Anàlisi de la cripto-moneda Bitcoin

MISTIC: Màster Interuniversitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions

Memòria del màster per als estudis de Màster Interuniversitari en Seguretat de les Tecnologies de la Informació i de les Comunicacions presentat per Isidro Pastor Jordà i dirigit per Jordi Herrera Joancomartí

UOC, València, 2014

Contingut

Índex d'Imatges

Índex de Taules

Introducció

Objectius

Al present treball final de màster ens plantegem estudiar la cripto-moneda Bitcoin a nivell teòric per conèixer com funciona i els seus mecanismes de seguretat i de confirmació de transaccions. Farem una part pràctica on hem d'aconseguir dades sobre transaccions reals i analitzar-les.

Esquema

Aquesta memòria s'estructura en tres parts diferenciades. En una primera part estudiem la part teòrica dels Bitcoins, el seu funcionament a nivell de xarxa distribuïda i la forma de gestionar les transaccions. En una segona part fem una anàlisi de transaccions reals per tal de veure si la primera part teòrica es compleix i sobre tot, tractar de veure si les comissions que es paguen en cada transacció tenen una relació directa en el temps que triguen en fer-se oficials. Finalment analitzarem els resultats amb una conclusió a la vista tant de la part teòrica com de la part d'anàlisi de dades.

Primera part – Anàlisi teòric

La moneda

L'intercanvi d'objectes sempre ha estat present a la història de la humanitat. En un primer moment era el bescanvi el que s'utilitzava però aquest tipus d'intercanvi era difícil en tant que el valors dels objectes a canviar potser no eren equiparables i establir un valor per al canvi resultava molt difícil. [1] Poc a poc es van començar a utilitzar metalls preciosos en diverses formes, encara que per la seua facilitat de transport la forma que va triomfar va ser la circular en diferents mides en funció de la quantitat de metall que es necessitava o es volia donar com a valor. Ja al 2500a.C. hi ha possibles restes de segells del que podrien ser monedes, però no va ser fins als segles VIII i IX quant es troben les primeres monedes impreses.

Les monedes al principi tenien una composició prou estable de metalls i per tant el seu valor era paregut en qualsevol part que es feien. Però era habitual que aquestes monedes es llimaren per treure part del seu contingut en or o plata. Per aquest motiu qui emetia les monedes terminava fent la moneda amb menys metall preciós mentre que el seu valor "facial" continuava sent el mateix, garantint-se més beneficis al posar en circulació la moneda. Existien experts que encunyaven la moneda, és a dir, posaven un segell que garantia la quantitat de metall preciós que hi contenia. Aquesta encunyació es fa a un lloc anomenat *seca* i tradicionalment es feia amb un martell que imprimia a la moneda un segell [2]. A l'edat mitjana (entre el segles V i el XV) la falsificació de moneda es va generalitzar, generalment fent passar per monedes amb cert valor en metalls preciosos altres que no hi tenien aquests metalls o els contenien en molta menor proporció. Al 1551 es va realitzar la primera encunyació mecànica de moneda en sèrie i amb valor uniforme al Tirol amb un molí hidràulic. Amb la revolució industrial (segle XVII) aquestes màquines per encunyar moneda es milloren i es generalitzen però no va ser fins al 1830 quant es va començar a encunyar les dues cares de la moneda i els cantons al mateix temps. Aquest fet d'encunyar els cantons va ser molt important per evitar la pràctica molt estesa de retallar o llimar les monedes perquè quant es feia aquesta encunyació desapareixia i era evident que el valor de la moneda havia estat modificat.

Paper moneda

El paper moneda va ser introduït per primera vegada en China al segle IX (entre els anys 618 i 907) per evitar portar grans quantitats de monedes metàl·liques. [3] Els primers bitllets a Europa apareixen al 1661 al banc d'Estocolm com a garantia de dipòsit d'or o metalls preciosos al banc i no va ser fins al 1780 amb el regnat de Carles III quant va arribar a Espanya. Aquest paper fins al 1970 estava recolzat pel que es diu "patró or" que garanteix que cada emissió de bitllets és recolzat per una determinada quantitat d'or. [4] L'emissor del bitllet, per tant , era propietari d'una quantitat d'or i garantia que podia canviar el valor del bitllet en or. Però a partir del 1971 es va deixar d'utilitzar or com a recolzament de la moneda, encara que des de 1944 les úniques divises que mantenien el patró or eren el dòlar i la lliura esterlina, la resta de monedes utilitzaven aquestes dos com a recolzament. Des

d'aquest moment **el diner és fiduciari** [5], es a dir, és diner que es basa en la fe o la confiança que es té en l'emissor i no en metalls preciosos. Els diners actualment són per tant un acte de confiança en el govern que encunya aquesta moneda i la fe en que en un futur també s'acceptarà. Sense aquesta confiança el diner de tipus fiduciari no té gens de valor. Els diners que avui utilitzem tenen valor en tant que podem fer intercanvis amb ells i són acceptats. Amb aquesta eliminació del patró or també va néixer el mercat de divises, un mercat d'intercanvi de moneda que aprofita les diferències de valor entre les diferents monedes, un mercat altament voluble i que ben jugat pot ser molt lucratiu.

Diner electrònic

Amb l'evolució de les tecnologies de la informació i sobre tot amb l'aparició d'Internet i la seua utilització massiva es comença a intercanviar diners de forma electrònica: Aquests intercanvis de diners normalment són només una representació de dèbits i de crèdits dins un marc de sistema. Aquestes transaccions estan recolzades per la moneda de cada país i normalment intervenen els bancs en aquestes transaccions. L'ús de targetes de crèdit pot ser considerat una mena de diner electrònic que ens permet fer pagaments al comerç sense portar ni monedes ni bitllets al damunt. Poc a poc els sistemes d'aquest tipus estan evolucionant a targetes xip que formen un moneder electrònic que permet recarregar el xip amb una quantitat d'efectiu que després es pot utilitzar per a pagar al petit comerç. [6]

Totes aquestes formes d'intercanvi digital de diners estan relacionades amb bancs on podem guardar els nostres diners i que ens faciliten la utilització dels diners sense haver de portar al damunt ni tan sols bitllets o monedes.

Monedes digitals

Al llarg del temps han anat sortint alguns jocs d'ordinador (World of Warcraft), mons virtuals (Second Life), tendes de videojocs (Nintendo Points) .... amb major o menor repercussió o èxit que han utilitzat un tipus de moneda pròpia per a fer intercanvis dins d'aquests mons. La moneda simplement es canviava per l'administrador del joc per les monedes virtuals que es feien servir en aquests llocs. Potser aquests són els primers contactes amb monedes digitals que tots nosaltres hem tingut. Però aquestes monedes no deixen de ser virtuals i que només es poden utilitzar dins un ecosistema molt concret, el videojoc.

