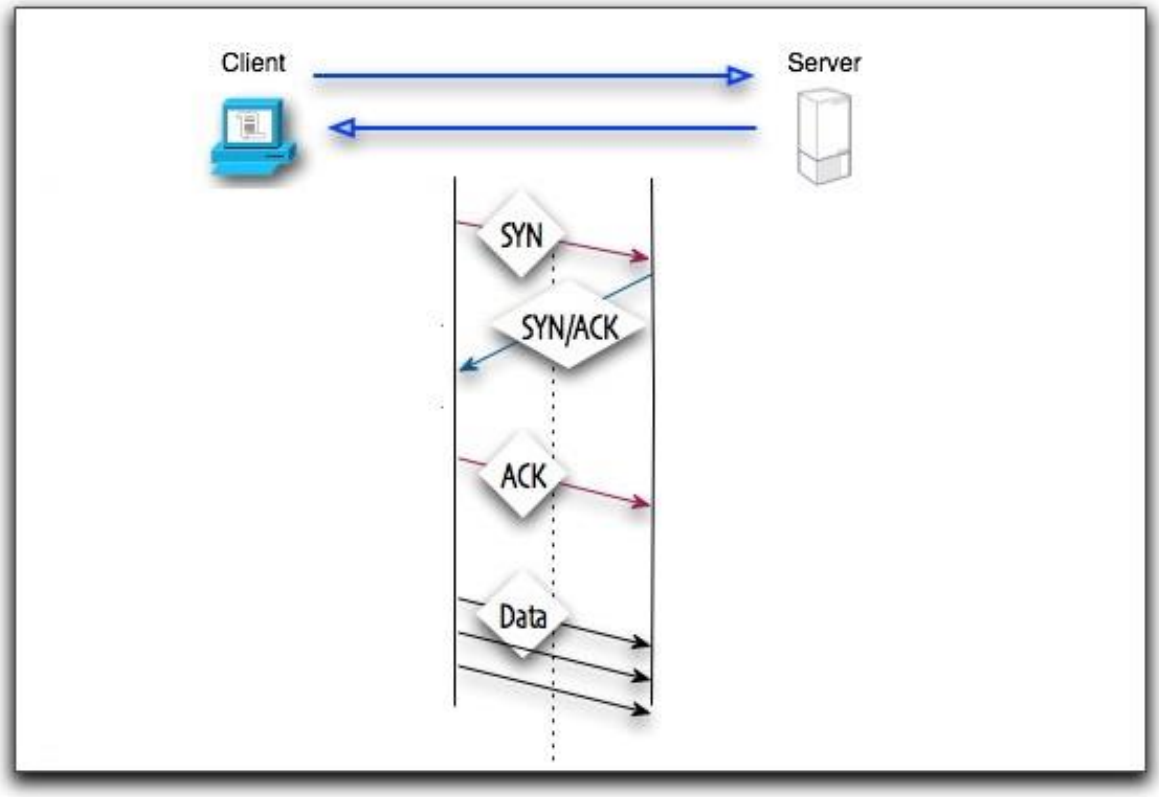

Imagen 14 Three Way Handsake

Como podemos ver en la imagen 14, para iniciar el primer dispositivo debe mandar un paquete con la bandera de SYN encendida, ejemplificándolo decimos que es un saludo “Hola”, al recibirla el segundo dispositivo responde con un SYN-ACK, es aquí cuando se responde con otro “hola” que es una confirmación de que se recibió el saludo. Y por último el primer dispositivo envía su confirmación mediante una bandera ACK y así comienza el envío de información. Esta es una forma muy sencilla y práctica de poder ver que para poder enviar un paquete tras otro es necesaria una confirmación de que el primero fue recibido.

Es esa parte la que aumenta la seguridad del protocolo TCP ante el UDP, no solo se puede inundar un dispositivo con paquetes TCP como sucedería con UDP.

Las redes P2P a su vez pueden clasificarse en distintos tipos, de entre los cuales hay tres que son los más importantes.

Red P2P centralizada, se tiene un servidor central en el cual se aloja una lista de los nodos que pertenecen a la red y el contenido que se busca. El defecto más grande que encontramos aquí es que si ese servidor central falla, la red misma dejaría de funcionar.

Red P2P descentralizada y estructurada, muchas veces se le conoce como red hibrida, en este caso no se tiene un servidor central, sino que son varios servidores actuando como un servidor central para agilizar el tráfico de la red.

Red P2P descentralizada y no estructurada, no existe un nodo o nodos centrales sino más bien cada nodo conectado en la red tiene la función de ser servidor y cliente al mismo tiempo con lo cual ganamos mucha más velocidad al momento de compartir archivos.

Entre las principales características que podemos encontrar en las redes P2P tenemos la descentralización, la cual nos lleva a que todos los dispositivos conectados serán iguales entre sí. Además tenemos la robustez, al tener conectados un gran número de equipos a la red más aumenta el tamaño de procesamiento que se comparte entre sí, aumentando los recursos en general de la red.

Ahora bien, al estar dentro de una red P2P, los usuarios aportan algo y reciben algo a cambio. Lo más común es compartir recursos a cambio de recursos, esto va dependiendo de uso que se dé a la red, los recursos van desde archivos, ancho de banda, ciclos de proceso o hasta almacenamiento en disco.

Entonces podremos observar que las redes P2P a pesar de ser muy comúnmente catalogadas como contenedoras de virus, también pueden llegar a ser utilizadas para el

bien como lo es la red destinada a almacenar, crear y utilizar bitcoins. Sin embargo además de compartir recursos, debemos proteger dicha información que viajará por la red.

2.5 Criptografía

Uno de los puntos más importantes dentro de las monedas virtuales descansa en la criptografía, y para ello es importante revisar en que conociste y cómo es que actúa para convertir a los coins en un sistema seguro para los usuarios.

Primeramente, el término criptografía proviene del griego criptos (oculto) y graphos (escritura), y se puede definir como una ciencia dedicada a estudiar mensajes ocultos. Aunque ese es el objetivo principal, no solo se encarga de esconder, sino que también resulta útil para proteger archivos contra modificaciones así como para comprobar su procedencia. En este caso, nos concentraremos más en ese tema.

Para poder ocultar un mensaje se utiliza una técnica llamada “cifrado”, la cual tiene como objetivo el mantener segura cierta información confidencial entre el emisor y el receptor.

El procedimiento para cifrar un mensaje es simple, el emisor envía el mensaje y lo hace pasar por un algoritmo para cifrar auxiliado por una llave o clave, que se utiliza para realizar proteger dicho cifrado, de tal modo que obtenemos un archivo prácticamente imposible de comprender a simple vista, de esta manera el mensaje puede viajar por un canal de transmisión con la seguridad de que a pesar de ser interceptado no podrá ser leído; y cuando el texto cifrado o cripto llega al destinatario, esté con ayuda de la llave y un algoritmo de descifrado, puede recuperar el mensaje.

Los orígenes de la criptografía datan desde la antigüedad, cuando surgió la necesidad de comunicarse con otras personas de una manera más privada y los primeros intentos por realizar esta actividad consistían en realizar escritos cifrados mediante ciertos algoritmos que realizan una mezcla de los caracteres del escrito; o tratar de esconder

el mensaje en algún lugar, como en los cuerpos de los esclavos o en algún objeto, para hacer imposible su localización.

Pero con el paso del tiempo ese tipo de algoritmos pasaron a ser obsoletos, y sin embargo las necesidades seguían existiendo, por lo cual el ingenio del hombre fue avanzando hasta puntos, en los que tratar de descifrar un mensaje, sin tener una llave llevaría años y años de trabajo.

Bien podemos definir que la criptografía está dividida en dos, criptografía simétrica y criptografía asimétrica. Ambas tienen el mismo propósito de cifrar archivos pero utilizando algún tipo de función matemática, pero la diferencia radica principalmente en la forma de hacerlo.

La criptografía simétrica suele ser aparentemente mucho más sencilla, ya que para poder realizar un cifrado de este tipo simplemente se necesita tener una llave para cifrar y con la misma para descifrar (más adelante veremos que son las llaves).

En cambio, para realizar un cifrado con criptografía asimétrica, se emplean dos llaves, una llave pública y una llave privada. Dichas llaves se utilizar para realizar el cifrado, utilizando una para cifrar y la otra para descifrar.

Además de utilizar los cifrados para poder ocultar información, también podemos emplearlos para realizar firmas digitales, porque gracias a esto podemos verificar la autenticidad de un archivo con la seguridad de saber que será demasiado difícil que alguna persona realice acciones maliciosas.

Antes de continuar con el tema, debemos de tener muy en cuenta las operaciones lógicas que podemos realizar con bits, es decir, operaciones AND, OR, NOT y XOR.

Dichas operaciones son de gran utilidad dentro de los cifrados.

X (AND) Y = Multiplicación bit a bit de las dos variables (1 AND 1 = 1 | 0 AND 1 = 0 | 1 AND 0 = 0 | 0 AND 0 = 0).

X (OR) Y = Suma bit a bit de las dos variables (1 OR 1 = 1 | 0 OR 1 = 1 | 1 OR 0 = 0 | 0 OR 0 = 0).

X (XOR) Y = Disyunción Exclusiva[1] bit a bit de las dos variables (1 XOR 1 = 0 | 0 XOR 1 = 0 | 1 XOR 0 = 1 | 0 XOR 0 = 0).

(NOT) X = Multiplicación bit a bit de las dos variables (NOT 1 = 0 | NOT 0 = 1).

Por ejemplo:

0110 1100 1011 1001 1101 0010 0111 1011

AND 0110 0101 1100 0001 0110 1001 1011 0111

= 0110 0100 1000 0001 0100 0000 0011 0011

0110 1100 1011 1001 1101 0010 0111 1011

OR 0110 0101 1100 0001 0110 1001 1011 0111

= 0110 1101 1111 1001 1011 1011 1111 1111

0110 1100 1011 1001 1101 0010 0111 1011

XOR 0110 0101 1100 0001 0110 1001 1011 0111

= 0000 1001 0111 1000 1011 1011 1100 1100

NOT 0110 0101 1100 0001 0110 1001 1011 0111

= 1001 1010 0011 1110 1001 0110 0100 1000

[1] La Disyunción exclusiva, se refiere a que la operación siempre será verdadera cuando X y Y tienen valores diferentes

Tenemos tres tipos de cifrados, cifrados simétricos, asimétricos e hibridos. El cifrado simétrico consta de utilizar una clave para cifrar y descifrar unmensaje, el punto débil de este sistema de cifrado es que la si la clave es interceptada cualquiera podría interpretar dicho mensaje, con lo cual podemos comprobar que laseguridad recae sobre la robuztes de la clave utilizada y no sobre el algoritmo.

Por otro lado, un cifrado asimétrico esta basado en el uso de dos distintas claves, una clave pública que puede ser compartida sin ningún temor y una clave privada que no debe ser revelada jamás. Ambas claves se generan simultáneamente y están ligadas la una con la otra.

Ahora bien, el cifrado asimétrico le podemos dar dos usos distintos, cifrar un mensaje, fimar un mensaje o autenticar un mensaje. De modo simple, para cifrar un mesaje simplemente se realiza con la clave pública de otro usuario y él será el único que podrá ver el contenido pues esa clave pública estará relacionada con su clave privada. Para realizar una firma con cifrado asimétrico, basta con realizar un cifrado con la clave privada y compartir la clave pública, cualquier persona con acceso a dicha clave pública puede comprobar si realmente del autor del mensaje, con lo cual obtenemos una firma o auteticación del usuario, ver imagen 15.

Imagen 15 Cifrado asimétrico

Existen diferentes tipos de algoritmos que permiten hacer el cifrado de un archivo o mensaje, en el caso de los bitcoins solamente nos centraremos en las funciones resumen SHA-256 y RIPMD-160, solamente tomaremos en cuenta como antecedente el algoritmo MD5 y SHA en sus dos primeras versiones, además, veremos como trabaja el algoritmo ECDSA. Los algoritmos de Hash o resumen, como su nombre lo indica, hace un resumen del mensaje, en este caso se utilizan dos algoritmos, RIPMD-160 para generar un resumen muy corto y SHA-256 para generar un resumen más

seguro, una vez que combinamos ambos algoritmos obtenemos una estructura que no es vulnerable tan fácilmente. Además, a dicho resumen, se le agrega un algoritmo más para comprobar la identidad de cada bitcoin, curvas elípticas o ECDSA, un algoritmo capaz de utilizar claves cortas y al mismo tiempo tener la misma seguridad de que otros algoritmos con claves grandes.

2.5.1 Hash

Los hash o también conocidos como función resumen, son algoritmos comúnmente utilizados para poder cifrar mensajes, así como, para poder realizar firmas digitales. A partir de una entrada (un mensaje, contraseña o incluso un archivo) se genera una salida alfanumérica la cual tiene una longitud fija, que es lo que se conoce como resumen.

Si bien este algoritmo es utilizado para la criptografía, tiene otro propósito que van más allá de eso. Entre sus varios cometidos podemos encontrar el aseguramiento de que un archivo no se ha modificado, mediante una firma digital o poder hacer ilegible una cadena de texto.

Para realizar este sistema de criptografía se realizan una serie de algoritmos los cuales aseguran que si se modifica el resumen el mensaje original no podrá ser recuperado. Con lo que se considera un algoritmo unidireccional, ya que a partir de un mensaje se genera un hash, pero no se puede recuperar un mensaje sin conocer la llave partiendo solamente de un hash.

Los hashes se utilizan comúnmente para determinar si se ha manipulado un mensaje que viaja a través de un canal no seguro, siempre y cuando el transmisor y el receptor compartan la clave secreta. Esto funciona con lo antes mencionado, si se altera el mensaje el valor del hash cambia y por consecuencia es fácil comprobar que se trata de una falsificación.

Existen varios tipos de hashes para poder comprobar la autenticación de los mensajes, entre los cuales se encuentran el MD5 (Message-Digest Algorithm 5 o Algoritmo de Resumen de Mensajes 5), SHA (Secure Hash Algorithm 1 o Algoritmo de Hash Seguro 1) y RIPEMD.

2.5.2 MD5 (Message-Digest Algorithm 5)

Una vez iniciada la época tecnológica, este algoritmo de cifrado fue uno de los más utilizados e incluso es base para las nuevas generaciones.

El MD5 es un algoritmo de reducción con una salida de 128 bits ampliamente creado en 1992 por Ronald Rivest, criptógrafo del MIT.

El algoritmo es una función de cifrado hash que acepta cadenas, de cualquier tamaño, como entrada y devuelve una salida fija de 128 bits. La más grande ventaja de este algoritmo es que prácticamente es imposible construir el mensaje a partir del resultado, así como, es casi imposible que dos cadenas o hashes se generen con el mismo contenido. Digo “casi” porque siempre habrá la manera poder hacer que el algoritmo falle. Pero cuando los contenidos coinciden se conoce como colisiones.

Para poder realizar el algoritmo de hash MD5 es necesario seguir los siguientes pasos:

1) *Adicionar bits de relleno.*

El mensaje en claro es llenado con n número de bits hasta que sea necesario, comenzando con un 1 y rellenando con 0 los lugares restantes hasta que su longitud sea exactamente 64 bits inferior a un múltiplo de 512.

2) *Adicionar bits correspondientes a la longitud del mensaje.*

Se añaden 64 bits correspondientes al número que representa la longitud del mensaje, colocándolos desde le byte menos significativo al más significativo, formando un número entero de bloques de 512 bits.

3) *Se inicializan los vectores A, B, C y D*

Para poder hacer la reducción se establecen cuatro registros de 32 bits con los siguientes valores.

A = 67 45 23 01
B = EF CD AB 89
C = 98 BA DC FE
D = 10 32 54 76

Dado que MD5 trabaja con un formato llamado little-endian, el cual se refiere a que el byte de menor valor se almacena en la dirección más baja de la memoria y el byte de mayor valor se almacena en la dirección más alta, el vector queda de la siguiente manera.

A = 01 23 45 67
B = 89 AB CD EF
C = FE DC BA 98
D = 76 54 32 10

4) *Se procesan los bloques*

En este paso es donde comienzan las operaciones principales, todo se divide en cuatro rondas de 16 iteraciones cada una, las siguientes operaciones corresponden a las operaciones que se realizan por ronda:

1. F(B,C,D) = (B AND C) OR ((NOT B) AND D)
2. G(B,C,D) = (B AND D) OR ((C AND (NOT D))
3. H(B,C,D) = B XOR C XOR D
4. I(B,C,D) = C XOR (B OR (NOT D)

La tabla 1 representa los valores de las funciones F, G, H e I previamente calculados.

Primera Ronda:	Tercera Ronda:
F F (D76AA478)	H H (FFFA3942)
F F (E8C7B756)	H H (8771F681)
F F (242070DB)	H H (6D9D6122)
F F (C1BDCEEE)	H H (FDE5380C)
F F (F57C0FAF)	H H (A4BEEA44)
F F (4787C62A)	H H (4BDECFA9)
F F (A8304613)	H H (F6BB4B60)
F F (FD469501)	H H (BEBFBC70)
F F (698098D8)	H H (289B7EC6)
F F (8B44F7AF)	H H (EAA127FA)
F F (FFFF5BB1)	H H (D4EF3085)
F F (895CD7BE)	H H (04881D05)
F F (6B901122)	H H (D9D4D039)
F F (FD987193)	H H (E6DB99E5)
F F (A679438E)	H H (1FA27CF8)
F F (49B40821)	H H (C4AC5665)
Segunda Ronda:	**Cuarta Ronda:**
G G (F61E2562)	I I (F4292244)
G G (C040B340)	I I (432AFF97)
G G (265E5A51)	I I (AB9423A7)
G G (E9B6C7AA)	I I (FC93A039)
G G (D62F05D)	I I (655B59C3)
G G (02441453)	I I (8F0CCC92)
G G (D8A1E681)	I I (FFEFF47D)
G G (E7D3FBC8)	I I (85845DD1)
G G (21E1CDE6)	I I (6FA87E4F)
G G (C33707D6)	I I (FE2CE6E0)
G G (F4D50D87)	I I (A3014314)
G G (455A14ED)	I I (4E0811A1)
G G (A9E3E905)	I I (F7537E82)
G G (FCEFA3F8)	I I (BD3AF235)
G G (676F02D9)	I I (2AD7D2BB)
G G (8D2A4C8A)	I I (EB86D391)

Tabla 1 Rondas MD5

Ahora, para poder realizar la compresión del hash, se representan otras cuatro funciones con las cuales se mezcla el mensaje con las variables previamente calculadas:

1. FF representa A= B + ((A + F (B, C, D) + M[i] +K[i]) <<< s)
2. GG representa A= B + ((A + G (B, C, D) + M[i] + K[i]) <<< s)
3. HH representa A= B + ((A + H (B, C, D) + M[i] + K[i]) <<< s)
4. II representa A= B + ((A + I (B, C, D) + M[i] + K[i]) <<< s)

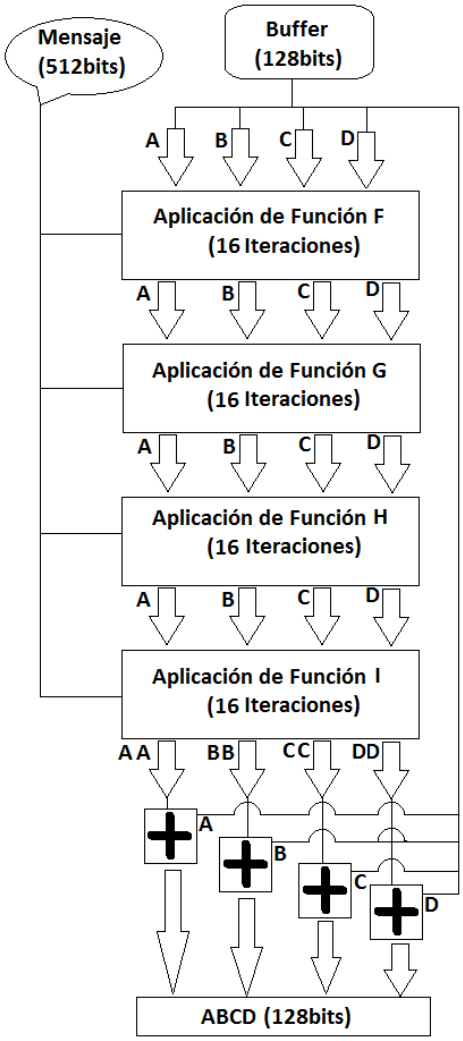

Imagen 16 Algoritmo MD5: Message-Digest Algorithm 5

La imagen 16 representa las operaciones realizadas en cada una de las iteraciones de cada ronda, además podemos como se desplaza circularmente el valor de "a" "s" bits a la izquierda.

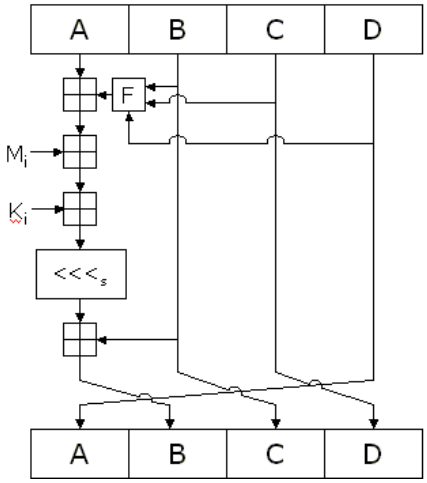

Imagen 17 Iteraciones MD5

5) Resumen

Para este último paso, solamente basta con sumar los valores resultantes de *A, B, C y D* con AA, BB, CC y DD. Procesando este último bloque, para el resultado o resumen, se concatena el valor de la última suma, como podemos ver en la imagen 18.

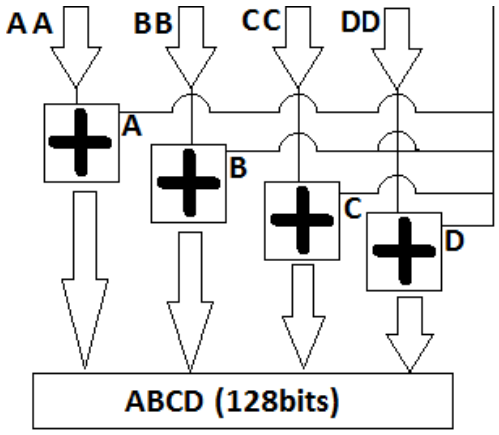

Imagen 18 Resumen MD5

Tenemos 8 valores almacenados, ahora simplemente para terminar el algoritmo y obtener nuestro hash, solo basta con realizar las siguientes operaciones y cada resultado lo iremos concatenando.

$$ABCD = \begin{cases} A' = A + AA \\ B' = B + BB \\ C' = C + CC \\ D' = D + DD \end{cases}$$

En los últimos tiempos el algoritmo MD5 ha ido mostrando debilidades, por lo cual tiende a ser el menos utilizado, sin embargo si lo utilizamos del modo correcto puede llegar a ser uno de los más seguros y confiables.

2.5.3 SHA (Secure Hash Algorithm)

Es un algoritmo diseñado por la NSA (National Security Agency), de los Estados Unidos, considerada como un estándar por el NIST, su primera versión es llamada SHA (ahora llamado SHA-0) y fue publicada en 1993, pero tiempo después cayó al ser descubiertas sus debilidades, en 1995 la NSA reemplazo el algoritmo por lo que se conoce como SHA-1.

SHA-0 es muy similar a MD5, pero la diferencia es que SHA-0 entrega un resumen de 160 bits, también cuenta con 4 rondas e igualmente se realiza un llenado del mensaje comenzando por un 1 seguido de los 0 que sean necesarios. A diferencia de MD5, SHA-1 emplea cinco registros de 32bits en lugar de cuatro.