L'evolució de la criptografia fa que els mecanismes de seguretat es millorin i que apareguin el que coneixem com a cripto-monedes, monedes que es basen en algorismes criptogràfics per garantir la seguretat del sistema i per tant de la moneda. D'aquest tipus de moneda ens endinsarem en una de les que més repercussió i expansió té, el Bitcoin, encara que existeixen altres monedes digitals en circulació tot i que amb menor implantació. Moltes d'aquestes monedes virtuals comparteixen un detall important, a part de l'ús de la criptografia, i és que no són controlades per cap organització central, com un banc o un govern.

bitcoin és possiblement la moneda digital més estesa de totes les que existeixen o han existit. Es basa en un model no centralitzat que produirà un total de 21 milions de monedes moment en el qual la economia no produirà més diners en circulació. Veurem en més detall les característiques d'aquesta moneda més endavant.

ripple es situa la segona moneda digital per capitalització amb 636 milions de dòlars (30 Maig 2014 [7]). També utilitza una tecnologia distribuïda i de codi obert. Implementa per a cada transacció una petita quantitat en forma de comissió de seguretat . En aquest sentit indiquen que es per evitar atacs de denegació de servei perquè si a cada transacció es lleva una petita part en forma de comissió de seguretat, es desincentiven els atacs, sobre tot els de denegació de servei. Ripple està pensat com a protocol d'Internet per fer transaccions financeres i de fet aquest seria el primer sistema de intercanvi de moneda descentralitzat. [8]

litecoin és una moneda digital basada en el protocol de Bitcoin encara que la mineria és més assequible en hardware que podria tindre qualsevol persona. Està preparada per generar fins a 84 milions de monedes digitals com a màxim. [9]

Peercoin és la cinquena moneda en capitalització (30 Maig 2014 [7]) i també comparteix bona part del codi font de Bitcoin i de la seua implementació tècnica. Aquesta moneda, a diferència de Bitcoin o Litecoin, no té una limitació tancada de la quantitat de monedes que poden formar part de l'economia. Ha millorat l'eficiència energètica del procés de mineria el que li dona una millor escalabilitat al llarg termini. [10]

Dogecoin va néixer com una broma entre uns amics de crear una criptomoneda divertida que poguera arribar a un sector demogràfic diferent al que arribava Bitcoin. Va néixer el 8 de desembre de 2013 i s'espera una producció massiva de monedes, del ordre de 100 mil milions al final de 2014 i uns 5.4 mil milions cada any posterior. Al març de 2014 ja s'havien minat uns 65mil milions de Dogecoin al sistema. És principalment utilitzada a xarxes socials per donar com una propina a gent que fa aportacions gracioses o considerades de valor. [11]

Mastercoin es basa també en el protocol de Bitcoin i va ser llançada el 31 de juliol de 2013 i a maig de 2014 aquesta és la desena criptomoneda per capitalització del mercat amb 11 milions de dòlars. (Maig 2014 [7]) [12]

La llista de monedes digitals és molt llarga i canvia constantment. Una bona referència és la seua capitalització de mercat, és a dir, quants diners es mouen dins d'aquestes monedes. És un senyal de l'acceptació dels mercats i sobre tot de la gent en aquestes monedes, així com la confiança que es diposita en elles. Podem trobar una llista d'aquest tipus a la web coinmarketcap.com [7].

Bitcoins

El Bitcoin és una moneda digital basada en protocol *peer to peer* (p2p) nascuda en 2009. El seu creador és Satoshi Nakamoto (encara que aquest no és el seu nom de veritat) i ja al 2008 va escriure el *white-paper* amb els detalls de la moneda [13]. El fet que el funcionament d'aquesta moneda es basi en una xarxa p2p aporta descentralització a la moneda, al contrari al que estem acostumats tradicionalment en les monedes amb un banc centralitzat.

La seguretat del Bitcoin es centra a la criptografia i l'ús de diferents mecanismes, que explicarem més endavant, per a garantir la seua seguretat. Al mateix temps el Bitcoin és una moneda electrònica (criptomoneda, diners electrònics ...) que permet mantenir anònim l'autor d'una transacció de diners.

El funcionament descentralitzat com a xarxa p2p implica que cada transacció que es realitza s'envia a tota la xarxa, és a dir, tots els nodes de la xarxa escolten totes les transaccions. Una volta es rep una transacció cada node treballa per construir un *block*, que més endavant veurem com es crea, però que anticiparem que es tracta d'un conjunt de transaccions. Cada node ha de fer un treball per a assegurar la creació d'aquest *block*, aquest treball s'anomena prova de treball i una volta fet el *block* es transmet per la xarxa p2p a tots els nodes. La resta de nodes acceptaran aquest *block* només si les transaccions que hi conté no han estat gastades en *blocks* anteriors al nou creat. Aquest punt és important per evitar que una quantitat de bitcoins sigui utilitzada dues voltes per pagar (evitar el que s'anomena doble despesa). Cada *block* conté el hash del anterior *block* formant una cadena (cadena de blocks), si el nou *block* és acceptat, tots els nodes començaran a treballar en el següent *block* incorporant-hi el *hash* de l'últim *block* acceptat.

Aquesta rutina és la mateixa una volta i un altra conforme s'avança en el temps i es generin noves transaccions.

Com hem pogut veure, les transaccions s'agrupen en blocs que al mateix temps s'agrupen formant una cadena (*blockchain*). Anirem poc a poc endinsant-nos en més detall en aquestes parts i com funcionen, però començarem per la part més bàsica, l'adreça que hem de fer servir tant per a fer pagament com per a rebre'ls.

Adreces i wallets

Com ja hem comentat, Bitcoin es basa en criptografia per al seu funcionament. Principalment fa servir criptografia de parells de claus publica-privada. Aquest sistema criptogràfic es basa en que una persona genera un parell de claus. Una és totalment privada, el que vol dir que el seu propietari no ha de distribuir-la ni fer-la pública per tal de mantenir la privacitat de les transaccions. L'altra part és pública i està disponible per a qualsevol persona que la necessiti. Mitjançant algorismes de signatura electrònica es pot aplicar la clau privada per fer la signatura d'una operació (o del hash d'aquesta operació), de forma que amb la clau pública qualsevol persona podria verificar que aquesta operació ha estat signada

per qui l'ha signat, en tant que la clau pública només podria aplicar-se per verificar signatures originades amb la clau privada parella a questa pública.

Als bitcoins per a originar una transacció o rebre-la és necessari una adreça electrònica. Aquesta adreça és una sèrie de nombres i lletres (tant majúscules com minúscules) que es deriva de la clau pública d'un parell pública-privada que genera l'usuari.

Un exemple de adreça bitcoin és aquesta

14dQ1DZCYTpPZRxjMqjDySZHF1Fz8J8VnN

Com ja hem dit, aquesta adreça es deriva de una clau pública generada com el algoritme de signatura digital de corba el·líptica [14]. A partir d'aquesta clau pública es fan una sèrie de transformacions, principalment hashes sobre parts de la clau pública, però el primer caràcter identifica la xarxa de bitcoin a la que pertany l'adreça. Un 1 correspon a la xarxa principal de bitcoin, mentre que altres identificadors a la direcció corresponen amb xarxes paral·leles, com ara el test network on es poden provar transaccions amb bitcoins de prova que es poden obtindre fàcilment. La següent adreça pertany a la xarxa TestNet de Bitcoin i com es pot comprovar el primer caràcter de l'adreça és una ***m*** en comptes d'un 1:

mxm76DN4ELyz7s7iKz8oYfxBAxmm8nSYCM

[15]

Però per a guardar aquestes adreces i les claus associades necessitarem algun tipus de contenidor. A la vida real tenim carteres on guardem els diners en metàl·lic que tenim, a aquest entorn digital també tenim carteres, o en anglès *wallets*, on hi guardarem els diners digitals. Aquestes carteres gestionen les claus criptogràfiques tant privades com públiques i ens permet al mateix temps generar adreces noves generant noves parelles de claus. De fet, per garantir l'anonimat de les transaccions, el recomanable es utilitzar una adreça de bitcoin diferent per a cada transacció que fem, en tant que generar una nova adreça no resulta costós i ens assegura la privacitat d'aquesta transacció.