Al igual que MD5, se emplean 5 pasos para realizar un SHA-0, los primero dos son idénticos a MD5. Se realiza un relleno comenzando por un 1 seguido de los 0 necesarios para de modo que la longitud sea de 64 bits antes de un múltiplo de 512. De la misma manera, el siguiente paso es rellenar esos 64 bits faltantes con un valor que represente la longitud del mensaje.

Para el tercer paso, se generan los siguientes registros auxiliares:

A= 67452301
B= EFCDAB89
C= 98BADCFE
D= 10325476
E= C3D2E1F0

Una vez inicializados, se asignan en cinco variables (A, B, C, D y E) respectivamente. Después se inician las operaciones con la ayuda de las siguientes expresiones.

F (B, C, D) = (B AND C) OR ((NOT B) AND D) → con t = 0 a 19
G (B, C, D) = B XOR C XOR D → con t = 20 a 39 y t = 70 a 79
H (B, C, D) = (B AND C) OR (B AND D) OR (C AND D) → con t = 40 a 59

Del mismo modo que MD5, cada función se repite por ronda. En la primera ronda se emplea la función F (toma valores de 0 a 19), en la segunda ronda la función G (valores de 20 a 39), en la tercera la función H (valores entre 40 y 59) y para este algoritmo en la cuarta ronda se emplea la función G nuevamente (valores entre 60 y 79).

Para poder tener un orden dentro de cada ronda se emplea una constante llamada K_t, la cual obtiene un distinto valor para cada ronda, donde se cumple que "t" se encuentra entre 0 y 79 y se le asignan los siguientes valores por ronda.

Ronda1 → K_t = 5A827999
Ronda2 → K_t = 6ED9EBA1
Ronda3 → K_t = 8F1BBCDC
Ronda4 → K_t = CA62C1D6

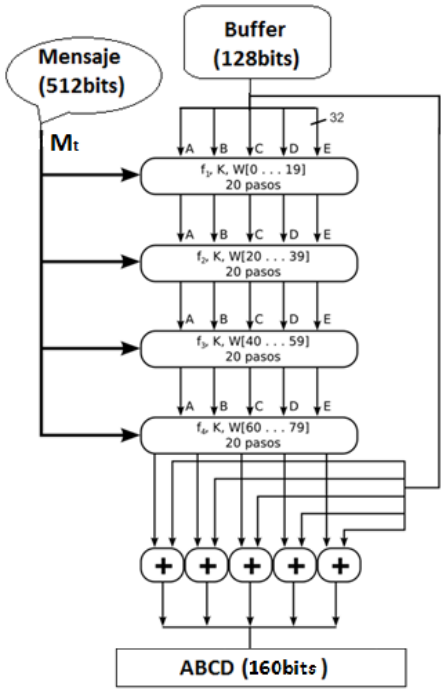

Imagen 19 Algoritmo SHA: Secure Hash Algorithm

En el caso de SHA-0, las funciones se aplican 20 veces por ronda de acuerdo con la imagen 19.

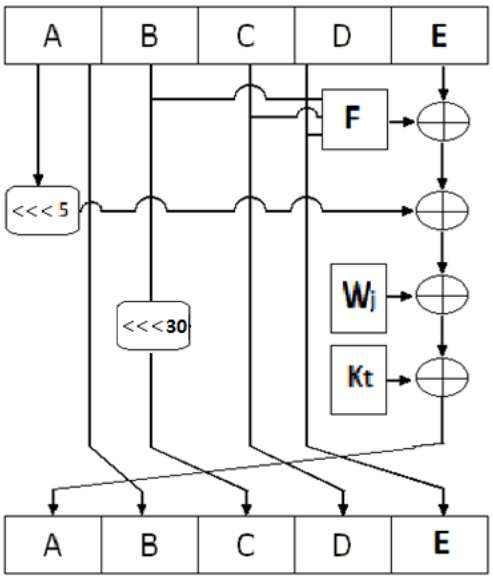

Imagen 20 Iteraciones SHA

A’ = E + F (B, C, D) + (A<<<5) + Wj + K_t

B’=A

C’= (B<<<30)

D’= C

E’= D

<<< Rotación circular a la izquierda

La manera de poder obtener el los valores de W_j es por medio de una función que se utiliza para expandir el mensaje, quedando de la siguiente manera.

Para 0 ≤ j ≤ 15

W_j=W_t

Para 16 ≤ j ≤ 79

W_j= $W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3}$

W_j Representa el número de iteración que en este caso va del 0 al número 79.

A<<<5 Representa el valor de A desplazado 5 veces a la izquierda.

El resultado final se obtiene al realizar la suma de las 5 variables finales de las 80 rondas, más las variables A, B, C, D, y E originales, obtenido una cadena de 160 bits se obtiene de la suma de la cadena de entrada con la cadena resultante del último bloque (la suma a realizar es en modulo 2^{32}). Para fines más prácticos, la cadena de 0’s y 1’s de longitud 160bits, es convertida a formato hexadecimal.

La más grande ventaja de SHA-0 con respecto a MD5, es que la salida de SHA-0 genera un resumen de 160 bits mientras que MD5 genera una salida de 120. Lo que hace que SHA-0 sea más robusto y al mismo tiempo más seguro que MD5, por otro lado, al emplear más operaciones SHA se convierte en un algoritmo más lento, pero mucho más confiable.

SHA-1

Como bien se mencionó, el algoritmo de SHA-0 comenzó a tener debilidades, más específicamente, comenzaron a encontrar el modo de poder realizar colisiones, con lo cual el tiempo para poder "romper" el algoritmo fue disminuyendo. Sin embargo, la NSA, para poder mejorar el algoritmo simplemente a la función que expande el mensaje, se tuvo que agregar un desplazamiento a la función, dando como resultado lo siguiente:

$$W_j = \left(W_{j-16} \oplus W_{j-14} \oplus W_{j-8} \oplus W_{j-3}\right) <<< 1$$

Después simplemente lo hace falta seguir los mismos pasos de SHA-0 para poder obtener un hash con más seguridad. Sin embargo, con el paso del tiempo investigadores australianos descubrieron un método con el cual se redice la probabilidad de tener dos hashes iguales.

Teóricamente, como de SHA-1 se obtiene una salida de 160bits, tenemos que la probabilidad de encontrar un hash es de 2^{160}, es decir, 1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,976 posibilidades de obtener el resultado. Todo ello utilizando fuerza bruta para obtener el mensaje en claro.

SHA-2

Después de varios años, el algoritmo SHA comenzó a tener debilidades, debido al aumento de potencia informática. Y finalmente en el 2005, detectaron que el algoritmo tenía fuertes debilidades matemáticas por lo cual fue necesario elaborar otro algoritmo que fuese más robusto, para poder brindar mayor seguridad. El resultado del nuevo algoritmo fue una serie de funciones que a las cuales se les llamo SHA-224, SHA 256, SHA-384 y SHA-512, todas diseñadas por la Agencia de Seguridad Nacional (NSA). Primero todos fueron conocidos como SHA-2, sin embargo, tiempo después cada uno fue separado y nombrado por su número de bits de salida.

Actualmente el algoritmo más utilizado en todo el mundo es el SHA-256, esto es debido a que en SHA-1 han encontrado vulnerabilidades que lo hacen cada vez menos seguro, y por ende, se utiliza SHA-256. Sin embargo, tenemos la opción de seleccionar algoritmos aún más robustos, pero, debemos recordar que para que esos algoritmos sean más fuertes, es necesario aumentar el tamaño de las salidas, y por consecuencia tenemos operaciones más grandes por lo que la velocidad de cálculo aumenta considerablemente haciendo que el algoritmo mientras más robusto se requiera tarde más en ser calculado.

Una vez teniendo presentes los algoritmos anteriores, es más sencillo entender el algoritmo de SHA-256 y a su vez podremos observar que efectivamente es más robusto respecto a sus antecesores.

Como SHA-256 está basado en sus antecesores, debemos comenzar de igual modo con un relleno del mensaje para hacer que cumpla con 512 bits. Para ello hacemos que el mensaje sea de longitud 448 mod 512.

Los siguientes 64 bits restantes, se utilizan para colocar el tamaño original del mensaje. Para final mente obtener una cadena de 512bits lista para ser cifrada y resumida.

El siguiente paso para obtener un hash con SHA-256 es establecer los valores de los vectores auxiliares utilizados para el proceso. En este caso se hace uso de ocho vectores.

A = 6A09E667
B = BB67AE85
C = 3C6EF372
D = A54FF53A
E = 510E527F
F = 9B05688C
G = 1F83D9AB
H = 5BE0CD19

El siguiente paso es operar los vectores A-H con las siguientes funciones auxiliares.

$$M_{aj}(A,B,C)= (A \wedge \mathrm{B}) \oplus (A \wedge \mathrm{C}) \oplus (B \wedge \mathrm{C})$$

$$Ch(E,F,G)= (E \wedge \mathrm{F}) \oplus (E \wedge \mathrm{G})$$

Las operaciones utilizadas para hacer la compresión del mensaje son las siguientes:

Para $0 \leq j \leq 15$

$$W_j = W_t$$

Para $16 \leq j \leq 63$

$$W_j = \sigma_1(W_{j-2}) \oplus W_{j-7} \oplus \sigma_0(W_{j-15}) \oplus W_{j-16}$$

Las funciones σ_0 y σ_1 se obtienen de la siguiente manera.

$$\sigma_0 (x) = (x >>> 7) \oplus (x >>> 18) \oplus (x >> 3)$$

$$\sigma_1 (x) = (x >>> 17) \oplus (x >>> 19) \oplus (x >> 10)$$

>>> Rotación circular a la derecha

>> Desplazamiento a la derecha

Además necesitaremos otras dos funciones más para poder operar todo el hash, las funciones son las siguientes.

$$\Sigma_0 (x) = (A >>> 2) \oplus (A >>> 13) \oplus (A >>> 22)$$

$$\Sigma_1 (x) = (E >>> 6) \oplus (E >>> 11) \oplus (E >>> 25)$$

Siguiendo el esquema de la imagen 21, realizamos la expansión del mensaje para obtener las 64 rondas.

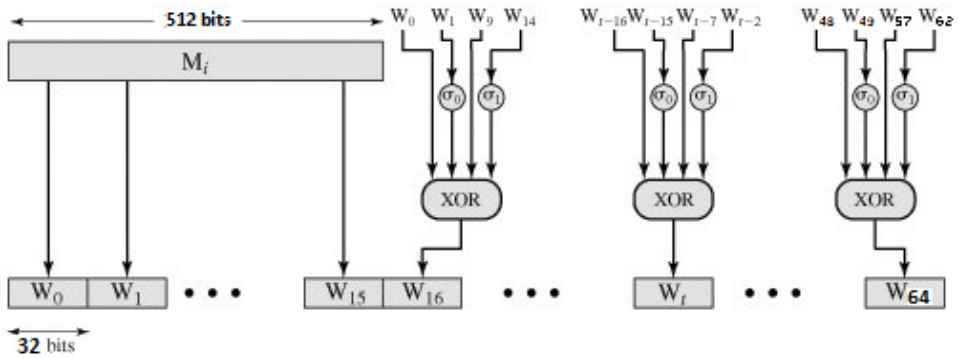

Imagen 21 Rondas SHA-256

Para el valor de K_t se emplean los primeros 32bits de las partes faccionarias de las raíces cubicas de los primeros 64 números primos. Pero para al ser valores constantes, los obtenemos de la siguiente lista (por columna).

0x428a2f98	0x71374491	0xc19bf174	0xd5a79147	0xc76c51a3	0x78a5636f
0x71374491	0xb5c0fbcf	0xe49b69c1	0x06ca6351	0xd192e819	0x84c87814
0xb5c0fbcf	0xe9b5dba5	0xefbe4786	0x14292967	0xd6990624	0x8cc70208
0xe9b5dba5	0x3956c25b	0x0fc19dc6	0x27b70a85	0xf40e3585	0x90befffa
0x3956c25b	0x59f111f1	0x240ca1cc	0x2e1b2138	0x106aa070	0xa4506ceb
0x59f111f1	0x923f82a4	0x2de92c6f	0x4d2c6dfc	0x19a4c116	0xbef9a3f7
0x923f82a4	0xab1c5ed5	0x4a7484aa	0x53380d13	0x1e376c08	0xc67178f2
0xab1c5ed5	0xd807aa98	0x5cb0a9dc	0x650a7354	0x2748774c	
0xd807aa98	0x12835b01	0x76f988da	0x766a0abb	0x34b0bcb5	
0x12835b01	0x243185be	0x983e5152	0x81c2c92e	0x391c0cb3	
0x243185be	0x550c7dc3	0xa831c66d	0x92722c85	0x4ed8aa4a	
0x550c7dc3	0x72be5d74	0xb00327c8	0xa2bfe8a1	0x5b9cca4f	
0x72be5d74	0x80deb1fe	0xbf597fc7	0xa81a664b	0x682e6ff3	
0x428a2f98	0x9bdc06a7	0xc6e00bf3	0xc24b8b70	0x748f82ee	

Tabla 2 Números primos para SHA

Ahora bien, las operaciones realizadas en cada ronda se harán como indica la imagen 22.

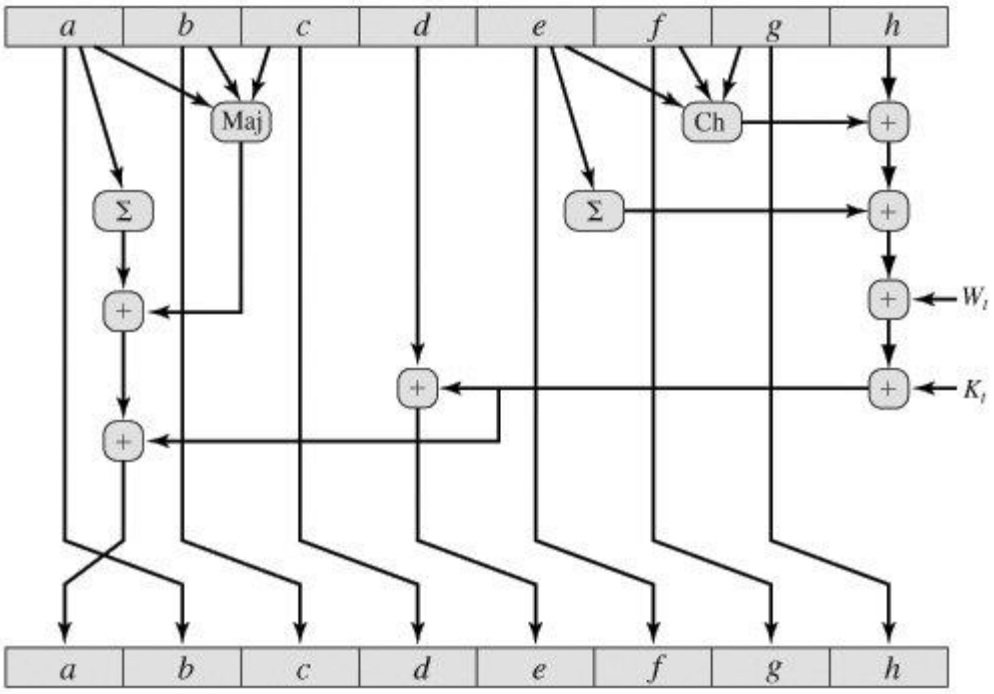

Imagen 22 Iteraciones SHA-256

Haciendo un resumen de todas las operaciones, guardando todo en dos variables temporales, obtenemos lo siguiente.

$$\mathrm{T}_1 = Maj + \Sigma_0$$

$$\mathrm{T}_2 = \mathrm{H} + Ch + \Sigma_1 + \mathrm{W}_t + \mathrm{K}_T$$

$$(\mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{F}, \mathrm{G}, \mathrm{H}) = (\mathrm{T}_1 + \mathrm{T}_2, A, B, C, D + \mathrm{T}_2, E, F, G)$$

Finalmente tenemos una estructura como lo muestra la imagen 23 para poder lograr nuestro hash.

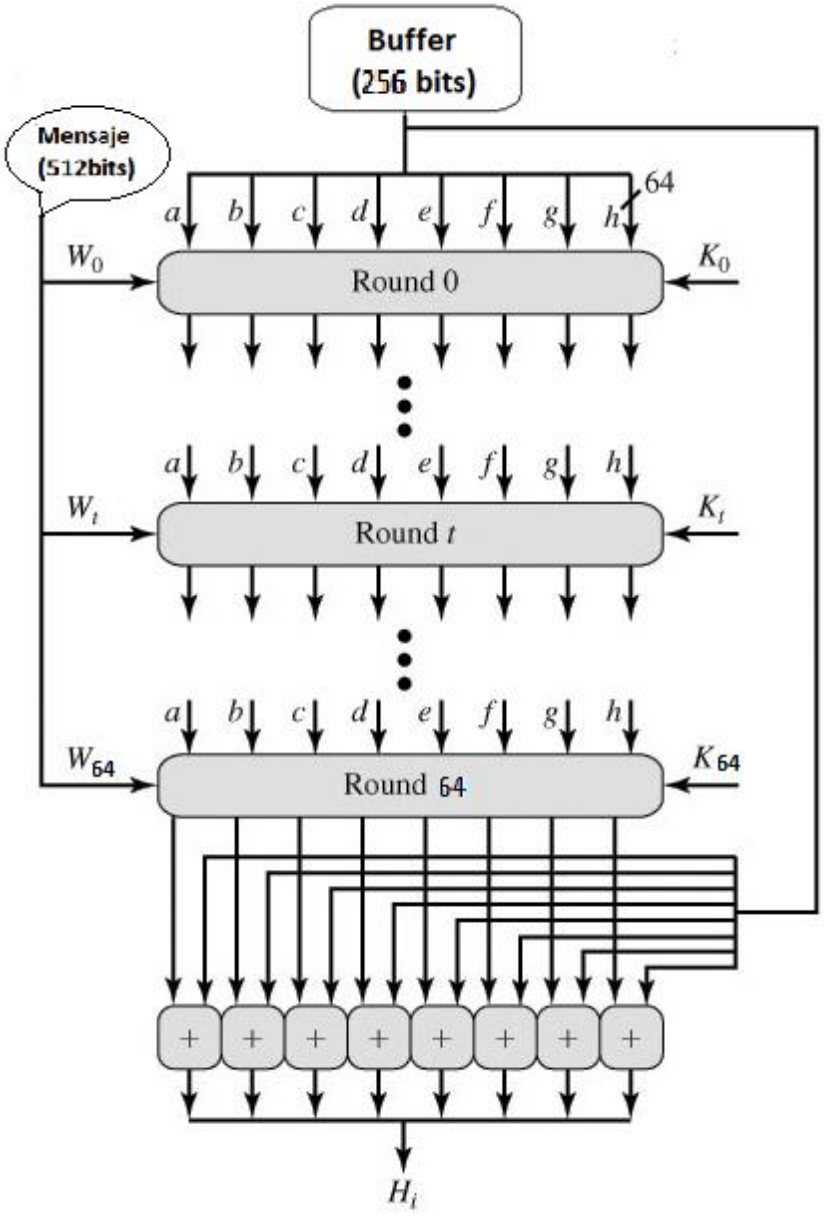

Imagen 23 Algoritmo SHA-256

Una vez completadas las 64 rondas, el paso final, como ya hemos visto en los demás algoritmos, es realizar una suma para obtener nuestro hash entre el último resultado de la ronda 64 y los valores iniciales del buffer.

2.5.4 RIPEMD-160

Cuando se descubrieron las debilidades del algoritmo MD5, surgieron muchos algoritmos que pretendían ser su sucesor, sin embargo, uno de ellos fue RIPEMD. Fue desarrollado casi a la par que el algoritmo de SHA, pero como SHA se convirtió en el algoritmo más usado, RIPEMD paso a segundo plano y fue quedando casi en el olvido. Pero todo eso ayudó mucho a mejorar su seguridad, al ser SHA el más usado, el mundo se dedicó a tratar de romper esas seguridad, y como RIPEMD era poco inusual, muy pocos eran los que le prestaban atención.

Partiendo de haber analizado un poco varios algoritmos Hash, es más sencillo ir entendiendo todos los algoritmos de este tipo. RIPEMD, al igual que SHA, contiene las mismas bases de un MD5.

Para comenzar con el algoritmo, seguimos los mismos pasos. El mensaje en binario se le hace un relleno comenzando con un 1, seguido de los 0 necesarios para que el tamaño coincida con 448 mod 512. Una vez que se realiza el relleno con 0, se añade el valor de la longitud del mensaje utilizando los últimos 64 bits, para así tener un mensaje de longitud final de 512 bit.

Ahora se inicializan las variables acomodadas como la forma little-endian, con las que se realizara el cifrado, de la siguiente manera.

A=67452301

B=EFCDAB89

C=98BADCFE

D=10325476

E=C3D2E1F0

Para procesar el mensaje, este algoritmo es un poco más especial que los anteriores. Se utilizan 10 rondas de 16 operaciones cada una, pero la diferencia de este algoritmo radica en el modo de operar las 10 rondas, dichas rondas se dividen en 2 para operarlas paralelamente, la imagen 24 ejemplifica mejor esto.

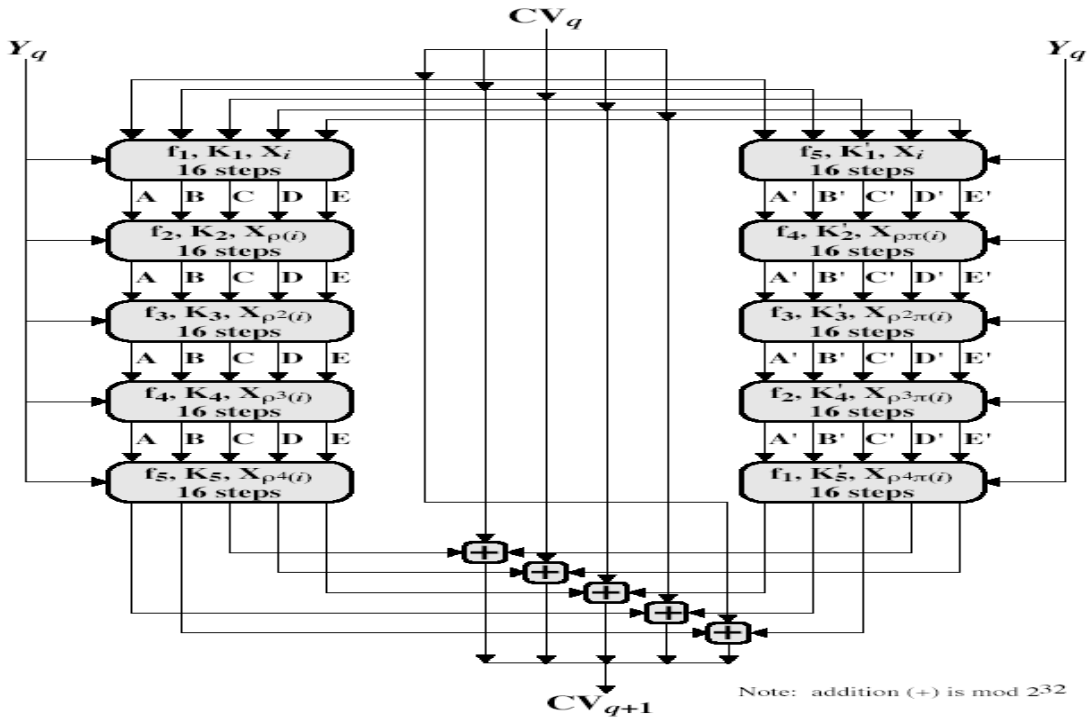

Imagen 24 Algortimo RIPEMD-160

Se toman 5 rondas por la izquierda, y 5 por la derecha para al final realizar una suma entre ambos lados que se operan y el contenido de las variables iniciales A,B,C,D,E.

La función de compresión se realizara como se indica en la imagen 25.

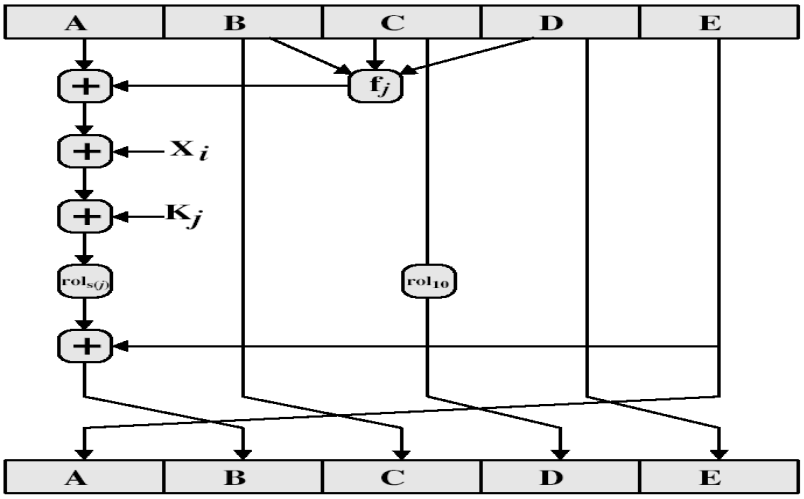

Imagen 25 Iteraciones RIPEMD-160

Para obtener el resultado de la función que opera a las variables que se inicializan, utilizamos las siguientes operaciones.

$F(A, B, C) = A \oplus B \oplus C$

$G(A, B, C) = (A \wedge B) \vee (\neg A \wedge C)$

$H(A, B, C) = (A \vee \neg B) \oplus C$

$I(A, B, C) = (A \wedge C) \vee (B \wedge \neg C)$

$J(A, B, C) = A \oplus (B \vee \neg C)$

En este caso, al tener 5 rondas por la derecha y 5 por la izquierda, cada una de cada lado se opera con una de las funciones anteriores como se infica en la tabla 3.

	Ronda 1	Ronda 2	Ronda 3	Ronda 4	Ronda 5
Derecha	F	G	H	I	J
Izquierda	J	I	H	G	F

Tabla 3 Orden de funciones RIPEMD-160

Ahora bien, resumiendo la imagen 25 obtenemos lo siguiente.

A = E

B = E + (A + F (B, C, D) + X + K) <<< s

C = B

D = C <<< 10

E = D

Después de haber realizado las operaciones y llegar a la última ronda se realiza la siguiente operación para obtener nuestro hash.

$$A' = C_{IZQ} + D_{DER} + B_{INI}$$

$$B' = D_{IZQ} + E_{DER} + C_{INI}$$

$$C' = E_{IZQ} + A_{DER} + D_{INI}$$

$$D' = A_{IZQ} + B_{DER} + E_{INI}$$

$$E' = B_{IZQ} + C_{DER} + A_{INI}$$

Finalmente solo basta con concatenar el resultado de cada una de las variables finales obtenidas y tenemos un Hash de RIPEMD-160.

Paradoja del cumpleaños

¿Cuántas personas debe haber en una sala para que la probabilidad de que dos de ellas celebren su cumpleaños el mismo día sea superior al 50%?

Entonces, sabemos que la probabilidad de que dos personas en un grupo de dos cumplan años cierto día se representa como:

$$P = \left(\frac{1}{365}\right)$$

Sin tomar años bisiestos, tomamos que el año tiene 365 días.

Para que en un grupo de tres personas, dos de ellas cumplan años el mismo día nos queda de la siguiente manera:

$$P = 1 - \left(\frac{364}{365}\right)\left(\frac{363}{365}\right) = 1 - \frac{364 \cdot 363}{365^2}$$

Desarrollando para que el valor de personas aumente, obtenemos la siguiente ecuación.

$$P = \left(\frac{365 \cdot 364 \cdot 363 \ldots (365 - n + 1)}{365^n}\right)$$

Por lo tanto:

$$P = \left(\frac{365!}{365^n \cdot (365 - n)!}\right)$$

En la tabla 4, podemos observar el comportamiento de la probabilidad, además podemos ver que a partir de que el valor de n = 23, la probabilidad de encontrar a dos personas es de aproximadamente el 50%.

n	p_n	n	p_n
1	0	23	0,5073
2	0,002740	...	...
3	0,008204	30	0,7063
4	0,01635	...	...
...	...	50	0,9704
10	0,01169	...	...
...	...	80	0,9999
22	0,4757	...	...

Tabla 4 Probabilidades paradoja de cumpleaños

Relacionando lo anterior con respecto a los hashes, tenemos algo muy similar a lo que llamamos colisiones. El término colisión, se refiere a la posibilidad de que existan dos hashes iguales pero con mensaje diferente. Y como podemos observar en la paradoja del cumpleaños, la probabilidad siempre aumenta mientras más elementos tengamos.

Sin embargo, el hecho de que haya un gran número de elementos esto beneficia para hacer que un ataque de fuerza bruta sea más tardado, y por ende, más impracticable.

Por ejemplo, supongamos que tenemos una llave de 10 dígitos en base 26, en número de operaciones quedaría de la siguiente manera.

$$26^{10} = 141{,}167{,}095{,}653{,}376\ operaciones$$

En el caso de MD5, teniendo una salida de 128bits

$$2^{128} = 340{,}282{,}366{,}920{,}938{,}463{,}463{,}374{,}607{,}431{,}768{,}211{,}456\ operaciones$$

Ahora bien, aplicando una búsqueda de colisiones MD5 quedaría de la siguiente manera.

$$2^{64} = 18{,}446{,}744{,}073{,}709{,}551{,}616\ operaciones$$

En el caso de SHA-1, que tiene una salida de 160bits, y utilizando solo la mitad, obtenemos lo siguiente.

$$2^{80} = 1{,}208{,}925{,}819{,}614{,}629{,}174{,}706{,}176\ operaciones$$

Entonces, reduciendo los bits de salida a la mitad podemos ver que el número de operaciones se reduce considerablemente, además de que con la velocidad de las nuevas tecnologías, es más fácil encontrar un hash repetido, con lo cual, una persona malintencionada puede interceptar y cambiar el mensaje original.

Debilidades de los algoritmos Hash

A pesar de tratarse de algoritmos de unidireccionales, donde es muy difícil obtener un mensaje a partir de un hash, y aparentar ser muy robustos tienen algún punto débil o una vulnerabilidad.

Podrá sonar muy rudimentario pero la manera más sencilla de poder lograr romper un HASH es a través de la fuerza bruta. Consiste en ir probando llave por llave hasta poder llegar a encontrar la que permita poder acceder a la información.

A pesar de que los algoritmos HASH son muy robustos, en la complejidad en la que genera el resumen durante varias rondas e incluso aumentando el número de bits de salida, su gran enemigo es el avance tecnológico que tenemos día a día. Mientras la tecnología siga avanzando los algoritmos HASH deben de hacer lo mismo ya que con procesadores tan complejos en los nuevos dispositivos los tiempos de prueba para cada llave se reduce considerablemente.

Por otro lado, existe otra vulnerabilidad que se conoce como colisiones, cada mensaje encriptado tiene muchísimas combinaciones que pueden generar el mensaje en claro, por medio de una función HASH podemos realizar lo anterior, obteniendo algo como r(m). Entonces el término colisión se refiere a que si algún extraño genera una función r(m') y al realizar una comparativa obtenemos que r(m)=r(m'), esto significa que tenemos dos funciones resumen idénticas, no importa si el mensaje es el mismo o no, este caso compromete mucho la información del mensaje y deja de ser un algoritmo seguro.

Colisiones

Hablamos del término colisión cuando nos referimos a que hay un mismo hash que corresponde a dos mensajes diferentes, es decir, tenemos un mensaje M y un mensaje N, cada uno diferente al otro, de tal modo que tenemos MD5 (M) = MD5 (N).

Recordando la paradoja de cumpleaños, esto sería similar a comprar los hashes de dos mensajes, si son diferentes se siguen comparando con otro, y así sucesivamente hasta encontrar dos mensajes donde sus hashes coinciden, con lo que nos referimos a encontrar una colisión. Y retomando un poco más el concepto, tenemos que el número de operaciones de una función hash disminuye a la mitad.

El encontrar una colisión es muy peligroso, ya que, a pesar de que nuestra información estará segura, puede haber una suplantación del mensaje original creando un gran conflicto.

Haciendo un resumen de los algoritmos que hemos visto, tenemos la tabla 5.

Algoritmo	Año	Desarrollador	Bits	Rondas	Núm. de posibilidades	Fuerza contra colisión
MD5	1992	Ronald Rivest	128	64	2^{128}	2^{64}
RIPEMD-160	1996	Hans Dobbertin Antoon Bosselaers Bart Preneel	160	80	2^{160}	2^{80}
SHA-0	1993	NSA	160	80	2^{160}	2^{80}
SHA-1	1995		160	80	2^{160}	2^{80}
SHA-256	2002		256	64	2^{256}	2^{128}

Tabla 5 Algoritmos de cifrado

2.5.5 Firma digital

Los algoritmos de firma digital son muy similares a lo hashes, se utilizan, como su nombre lo indica, para realizar una firma a un archivo o mensaje y para comprobar la validez de un mensaje y con ello poder evitar que no se trata de un mensaje falso.

Simplemente se trata de un proceso con el cual se puede hacer la verificación de un objeto realmente es original. Es muy similar a una firma en un documento, la firma digital indica a todos que el objeto es realmente válido generado por la persona que agrega su firma. Para realizar una firma digital, de igual modo que en los hashes, se utilizan dos llaves, una llave pública y otra privada, en este caso es un proceso distinto a la generación de hashes, aquí para poder agregar la firma al archivo se utiliza la llave privada, a la llave pública todos pueden tener el acceso y es con está con la que se realiza la comprobación de que el objeto realmente es originario del usuario dueño de la llave privada.

Cabe destacar que más que un algoritmo de cifrado, las firmas digitales son utilizadas para comprobar que realmente los archivos son válidos que vienen de cierta persona en confianza y no han sufrido modificaciones. No hay que confundir los hashes con las firmas digitales, pero si hay que recalcar que trabajan de la mano. Si bien los hashes se utilizan para generar un resumen y también es muy difícil encontrar dos hashes con el mismo resumen, para brindar mayor seguridad y confiabilidad, se agrega el algoritmo de firma digital para la verificación de que se trata de un algoritmo original.

Hablando en términos un poco más sencillos, la complejidad de las firmas digitales radica el su algoritmo, un ejemplo muy fácil, supongamos que tenemos la siguiente ecuación, P = n * q, computacionalmente es muy sencillo obtener el valor de P, si se conocen las variables n y q, sin embargo, el problema comienza cuando solamente conocemos el valor de P y necesitamos obtener el valor de n y q. En las firmas digitales, esos valores de n y q representan nuestras llaves, y la complejidad aumenta cuando utilizamos números grandes para generar esos valores de n y q.

2.5.6 ECDSA (Elliptic Curve Digital Signature Algoritm)

Uno de los algoritmos más conocidos y que últimamente se ha convertido en los más famosos es el algoritmo de curvas elípticas, ECDSA (Elliptic Curve Digital Signature Algorithm - Algoritmo de Firma Digital de Curva Elíptica).

Como bien su nombre lo dice, son algoritmos basados en la obtención de puntos dentro de una curva elíptica, mediante un algoritmo que hará difícil la obtención de dichos puntos de una manera muy sencilla.

Las curvas elípticas pueden llegar a ser definidos sobre números reales, complejos o cualquier otro tipo de campo, pero desde el punto de vista criptográfico, utilizamos curvas sobre campos finitos.

En criptografía, al hablar de curvas elípticas es en referencia a una ecuación $y^2 = x^3 + Ax + B$ que cumple con que $4A^3 + 27B^2 \leq 0$. Para diferentes valores de A y B podemos obtener todo un conjunto de curvas, que al ser dibujadas en una gráfica una forma especial. Una de las características consiste en la posibilidad de obtener un punto dentro de una curva a partir de dos puntos anteriores, esto se ve reflejado en la imagen 26.

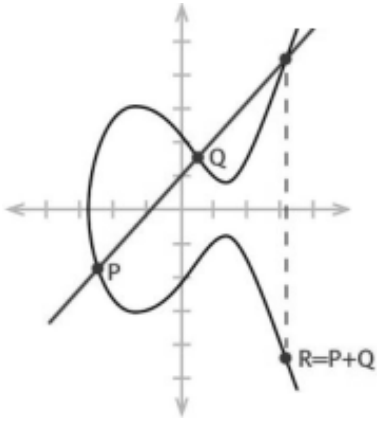

Imagen 26 Obtener nuevo punto

Para este ejemplo partimos de que se conocen dos puntos, P y Q, ahora trazamos una línea para unir dichos puntos y en el caso donde la línea corta la curva en un tercer punto, obtenemos como resultado un nuevo punto R. Para representar lo anterior lo hacemos como P + Q = R. Hay otro caso donde la línea entre los puntos P y Q jamás vuelve a cortar la curva, en este caso hablaremos de qué se trata de un punto 0 en el infinito y se representa como P + Q = 0.

Si bien, ese concepto se escucha muy fácil, vamos a definir algunas propiedades de las curvas elípticas para que quedemos claro.

La primera propiedad ya la hemos visto, obtener un tercer punto partiendo de contamos con dos, para ello tomamos los puntos P y Q y al realizar una suma entre ellos podemos obtener un tercer punto, como podemos ver en la imagen 27.

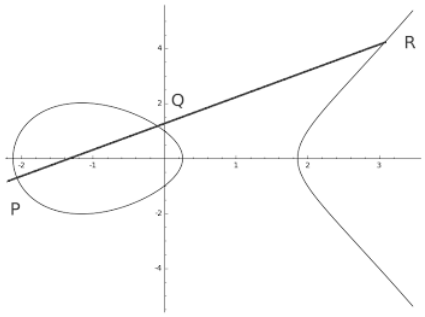

Imagen 27 Suma de puntos

Por definición vamos a tomar el valor simétrico de nuestro punto R respecto al eje de las abscisas y así obtendremos el inverso de R formado por los puntos P y Q, es decir, P + Q = -R, ver imagen 28.

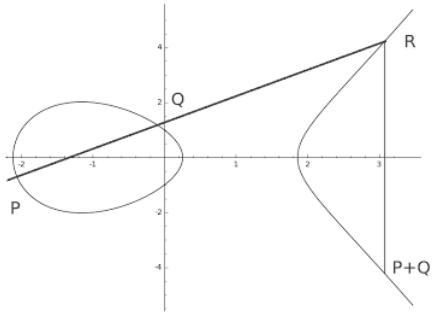

Imagen 28 Resultado de suma de puntos

Pero tal vez nos preguntamos, ¿Qué pasa con los puntos en los que no obtenemos ese tercer punto?

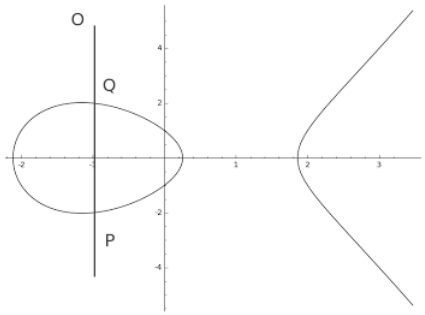

Imagen 29 Elemento cero o neutro

Pues bien en estos casos a simple vista tenemos que al parecer no cumplimos con la suma que hemos venido planteando, sin embargo, para este tipo de casos decimos que los puntos que definen esa recta son iguales a cero. Este valor que obtenemos de esa suma vamos a llámalos elemento cero o neutro, ver imagen 29.

Otro caso que podemos encontrar para la obtención de un tercer punto es el siguiente, ver imagen 30.

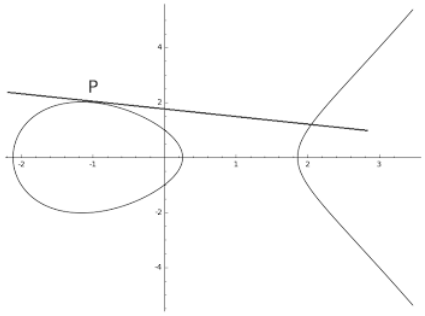

Imagen 30 Suma de un puto igual

Muchos podríamos pensar que si una recta no toca dos puntos no vamos a obtener una suma, sin embargo, podemos expresarlo de la siguiente manera, P + P = 0, o bien, 2P = 0. Incluso de este tipo de operación podemos obtener el inverso -R y expresarlo como P = Q.

Partiendo del concepto de dicha suma, tenemos que es muy sencillo encontrar la manera de realizar una multiplicación por un escalar, esto es kP o kQ. Por ejemplo, para llegar al resultado de 15P, tenemos distintas maneras de obtener el mismo resultado, esto es que:

$$15P=2P+3P+10P=5P+5P+5P\ldots$$

Este simple mecanismo utilizado para obtener nuevos puntos sobre una curva elíptica es la base de muchos criptosistemas modernos con mayor seguridad hasta el momento.

En criptografía, las curvas elípticas se utilizan sobre campos finitos F (q), y para obtener mucha más seguridad se utilizan valores para “q” muy grandes de modo de que tome años realizar un ataque, un ejemplo de campo finito es, F (7)= {0, 1, 2, 3, 4, 5, 6} de tal manera que para representar el número 9 dentro del campo finito se realiza la siguiente operación, 9 mod 7 = 2. Cuando hablamos de utilizar campos finitos, también decimos que los puntos de la curva son finitos. Este número se conoce como orden de la curva y lo podemos representar con la letra E.

Como ya hemos visto, la fuerza de la criptografía siempre radica en los algoritmos matemáticos utilizados. Uno de los más usados es el problema del logaritmo discreto. Este problema se basa en resolver ecuaciones, en este caso, del tipo *x = ay mod n*. Donde "x", "a" y "n" son variables conocidas y tratamos de encontrar el valor que la variable "y". Simplemente basta con tener valores lo suficientemente grandes de "y" y "n" para hacer computacionalmente imposible la obtención de un resultado correcto de la ecuación, al menos con los algoritmos o computadoras que tenemos en la actualidad.

Por ejemplo.

Consideremos la curva $y^2 = x^3 + x^2 + 2$ con modulo 13.

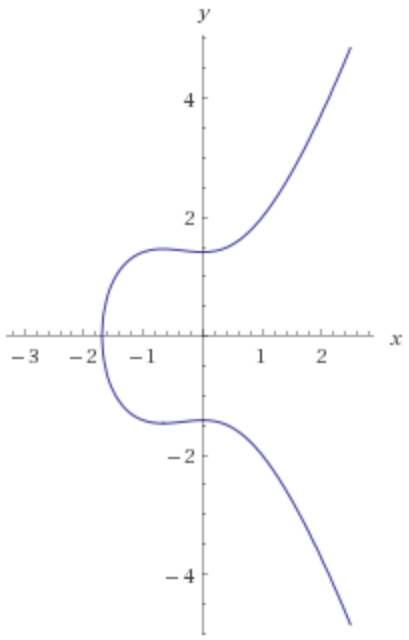

Imagen 31 Grafica de la curva y^2=x^3+x^2+2

Para simplificar el ejemplo, comencemos haciendo la tabla 6.

x	0	1	2	3	4	5	6	7	8	9	10	11	12
x^2	0	1	4	9	3	12	10	10	12	3	9	4	1
x^3	0	1	8	1	12	8	8	5	5	1	12	5	12
$x^3 + x^2 + 2$	2	4	1	12	4	9	7	4	6	6	10	11	2

Tabla 6 Válores de la curva

Comenzamos con darle valores permitidos, en este caso al tratarse del modulo 13 tenemos un rango que va desde el 0 al 12. Calculamos el valor de x^2 sin olvidar el modulo, y del mismo modo operamos para x^3. Ahora realizamos la suma de los valores correspondientes obteniendo el resultado de la ultima fila.

Ahora bien, debemos poner enfacis en la segunda fila (x^2), ya que esos serán los varoles que tomarán las raíces modulo 13, en este caso son 0, 1, 3, 4, 9, 10, 12. Una vez identificados esos valores, realizamos una comparación entre la ultima fila y la segunda, para poder descartar los valores que no coinciden. En nuestro ejemplo podemos ver que el 2, 7, 6 y 11 no coinciden y por lo tanto no son soluciones de nuestra curva.

Entonces, con todo lo anterior tenemos definidos los valores que tomara x. Sin embargo, ahora simplemente basta con obtener los puntos de y. Debemos tener muy presente que cuando se trata de encontrar las raíces una variable con exponente, esté mismo indica en numero de soluciones, en el caso particular de la raíz cuadrada, obtendremos dos valores iguales pero de signo diferente.

En los caso en donde no tenemos una raíz cuadrada directa, debemos encontrar el valor obtenido igualando de la siguiente manera.

Para el caso del 3.

$$y^2 = 12 mod 13$$

Debemos encontrar un valor para y, de tal modo que se cumpla la igualdad.

$$y = 5$$

$$5^2 mod 13 = 12 mod 13$$

$$12 mod 13 = 12 mod 13$$

$$y = 8$$

$$8^2 mod 13 = 12 mod 13$$

$$12 mod 13 = 12 mod 13$$

Los puntos al final que obtenemos son:

(1,2),(1,11), (2,1), (2,12), (3,5), (3,8), (4,2), (4,11), (5,3), (5,10), (7,2), (7,11), (10,6), (10,7)

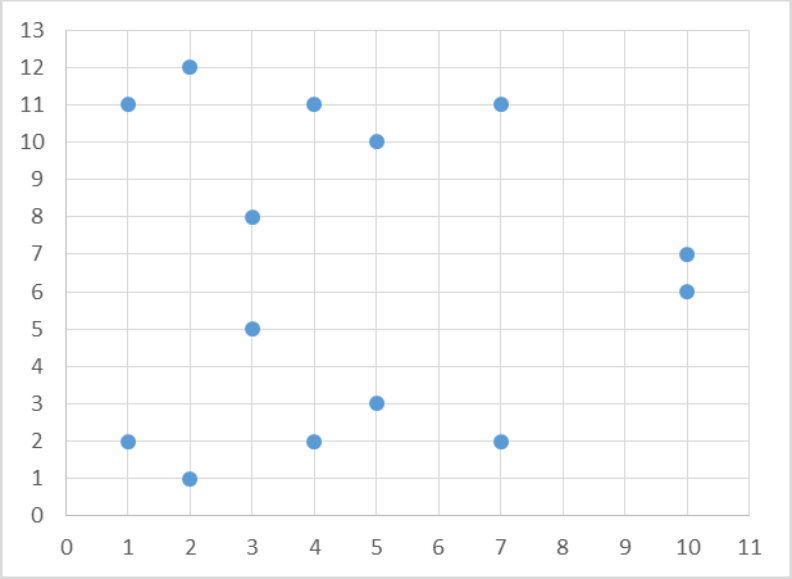

Imagen 32 Puntos de la curva definidos en mod13

Como podemos observar, los puntos encontrados con números pequeños son pocos, sin embargo haciendo que esos números sean mucho más grandes es practicamente imposible encontrar la solución solo conociendo el resultado.

Ahora bien, relacionando el problema del logaritmo discreto con el algoritmo de curvas elípticas, existe un problema muy similar entre ellos. Ambos algoritmos tratan de encontrar el valor de una variable que cumpla con el algoritmo correspondiente. De tal modo que para obtener un resultado podemos llegar de una infinidad de números, lo cual hace que el valor de dicha variable puede ser muy pequeño y al mismo tiempo garantizarnos una seguridad casi comparable con algoritmos más robustos con la ventaja de que al utilizar llaves pequeñas optimizamos la parte de la memoria utilizada y la capacidad de proceso que tendrán.