Hi ha gran quantitat de *wallets* però es destaquen dues formes de guardar els diners digitals. Per una banda estan els *wallets* que s'instal·len en l'ordinador o telèfon mòbil de cada persona i per altra *wallets* que estan "al núvol".

Hi ha molts programes *wallet* que podem descarregar d'Internet. Per a tots els sistemes operatius (Windows, linux, Mac, Android, Windows Phone) excepte iOS. Aquests programes emmagatzemen a nivell local les claus privades i publiques i les direccions de bitcoin, com també els mateixos diners digitals. Això vol dir que si perdem les dades relacionades amb aquestes claus no disposarem dels diners que hi tenim a les adreces que gestionem. És per això que es fa molt recomanable fer còpies de seguretat per tal de no perdre aquesta informació. El client original de bitcoin, Bitcoin-Qt és un *wallet* a més de ser un node més de la xarxa p2p de bitcoin retransmetent transaccions. Hi ha altres *wallets* per a ordinadors com ara Multibit,

Hive, Armory, Electrum .... pel que fa als clients mòbils hi ha aplicacions per a Android com Bitcoin Wallet, Mycellium, Gliph o blockchain, inclús pròpies per a la xarxa de TestNet com Bitcoin Wallet for Testnet. Moltes de les aplicacions són multi-sistema, és a dir, la mateixa aplicació ofereix diferents versions per als diferents sistemes operatius i sistemes mòbils.

Les *wallets* al núvol treballen com ho faria una cartera local però guardant les claus privades també a la web. Aquest mode de treball té l'avantatge de que el wallet és sempre disponible (no necessites estar al teu ordinador o portar el teu telèfon al damunt) però per altra banda també té altres punts febles, com que li deixes el control de les teues claus privades i dels teus diners, casi com si fos un banc però sense les garanties que aporta un banc, en tant que si hi ha un robatori o tenen qualsevol altre problema ningú no els obliga a tornar els diners.

Alguns dels serveis de *wallet online* són blockchain.info, coinbase.com ó strongcoin.com. Tots indiquen que les claus privades s'emmagatzemen xifrades i ells no tenen accés a utilitzar-les, només el seu propietari, però fins i tot la mateixa pàgina oficial de bitcoin adverteix que aquestes webs poden sofrir atacs i els bitcoins que hi tenim en aquestes webs poden perdre's.

Una alternativa a aquestes *wallets* són les carteres de paper. Seria com traslladar les dades de cada adreça de bitcoin amb la seua clau privada a paper. Normalment aquesta opció no és pràctica i a més també pot sofrir els problemes tradicionals del paper (amb el temps es fa malbé, pot cremar-se, es pot perdre ...) però per altra banda també aporta un nivell de seguretat addicional i és que no te'l poden furtar per Internet. Una web que genera aquest tipus de *wallet* és bitaddress.org on podem generar adreces bitcoin i després imprimir-les per guardar-les. Cada adreça generada porta l'adreça bitcoin i la clau privada en codis QR que es poden capturar fàcilment en un telèfon mòbil per importar-la a una *wallet* tradicional de bitcoin (per exemple amb blockchain per a Android).

Imatge 1 Bitllet amb codi QR amb els certificats per imprimir en paper

[16]

Transaccions

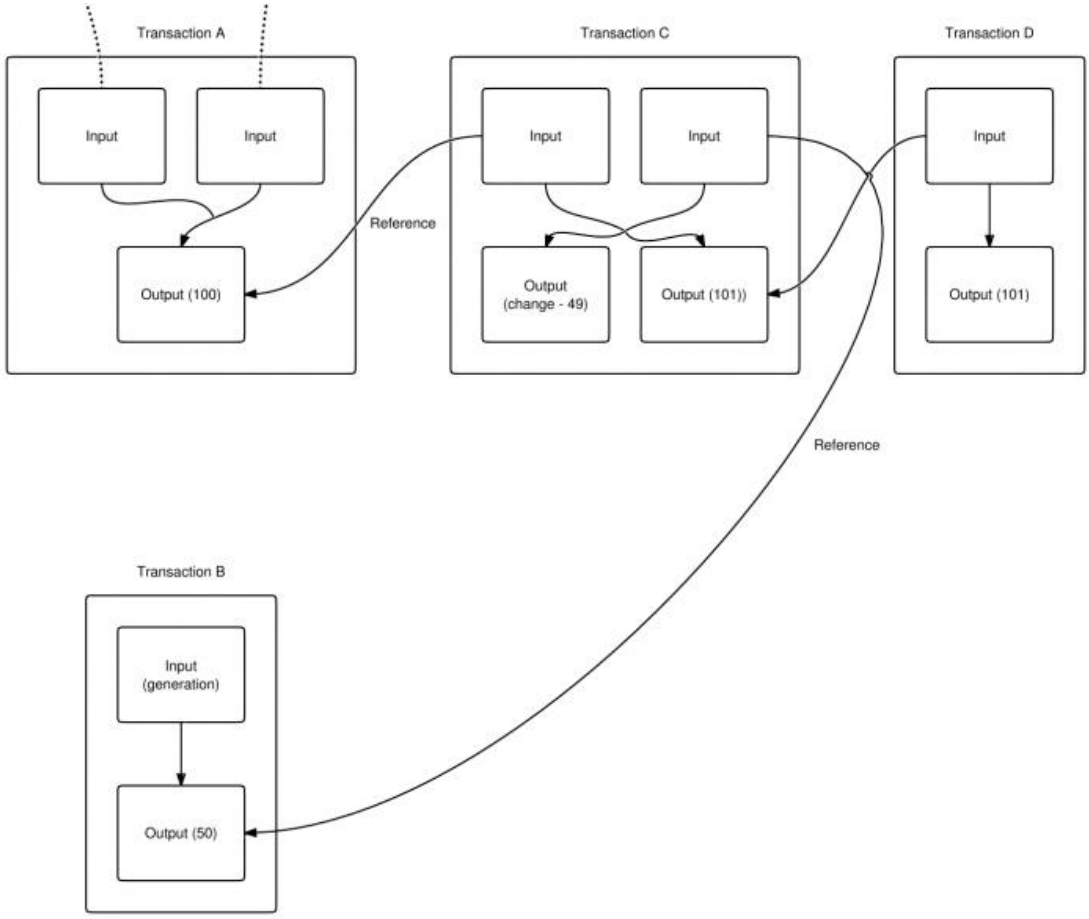
Les transaccions són la base dels bitcoins. De fet són en si mateixa els bitcoins. Quan una persona fa un pagament es genera una transacció de bitcoins. Aquesta transacció, com expliquem seguidament, conté una referència a les transaccions anteriors de les que s'han obtingut els bitcoins que utilitzem en aquesta nova transacció.

Estructura d'una transacció

Una transacció és l'acció de transferir bitcoins d'un compte a un altre. Una transacció està composta per diverses entrades i diverses eixides. També consta del valor de la transacció.