Entonces basándonos en los sencillos ejemplos anteriores, el algoritmo ECDSA en simples palabras es eso, un algoritmo que utiliza puntos de una curva elíptica, dichos

puntos elegidos son conocidos como la llave pública y la llave privada. Con una firma digital podemos llevar a cabo una verificación y firma de archivos para comprobar su autenticidad.

Podemos dividir el algoritmo de ECDSA, en tres subalgoritmos, generación de un par de llaves, generación de la firma y verificación de la firma.

3. Marco práctico

Ahora que ya tenemos el conocimiento de todo lo que está detrás de las monedas virtuales, es hora de ver qué son, de dónde vienen, cómo funcionan, cómo se utilizan e ir aplicando los conceptos que hasta el momento hemos visto.

Sin duda es hora de poner en práctica nuestros conocimientos para poder ver que las monedas virtuales después de todo son muy seguras, y a pesar de sus altibajos, en comparación con el dinero que conocemos comúnmente, serán el futuro de nuestra economía sin tener un representante que controle el flujo de la nueva moneda.

3.1 Origen de las monedas virtuales

El concepto de moneda virtual puede sonar muy reciente y dado a varios problemas, ha comenzado a ser adoptado como medio de pago en muchos países dando una forma más sencilla de pago en distintos sitios, tanto físicos como virtuales.

En 1998, el concepto de criptomoneda es introducido por Wei Dai con la idea de crear un método de pago online basado en la criptografía para controlar su creación y las transacciones realizadas, pero la idea nunca progreso hasta que en el año de 2009, Satoshi Nakamoto retoma la idea de utilizar este método de pago.

Cuenta la leyenda que en el 2007, ya que el origen sigue sin conocerse, que Satoshi Nakamoto comenzó a trabajar en el concepto bitcoins, y se especula que Satoshi corresponde efectivamente a una persona de origen Japonés, pero el término Nakamoto es el sobrenombre de un grupo de personas que trabajaban junto a él.

La primera moneda, y muy utilizada, lleva por nombre bitcoin. Entre el período de 2008-2009, Satoshi Nakamoto decidió lanzar al mundo una moneda electrónica mediante el protocolo llamado Bitcoin. Y el 3 de enero de 2009 entra por primera vez en funcionamiento una red Peer-to-Peer de bitcoins para llevar a cabo pagos de una manera mucho más sencilla.

Muchas personas consideran a Nakamoto como el creador de bitcoins, pero cabe destacar que desde su aparición del concepto el código natural de los bitcoins siempre ha sido abierto para cualquier desarrollador. Por lo cual a ciencia cierta hay la posibilidad de que realmente Nakamoto no sea el verdadero creador.

Antes de su aparición, todas las compras de este tipo eran obligadas a realizarse mediante un intermediario, generalmente bancos o empresas financieras, pero con el nacimiento de esta nueva tecnología esto pasó a ser distinto.

Los bitcoins son una moneda virtual, independiente y descentralizada, es decir, ninguna institución gubernamental, organización financiera o empresa la controla. Por

lo cual toda la información se maneja de una forma mucho más anónima para tener la confianza al realizar distintos pagos o tener algún tipo de ahorro.

El 18 de agosto de 2008, es registrado el sitio con el dominio "bitcoin.org", primer sitio donde podías comenzar a utilizar bitcoins, fue registrado en anonymousspeech.com, un sitio que permite a distintos usuarios la oportunidad de registrar dominios de manera anónima (En la actualidad acepta bitcoins).

Unos meses más tarde, el dominio es registrado en SourceForge.net, sitio dedicado al desarrollo y distribución de software de código abierto.

El 3 enero de 2009, el protocolo Bitcoin (BTC) se publica y las primeras monedas virtuales son creadas, se genera el primer bloque, bloque cero o bloque de génesis. Y para el 9 de enero de ese mismo año, se crea la versión 0.1 del cliente bitcoin, compilado en Visual Studio.

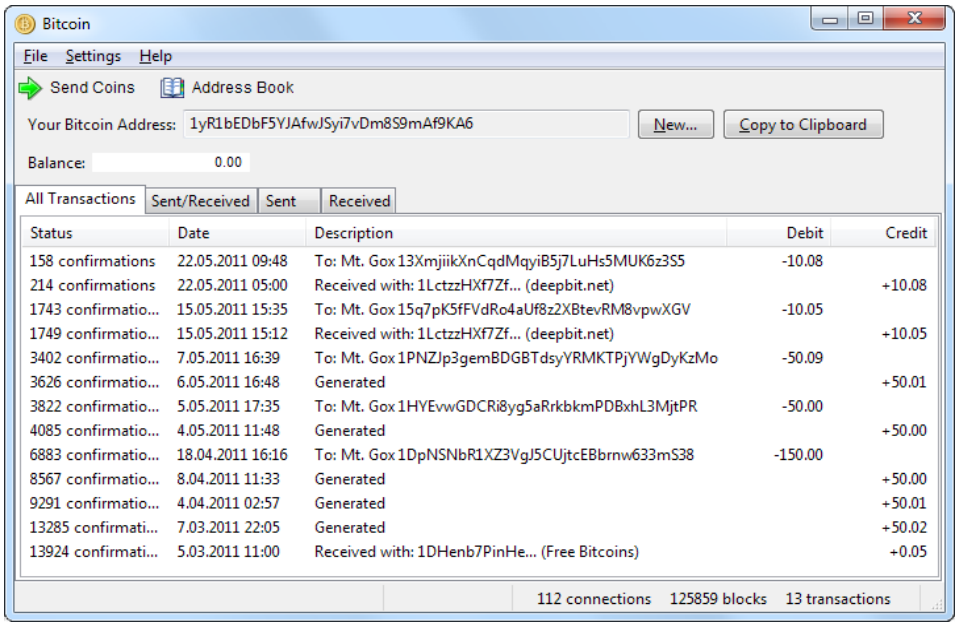

Imagen 33 Software cliente Bitcoin

El primer movimiento de bitcoins fue el bloque 170 el 12 de enero de 2009 y se lleva a cabo entre Satoshi y Hal Finney, un desarrollador y activista criptográfico. Y para el 5 de octubre, New Liberty Standard asigna el primer valor de la moneda, 1309.03 bitcoins

era el equivalente a 1 dólar. Este valor fue calculado a partir de una ecuación que relacionaba el consumo de electricidad que gastaba el equipo que generaba el Bitcoin. Y al final del 2009, se libera la segunda versión del software Bitcoin (0.2).

En el 2010, el 6 de marzo surge el primer mercado de bitcoins donde era posible realizar el cambio de divisas, el 17 de mayo, Laszlo Hanyecz fue la primera persona en utilizar los bitcoins para realizar una compra publicando el 22 de mayo que la compra había sido exitosa, pagó 10000 bitcoins por dos pizzas. Para entonces el bitcoin ya se cotizaba 0.080 dólares. Durante ese mismo año, el 17 de julio comienza a operar Mtgox, un sitio para realizar el cambio de bitcoins, para ese entonces el valor de los bitcoins había subido hasta 10 veces su valor, y para el mes de noviembre el precio en Mtgox de un bitcoin ya era de 0.50 dólares.

El 15 de agosto de 2010 fue descubierta la primera vulnerabilidad del sistema bitcoins, se trataba de la creación de bitcoins sin la necesidad de ser verificados lo cual terminó en la generación de 184 mil millones de bitcoins. Para finales de año, los bitcoins ya superan la cantidad de 1000.

En el año de 2011, abren un sitio de nombre Silk Road, el cual se utilizaba para la venta de drogas, mediante el pago de bitcoins. Para febrero del 2011, el bitcoin alcanza a la par el valor de un dólar (1 bitcoin = 1 dólar). Durante ese año, en junio registró su primer pico llegando hasta los 32 dólares, aunque después de unos meses de estabilizó en 20 dólares por bitcoin. Para estas fechas, ya se tenían creados el 25% de los bitcoins totales que se tienen estimados (21 millones).

En marzo de 2011, la velocidad de proceso para generar bitcoins crece demasiado, tanto que alcanza una velocidad de 900 Ghash/s aunque días después esa velocidad disminuye hasta llegar a 500 Ghash/s. En ese mismo mes, se realiza la primera venta de un automóvil, un miembro australiano logra vender su auto Celica Supra modelo 1984 en 3000 bitcoins.

En junio de 2011, un usuario con un seudonombre Allinvain anuncia que de su computadora personal alguien roba una cantidad de 25,000 bitcoins equivalentes en ese tiempo a 375,000 dólares. Durante ese mismo año, Wikileaks comienza a aceptar donativos a través de bitcoins lo que poco a poco comienza a generar confianza entre los usuarios de internet, con lo cual el valor del bitcoin llega a estabilizarse en 10 dólares. Para este año, la compañía MtGox, realiza el 90% de las transacciones de bitcoins, llegando con un máximo para el mes de julio de 31.91 dólares por unidad y es en ese año cuando sufre la primera vulnerabilidad. Alrededor de 600 clientes pierden su dinero de sus cuentas lo cual genera nuevamente la desconfianza y el precio del bitcoin vuelve a caer, finalmente ese año cierra con un valor de 5,27 dólares por unidad.

Durante el 2012, era un año nuevo para volver a conseguir la confianza pérdida y podemos decir que así fue. Muchas empresas comienzan a hacer el bitcoin una divisa con valor. Este es el año en el que WordPress comienza a aceptar pagos con bitcoins, el primer álbum de música comienza a venderse en bitcoins, servicios de taxis comienzan a aceptar pagos con la moneda virtual y comienza a operar un juego de nombre "Satoshidice" un juego de casinos online que comienza a utilizar bitcoins. Para finales de ese año, el itcoin cierra con un precio de 13,30 dólares por bitcoin.

Y finalmente, el año 2013, comienza la comunidad cibernética a darle la confianza a la moneda. Poco a poco con el constante uso comienza a adquirir valores muy significativos, para febrero de ese año ya tenía el mismo valor que una onza de plata, y como consecuencia los usuarios comienzan a hacer parte de su día el uso de bitcoins.

Otras compañías de transacciones comienzan a surgir, por ejemplo, BitPay entra al negocio y para marzo de 2013 ya supera las 10,000 transacciones. El sitio Pizzaforcoins.com entra en funcionamiento, comienzan a comercializar pizzas y el pago se maneja a través de bitcoins. Pero los problemas seguían surgiendo, el sitio de BitInstant anuncia que hackean su sistema y el monto robado asciende a 12,000 dólares en bitcoins. Pero sin importar eso muchas más empresas comienzan a seguir aceptando donativos con bitcoins, y esto va generando mayor confianza en los

usuarios de los diferentes usuarios. Durante el mes de marzo el valor era muy variado, en ocupaciones estaba cerca de los 74 dólares y de un día para otro podía llegar a valer 30. Durante el mes de mayo llegan a tener un valor de hasta 100 dólares por primera vez en su historia. En ese mismo mes, inauguran el primer cajero automático del mundo, ubicado en San Diego, California. El sitio PrimeDice.com lanza el primer casi online donde aceptan apuestas con bitcoins. Durante el mes de octubre, el FBI pone fin al sitio "Silk Road", un sitio utilizado para la venta ilegal de drogas, el cual tenía un valor de 3.6 millones de dólares en bitcoins. Pero a pesar de todo lo sucedido, el precio del bitcoin comenzaba a ser más estable, incluso en vez de bajar, con la confianza que iba obteniendo y la popularidad que iba generando, todo indicaba que sería un gran éxito y su valor comenzaría a ir incrementando. En diciembre una de las famosas tiendas Subway ubicada en Allentown, Pensilvania, comienza a aceptar pagos con bitcoins. Y con tanta popularidad el precio se elevó hasta los 503 dólares y después de unos días el valor de los bitcoins subió hasta más de mil dólares por unidad, dejando sorprendidos a muchos que aún tenían la desconfianza en la moneda.

El problema surge cuando en ese mismo año, las autoridades chinas prohíben a los bancos y entidades de pagos realizar transacciones con bitcoins, lo cual fue un gran golpe para la moneda. Tanta fue la afectación de que prohibieron su uso en China que el valor tuvo una gran caída, después de valer más de mil dólares cayó hasta los 600, aunque esta cifra aún no eran tan mala.

El peor año para los bitcoins y sus propietarios fue en el 2014, la compañía que controlaba la gran mayoría de transacciones de bitcoins, MtGox, sufre el más terrible robo de la historia de la moneda, una cifra de aproximadamente 850,000 BTC con un valor casi de 500 millones de dólares había sido sustraída de varias de las cuentas que alojaban en sus servidores. Lo cual afectó mucho a la moneda y la popularidad que había ganado.

De llegar a un valor máximo de 1130 dólares por bitcoin, cayó de manera muy alarmante, su precio rondaba los 300 dólares.

En abril del 2014, el banco central de Colombia anuncia que el bitcoin no es considerado como una moneda de uso legal en el país. Pero no todo seguiría siendo malo, Bulgaria en ese mismo año reconoce como moneda al bitcoin y le comienza a dar una buena imagen. Bitso, el primer Exchange de bitcoin de México permite cambiar la criptomoneda por pesos mexicanos. En ese mes el bitcoin rondaba por los 500 dólares.

En diciembre de 2014, el gigante tecnológico Microsoft comienza a aceptar pagos con bitcoin. Con la nueva moneda se podía comprar aplicaciones de Windows Store, videojuegos para la consola Xbox, así como música y videos, series y películas. Sin embargo, las compras podían ascender hasta 1000 dólares y la plataforma de pago sería BitPlay.

El 4 de enero de 2015, la plataforma que manejaba la mayor cantidad de bitcoins, BitStamp, sufre una gran pérdida de 18,866 bitcoins que eran aproximadamente 5 millones de dólares, lo que es equivalente al 12% total de los bitcoins manejados por BitStamp. Según declaraciones de la compañía, la cantidad robada era solo una pequeña parte de las reservas de bitcoin con las que contaban por lo cual no afectó el dinero de los clientes. Sin embargo, el CEO de BitStamp Nejc Kodric jamás hace mención sobre la palabra hackeo para no alarmar a sus clientes.

Pero esto no tenía mucho sentido, en marzo del mismo año, BitStamp informa a sus clientes de otros ataques que se realizaban contra sus cuentas, los cibercriminales enviaban emails con **phishing**[2] con la intención de robar usuarios y contraseñas para realizar sus robos.

El 19 de Mayo del 2015, Ross Ulbricht, operador del mercado Silk Road, es declarado culpable en 7 cargos de lavado de dinero, piratería informática y tráfico de narcóticos. El precio del bitcoin ronda cerca de los 234 dolares. Para el primero de julio del 2015, dos agentes federales, Carl Force IV y Shaun Bridges, son declarados culpables por el robo de bitcoins durante su investigación activa de Silk Road. Carl desvió un monto de

[2] El phishing es un término que se utiliza para referirse a la suplantación de la identidad de una persona o empresa.

$50,000 dólares a su cuenta personal, mientras que Shuan desvió $800,000 dólares a su cuenta personal.

El 22 de Octubre del 2015, el Tribunal de Justicia Europeo, dictamina que el intercambio de bitcoins no está sujeto al impuesto al valor agregado (IVA). Lo cual hizo que el valor del bitcoin se elevara aproximadamente 3 por ciento, quedando con un valor de 273 dólares.

El comité Unicode, el 3 de noviembre del 2015, acepta el símbolo de bitcoin, una B mayúscula con una barra a través de ella, ***Ƀ*** en mayúscula con un valor de 579 en decimal y ***ƀ*** en minúscula con un valor de 389 en decimal.

El 8 de diciembre del 2015, la revista estadounidense Wired, publica un artículo donde mencionan que la verdadera identidad de Satoshi es un hombre australiano llamado Dr. Craig Steven Wright. Sin embargo, a pesar de la evidencia presentada, Wright no confirmó nada sobre si realmente es o no Satoshi Nakamoto. Para estas fechas, el valor del bitcoin iba poco a poco incrementando, obteniendo un valor de 397 dólares.

El 4 de abril del 2016, es lanzado al mercado el primer software con la intención de crear un mercado utilizando bitcoins. OpenBazaar, es el nombre del software que permite crear tiendas virtuales donde la moneda utilizada es el bitcoin, esto con el fin de facilitar el comercio P2P sin intermediarios.

El 27 de abril del 2016, Steam, que es una compañía de videojuegos, entra a la lista del comercio con bitcoins. Beneficiando a clientes de todo el mundo, generalizando una moneda para poder realizar pagos para los videojuegos.

Para el 9 de julio del 2016, el valor por un boque minado es de 12.5 bitcoins. El precio de la moneda ronda cerca de los 652 dólares, y cada vez va ganando más popularidad.

2 de agosto del 2016, Bitfinex, una de las compañías con mayor volumen de bitcoins, anuncia que 119,756 bitcoins han sido robados de las cuentas de sus clientes, el robo

es de $72 millones de dólares. Bitfinex estaba asociado a otra compañía de nombre BitGo, sin embargo, aún no está claro si el dinero será reembolsado a los clientes. El valor del bitcoin cayó un 20 por ciento, quedando cerca de los 480 dólares, pero tuvo una pequeña recuperación para rondar cerca de los 594 dólares.

El 9 de noviembre del 2016, el hecho más importante de los últimos tiempos, la elección de Donald Trump como presidente de los Estados Unidos también tuvo un impacto sobre los bitcoins. Muchas de las monedas comenzaron a tener distintas variaciones respecto al dólar, lo cual fue muy beneficiario para los bitcoins. Se ha registrado un aumento la noche de ese día, de hasta el 5 por ciento del valor en menos de 24 horas, pero después el precio se estabilizó dando un incremento del 2.5 por ciento quedando con un valor aproximado de 715 dólares.

Imagen 34 Valor del bitcoin (15/11/2015 - 15/11/2016)

Imagen 35 Valor del bitcoin desde su creación (02/01/2009 – 15/11/2016)

3.2 Monedas virtuales, tipos y características

Moneda virtual, es una unidad de divisa digital utilizada para realizar operaciones en línea a través de una red sin la necesidad de contar con un organismo gobernador central. Para llevar a cabo las transacciones de bitcoins se utiliza el protocolo de Bitcoin (BTC). Cada intercambio se registra en cadenas de bloques, una base de datos pública y en línea. Para poder añadir más bitcoins al sistema, se utiliza un procedimiento llamado "minería de bitcoins", es un proceso que requiere operaciones matemáticas complejas. Se estima que el total de bitcoins que puede ser generado es de 21 millones; este límite se debe a la facilidad que tiene el bitcoin de ser dividido y a que no es manejado por ninguna organización.

Bitcoin, es la primera red de pagos Peer-to-Peer descentralizada donde usuarios de todo el mundo pueden acceder para poder realizar compras sin la necesidad de contar con algún intermediario que maneje el movimiento del dinero. Es una moneda digital con un valor como las demás, que tiende a crecer y decrecer dependiendo de la

popularidad que tengan. Generalmente, utilizamos “Bitcoin” para referirnos al protocolo y “bitcoin” para hablar sobre la moneda.

Litecoin, esta moneda utiliza el mismo sistema de Bitcoin, la diferencia es que el sistema de confirmación de una transacción se realiza en menos tiempo. El proceso de obtención puede realizarce con equipos con poca capacidad de procesamiento.

Namecoin, esta moneda tiene la característica de utilizar un dominio de internet que no esta registrado en la Corporación de Internet para la Asignación de Nombres y Números (ICANN), por lo que las operaciones pueden ser espiadas por cualquier usuario. La cantidad máxima que puede existir es de 21 millones, el mismo limite de bitcoins, y el proceso de creación de igual modo es muy similar.

Ether, no solo es una moneda virtual a pesar de su gran similitud con Bitcoin, su creación fue debido a querer solucionar algunos problemas de Bitcoin, tales como los tiempos largos para las confirmaciones de transacciones, entre 14 y 15 segundos mientras que Bitcoin tarda 10 minutos. Además, disminuye la dificultad para poder hacer las comprobaciones de transacciones y es más fácil crear competencia entre los mineros.

La principal característica de las monedas virtuales, es que son descentralizadas, es decir ningún estado, banco o institución las controla, esto permite que no sea posible generar inflación, sino que la generación mediante la minería permite que los usuarios obtengan dicha descentralización. Al no tener a algún intermdiario, las transacciones se hacen persona a persona haciendo que sean de modo casi instantáneo y disponible a cualquier hora del mundo.

Además, algo que hace aun más especiales a las monedas virtuales es el gran sistema de seguridad con el que cuentan. Su sistema criptográfico protege a todos los usuarios y al mismo tiempo simplifica las operaciones. Por otro lado, el tipo de red utilizada por el protocolo mantiene seguras las transacciones y el tipo de seguridad que se utiliza

para proteger cada monedero, hace que nuestros intereses estén seguros contra cualquier persona.

Otra característica de las monedas, es las transacciones son únicas e irreversibles, esto es, cada vez que se realiza un pago no puede ser anulado e inclusive no puedes hacer dos o más pagos con una misma dirección. Por otro lado, el dinero le pertenece directamente al usuario, y al realizar una transacción la identidad de dicho usuario jamás será revelada, lo cual brinda anonimato para realizar movimientos, de modo de que nadie sabrá cuanto dinero tiene algún usuario.

Por ultimo, una característica única, la volatidad de los precios. Las monedas virtuales sufren altas y bajas en su valor con mucha frecuencia, debido a que no dependen de ningún mercado su valor se vuelve inestable. Ahora bien, al ser una moneda digital la primera capa de seguridad depende directamente de los proveedores de monederos y estos al sufrir diversos ataques en contra, incrementan la desconfianza entre compadores y vendedores por lo que su valor se ve afectado. A medida en que creesca el número de usuarios, el valor irá a la alza y se comenzará a estabilizar.

Al ser bitcoin la moneda más usada, nos basaremos en ella para poder entenderla más a fondo y del mismo modo comprender la tecnología utilizada desde su creación hasta el momento en que se transfieren entre monederos.

3.3 Minería de bitcoins

Para poder entender mejor el concepto de bitcoins, debemos primero saber cuál es el proceso que se lleva a cabo para generarlos. Los bitcoins no salen de la nada a la red que los maneja, sino que llevan un proceso de construcción conocido como minar.

Primeramente debemos saber que es una cadena de bloques. La canela de bloques o Blockchain, es la base de datos donde todos los nodos que participan dentro de la red, comparten la información de los hashes que van generando u obteniendo. Una copia completa de dicha cadena, contiene información de cada transacción realizada desde

el comienzo de la historia, con lo cual podemos averiguar el precio de cada dirección en cualquier momento.

El nombre de cadena de bloques hace referencia a que cada hash es un bloque único y a su vez cada bloque contiene el hash del bloque anterior, con ello se garantiza que los hash siguientes dentro de la cadena no puedan ser repetidos y evitar el doble gasto. Además, con todo lo anterior sería computacionalmente imposible modificar un bloque que ya ha sido procesado al no conocer el bloque siguiente la cadena. Para poder validar que un bloque es legítimo, se valida que los bloques y las transacciones sean válidas, además de que el bloque inicial, conocido como génesis, debe de ser el mismo siempre. Con lo cual la seguridad de los bloques siempre será muy fuerte, el único método para poder crear bitcoins falsos será tener un procesamiento muy grande capaz de reconstruir toda la cadena, sin embargo con todos los usuarios existentes que aportan tu trabajo en procesamiento es imposible de alcanzar.

Cada bloque contiene información que ayuda a los mineros a comprobar si es correcta o es información fraudulenta. Es un poco similar a lo que hemos visto en los modelos de red. Tiene una estructura de la siguiente manera.