Les entrades o inputs de la transacció corresponen a les transaccions anteriors que ens van donar la possessió dels bitcoins que anem a gastar. És a dir, una entrada de la transacció actual sempre ha de fer referència a un output d'una transacció anterior en el temps. Si amb un input no hi tenim prou bitcoins per fer front a la transacció que volem fer, simplement hi afegim més inputs fins que arribem a la quantitat de bitcoins que volem transferir. No hi ha problema si ens passem de quantitat, de fet el més normal és que no hi tinguem transaccions amb les quantitats exactes. Això és perquè el valor d'una transacció no es pot dividir, és a dir, quant gastem una transacció que ens han fet ho hem de fer amb tot el seu valor, no podem dividir-la. Per tant, quant fem una transacció el més habitual és que una de les eixides o outputs d'aquesta transacció, com ho seria si la férem físicament, és el canvi, que seria un output que anirà a una de les nostres adreces i serà del valor que hi sobri d'aquesta transacció una volta sumats tots els inputs i restat el valor que hi volem pagar al destinatari (llevant també la comissió que veurem mes endavant).

En la següent imatge [17] podem veure més gràficament els inputs i els outputs d'una transacció. Si ens fixem en la transacció C (volem pagar 101 BTC) hi apareixen dos inputs que fan referència a dues transaccions que apareixen abans en el temps, una de 100BTC i un altra de 50BTC, 150BTC en total. Aquests inputs generen dos outputs, un és el valor que volem pagar (101 BTC) i un altre que seria el canvi de 49BTC que aniria a una adreça de qui origina la transacció C.



Imatge 2 inputs i outputs d'una transacció

En un futur, s'hi farà referència a aquesta transacció C des d'una nova transacció quant es vulgui gastar els bitcoins que s'hi han passat en aquesta transacció.

Però, què ocorre si la suma dels outputs és inferior a la dels inputs? Aquesta diferència, si existeix, va a parar al creador del *block* en el que posteriorment s'inclourà la transacció i és una comissió o *fee* que qui genera el *block* cobra per fer la feina de incloure aquesta transacció en un *block*. Aquest detall ho veurem més endavant.

Però si mirem les parts que té cada input veurem que aquest input fa referència a una transacció anterior en forma de *hash* d'aquesta transacció anterior. A més hi trobem un *index* que indica quin output d'aquesta transacció prèvia es fa referència (en tant que hi pot haver més d'un output per transferència) i hi trobem un tercer element anomenat ***ScripSig***. És un guió d'accions que indica com la persona que vol gastar en el futur aquesta transacció com a input ha de realitzar. Bàsicament una persona que vulgui gastar els *bitcoins* rebuts en una transacció ha de facilitar dues coses, per una part una clau pública de la que deriva l'adreça del destinatari de la transacció i per altra banda una signatura per evidenciar que es posseeix la clau privada corresponent a la clau pública que hem gastat en la primera part del guió. Aquesta forma de fer referència als inputs amb guions o *scripts* fa molt més versàtil aplicar diferents accions a realitzar per poder gastar una transferència, en tant que es pot modificar la llista d'accions a realitzar. [18].

L'output, a més del valor d'eixida de la transacció, incorpora un guió d'accions anomenat *ScripPubKey*. La combinació del *ScripSig* i del *ScripPubKey* proporciona una sèrie d'accions que s'han de complir i que per tant han de retornar el valor *True* per a que la transacció sigui vàlida.

Per tant un input tindrà la següent informació:

hash de l'anterior transacció
Index del output de l'anterior transacció
Longitud del guió de transacció
Guió d'entrada: *scriptSig*
Nombre de seqüència (normalment amb valor 0xFFFFFFFF)

i per la seua part un output tindrà aquesta informació:

Valor d'eixida
Longitud del guió de eixida
Guió d'accions d'eixida: *scriptPubKey*

Unitats de moneda

Totes les monedes tenen divisors i múltiples de la moneda base. Al cas dels bitcoins, la base és 1 bitcoin i a partir d'aquest punt es poden fer transaccions de menys valor o múltiples d'aquest valor. Es segueix el sistema de múltiples i submúltiples del sistema internacional de mesures encara que es poden usar altres tipus de notacions o múltiples/divisors. En definitiva podem dividir un bitcoin fins a la unitat més menuda anomenada **satoshi** que correspon a 0.00000001 Bitcoins.

A la següent taula tenim les unitats més utilitzades, encara que podem utilitzar altres unitats si el client que utilitzem de bitcoin està adaptat a utilitzar-les.

Abreviatura	Nomenclatura	Decimal (BTC)
MBTC	megabitcoin	1.000.000
kBTC	kilobitcoin	1.000
hBTC	hectobitcoin	100
daBTC	decabitcoin	10
BTC	bitcoin	1
dBTC	decibitcoin	0,1
cBTC	centibitcoin	0,01
mBTC	millibitcoin	0,001
µBTC	microbitcoin	0,000001
	satoshi	0,00000001

Taula 1 Unitats més utilitzades en Bitcoin

[19]

Fees o comissions

Una transacció per ser reconeguda pel sistema s'ha d'incloure a un *block*. El temps que una transacció espera per ser inclosa a un *block* depèn moltes voltes dels miners que generen aquests *blocks*, que tractaran d'incloure les transaccions més interessants per a ells, que a la fi són les que incorporen *fees* o comissions que el miner es quedarà pel treball que fa d'incloure la transacció al *block*. Normalment són transaccions que incorporen nombrosos inputs, el que fa que la transacció sigui prou gran a nivell de grandària de dades. Normalment el límit per fer transaccions sense *fees* seria una transacció de menys de 1000 bytes amb outputs de més de 0.01 BTC. En aquest punt també entra en joc la prioritat de la transacció, que depèn proporcionalment del valor dels inputs i el temps que tenen i inversament proporcional a la grandària en bytes de la transacció:

```
priority = sum(input_value_in_base_units * input_age)/size_in_bytes
```
[20]

La prioritat d'una transacció ha de ser al menys de 0.576 per no necessitar pagar *fees*, el que correspondria amb 1 BTC amb antiguitat de 1 dia (144 *blocks* per dia estimats) amb una mida de 250 bytes (1 * 144 / 250).
Per tant el que es prioritza són les transaccions més antigues i les que tenen major valor, mentre que les de menys valor o les més noves no arribem a tindre suficient prioritat per a ser incloses als *blocks* més nous

El que normalment es cobra és 0.1mBTC cada 1000 bytes de transacció(uns 3 cèntims d'euro amb el canvi d'abril de 2014). Aquesta *fee* de 0.1mBCT/kB es pot ampliar però normalment és la mínima que les *wallets* tenen configurada per defecte. Les transaccions que més *fees* paguen per kB són les que s'inclouran abans a un *block* de transaccions. [21]

Blocks

Un *block* és un conjunt de transaccions agrupades amb unes característiques especials. Aquest *block* conté, a més d'una sèrie de transaccions, el *hash* del *block* validat anterior a ell, amb el que formarem la cadena de blocks que veurem en el següent punt.

A la següent taula podem veure l'estructura d'un block amb la informació que conté: [22]

Camp	Descripció
Magic no	Valor que sempre és 0xD9B4BEF9
Blocksize	Mida del block en bytes
Blockheader	Capçalera amb 6 elements
Transaction counter	Número positiu
transactions	Conjunt de transaccions al cos del block

Taula 2 Estructura d'un block

La capçalera d'un block conté una sèrie de camps amb informació mentre que el cos del block conté les transaccions.

Camp	Objectiu
Version	Versió del block actual
hashPrevBlock	hash del anterior block 256bits
hashMerkleRoot	hash de les tx incloses al block 256bits
Time	Timestamp (actualitzat cada pocs segons)
Bits	Target actual
Nonce	numero de 32 bits aleatori (començant en zero)

Taula 3 Capçalera d'un block

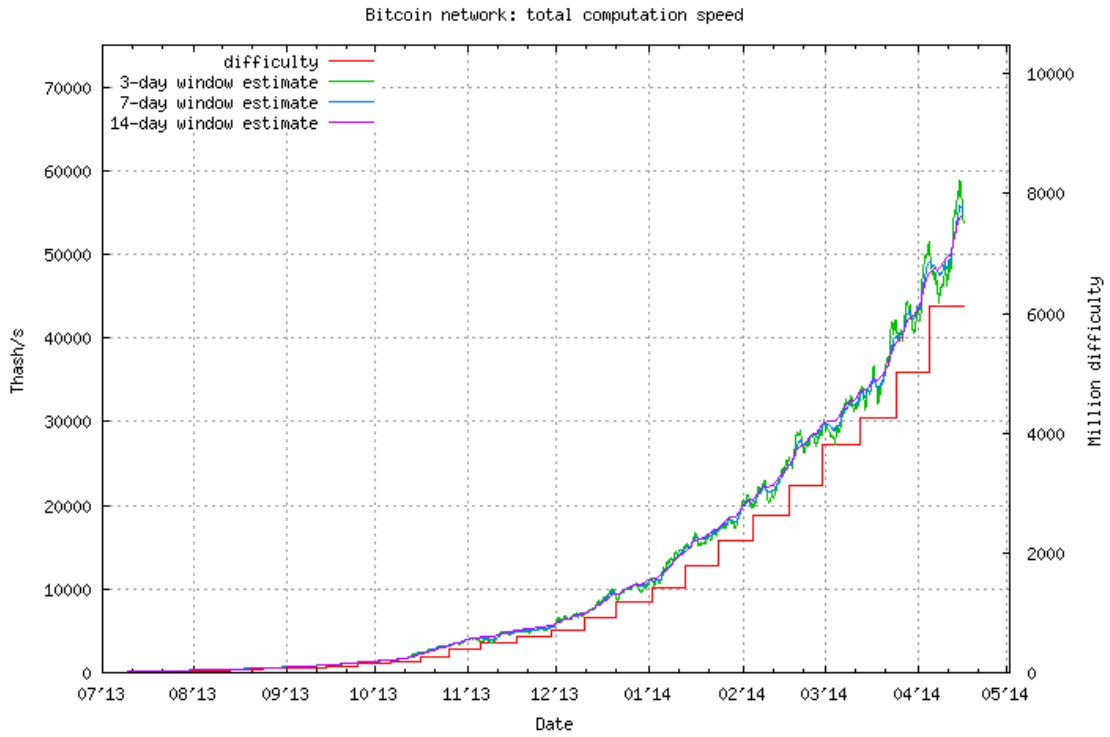
Per a considerar un *block* acceptat s'ha de solucionar un problema matemàtic que consisteix en fer un *hash* de la capçalera del block [23] [24] que cal tractar que s'ajusti a una dificultat objectiu [25] o ***target***. Aquesta *target* és un numero de 256 bits que tots els clients de bitcoin comparteixen. El *hash* obtingut amb mecanisme SHA-256 de la capçalera del *block* ha de ser inferior o igual a aquest *target* per a que el *block* sigui acceptat, és a dir, aquest *target* és un màxim a obtindre. Aquest *hash* es calcula com el doble *hash* de la capçalera de la forma SHA256(SHA256(Block_Header)).

Normalment aquest *target* té la forma d'una sèrie de zeros al principi, de manera que es força al *hash* a ser més menut i per tant també començar amb zeros. Un exemple de valor per al *target* seria el següent:

```
0x00000000000404CA000000000000000000000000000000000000000000000000
```

Amb aquest *target* es defineix un paràmetre que es sol utilitzar per a calcular la capacitat de computació necessària per a resoldre aquest problema d'obtenir un *hash* inferior o igual al *target* anomenat dificultat (*difficulty*). [26] Aquesta dificultat s'obté de dividir la dificultat de valor 1 (la màxima dificultat) entre el *target* actual (*difficulty = difficulty_1_target / current_target*). Tenim pàgines web com http://blockexplorer.com/q/getdifficulty que ens torna la dificultat actual.

Es produeixen uns 6 *blocks* per hora (un cada 10 minuts) i cada 2016 *blocks* (que solen tardar unes dues setmanes) es compara si amb el *target* que s'ha usat durant aquestes dues setmanes s'ha complert la generació d'un block cada 10 minuts, en cas de que no es compleixi aquesta cadència de *blocks* es genera un nou target que difereix en un percentatge igual a la desviació del temps objectiu. D'aquesta forma s'intenta ajustar la dificultat amb la capacitat de computació de la xarxa. Existeixen altres webs en les que es pot consultar l'evolució de la dificultat així com la dificultat estimada per al futur en funció d'aquesta desviació, com http://bitcoindifficulty.com/ (gràfica obtinguda de la sèrie que es pot consultar a Bitcoin Network Graphs http://bitcoin.sipa.be/)



Imatge 3 Evolució en el temps de la velocitat de computació i la dificultat

Qui genera el *block* rep una gratificació per haver-ho generat. Aquesta gratificació va començar en 50BTC però va disminuint cada quatre anys a la meitat (actualment és de 25BTC). La primera transacció que apareix al cos del *block* correspon a aquesta transacció (anomenada **transacció de generació**). Com aquesta adreça normalment és una adreça bitcoin de qui genera el *block*, fa que el conjunt de transaccions que es genera és diferent en per a cada generador (miner) de *blocks* i per tant el *hash* que s'obté de la capçalera també és diferent, produint-se la "lluita" per obtindre el *hash* més prop del *target* del moment. Aquesta transacció és lleugerament diferent en quant a estructura amb les que ja hem comentat

Per aconseguir aquest *hash* es juga, com ja hem comentat, amb el camp de ***nonce***. Aquest camp és un número amb el que es pot anar jugant per ajustar el resultat del *hash* de forma que ens acostem el màxim possible al *target* per baix. S'incrementa de forma lineal cada volta que es calcula un *hash* de la capçalera començant en 0. Quant per a un conjunt de transaccions s'han provat tots els valors de *nonce* i no s'ha trobat solució per al problema plantejat, a la transacció de generació s'incrementa un camp especial anomenat ***extraNonce*** (amb una mida de entre 2 i 100 bytes) [17]. Aquesta modificació fa que el *hash* de les transaccions canviï (*hashMerkleRoot*) i per tant el hash de la capçalera canvia totalment i es pot començar de nou amb un nonce de 0.

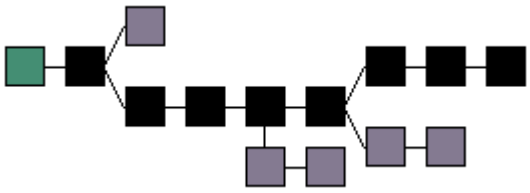
Aquesta forma de calcular el hash, juntament amb que cada generador de blocks ho calcula sobre una base de transaccions diferent (al menys la transacció de generació serà diferent) fa que aconseguir generar el *block* amb el *hash* més a prop del *target* sigui una qüestió quasi de sort i un mecanisme en el que tots tenen les mateixes oportunitats d'aconseguir la millor solució del problema.