Tamaño	Campo	Descripción
4 bytes	Tamaño del bloque	El tamaño del bloque, en bytes
80 bytes	Cabecera	Varios campos de la cabecera
1-9 bytes(Variable)	Contador de transacciones	Indica la siguiente transacción
Variables	Transacciones	Transacciones generadas en este bloque

Tabla 7 Estructura de un bloque

Ahora bien, el campo de la cabecera lo conforman los siguientes datos.

Tamaño	Campo	Descripción
4 bytes	Versión	Versión de software para rastrear actualizaciones
32 bytes	Hash anterior	Es la referencia al Hash anterior (padre) en la cadena de bloques
32 bytes	Raíz de Merkle	Un Hash dela raíz del árbol de Merkle de las transacciones de este bloque
4 bytes	Marca de tiempo	Tiempo de creación aproximada del bloque (Medida en tiempo Unix 00:00 1/Enero/1970)
4 bytes	Dificultad	Dificultad para el algoritmo proof-of-work
4 bytes	Nonce	Un contador utilizado por el algoritmo proof-of-work

Tabla 8 Cabecera de un bloque

Cada bloque, dentro de la cadena de bloques, es identificado por el Hash que tiene dentro de su encabezado. Cada bloque hace referencia al bloque anterior, que se considera como el bloque padre, por lo cual podemos ver qué direcciones hicieron cuál transacción. Pero jamás podremos saber el nombre de la persona que realiza la transacción.

Proof-of-work o Prueba de trabajo

La prueba de trabajo es un dato difícil y costoso para producir, pero muy sencillo para que otros puedan comprobar que cumpla con ciertos requisitos. La prueba de trabajo

utilizada por bitcoins, es el HashCash, el cual era una solución para evitar el correo basura.

Consiste en agregar una marca en el correo que solo si se está dispuesto a “pagar” con tiempo de CPU se puede agregar. De tal forma que si una persona malintencionada quiere enviar correos basura en grandes cantidades, será tendrá que hacer una muy grande inversión de dicho tiempo. La idea fue utilizada en el concepto de bitcoins, debido a que era necesario tener una manera de comprobar que cada bloque era realmente valido.

La cadena de bloques, está constituida por pequeños host anónimos que le dan el mantenimiento, es decir, todos los host de la red comprueban que cada bloque es verdadero, sin embargo, los host se dividen en dos, los que colaboran con más trabajo para verificar los bloques y los que simplemente se dedican a hacer transacciones.

Para hacer notar esta diferencia entre los host, se agrega un valor que lleva por nombre dificultad. Este apartado es utilizado para corroborar que parte de los host trabajan más que otros. Al ir encadenando los bloques es casi imposible querer realizar alguna modificación, por lo cual vemos que por cada nuevo bloque añadido, la dificultad va a ir subiendo.

Para producir o comprobar, a cada bloque se le asigna un valor de dificultad. Para poder procesar dicho bloque, pequeños grupos de host se unen para resolver el problema, y es entonces cuando la dificultad obtiene un valor y es pagado a todo el grupo, dependiendo del trabajo de CPU invertido.

El HashCash para bitcoin, simplemente consiste en agregar 20 bits en cero a la izquierda. De modo que, el costo computacional de alguien que quiera falsificar un bloque aumenta considerablemente, ya que, tiene que aplicar 2^{20} Hashes para encontrar un resultado valido.

La privacidad de un minero está dada por el contador nonce, dicho contador representa el esfuerzo que se realiza para comprobar le bloque, y si un minero tiene un gran poder

de trabajo, después de terminar se reinicia. De lo contrario, esto propicia revelar el nivel de esfuerzo de un minero y por tanto implica que los bitcoin le pertenecen. Para mantener el anonimato y así nadie podrá saber si se trataba de alguien que trabajaba duro o simplemente algún afortunado, el contador siempre debe de ser regresado a cero.

Una vez teniendo claro todo lo anterior, la minería de bitcoins consiste en agregar nuevos bloques a esa enorme cadena principal. Esta minería se divide en dos partes, la primera es donde un minero intenta generar un nuevo bloque por su cuenta obteniendo su recompensa o comisión que solo es destinado para sí mismo.

La segunda opción, consiste en juntar un grupo de mineros para generar nuevas cadenas y al final la recompensa es distribuida dependiendo de número de hash procesados de cada uno.

La diferencia entre estas dos opciones de minado es una relación entre el tiempo invertido y la recompensa obtenida. Como podemos observar, cuando una sola persona trata de minar bitcoins, el tiempo de procesamiento será mucho mayor al tiempo que se invierte si se genera un grupo que puede aportar mucha más velocidad de procesamiento, al final siempre se obtiene una recompensa, solamente es cuestión de ser pacientes y esperar.

Cada uno de los nodos se convierten en generadores, compiten en la red, solos o en grupo para resolver problemas criptográficos con la finalidad de obtener un nuevo bloque legítimo, todo esto requiere de una gran cantidad de soluciones por segundo, utilizando la fuerza bruta. Es aquí donde radica una parte de su seguridad, al no generar todo de manera determinista, los mineros con un nivel de procesamiento no quedan fuera y por ello la probabilidad de encontrar un bitcoin depende más del poder computacional con el que contribuyen a la red, así mismo, para poder realizar bloques ilegítimos necesitamos superar el nivel de procesamiento para obtener con éxito dichos bloques.

3.4 Árboles de Merkle

Un árbol de Merkle o árbol de hashes, es una estructura de datos en forma de árbol, en donde cada nodo interior es el resultado de aplicar una función hash a uno o más nodos hijos hasta poder llegar a un nodo principal o nodo raíz. Esta dicha estructura es muy útil para el protocolo Bitcoin, pues permite que una gran cantidad de datos sean separados para así operar independientemente nodo por nodo haciendo que la verificación de nodos sea más pequeña, rápida y eficiente debido a que la información por comprobar será más breve.

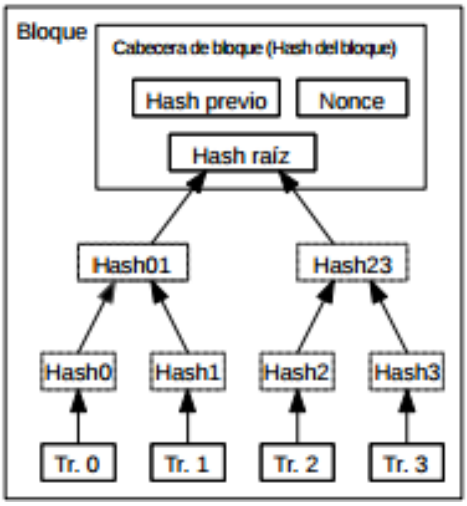

Imagen 36 Transacciones hash en un árbol de Merkle

Además de ayudar a facilitar la verificación de cadenas de hashes, tiene otra utilidad. Ahorrar espacio de disco, como bien hemos visto, el árbol almacena hashes interiores y estos, al no tratarse del nodo raíz, a su vez son nodos hijos. Tomando esto en cuenta, los hashes almacenados y muy antiguos también se les pueden aplicar una función hash y así ir reduciendo nodos del árbol y por tanto reducir espacio de memoria.

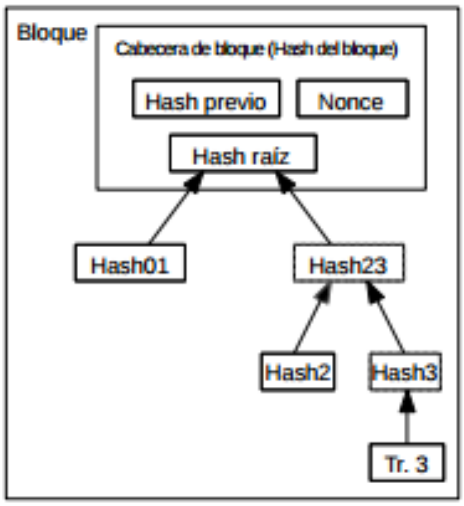

Imagen 37 Eliminando Hashes antiguos

"La cabecera de bloque sin transacciones pesaría unos 80 bytes. Si suponemos que los bloques se generan cada 10 minutos, 80 bytes × 6 × 24 × 365 = 4.2MB por año. Siendo habitual la venta de ordenadores con 2GB de RAM en 2008, y con la Ley de Moore prediciendo un crecimiento de 1.2GB anual, el almacenamiento no debería suponer un problema incluso si hubiera que conservar en la memoria las cabeceras de bloque". (Nakamoto, 2008, pág. 4)

3.5 Transacciones

Como bien hemos visto, las monedas virtuales simplemente son una cadena de hashes, donde cada propietario al transferir una moneda o cierta parte, realiza una firma digital sobre el hash a transferir y la llave pública del usuario que recibirá el monto. Con este proceso, es mucho más sencillo realizar la verificación para comprobar la validez del hash recibido.

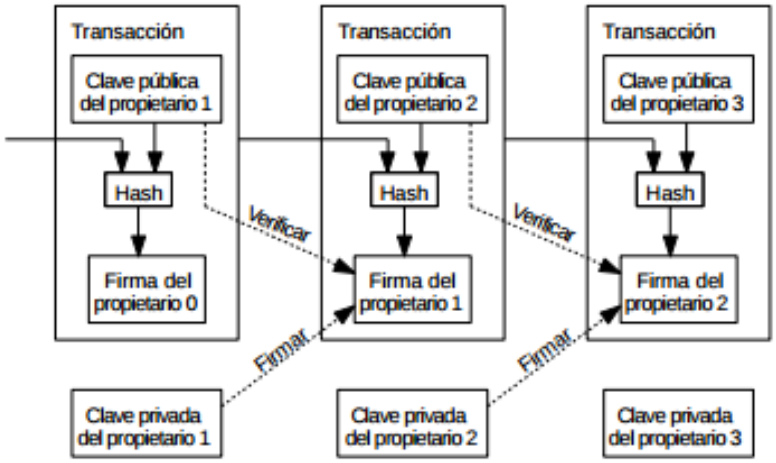

Imagen 38 Transacciones Hash

Un pregunta que muchas personas llegan a pensar es, ¿Cómo sabemos si realmente un hash no ha sido utilizado dos veces? Pues bien, con una moneda convencional, simplemente contamos con una casa de monedas, banco o entidad gubernamental que hace esa verificación y con ello estamos tranquilos. Pero en el caso de las monedas virtuales no tenemos a alguien es especifico que realice esta tarea. Para poder comprobar de una mejor manera los hashes, es necesaria hacer esa transferencia pública (recordemos que se trata de una red P2P), ahora bien, todos los equipos que colaboran en la red, usuarios o mineros, trabajan en conjunto para llevar a cabo la verificación de un hash y poderlo agregar a la cadena. Cuando una cadena fue verificada y agregada ya no puede ser cambiada, es por ello que aumenta el nivel de seguridad mientras más usuarios están conectados a la red P2P, para que alguien pueda realizar una cadena falsa, es necesario superar el nivel de procesamiento total de la red para así agregar primero la cadena. Y por ende, cada que uno o varios mineros realizar una verificación, se les otorga un incentivo, que en nuestro caso cubre el uso de CPU o más específicamente el gasto de luz eléctrica.

Pero, ¿Qué pasa cuando no podemos hacer pública la transacción para ser verificada? Es aquí donde nos es más útil el uso de los árboles de Merkle, para ello, el nodo que va realizar la verificación debe de tener consigo una copia de las cabeceras de la cadena, la más larga para que el bloque sea añadido después sin problemas, y

después verificar su hash en una de las ramas para posteriormente añadirlas a la red principal. Sin embargo es recomendable realizar la sincronización lo más antes posible para ser agregada a la cadena principal.

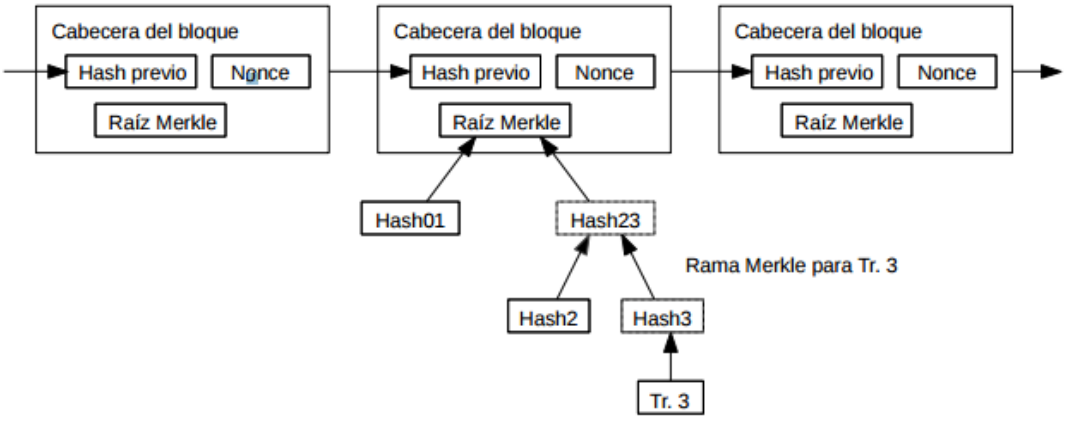

Imagen 39 Copia de las cabeceras de la cadena

En la imagen 40 se puede observar la información de un grupo de mineros realizando la transacción número 2082, resolviendo un bloque y obteniendo uno nuevo, el valor obtenido por minar dicho bloque fue de 12.5 Ƀ. Además de observar el hash anterior y el obtenido, también se muestra la raíz de Merkle de donde surgió.

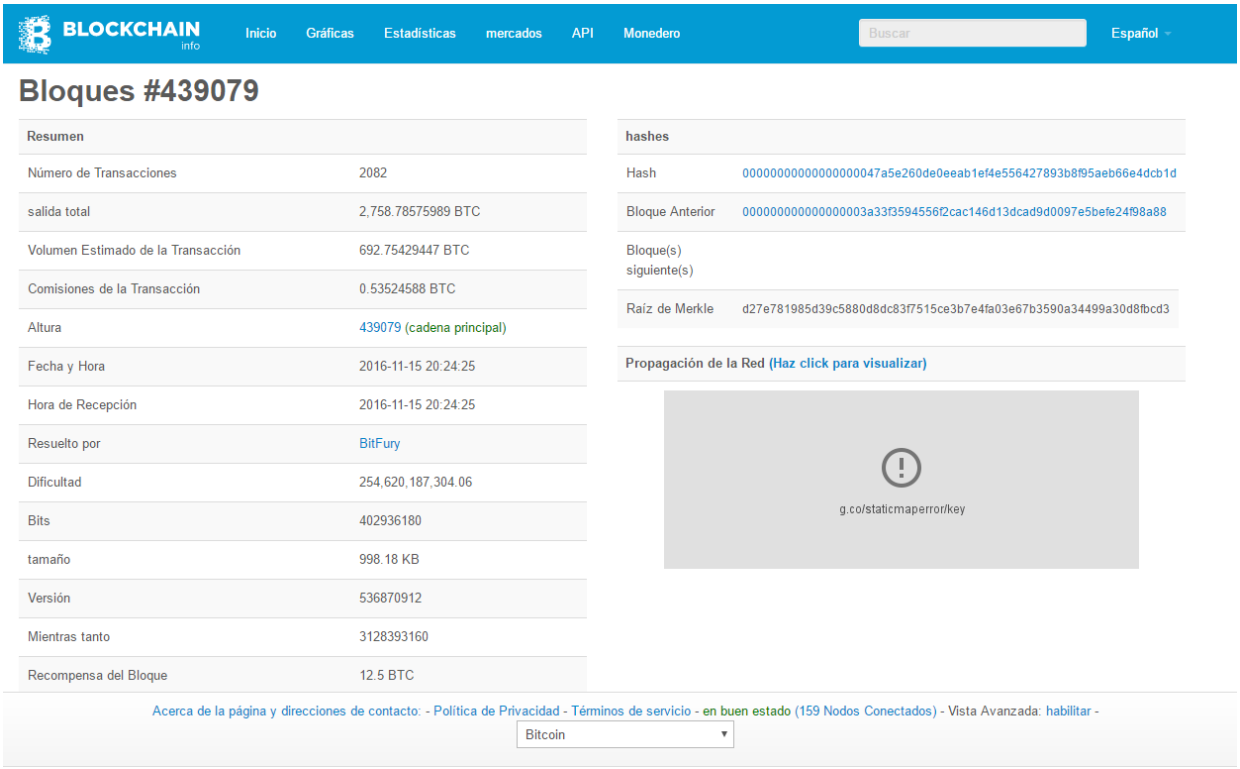

Imagen 40 Ejemplo de bloque (15/11/16 - 14:32)

Toda esa información es pública, sin embargo no se sabe que personas intervinieron en la transferencia, en este caso, BitFury, muy seguramente se encarga de distribuir esa recompensa entre sus mineros.

3.6 ¿Cómo trabajan los bitcoins?

Primeramente, como ya se ha mencionado, el número máximo de bitcoins que estarán en circulación son 21 millones, la siguiente gráfica nos da un aproximado del crecimiento que habrá hasta llegar a esa cantidad.

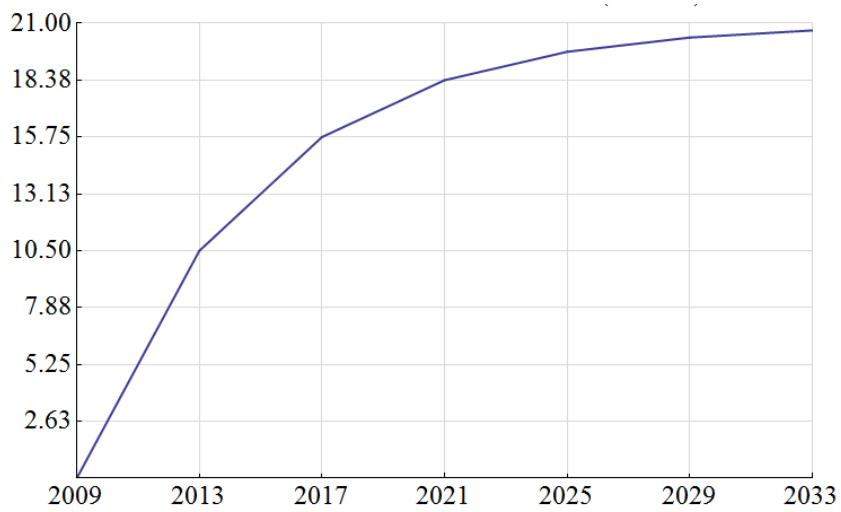

Imagen 41 Estimado del total de bitcoins

Un bitcoin no es más que una serie de caracteres cifrados a los cuales les corresponde un número que representa los bitcoins. Esa cadena de caracteres normalmente es de entre 27 y 34 caracteres, en su mayoría 34 caracteres representados por dígitos y letras, tanto mayúsculas como minúsculas. Para evitar confusiones, los caracteres omitidos son la letra “O”, el número “0”, la letra minúscula “l” y la letra mayúscula “i”.

Ahora bien, las direcciones más cortas a 34 caracteres, en teoría con un mínimo de 27, son el resultado de obtener valores numéricos que iniciaban con ceros. Aunque ya no es muy común encontrarlas.

Pero la duda más importante es, ¿Cómo se crean las direcciones de bitcoins? Pues bien, solo basta con aplicar los algoritmos de cifrado antes vistos de la siguiente manera en solo 10 pasos.

1. Tener una llave privada ECDSA:

   18E14A7B6A307F426A94F8114701E7C8E774E7F9A47E2C2035DB29A206321725

2. Tomar la correspondiente llave publica

0450863AD64A87AE8A2FE83C1AF1A8403CB53F53E486D8511DAD8A04887E5B23522CD470243453A299FA9E77237716103ABC11A1DF38855ED6F2EE187E9C582BA6

3. Se aplica el algoritmo SHA-256 a la llave pública

   600FFE422B4E00731A59557A5CCA46CC183944191006324A447BDB2D98D4B408

4. Al resultado del algoritmo SHA-256 se le aplica el algoritmo RIPEMD-160

   010966776006953D5567439E5E39F86A0D273BEE

5. Se añade el byte de versión al inicio de la cadena (0x00 para Main Network)

   00010966776006953D5567439E5E39F86A0D273BEE

6. Al resultado anterior de RIPEMD, se le aplica el algoritmo SHA-256 nuevamente

   445C7A8007A93D8733188288BB320A8FE2DEBD2AE1B47F0F50BC10BAE845C094

7. Una vez más se aplica el algoritmo de SHA-256 sobre el resultado anterior

   D61967F63C7DD183914A4AE452C9F6AD5D462CE3D277798075B107615C1A8A30

8. Se toman los primero 4 bytes del punto anterior, el cual será el identificador de nuestro Hash

   D61967F6

9. Al resultado del punto 5, añadir al final de la cadena el identificador anterior

00010966776006953D5567439E5E39F86A0D273BEED61967F6

10. La cadena anterior es el resultado en bytes, simplemente hay que transformada en una cadena en base58, que es el formato de direcciones más utilizado.

16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM

EL motivo por el cual el algoritmo RIPEMD-160 es utilizado, es debido a que produce una salida de Hash más corto, esto permite que una dirección bitcoin pueda ser más corta y sin tener que comprometer la seguridad. Por otro lado, se utiliza la codificación en base 58 para convertir la cadena final en un texto mas legible para el ser humano, es decir, simplemente se trata de retirar los caracteres 0 (cero), O, I (i mayúscula) y l (L minúscula), con lo que se pretende evitar confusiones visuales.

Para obtener una representación en base58, simplemente tomamos el resultado del paso número 9, lo convertimos a decimal y obtenemos el siguiente resultado (Recordar que la conversión será de hexadecimal a decimal).

25420294593250030202636073700053352635053786165627414518

Ahora el resultado anterior debemos dividirlo entre 58, el resultado de esa división, debemos aplicarle la operación de modulo 58 con lo que obtenemos como resultado 20. A continuación debemos dividir entre 58 nuevamente el resultado 25420294593250030202636073700053352635053786165627414518 y obtener el modulo 58 con lo cual obtenemos el valor de 53. Repitiendo la división y la operación modulo hasta que no se pueda más obtenemos los siguientes resultados de cada modulo 20, 53, 42, 51, 20, 54, 6, 23, 10, 44, 16, 38, 46, 18, 53, 27, 10, 48, 22, 38, 23, 2, 35, 50, 41, 24, 8, 19, 19, 54, 27 y 5.

Con ayuda de la tabla 7 haremos la representación de cada carácter para obtener nuestra cadena final, en donde nuestro ultimo valor encontrado (5) será el valor más significativo.

Valor	Carácter	Valor	Carácter	Valor	Carácter	Valor	Carácter
0	1	1	2	2	3	3	4
4	5	5	6	6	7	7	8
8	9	9	A	10	B	11	C
12	D	13	E	14	F	15	G
16	H	17	J	18	K	19	L
20	M	21	N	22	P	23	Q
24	R	25	S	26	T	27	U
28	V	29	W	30	X	31	Y
32	Z	33	a	34	b	35	c
36	d	37	e	38	f	39	g
40	h	41	i	42	j	43	k
44	m	45	n	46	o	47	p
48	q	49	r	50	s	51	t
52	u	53	v	54	w	55	x
56	y	57	z				

Tabla 7 Valores de Base58

Una vez que tenemos el resultado (6UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM), se agrega el carácter que se utiliza para indicar cual es la versión utilizada para distinguir entre cada moneda al principio de la cadena, a continuación tenemos en la tabla 8 algunos de los prefijos de diferentes direcciones de otras monedas.

Versión decimal	Símbolo inicial	Uso	Ejemplo	Longitud de la dirección
0	1	Hash de la clave pública bitcoin	12CPLrAUPvhVwjZqBgww3sLdEg4Z888R1j	Hasta 34
5	3	Hash del script bitcoin	3EktnHQD7RiAE6uzMj2ZifT9YgRrkSgzQX	34
48	L	Hash de clave pública de Litecoin	LhK2kQwiaAvhjWY799cZvMyYwnQAcxkarr	34
52	M o N	Hash de la clave pública Namecoin	NATX6zEUNfxfvgVwz8qVnnw3hLhhYXhgQn	34
111	m o n	Hash de la clave pública bitcoin en testnet	mkJ7Bf5chdfw61d1m7gnDVAQV3EQQAb8iz	34

Tabla 8 Prefijos de direcciones

Finalmente es así como llegamos a obtener el resultado, para el caso de BITCOINS (16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM), utilizando el prefico correspondiente.

Los bitcoins pueden ser comparados con cualquier moneda en circulación, la diferencia radica en que cualquiera moneda es manejada por una autoridad central, alguna

institución intermedia que tiene el control sobre la moneda, en cambio con los bitcoins no pasa lo mismo. Estos son manejados por los usuarios en términos de creación, compra y venta, es decir, los usuarios de bitcoins al generar y hacer uso de la moneda, son considerados como los “controladores”. Por lo cual los bitcoins son considerados como una divisa descentralizada al no tener a alguien que los controle.

Bitcoin es una criptomoneda ya que para realizar la creación de una pieza se utiliza un protocolo mediante el uso de la criptografía. Desde un punto de vista, es una moneda considerada anónima, porque el usuario no necesita agregar información adicional para realizar una transacción, y al utilizar la criptografía pasa generación, solo basta con tener un seudónimo con el que se llevará a cabo el cifrado para hacerlo anónimo, y al momento de ir circulando de usuario en usuario lleva siempre un proceso de cifrado para cada movimiento realizado, por lo cual siempre que tenemos un Bitcoin, anteriormente llevó un proceso de cifrado que lo hará más fuerte e imposible de rastrear su origen al realizar tantos cifrados cuando pasa de persona en persona. Por lo tanto, este proceso que se lleva a cabo hace que la moneda sea irreversible, aumentando la seguridad de la transacción por parte del comprador y el vendedor, dándoles el papel de ser anónimos para el mundo.

3.7 ¿Qué es un monedero electrónico?

Un monedero de bitcoins es un sistema diseñado para facilitar el uso de la criptomoneda que realiza las transacciones en tiempo real, siempre que se tenga conectado a la red Peer-to-Peer. Un usuario de bitcoins puede tener distintos tipos de monederos, monederos web, monederos basados en un software, monederos basados en un hardware y monederos para teléfonos inteligentes. La diferencia entre cada uno de ellos radica en el tipo de uso y los niveles de seguridad que ofrecen.

3.7.1 Monederos web

Muchas personas prefieren usar esta alternativa por lo sencillo que es su uso. Simplemente basta con iniciar sesión y realizar la transacción deseada. Sin embargo, los monederos web carecen de seguridad frente a otro tipo de monedero ya que existe la posibilidad de que una persona ajena entre al sitio web simplemente instalando algún tipo de **script** en el servidor donde se aloja el sitio web y sustraiga información muy sensible, incluyendo la llave privada.

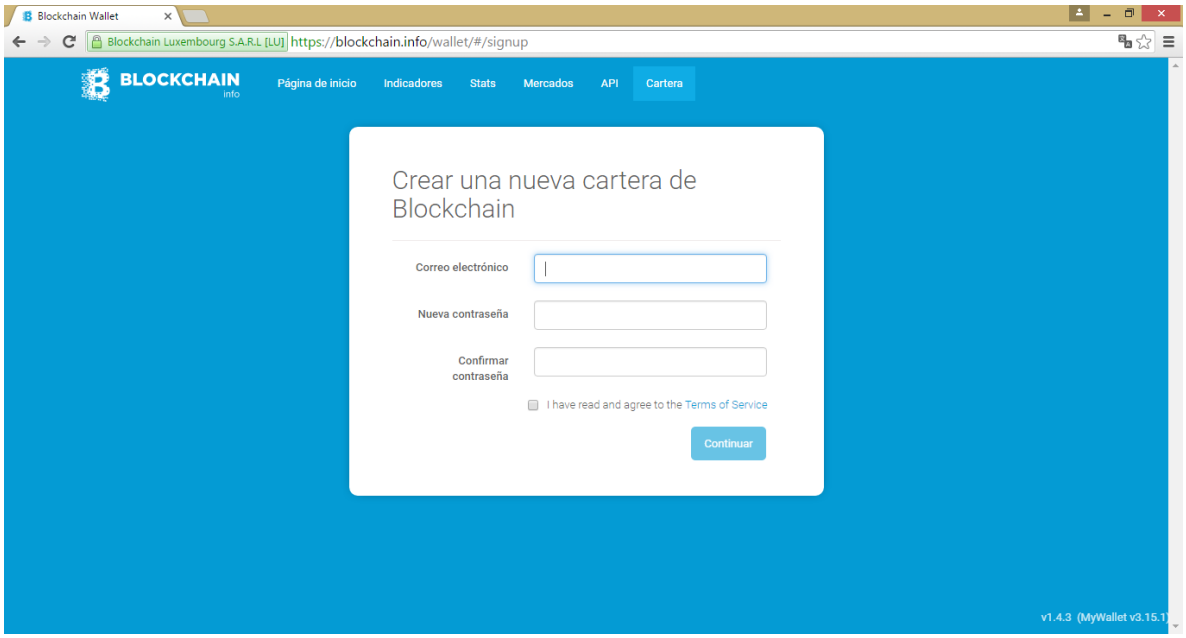

Imagen 42 Monedero Web

3.7.2 Monedero basado en software

Este tipo de monedero, a diferencia de los monederos web, son más seguros. Su uso de igual manera es muy sencillo, simplemente debemos hacer una descarga de un software destinado para clientes e iniciar sesión para poder comenzar a realizar

transacciones. Una de las desventajas es que para que el equipo de cómputo que se pretenda utilizar, es necesario realizar una única descarga de lo que se conoce como la cadena de bloques, para poder estar actualizada en la red Peer-to-Peer.

Pero la seguridad aumenta a comparación de los monederos web, ambos monederos están expuestos al mundo ya que están conectados a internet, sin embargo, sabemos que el servidor que aloja el sitio web debe de ser más popular que una simple computadora, lo que convierte al sitio web en un blanco mucho más fácil de atacar, y al mismo tiempo se convierte la computadora del usuario en un lugar más seguro para almacenar nuestras llaves para realizar los cifrado de nuestros bitcoins.

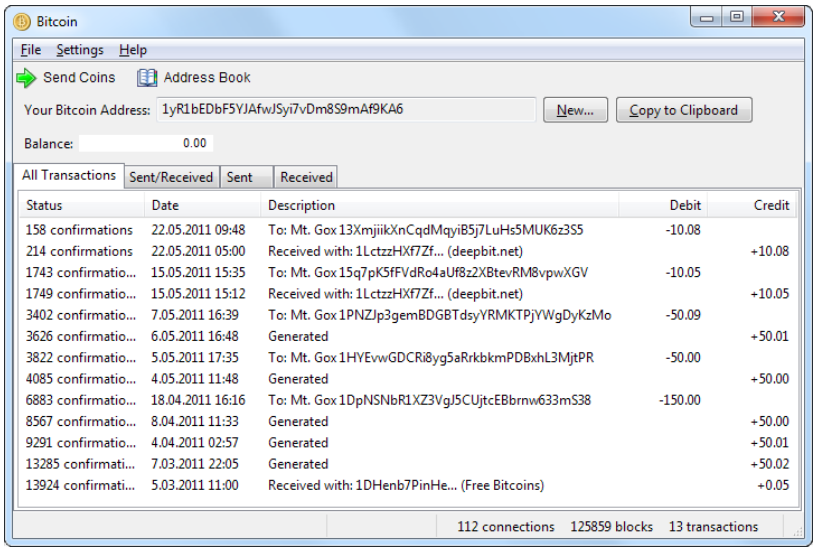

Imagen 43 Monedero basado en software

3.7.3 Monedero basado en hardware

Un monedero basado en hardware, es un dispositivo en donde se almacena la información de cada usuario de Bitcoin, sus bitcoins y su llave privada. Podría incluso decirse que es un equipo de cómputo, capaz de realizar los procesos necesarios para la transacción de la criptomoneda incluso sin la necesidad de estar conectado a la red

Peer-to-Peer. Se puede usar de manera offline y después basta con solo conectarlo un momento a la red para poder actualizar los datos y las transacciones.

Generalmente la información importante (llave privada) se almacena en una parte de memoria que es protegida por un microprocesador que se encarga de que la llave no salga del dispositivo en texto en claro, sino que la única forma de poder obtener la llave será de tal manera que la llave salga cifrada del dispositivo. Esto con la finalidad de tener una mayor seguridad en dispositivos de este tipo.

Imagen 44 Monedero basado en hardware

Comparado con los demás monederos, este tipo es considerado como el más seguro ya que como hemos visto, la posibilidad de obtener la llave en texto plano es casi imposible, y además estos monederos son implementados para ser inmunes a los virus informáticos que traten de robar credenciales de software.

Por otra parte, con tantos avances en la tecnología, se han vuelto muchísimo más sencillos de utilizar además de ser muy intuitivos para los usuarios finales. Simplemente basta con conectarlo por medio de un puerto USB a un computador para

poder lograr hacer una transferencia. Incluso con tantos avances en vez de realizar la conexión con una computadora, se puede conectar a un teléfono inteligente por medio de un cable y realizar las operaciones por esté medio. Hoy en día, además de realizar la comunicación entre el monedero y algún dispositivo por medio de un cable, existe la posibilidad de utilizar la tecnología NTFS o Bluetooth, para realizar la sincronización del monedero por medio de una conexión inalámbrica.

En pocas palabras, todo el procesamiento se genera dentro de cada monedero de hardware, incluso el almacenamiento y generación de llaves, la conexión hacia otros dispositivos se utiliza simplemente para realizar una sincronización con la red para contribuir con los bloques de hash.

3.7.4 Monederos para teléfonos inteligentes

En la actualidad, los teléfonos inteligentes o Smartphones son muy populares en cualquier parte de planeta, inclusive niños de muy pequeñas edades cuentan ya con un dispositivo de estos. Y así como podemos darnos cuenta que todo mundo tiene o simplemente conoce un Smartphone, podemos observar como sus sistemas operativos son muy intuitivos y por ende las aplicaciones destinadas para usuarios deben de ser muy sencillas. En el caso de los monederos utilizados con un Smartphone son muy fáciles de manipular, simplemente podríamos compararlos con los monederos web, sin embargo, con una aplicación es más enfocado a un solo propósito.

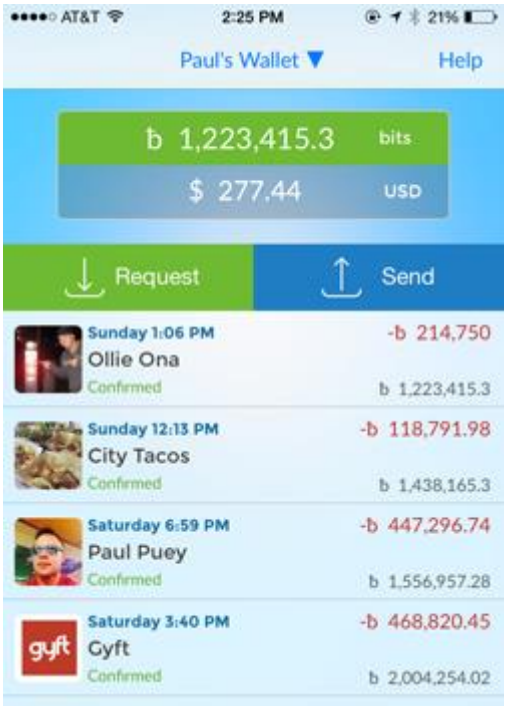

Imagen 45 Monederos para teléfonos inteligentes

La principal ventaja que tienen los monederos virtuales es la portabilidad. Es mucho más sencillo llevar un monedero a cualquier lugar que llevar una computadora, además, los recursos de los Smartphones generalmente no son tan grandes y es por ello que para mejorar el rendimiento no es necesario descargar nada ya que todas las transacciones son procesadas en un servidor de la aplicación que brinda el servicio.

3.8 ¿Cómo comprar bitcoins?

Para poder entrar al mundo de los bitcoins, es necesario ubicar un Exchange de bitcoins, es decir, alguna empresa que se dedique a la compra y venta de bitcoins. Para este ejemplo utilizaremos el Exchange Bitso.

Bitso, en simples palabras, es una página web donde puedes comprar y vender bitcoins con pesos mexicanos. Además podemos utilizarlo como monedero web para almacenar nuestros bitcoins comprados.

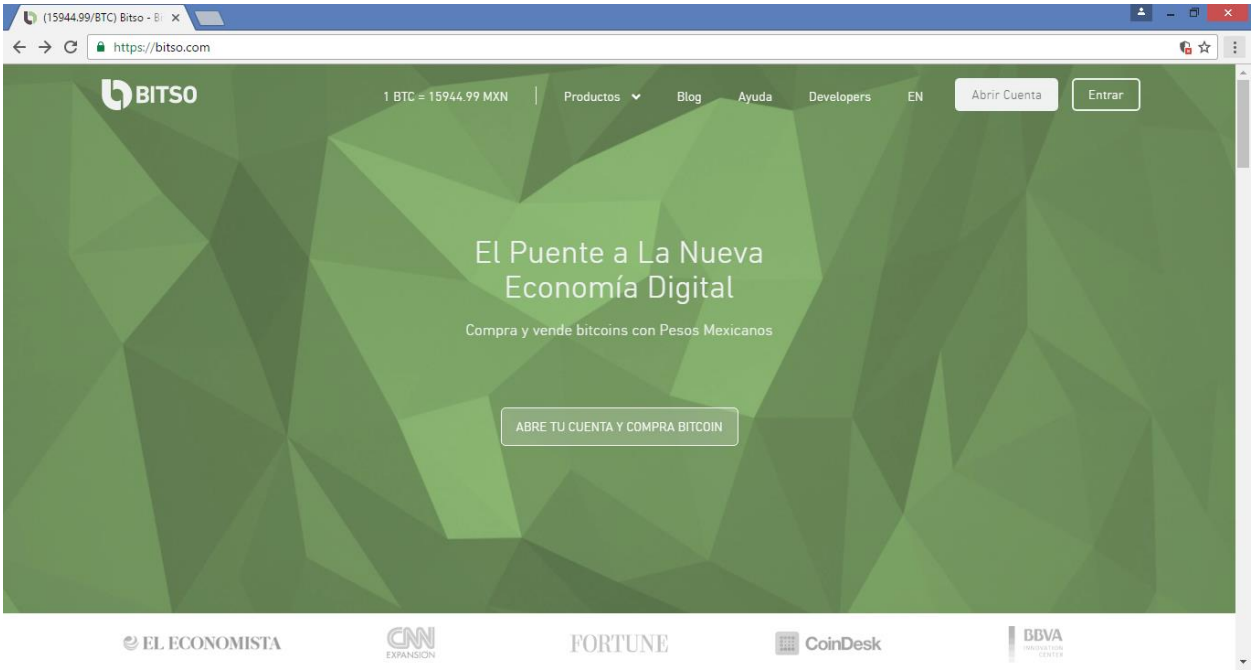

Imagen 46 Bitso.com

Podemos encontrar información sobre el costo del bitcoin en pesos mexicanos. Ahora para poder realizar nuestra primera compra, debemos registrarnos en la página, es muy sencillo.

Una vez que tenemos acceso a nuestra cuenta veremos algo así.

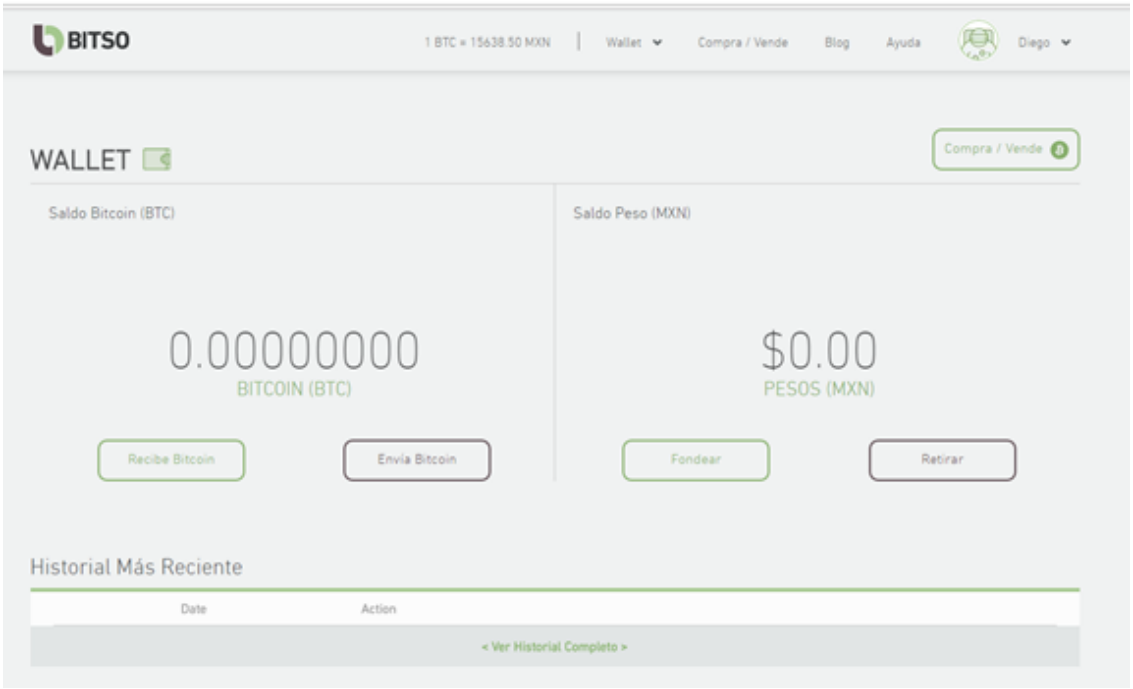

Imagen 47 Monedero Bitso

En esta parte, tenemos dos secciones, nuestro saldo en bitcoins y nuestro saldo en pesos. Primero debemos Fondear la cuenta, esta es la manera en la que se refiere a abonar dinero en pesos mexicanos para realizar la compra de bitcoins.

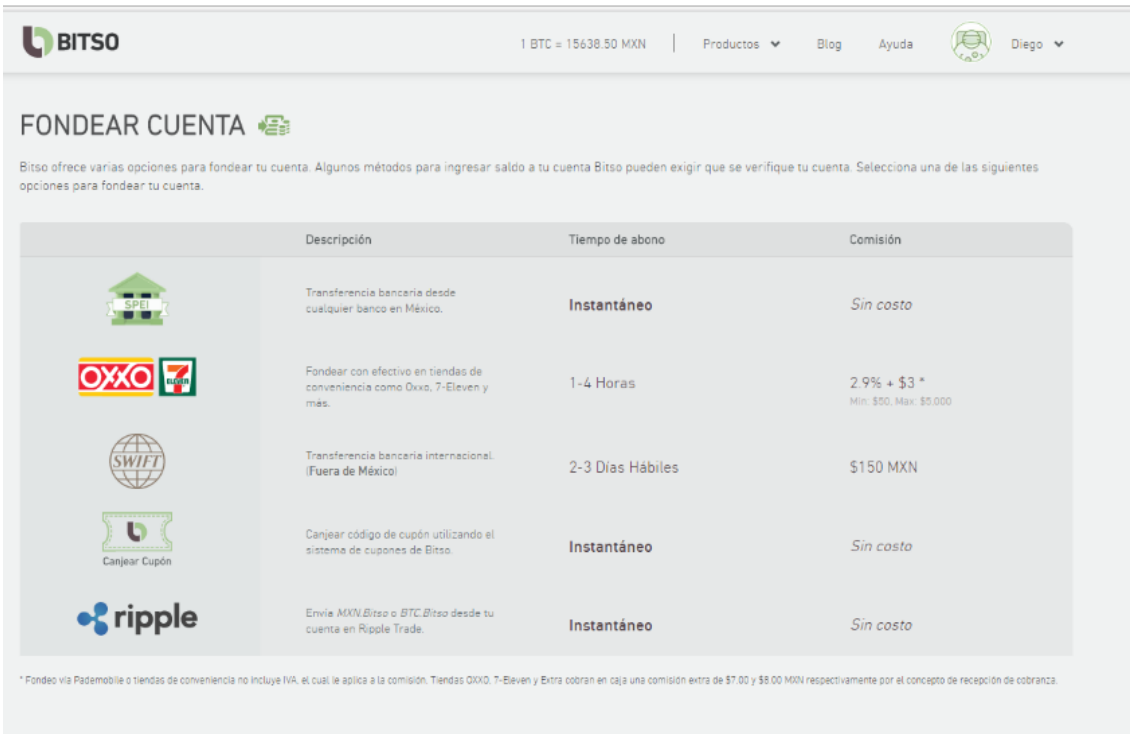

Imagen 48 "Fondear" cuenta

Tenemos cinco diferentes maneras para agregar saldo en pesos mexicanos. Para este ejemplo utilizaremos una transferencia bancaria.

Imagen 49 Datos para Fondear cuenta

En la imagen 49, vemos los datos necesarios para llevar a cabo la transferencia. Y una vez hecha, vemos una notificación en la parte inferior de la página corroborando nuestra transacción.

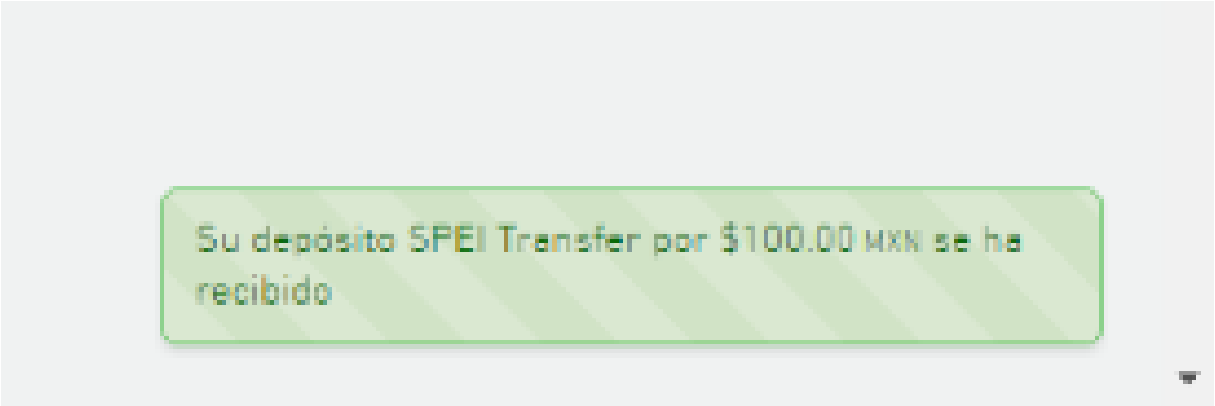

Imagen 50 Transferencia exitosa

Ahora que ya tenemos dinero en nuestra cuenta, simplemente regresamos a la página principal.