Blockchain

Com hem vist, a la capçalera d'un *block* hi ha un camp que fa referència al *hash* del *block* anterior. Això crea una cadena de *blocks* en el que existeix un *block* inicial anomenat *genesis block* que únicament conté una transacció que correspon amb la de generació d'aquest block. Aquest block té un índex cero i a partir d'aquest *block* es comença a formar la cadena de *blocks* de bitcoin. Com que el següent *block* ha de contenir el *hash* del *block* anterior, hi ha seguretat en l'ordre de creació dels *blocks* en el temps. A més, aquesta part que fa referència al block anterior forma part de la capçalera del *block* que ha de complir amb les condicions de creació del *block* (complir amb el problema plantejat d'obtenir un *hash* per sota, però el més a prop possible, del *target* del moment).

El fet d'incloure el *hash* de la transacció anterior formant una cadena també ens assegura que alterar un *block* anterior en la cadena resultaria prou costós computacionalment parlant. Com que cada *block* conté el *hash* del anterior, alterar-hi un *block* a meitat cadena ens faria regenerar tots els *blocks* següents, el que implica tornar a calcular per a cada *block* posterior el *hash* per que s'acompleixi que el seu *hash* sigui el més prop del *target* amb el nou *hash* del *block* anterior. Aquest cost computacional és molt elevat i fa pràcticament impossible que la cadena s'alteri.

Conforme es generen els bitcoins és possible que es generin dos *blocks* amb molt poc temps de diferència. Els nodes que reben aquest *block* comencen a formar una cadena amb una ramificació amb dos nodes a la cadena. Quan es generi el següent *block* es farà referència a un dels dos *blocks* generats prèviament generant-se una cadena més llarga que serà la que anirà fent-se més llarga poc a poc. És aleshores quant el *block* "alternatiu" que s'ha creat com a ramificació de la cadena es descarta i deixa de formar part de la cadena. Totes les transaccions que hi contenia, si no han estat ja incloses en altre *block*, tornen a formar part de les que els nous generadors de *blocks* han de incorporar en el futur.



Imatge 4 Construcció d'una cadena de blocks

[27]

Existeixen algunes eines web per a mirar el contingut de la cadena de bitcoin i examinar en detall els *blocks* que hi conté, com les transaccions que hi inclouen. Per exemple blockchain.info, blockexplorer.com, blockr.io, donen informació a la pàgina principal dels últims *blocks* creats incorporats al *blockchain* i de les transaccions realitzades, així com de les *fees* i de la transacció de generació que rep

qui genera el block. Com la xarxa és *peer to peer*, totes les transaccions incloses en un block són públiques i per tant podem fer un cop d'ull a aquestes transaccions.

Block 296439[2]

Short link: http://blockexplorer.com/b/296439
Hash[2]: 000000000000000008f1718ae5a466d5bc57d01e8bbf32087e21dd674cc6f367a
Previous block[2]: 00000000000000000eebb75c3332d5af3642fefcb685aa8c9bfef0405be982ca
Time[2]: 2014-04-18 07:41:23
Difficulty[2]: 6 978 842 649.592383 ("Bits"[2]: 19009d8c)
Transactions[2]: 55
Total BTC[2]: 581.3948724
Size[2]: 46.122 kilobytes
Merkle root[2]: 8fbaeccf6c467486c0404087b760b3a625ab74e9fd0b124f70d7d0f5c6d85163
Nonce[2]: 596755628
Raw block[2]

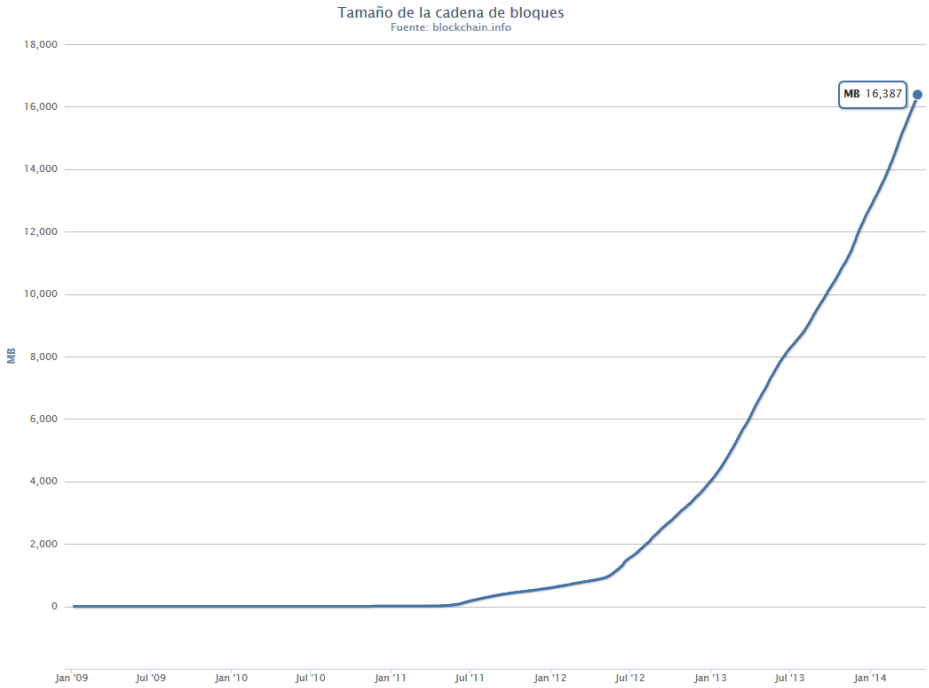
Transactions

Transaction[2]	Fee[2]	Size (kB)[2]	From (amount)[2]	To (amount)[2]
0f78f0ebd8...	0	0.102	Generation: 25 + 0.00778 total fees	1A6CwRa3QTPrwU5RyBmfKZXdvPpPvBbUVM: 25.00778
35002616f1...	0.0005	0.225	1EWCNY9fiNvWnAAEknTfiQ3X4GRUetqtb5: 423.25022123	1DVbTnr1cSfvFL4bgMyvoHReCUH9dbkdVV: 423.23072123 1Lio597R8PegSq2znq6YSPsVdeS7GgLTvC: 0.019
82be04149c...	0.0001	0.226	1gFQAoevYDbWciiphoGKQYWUemvgz1TdY: 4.43285051	194fAvzgNeZMawcznkN3ZZCSSDR2Zt5Tb4: 0.3005 1PmBSgKMbGcHVCmveGNwxpaMwyUdbgiV4: 4.13225051
			14Hg9oMQGW8KBCWHzV2gGMmGwMdLqRn6bB: 0.01002525	

Imatge 5 Informació d'un block amb Blockexplorer.com

Cada bloc dins de la cadena porta un número d'índex dins de la cadena, començant com hem vist amb el bloc zero o gènesis (creat el 3 de gener de 2009 a les 18:15:05) fins a l'últim block incorporat. Com es genera un bloc cada 10 minuts aproximadament aquest número creix constantment. Actualment existeix una cadena amb més de 296.000 blocs.