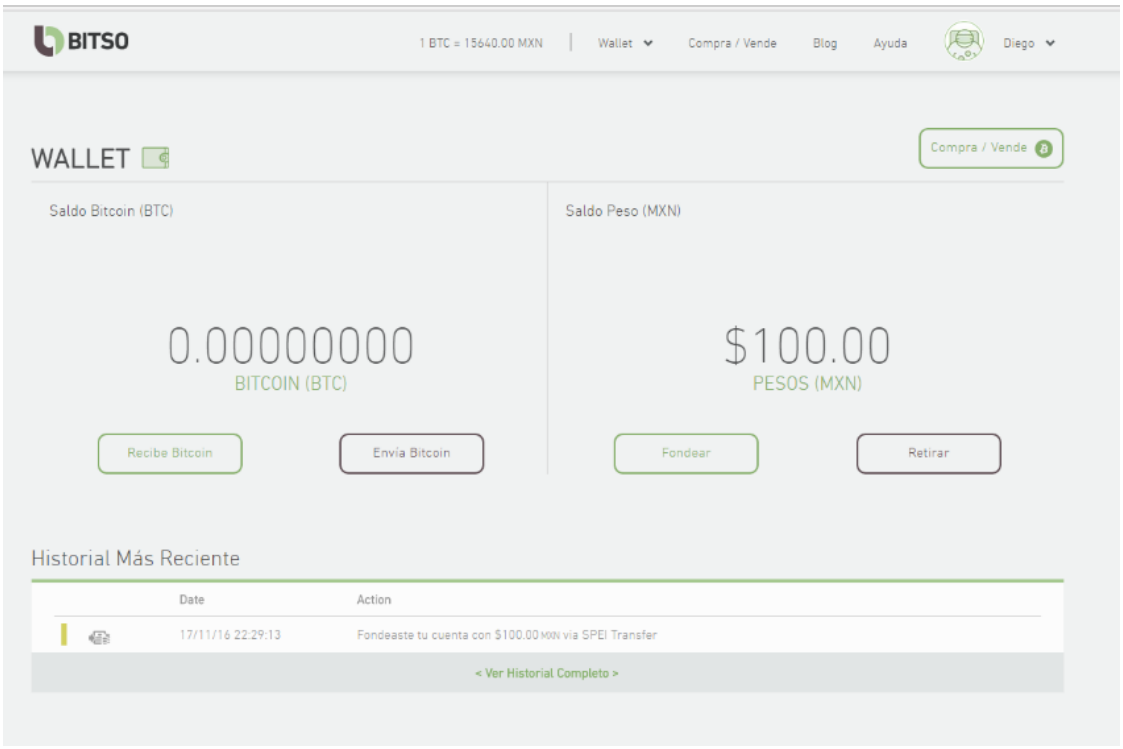

Imagen 51 Monedero con saldo en pesos mexicanos

Para realizar nuestra primera compra, vamos a la sección donde dice “Compra / Venta”. Una vez ahí debemos seleccionar la cantidad en pesos que deseamos comprar.

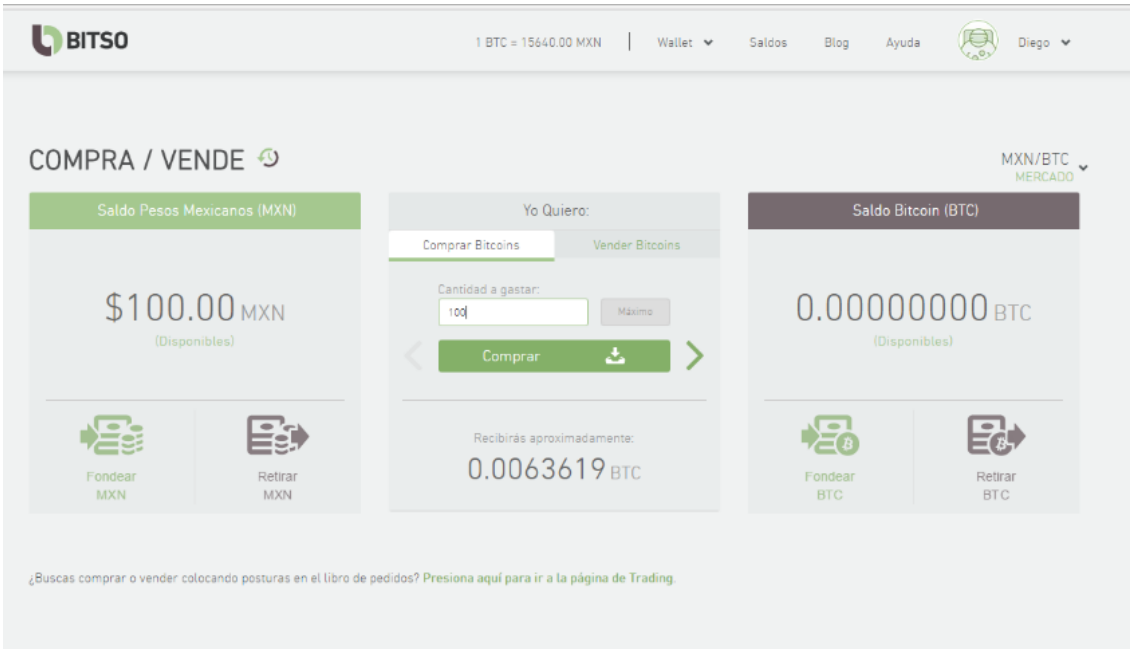

Imagen 52 Compra de bitcoins

La interfaz es muy amigable con los usuarios, como podemos ver, cada que se ingresa una cantidad automáticamente se hace la conversión a bitcoins.

Ahora bien, para realizar la compra solamente se ingresa la cantidad y se da en comprar.

Regresando a la página principal, podemos ver que nuestro saldo en pesos se le resto la cantidad comprada en bitcoins, además, podemos ver ya reflejados los bitcoins en nuestro monedero.

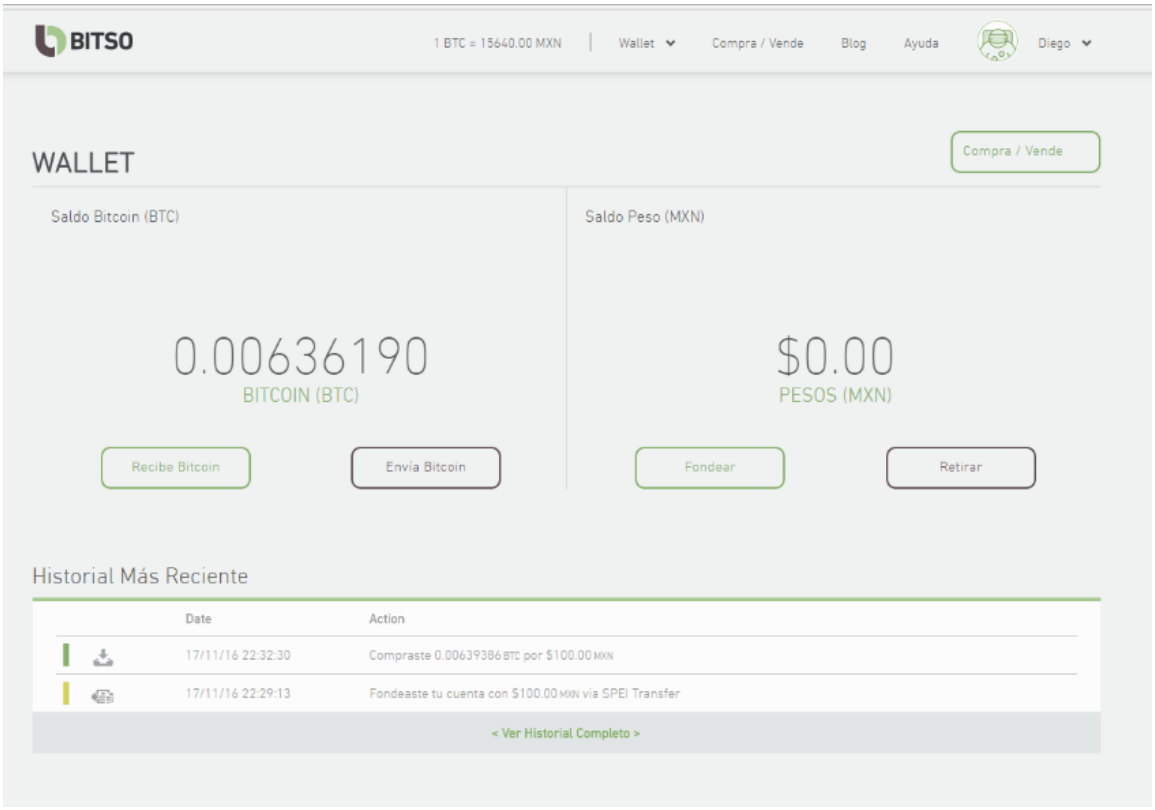

Imagen 53 Monedero con saldo en bitcoins

Con ello podemos realizar compra de bitcoins. Sin embargo, para realizar una compra, simplemente hace falta realizar un envío hacia otra cuenta. Para ello seleccionamos en "Envía bitcoins".

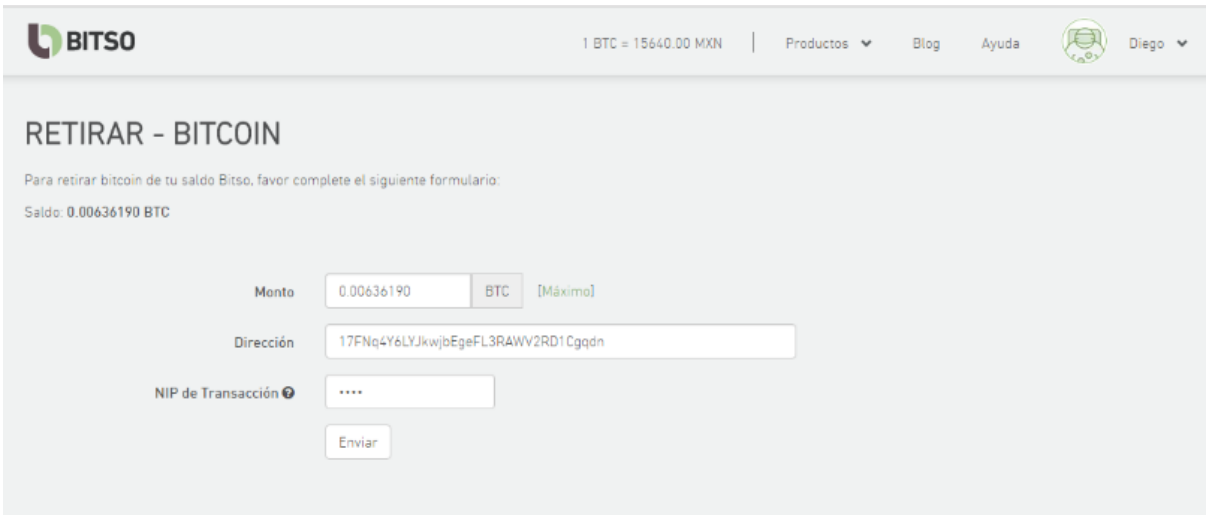

Imagen 54 Datos para envío de bitcoins

En esta sección debemos elegir el monto que vamos a enviar a otra cuenta, así como la dirección a donde serán agregados los bitcoins, y en NIP que autoriza la transacción. Mientras tanto, en el otro monedero que va a recibir los bitcoins, debemos de buscar la dirección para recibirlos.

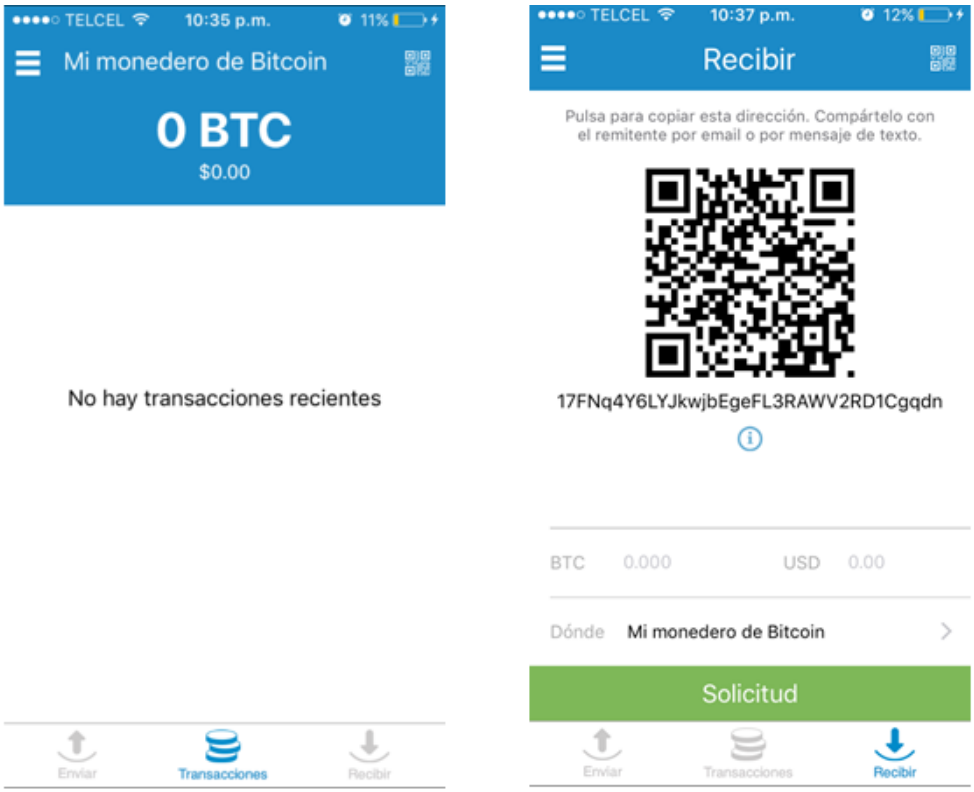

Imagen 55 Monedero y dirección para recibir bitcoins

Esa dirección la podemos intercambiar, como podemos ver, con un código QR, o bien, compartiendo la dirección. Y después cumplir con la transacción, veremos reflejado en la otra cuenta nuestro dinero en bitcoins.

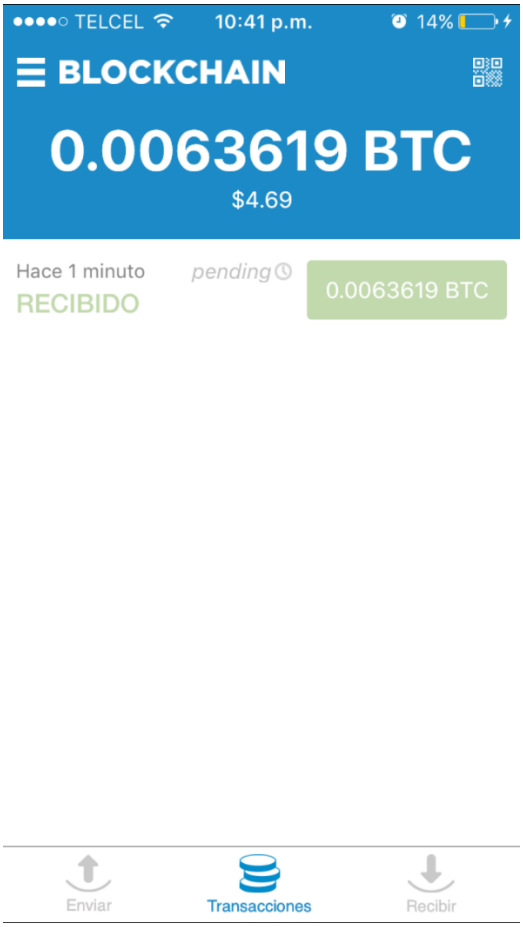

Imagen 56 bitcoins recibidos

3.9 Monedas virtuales alternas

Después de ver el éxito que tuvo la moneda bitcoin, diferentes desarrolladores comenzaron a realizar muchos tipos de monedas para dar más alternativas. Sin embargo, la gran mayoría están basadas en el protocolo de Bitcoin, utilizando parámetros de diseño e incluso tomando el código base para su creación.

Imagen 57 Diversas monedas virtuales

La diferencia entre todas las monedas que existen hoy en día son muy pocas, como por ejemplo:

Los algoritmos utilizados. En algunos casos, el algoritmo de cifrado SHA-256, utilizado por Bitcoin, Peercoin, Namecoin, entre otras, es cambiado por un algoritmo de nombre Scrypt, utilizado por Litecoin. Dicho nuevo algoritmo es muy similar a los ya antes mencionados, la diferencia radica en que SHA-256 utiliza la capacidad del procesador, mientras que scrypt utiliza la memoria RAM disponible. Cabe destacar que es más rentable ser minero utilizando un procesador, porque utilizar memoria RAM aún es muy caro. Por otro lado, el hecho de tener limitado el consumo de memora RAM, es más que nada con la finalidad de que en su momento, cuando divisa llegue a su tope establecido, cualquier usuario con una computadora convencional sea capaz de realizar la verificación, así como, tener a los verificadores mejor distribuidos.

Otra diferencia que existe, es el tiempo de espera entre confirmaciones. El tiempo estimado en Bitcoin es de 10 minutos, por lo que es más fácil tratar de realizar un “doble gasto”, mientras que en Litecoin, el tiempo es de 2.5 minutos. Dogecoin es de 1 minuto. Es una gran ventaja ser más veloz, ya que ese tiempo de confirmación es el tiempo que tarda en ser escrita en la cadena de bloques.

3.10 Beneficios

El mayor de los beneficios que obtenemos al utilizar la moneda virtual es sin duda el ser descentralizada, manejado única y exclusivamente por los usuarios que la utilizan día a día, sin la necesidad de contar con algún intermediario, banco o autoridad central que sea la encargada de su manejo y al ser una economía descentralizada, supera con creces todas las ventajas del dinero actual y soluciona muchos de sus problemas, por lo cual podemos enviar y recibir dinero instantáneamente, en cualquier momento y sin un monto límite sin tener un horario para realizarlo.

Otro punto importante dentro de los beneficios que obtenemos al ser usuarios de bitcoins es el anonimato que genera la moneda. Con todos los algoritmos de cifrado que son aplicados a los bitcoins, es muy complicado saber quién era el dueño de un bitcoin ya que al haber pasado varias veces por un algoritmo de cifrado no es nada sencillo encontrar al el dueño original o algún dueño que tuvo anteriormente. Lo cual, para muchos de los usuarios, es un gran alivio pues su información económica no queda al descubierto para todo el mundo.

Gracias al protocolo de Bitcoin, el hecho de que el número de usuarios aumente hace que todo sea mucho más seguro, haciendo posible que los algoritmos de cifrado sean capaces de mantener la información oculta, otra parte de la seguridad se basa en la una cadena principal y con ayuda de los mineros se realiza una comprobación de que el bitcoin de una transacción sea legítimo, hace aún más seguro nuestro protocolo.

La forma de transporte de la moneda virtual es mucho más sencilla gracias a los protocolos de red que hacen que la información viaje por todos los lugares teniendo un alcance mundial y la sincronización con la cadena de bloques sea casi de inmediato. Con ayuda de los protocolos la comunicación entre hardware y software es mucho más sencilla ya que se encargan de hacer una traducción, independientemente del dispositivo o el sistema operativo que se maneje, es decir, no importa si tienes un monedero de software o hardware, con los protocolos antes vistos, podemos hacer que ambos tengan la misma transferencia sin importar los recursos computacionales con

los que cuenten. Lo cual es de gran ayuda para que los equipos se comuniquen unos entre otros para hacer crecer la red que mantiene viva la moneda.

Otra ventaja que podemos resaltar, es la facilidad con la que un bitcoin puede ser dividido, podemos dividir un bitcoin hasta ocho decimales, es decir, podemos obtener el valor de 0.00000001 lo cual representa la unidad mínima de un bitcoin, también se le da el nombre de sathosi a dicha unidad en honor a su creador, Satoshi Nakamoto.

Además de todo lo anterior, un punto a favor de Bitcoin sobre otras monedas, físicas y digitales, es que los gastos por transacción son casi nulos a comparación de los cobros que efectúan dentro de otras plataformas que a la larga se convierte en mucho dinero gastado solo en la transacción. Por otro lado, tenemos que nuestro dinero en bitcoins estará disponible a cualquier hora y momento en el que se necesite hacer un movimiento.

Y lo más importante, toda la información es pública para todo mundo sobre las transferencias realizadas, más no los usuarios que la generaran, creando un ambiente neutral, seguro y transparente para todos.

3.11 Debilidades

El talón de Aquiles de los bitcoins sin duda alguna son los monederos donde los clientes guardan su información. El problema en sí no es el protocolo ni la moneda, más bien es el lugar donde se guardan los bitcoins. Como hemos visto, los más populares son los monederos online, y por ende son los más buscados por personas con malas intenciones. En los robos que se han generado a lo largo de la historia, podemos observar que la seguridad implementada en los monederos, mas no en el protocolo Bitcoin, es la culpable de que las personas no crean en el potencial que tenemos frente a nosotros. La única solución que es más factible es que esos sitios,

aplicaciones o dispositivos que ayudan a los usuarios a tener su dinero virtual a salvo es aumentando la seguridad de sus aplicaciones.

A pesar de que es una discusión muy grande la privacidad de las monedas, el hecho de que sean anónimas entra mucho en controversia, ya que muchas personas le dan un mal uso a la moneda y por tanto un mal aspecto hacia la sociedad. En los mercados negros es muy común encontrar grandes ejemplos del mal uso de los bitcoins. La gran mayoría de los mercados negros en internet, se manejan a través de la DeepWeb, que es la parte de internet por el cual se navega anónimamente por medio del navegador de nombre Tor, y al ser una navegación anónima se presta para realizar un comercio de armas o drogas, en el mejor de los casos, en inclusive trata de personas o contratar servicios de hackers o personas malintencionadas. Y todo se paga mediante bitcoins y así aumentan el anonimato y nadie sabe quién compra o vende.

Un curioso caso es el de los Ransomware, un virus que en los últimos años se ha vuelto cada vez más popular conforme pasa el tiempo, el objetivo de dicho virus es tener acceso a algún equipo y cifrar archivos del usuario afectado, y una vez que lo logra, manda un mensaje en la pantalla para pedir una recompensa por la llave privada con la cual se han cifrado los archivos.

Imagen 58 Ejemplo Ransomware

En la imagen 58, tenemos un ejemplo del mensaje generado por el Ransomware indicando que documentos fueron cifrados, el algoritmo ocupado para realizar el cifrado y enseguida te dan una descripción de lo que lo sucedido; aparentemente guardan la llave privada con la que es posible descifrar los archivos, piden una recompensa en bitcoins y si en el lapso de tiempo indicado no se realiza el pago, dicha llave será destruida.

Es entonces en donde cae el mal aspecto de la moneda, ya que ese dinero jamás se sabrá a qué lugar o para quién será destinado convirtiéndose en un uso ilegal.

Otro curioso caso de mal uso de bitcoins se registró en la famosa página de Yahoo! en la comunidad europea a finales del 2013, cuando el sistema de publicidad del sitio se vio comprometido; cuando alguien ingresaba y hacia clic en algún anuncio de publicidad, con una vulnerabilidad en el software Java, se instalaba software malicioso que abría una puerta trasera con muchos malwares y entre todo esto tenía un programa para minar bitcoins sin que el dueño del equipo se diera cuenta de su presencia, el malware tuvo la oportunidad de minar aproximadamente 700,000 euros en bitcoins, los cuales fueron legítimos para la red P2P y todos los mineros, pero extraídos a cosas de personas que no sabían de su existencia.

Algo que puede llegar a parecer una debilidad y un gran temor, es el hecho de tener el computo cuantico cada día más cercano. En palabras más claras, se trata de una computadora 3,600 veces más rápida que una computadora convencional, por lo que, los sistemas criptográficos resultan afectadas con respecto a la seguridad que ofrecen hoy en día.

Simplemente al tener un dispositivo muy veloz, el algoritmo ECDSA estaría roto. Los dispositivos cuánticos serían la opción sencilla para obtener una clave privada a partir de una clave pública. El algoritmo que se utiliza para realizar el resumen, SHA-256, no sería lo suficientemente seguro, así que comparándolo con lo que tenemos, sería el equivalente a descifrar un SHA-128 en un dispositivo actual.

Pero además de todo, ocurriría una centralización en la red, pues al ingresar a la red un equipo capaz de superar a todos los nodos conectados, pues en principio al comenzar a minar, la dificultad aumentaría draásticmente dejando a los demás mineros sin la oportunidad de competir. Si dicho equipo cuantico llega a superar el 51% de la potencia de la red, puede controlar todo. Podrá realizar transacciones a placer, bloquear transacciones, evitar que los mineros trabajen, revertir transacciones al con cumplir las confirmaciones, generar bloques de la nada e incluso enviar monedas que no le pertenecen.

Entonces, en sí todos los sistemas basados en criptografía, incluyendo los sistemas de banca en línea, son vulnerables a la computación cuántica, sin embargo, es un tema que probablemente tardará mucho en ser más estable para dichos fines, además de ser muy costoso por lo que las ganancias hasta ahora no serían tan buenas. Por otro lado, cuando llegue el momento de tener que el computo cuantico llegue a ser una amenaza, el protocolo podrá ser actualizado para lograr funcionar con criptografía post-cuantica, que es otro sistema que utiliza la física cuántica para poder cifrar y descifrar mensajes.

Y así podemos poner muchos del mal uso de la moneda, sin embargo debemos dejar que sea la ética de las personas la encargada de hacer cambiar a todos para obtener el beneficio y la libertad de usar bitcoins como una moneda no solo nacional, sino internacional haciendo los beneficios muchos mayores.

Siempre será más fácil hablar mal de algo que es muy nuevo para la sociedad, pero veremos que si adaptamos la moneda a nuestra vida cotidiana se volverá mucho más segura y todos los puntos negativos, si bien no van a desaparecer, pero se convertían en muy mínimos.

4. Conclusión

Como hemos visto a lo largo del escrito, las monedas virtuales son una tecnología que poco a poco será usada en un futuro no muy lejano y qué mejor que ponernos al día para poder ver el funcionamiento e ir adaptando nuestros recursos y necesidades sin tener que estar desprevenidos.

Para ello, es necesario realizar un análisis de la tecnología empleada. Partiendo desde lo más básico, tenemos que el medio en el que se hace posible realizar todos los intercambios es mediante una red denominada Peer to Peer, donde cada uno de los integrantes aporta cierta cantidad de recursos computacionales para mejorar el rendimiento. Sin embargo, podemos ver que el funcionamiento interno de la red contiene protocolos que mejoran el rendimiento y la seguridad para poder realizar operaciones desde cualquier lugar del planeta con acceso a internet.

Ahora bien, para poder confiar más en el uso de las monedas virtuales, se emplean métodos criptográficos, que a la fecha son muy fuertes; tal vez habrá quienes piensen que un solo método sería necesario para poder ocultar nuestra información a personas ajenas, pero como hemos visto, nuestras monedas virtuales ocupan por lo menos 3 métodos, para asegurar que solamente las personas que hagan alguna transacción sean las que puedan ver los datos enviados y recibidos.

Entonces, si juntamos un canal seguro por donde existe una comunicación e información que muy difícilmente alguien pueda llegar a leer, tenemos lo que se conoce como bitcoins, Litecoins, Dogecoin, o en simples palabras, coins.

Hoy en día, las monedas más utilizadas son los bitcoins, existen en total 16305237.5 bitcoinsen circulación de los 21 millones que representan el limite[3]. La gran ventaja que tiene dicha moneda es que puede ser fraccionada hasta en ocho decimales, es decir, puedes tener en tu posesión una cantidad de 0.00000001 bitcoin. Esto es de gran ayuda, ya que aproximadamente en el año 2040 se habran generado casi todos los bitcoins totales, entonces si las recompensas para los mineros por su aporte a la red disminuyen a la mitad cada 4 años, a la larga solamente ingresaran pequeñas fracciones de bitcoin de modo que la representación seria asintotamente cercana a 21 millones.

Al conocer todo lo anterior, un total de 21 millones es suficiente para no sufrir de una escacez de bitcoins, y al no poder ser superado ese limite no puede ser alterado el ritmo en que incrementan, y por lo tanto hablamos de, como minimo, cuatrillones de unidades por un solo bitcoin por lo que con uno, de ser necesario, puede abastecer las necesidades de miles de usuarios.

[3]BLOCKCHAIN. Consultado el 02 de mayo de 2017 de https://blockchain.info/es/charts/total-bitcoins?timespan=all

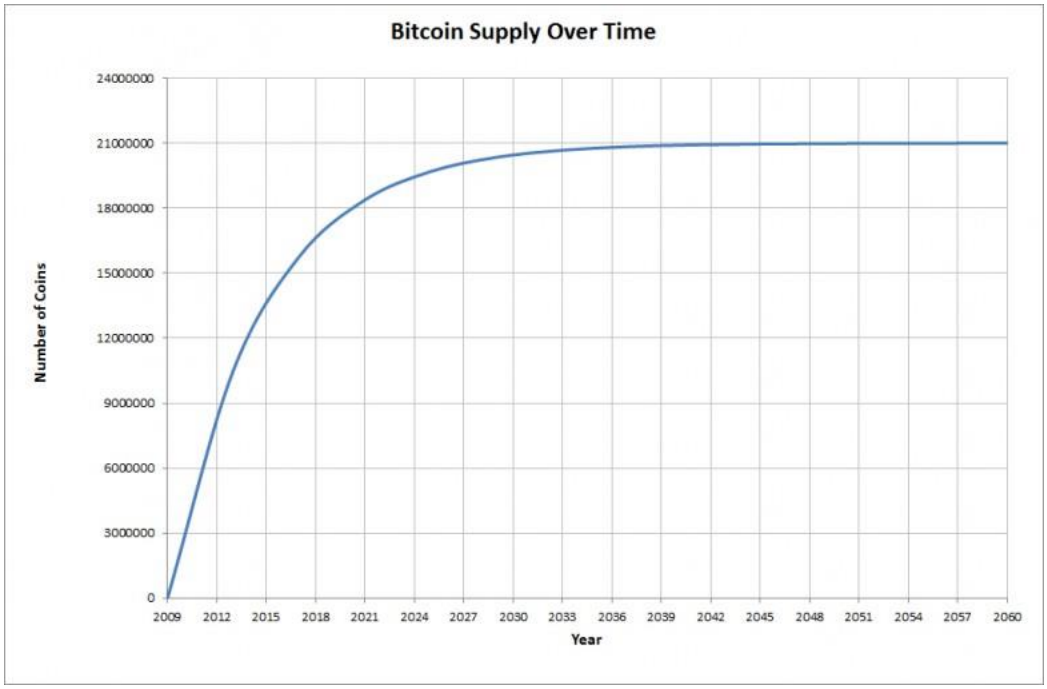

Imagen 59 Comportamiento del total de bitcoins

Además, podemos ver que no es tan diferente al dinero que ocupamos actualmente, simplemente las transacciones son más discretas al tratarse de un proceso anónimo, pero con la confianza de que los algoritmos hasta el momento son muy seguros para proteger nuestros intereses. Por otra parte, como vimos a lo largo del escrito, es muy difícil lograr generar monedas ficticias o utilizar un código para utilizarlo una segunda vez, sin embargo, para estar más protegidos es muy bueno contar con un monedero muy de confianza para poder almacenar nuestros códigos sin preocupación.

Una vez conociendo una de las nuevas tecnologías, podemos ver que tiene muchas ventajas sobre el hecho de ocupar otro tipo de forma de copra-venta (monedas, billetes o tarjetas), además es mucho más práctico contar con diferentes medios para acceder a tu cuenta y realizar un pago o algún cobro sin la necesidad de exponerte, ya que todos los movimientos son realizados de forma anónima para asegurar la integridad del usuario.

Por ultimo debemos recordar, el valor que adquiere un bitcoin se ve reflejado por el número de usuarios que están conectados a la red. Al no tener una entidad única que los controle, todo el conjunto de usuarios llevan a cabo esa tarea; al tratarse de

algoritmos de cifrado bastante fuertes, las transacciones son transparentes para el resto de los usuarios, por lo cual podemos compararlo al hecho de tener dinero físico, nadie sabra por quien ha pasado anteriormente. Además, la facilidad de portar bitcoins en esta época, facilita todos los movimientos de cada usuario.

Es por ello, que puedo afirmar que la desconfianza que ejercen las monedas hasta ahora debe ir desapareciendo, esto debido a que los métodos para protegerlas son, hasta ahora, de lo más nuevo y seguro, además de contar con una red P2P donde todos los usuarios prestan parte de su poder de procesamiento para poder tener una seguridad más firme y al mismo tiempo evitando que los usuarios circulen por las calles con el temor de que alguien conoce los movimientos que realizan con su dinero. Así bien, una vez que la sociedad vaya obteniendo más confianza, será una de las divisas más cotizadas en el mercado sin tener una entidad que las controle.

5. Glosario

A

Ataque de 51%: Un ataque informático que pudiese ser hecho por una entidad o grupo de minería que posea la mayoría del procesamiento de transacciones de la red blockchain (51% o más) para prevenir que nuevas transacciones se confirmen.

B

Bitcoin - con B mayúscula, se utiliza para referirse al protocolo.Ejemplo, “he aprendiendo sobre el protocolo Bitcoin”.

bitcoin – con b minúscula, se utiliza para referirse a la moneda. Ejemplo, "Hoy he comprado diez bitcoins".

BitStamp: Casa de cambio con sede en Eslovenia que permite a los usuarios cambiar bitcoins por dinero fiduciario y otras criptomonedas.

Blockchain: Se refiere a la secuencia de bloques que almacenan información y que han sido verificados por los usuarios de la red desde sus inicios. El término blockchain (cuya traducción literal es “cadena de bloques”) proviene del hecho de que cada bloque contiene un apuntador hash hacia su bloque predecesor, creando una red interconectada. Es importante destacar que existe una empresa de nombre Blockchain y cuyo principal producto es un explorador de bloques que posee el mismo nombre.

Bloque Génesis: Nombre dado al primer bloque creado y verificado de la blockchain de una criptomoneda.

Bloque huérfano: Bloque de información que no es parte de la red distribuida. Se crea cuando dos o más mineros producen bloques casi al mismo momento pero uno de ellos es propagado por la red con mayor rapidez y aceptado por los nodos, dejando fuera de la cadena a los demás.

Bloque recompensa: Beneficio que obtiene un minero por resolver con éxito un acertijo hash y crear un bloque. La red Bitcoin actualmente otorga 12,5 bitcoins por cada bloque minado. Esta recompensa se reduce a la mitad cuando se ha extraído un cierto número de bloques. En el caso de Bitcoin, el cambio se produce cada 210.000 bloques.

BTC: Abreviatura para referirse a las unidades de bitcoins.

C

Casa de cambio o exchange: Operadora de criptomonedas dedicada al cambio por otro tipo de monedas.

Checksum: Método de detección de errores que actua sobre conjuntos de información. Para elementos de información de n bits se obtiene la suma de todos los elementos, incluyendose esta junto con la información, siendo el resultado de módulo m de la forma m=2^n. Cuando sea preciso realizar una comprobación se vuelve a realizar el cálculo, comparándolo con el resultado previamente obtenido.

Código QR: Es un gráfico bidimensional conteniendo un patrón de puntos monocromáticos distribuidos de cierta forma tal que contengan información para cualquiera que sepa leerlos. Son muy utilizados en el mundo del bitcoin por ejemplo para transmitir de forma sencilla nuestra dirección a una persona para que pueda pagarnos.

Colisión: Situación en la que una función hash produce el mismo resultado para valores distintos.

Confirmación: Verificación por parte de los nodos de la red de que un bloque contiene únicamente transacciones válidas realizadas con cripmonedas que nunca antes habían sido usadas. El tiempo de confirmación en la red Bitcoin varia de 10 a 60 minutos, generalmente.

Criptografía: Conjunto de técnicas y métodos matemáticos que protegen la información de los datos registrados en la blockchain, dotandolos de seguridad y garantizando su inmutabilidad.

Criptomoneda: moneda basada exclusivamente en la criptografía. A diferencia de las monedas emitidas por países, se genera con la resolución de problemas matemáticos basados en criptografía. Su valor, no obstante, está sujeto a variación de precios y dependiendo de la oferta y demanda en los mercados.

D

Dificultad: número que determina la complejidad del acertijo hash a resolver en cada bloque. Varía en función de la potencia de cálculo de los mineros en la red y se ajusta automáticamente cada cierta cantidad de bloques minados. En el caso de Bitcoin, se ajusta cada 2016 bloques.

Dirección: Secuencia de caracteres alfanuméricos que señala la ubicación de una cartera a la que pueden enviarse la cantidad deseada de criptomonedas.

Doble gasto: Acto de realizar dos pagos con una misma criptomoneda. Supone una operación fraudulenta y, aunque no resulta fácil de hacer en la red Bitcoin, se evita esperando al menos una confirmación de la red antes de dar por finalizada la transacción.

E

ECDSA: Elliptic Curve Digital Signature Algorithm es el algoritmo usado para firmar las transacciones en el protocolo bitcoin.

F

Firma digital: Proceso matemático que permite verificar la autenticidad del remitente de bitcoins. Hasheando en conjunto la clave pública y la clave privada del remitente, el receptor puede comprobar que el pago fue realizado por ese remitente y que, además, no fue alterado por nadie más.

Función Hash: Función matemática que transforma valores de un conjunto grande en otro conjunto de valores más pequeño. Es una función resistente a colisiones. Las funciones resumen dependen de una clave criptográfica. La función hash puede tomar valores de tamaño variable, pero siempre produce una salida de longituda fija.

G

Gigahashes / sec: El número de intentos de hash posible en un segundo dado, medido en miles de millones de hashes (miles de Megahashes).

GPU: Unidad de procesamiento gráfico. Chip de silicio diseñado específicamente para realizar cálculos matemáticos complejos necesarios para interpretar los gráficos visuales de juegos de ordenador. Son muy adecuadas para hacer cálculos criptográficos necesarios en la minería criptomoneda.

H

Hash: Función algorítmica que emite una dirección alfanumérica que resume y protege la información insertada a través de una entrada. Sirven también para garantizar la inmutabilidad de una unidad de información, ocultar una contraseña o servir como firma digital.

K

Kilohashes / sec: Es el número de intentos de hash posible en un segundo dado, medido en miles de hashes.

M

Megahashes / sec: El número de intentos de hash posible en un segundo dado, medido en millones de hashes (miles de Kilohashes).

Minería: Es el acto de resolver un bloque, validando todas las transacciones que contiene.

N

Nodo: Es un ordenador conectado a la red Bitcoin que transmite transacciones a otros.

P

P2P: Hace referencia a una red peer-to-peer, es decir, una red descentralizada donde todas las partes interactúan entre sí.

Private Key (clave privada): Es un texto alfanumérico asociado matemáticamente a una dirección y que debe ser conocido sólo por el dueño de esa dirección. Para poder

acceder a los bitcoins depositados en la dirección y disponer de ellos es necesario conocer la clave privada.

Prueba de trabajo: Es el sistema mediante el cual se minan bitcoins. La prueba consiste en la resolución de problemas matemáticos (un hash) que tiene una variable que lo dificulta. Resolver la prueba con éxito suele requerir tiempo y es por esto que en última instancia, este sistema condiciona la capacidad de minado al poder computacional del usuario.

Public Key (clave pública): Es un texto alfanumérico del cual se obtiene una dirección conocida por todos los usuarios. Al ser conocida, cualquiera puede enviar bitcoins a la dirección asociada, pero sólo quien tenga la clave privada podrá acceder a ellos.

S

Satoshi: Es la subdivisión más pequeña que puede obtener de un bitcoin, a saber: 0.00000001 BTC.

Satoshi Nakamoto: Es el pseudónimo utilizado por la persona o grupo de personas que desarrollaron el protocolo de Bitcoin. Está retirado desde 2010.

Silk Road: Fue un mercado en línea (ubicado en la Deep web) utilizado para la compra de productos ilícitos y en la cual, la principal forma de pago fue el bitcoin. Fue cerrada a finales del año 2013 luego de que el FBI arrestara a su propitario, Ross Ulbricht.

SHA-256: Es la función criptográfica utilizada como base para la prueba de trabajo que permite minar bitcoins.

W

Wallet (monedero): Es donde se almacenan los bitcoins, por lo que a través de él se reciben y envían de unos usuarios a otros sin necesidad de intermediarios. En realidad, un monedero es un archivo que contienen claves criptográficas. Existen cuatro tipos: monederos para ordenadores, monederos para móviles o tablets, monederos online y dispositivos físicos o monederos de papel.

6. Referencias bibliográficas

- Daniel, D. A. (2013). Instituto nacional de investigación y capacitación de telecomunicaciones. Consultado el 16 Marzo de 2016, de http://slideplayer.es/slide/17470/
- MikroTik (n.d.) Conocimientos Básicos. Consultado el 16 de marzo de 2016, de http://www.mikrotikxperts.com/images/informacion/
- Transmission Control Protocol (1981). Consultado el 30 de marzo de 2016, de https://www.ietf.org/rfc/rfc793.txt
- User Datagram Protocol (1980). Consultado el 3 de abril de 2016, de https://www.ietf.org/rfc/rfc768.txt
- Internet Protocol (1981). Consultado el 24 de abril de 2016, de https://www.ietf.org/rfc/rfc791.txt
- Eduardo, A. (2014). New Crypto-Ransomware Emerge in the Wild. Consultado el 24 de julio de 2016, de http://blog.trendmicro.com/trendlabs-security-intelligence/new-crypto-ransomware-emerge-in-the-wild/
- Nermin, H. (2014). Yahoo Infects 2 Million European PCs with Bitcoin Malware. Consultado el 13 de diciembre de 2015, de http://www.coindesk.com/yahoo-infects-2-million-european-pcs-bitcoin-malware/
- Elige tu monedero (n.d.). Consultado el 20 de diciembre de 2015, de https://bitcoin.org/es/elige-tu-monedero
- Todo sobre P2P (n.d.). Consultado el 9 de febrero de 2016, de http://www.elotrolado.net/wiki/Todo_sobre_P2P
- Microsoft ya acepta el uso del Bitcoin (2014). Consultado el 25 de febrero de 2016, de http://www.elpais.com.uy/informacion/microsoft-ya-acepta-bitcoin-moneda.html
- Nick, S. (2013). A (Relatively Easy To Understand) Primer on Elliptic Curve Cryptography. Consultado el 14 de julio de 2016, de https://blog.cloudflare.com/a-relatively-easy-to-understand-primer-on-elliptic-curve-cryptography/
- The Use of HMAC-RIPEMD-160-96 within ESP and AH (2000). Consultado el 18 de octubre de 2016, de https://www.ietf.org/rfc/rfc2857.txt
- The MD5 Message-Digest Algorithm (1992). Consultado el 14 de octubre de 2016, de http://www.ietf.org/rfc/rfc1321.tx
- Satoshi, N. (n.d.). Bitcoin: un sistema de dinero en efectivo electrónico peertopeer. Consultado el 27 de febrero de 2016, de https://bitcoin.org/files/bitcoin-paper/bitcoin_es.pdf
- Ralph C. Merkle. (1980). Protocols for public key cryptosystems. Sunnyvale,Ca.: 1980 Symposium on Security and Privacy, IEEE Computer Society.
- Norbert, P., Chrestian, R., Vincent, R. (n.d.). Impact of Rotations in SHA-1 and Related Hash Functions. Consultado el 16 de octubre de 2016, de http://csrc.nist.gov/groups/ST/hash/documents/Rechberger_ImpactOfRotations.pdf
- Adam, B. (2002). Hashcash - A Denial of Service Counter-Measure. Consultado el 16 de octubre de 2016, de http://www.hashcash.org/papers/hashcash.pdf
- Bill, B (2005). NIST Hash Function Standards Status and Plans. Consultado el 18 de octubre de 2016, de http://csrc.nist.gov/groups/SMA/ispab/documents/minutes/2005-12/B_Burr-Dec2005-ISPAB.pdf
- Vicent, G. B (2004). El problema del cumpleaños. Consultado el 18 de agosto de 2016, de http://personales.upv.es/~vigibos/ProblemaCumple.pdf
- Sergio, S (2009). Se reduce la complejidad para provocar colisiones en SHA1. Consultado el 26 de julio de 2016, de http://unaaldia.hispasec.com/2009/06/se-reduce-la-complejidad-para-provocar.html
- US Secure Hash Algorithms (SHA and HMAC-SHA) (2006). Consultado el 19 de julio de 2016, de https://tools.ietf.org/html/rfc4634
- US Secure Hash Algorithm 1 (SHA1) (2001). Consultado el 20 de julio de 2016, de https://tools.ietf.org/html/rfc3174

- Daniel, B., Richard, S., Robert, B. (2005). SSH, The Secure Shell: The Definitive Guide. Estados Unidos: O'Reilly Media.
- Florian, M., Thomas, P., Martin, S., Lei, W., Shuang, W. (n.d.). Improved Cryptanalysis of Reduced RIPEMD-160. Consultado el 9 de noviembre de 2016, de https://eprint.iacr.org/2013/600.pdf
- Bart, P., Hans, D., Antoon, B. (1997). The Cryptographic Hash Function RIPEMD-160. Consultado el 9 de noviembre de 2016, de https://securewww.esat.kuleuven.be/cosic/publications/article-317.pdf
- Bart, P., Hans, D., Antoon, B. (1996). RIPEMD-160: A Strengthened Version of RIPEMD. Consultado el 9 de noviembre de 2016, de http://homes.esat.kuleuven.be/~bosselae/ripemd160/pdf/AB-9601/AB-9601.pdf
- Hugo Daniel, S., Juan Pedro, H. (2004). Impacto de recientes ataques de colisiones contra funciones de hashing de uso corriente. Consultado el 16 de agosto de 2016, de https://www.certisur.com/sites/default/files/docs/ataques_funciones_hashing.pdf
- Codificación Base58Check (2012). Consultado el 20 de octubre de 2016, de https://es.bitcoin.it/wiki/Codificaci%C3%B3n_Base58Check
- Lista de prefijos de direcciones (2012). Consultado el 20 de octubre de 2016, de https://es.bitcoin.it/wiki/Lista_de_prefijos_de_direcciones
- Blockchain (n.d.) Consultada el 24 de noviembre de 2015, de https://blockchain.info/es/
- M. Magdalena, P. C., Andreu Pere, I. D., Macià M. P. (2014). Lección 3. Introducción a Bitcoin. Consultado el 24 de enero de 2016, de http://www.criptored.upm.es/crypt4you/temas/sistemaspago/leccion3/leccion03.html#apartado32
- ¿Cómo sabemos que nunca habrá más de 21 millones de bitcoins?. Consultado el 2 de mayo de 2017, de http://elbitcoin.org/como-sabemos-nunca-habra-mas-21-millones-bitcoins
- CriptoNoticias. Glosario. Consultado el 13 de mayo de 2017, de https://criptonoticias.com/informacion/glosario/#axzz4hYHMyAjq
- Sergio de Luz. Criptografía : Algoritmos de cifrado de clave asimétrica, 13 de noviembre de 2010. Consultado el 16 de mayo de 2017 de https://www.redeszone.net/2010/11/16/criptografia-algoritmos-de-cifrado-de-clave-asimetrica/
- Glosario sobre minería para principiantes, 20 abril de 2014. Consuldato el 16 de mayo de 2017 de https://bitcointalk.org/index.php?topic=577566.0
- Glosario Terminología informática (1997). Consultado el 17 de mayo de 2017 de http://www.tugurium.com/gti/index.php