Aquesta cadena per tant cada volta és més gran i com conté totes les transaccions des de l'origen comença a tenir una mida considerable. Segons la gràfica de blockchain.info és d'uns 16 GB i actualment creix a un ritme de 1GB cada dos mesos aproximadament. Per a la majoria de clients de bitcoin tenir la cadena completa de transaccions des del gènesi no és interessant, en tant que el que els interessa principalment són les adreces que tenen bitcoins actualment, no tota la cadena de transaccions. És per això que es pugui treballar sense conèixer la cadena completa i fer ús d'informació amb les transaccions no gastades i per tant que poden generar nous pagaments. En aquest cas tota la informació referent a aquestes transaccions són uns 100MB, el que fa molt més àgil el tractament de les transaccions i el treball de clients mòbils amb un espai limitat de emmagatzemant. [28]



Imatge 6 Evolució del tamany de la cadena de blocks principal

Miners (el treball que genera nous diners)

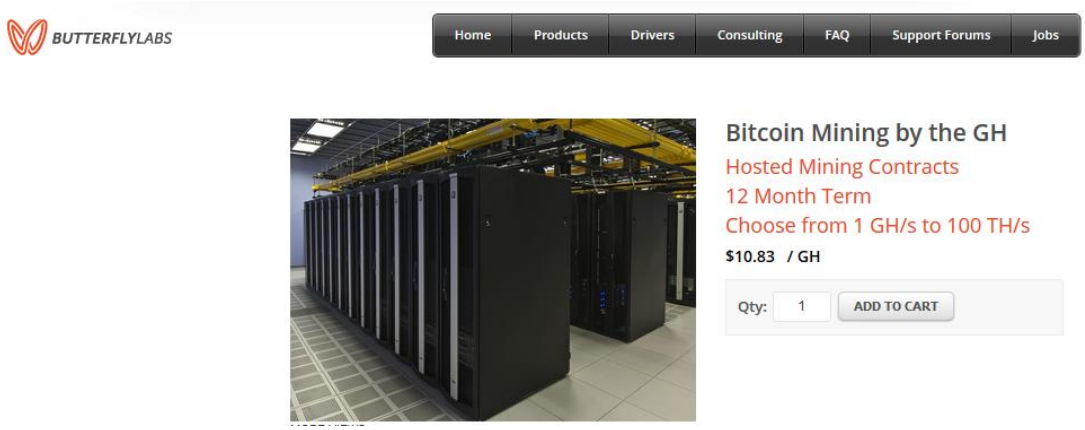
Els miners són qui generen nous *blocks* amb transaccions. El terme de mineria prové precisament del treball que els miners realitzen, d'obtenir els metalls preciosos que s'utilitzaven com a canvi per a comprar diferents productes. Igual que aquests miners originals, els miners de bitcoin fan el treball de resoldre el problema plantejat per generar un nou *block* amb transaccions. El principal objectiu de la mineria és aconseguir un consens global a la xarxa sobre quines transaccions es consideren acceptades i al mateix temps és el mecanisme d'introduir nous *bitcoins* al sistema econòmic. Quant un miner aconsegueix generar un nou *block* la resta de xarxa pot comprovar la prova de treball (*proof of work*) [29] que és simplement el *hash* de la capçalera que com ja em vist ha de ser menor que un valor objectiu. Aquesta comprovació és molt senzilla per a la resta de xarxa que només ha de fer el hash de la capçalera del *block* candidat i introduir-ho al *blockchain* i retransmetre a la xarxa aquest *block* com a vàlid.
A més de la recompensa per haver generat el *block* el miner es queda amb les *fees* que les transaccions que ha inclòs al *block* han generat. Com que la recompensa per generar un *block* disminuirà a la meitat cada quatre anys, en un futur aquesta recompensa serà prou menuda i per tant les *fees* que es cobren per cada transacció serà la principal motivació dels miners per fer aquest treball.

Per tant la mineria es basa en fer moltes operacions de *hash* per aconseguir un *block* que tingui un hash inferior a l'objectiu marcat en cada moment. Habitualment sempre hem fet ús de la CPU de les màquines per fer aquest treball però amb el temps s'han anat creant circuits especialitzats en fer l'operació de doble *hash* que es requereix com a proof of work de la generació del *block*. S'ha vist que els processadors gràfics són molt més ràpids a l'hora de fer el càlcul d'aquest

hash. També l'ús de FPGA, comparable en velocitat de càlcul a les GPU, s'ha utilitzat molt per a la mineria perquè aporten un consum mínim d'energia amb l'estalvi que això permet. A partir de 2013 apareixen xips específics per fer mineria de *bitcoin* amb productes de tipus ASIC (*Application Specific Integrated Circuit*). Aquests ASIC són directament dissenyats per fer la tasca d'obtindre el doble *hash* del *block*, el que ha anat deixant enrere les anteriors tecnologies per l'avanç tant en capacitat de processament com en consum d'energia. Al wiki de bitcoin podem trobar una llista comparativa de la capacitat d'obtindre bitcoins (mesurat normalment en milions de hashes per segon o Mhash/s o inclús en Ghash/s) que és prou interessant o es pot comparar la quantitat de hardware utilitzat avui dia per a la mineria de bitcoins (https://en.bitcoin.it/wiki/Mining_hardware_comparison)

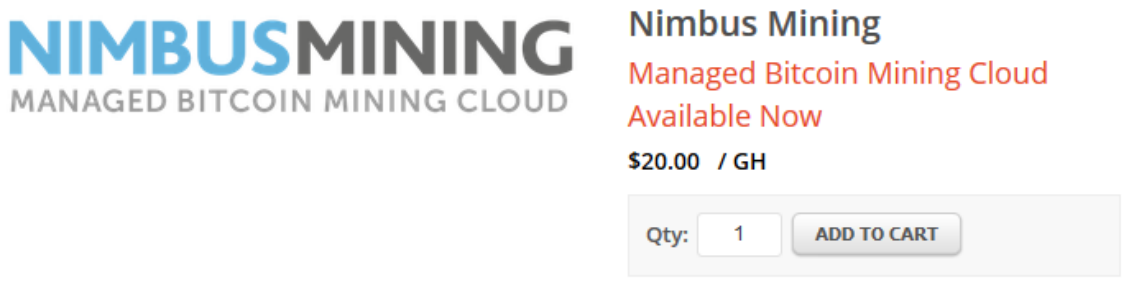
Existeixen altres mètodes de formar part del procés de mineria sense haver de dedicar-hi recursos hardware als nostres equips. Són els anomenats *cloud mining* que permeten "comprar" hores de treball o amb un lloguer mensual de capacitat de computació d'un altre miner (una mica similar al que seria el servei al núvol de Amazon AWS). També es pot llogar directament hardware específic per a minar bitcoins en forma de servei, el que permet a grans empreses invertir en grans sistemes de generació de bitcoins al mateix temps que els permet llogar aquests serveis en parts més menudes. [30]

Com exemple podem trobar el servei que dona butterfly labs, que ofereix targetes hardware per fer mineria de bitcoins però que ha llançat un servei per llogar la potència de mineria en forma de quotes anuals a pagar per cada GHash/s de capacitat.



Imatge 7 pàgina de contractació de Butterfly Labs dels serveis de mineria llogats

Com exemple de serveis de mineria al núvol tenim Nimbus Mining, al qual podem contractar per un any una quantitat de Gh/s, aquesta capacitat la associen a un grup de mineria distribuïda que serà l'encarregada de repartir després els beneficis en bitcoins d'haver creat blocks acceptats pel sistema.



Imatge 8 Pàgina per contractar serveis de mineria al núvol

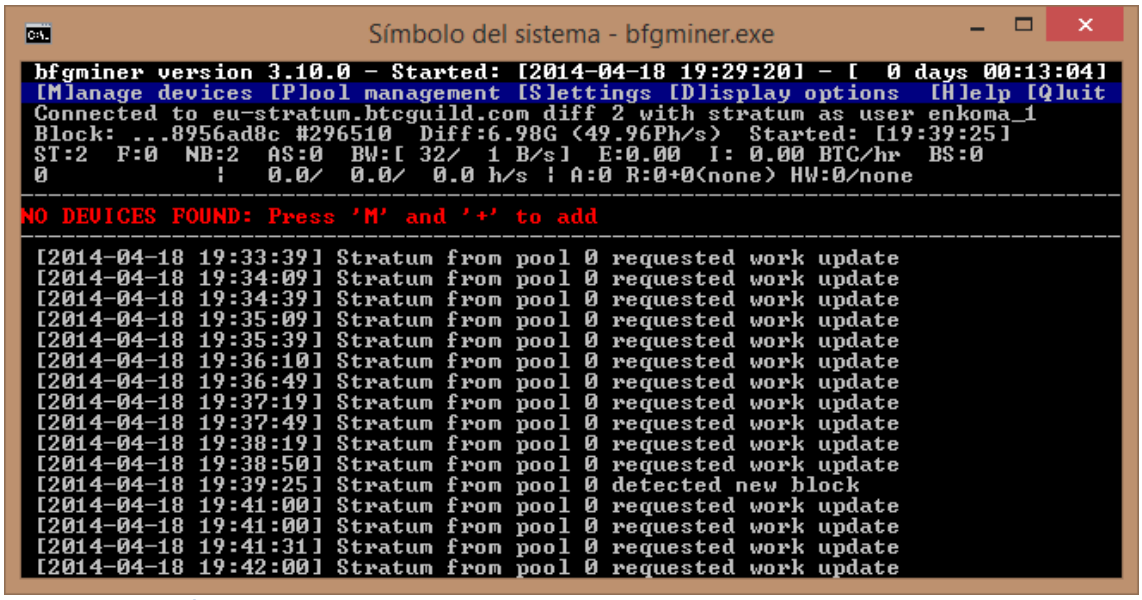
Mineria distribuïda (Pooled mining)

La mineria distribuïda és una forma de mineria molt interessant en tant que permet que qualsevol persona, encara que no tingui hardware dedicat o amb una capacitat de processament elevat, tingui més possibilitats de veure el seu esforç per obtindre hashes de transaccions recompensat. El que es fa és distribuir el treball de mineria entre un grup (pool) de miners que aporten treball per generar un block. Aquest grup es veu com un únic generador de cara a la xarxa de bitcoin, tots comparteixen la mateixa direcció per a la generació de bitcoin i el treball de fer els hash per a generar el block es reparteix entre tots els components del grup. Quan s'aconsegueix generar un block la recompensa per crear-ho es reparteix entre els membres del grup en funció de la seua contribució. Les formes de treball amb els pools són variades, des de les que utilitzen un sistema de punts, passant per les que paguen una quantitat fixa per els *hashes* que es generen sense esperar si s'obté o no un *block* amb la seua recompensa, o les que, per evitar transferències molt menudes en les que les *fees* a pagar serien superiors al valor de la transacció, van sumant el total que guanya un miner del pool fins haver acumulat una quantitat determinada de bitcoins. [31] A l'argot de bitcoin les participacions que una persona aporta al grup s'anomenen *shares* i en funció de com es reparteixen els *shares* es repartirà el valor de bitcoins que es genera. Un mètode molt utilitzat és el anomenat ***pay-per-share***, al que cada *share* enviat al sistema val una quantitat de bitcoins. Com que aquesta forma de pagament té molt de risk per als operadors (en tant que el miner cobra sempre una quantitat encara que no es generin *blocks* vàlids) les comissions que es cobren solen ser prou elevades. Altres alternatives serien les conegudes com ***proportional***, al qual, quan es genera un *block* vàlid, els bitcoins generats es reparteixen entre els miners de forma proporcional. Un altre molt utilitzat es el PPLNS (Pay-per-last-N-shares) que es paregut al proporcional però la recompensa es reparteix entre les últimes *shares* aportades al sistema, sense restringir aquests *shares* a les que han generat el block vàlid. [32]

Existeixen molts grups de mineria als quals ens podem afegir. Cada grup té una sèrie de normes per regular el treball dels miners. Hi ha grups que per cada block aconseguit es queden una comissió o bé es queden directament les fees que genera el block mentre que els bitcoins generats pel block (transacció de generació) són els que comparteixen amb el grup. Un dels que més capacitat de computació té actualment és BTC Guild amb uns 8000 TH/s. Encara que aquesta xifra no és indicativa de la quantitat de blocks vàlids que es generen al menys sí és una referència de la seua popularitat. Utilitza un repartiment dels beneficis de tipus

Pay-per-share (amb una comissió del 5%) i Pay-per-last-N-shares (amb comissions del 3%).

Per contribuir a un grup de miners per descomptat fa falta software que instal·larem al nostre equip i que permeti connectar amb els servidors del grup per gestionar la distribució del treball. Per exemple BTC Guilt utilitza un software anomenat cgminer que te una versió en la majoria de sistemes operatius, o be bfgminer. Tots dos amb suport per a utilitzar ASIC i dispositius USB per fer el hashing dels blocks.



Imatge 9 Execució de bfgminer per minar bitcoins en sistemes MS Windows

Mecanismes de seguretat

Un dels principals problemes que cal solucionar en qualsevol moneda és la doble despesa dels diners (double-spending). [33] El que es tracta d'evitar és que una persona gasti més d'una volta els mateixos diners. Aquest problema en bitcoin es resol amb l'existència de la cadena de blocs. Com que en aquesta cadena hi són totes les transaccions que es realitzen, sempre es pot comprovar que una transacció prové d'altres que no han estat gastades encara. Però, què ocorre quant el temps que tardem en fer dues transaccions es mínim i és abans de que la transacció sigui inclosa en un *block*? En aquest cas el que es fa normalment és esperar un temps prudencial per fer efectiu el servei pel que està pagant-se. És per aquest motiu que algunes transaccions, que necessiten ser incloses en blocs el més ràpid possible per la urgència de rebre un servei, portin més diners en *fees*, de manera que els miners tractaran d'incloure aquestes transaccions abans als blocs que estan creant. Una transacció sense *fees* trigarà molt més en incloure's en un *block* i per tant rebre el servei pel que s'està pagant trigui més. Què ocorre amb els bitcoins que s'han generat amb la creació del block (transacció de generació)? Hi ha una limitació al sistema que inclou un període de cadència per a gastar una transacció de generació que equival a 100 blocks, és a dir, des de la creació del block aquesta transacció de generació no es pot gastar fins que transcorren 100 blocks nous (quasi 17 hores si es genera un block cada 10 minuts). D'aquesta manera, si la transacció de generació finalment no s'inclou dins la cadena principal

(la més llarga), aquesta transacció no serà reconeguda pel sistema i per tant es perdrà totalment, mentre que només les transaccions de generació acceptades en la cadena més llarga formaran part del blockchain de bitcoin

Existeixen atacs que generen *blocks* al sistema de forma massiva per tractar de que siguin inclosos a la cadena principal. Aquests intents ràpidament són descartats en tant que, com hem vist, la cadena principal només accepta la cadena més llarga i quant el *proof-of-work* (resultat del *hash* de la capçalera) no arriba a l'objectiu directament pot descartar-se. Però, en cas que hi hagi dos *blocks* creats simultàniament, el sistema normalment espera un temps per validar la branca més llarga de la cadena.

Les wallets també són objectiu d'atacs; normalment guarden la informació sense xifrar a l'ordinador i això les fa vulnerables a robatoris. Actualment moltes *wallets* ja inclouen mecanismes de xifrat de les dades. Per exemple hi ha *scripts* com Pywallet que poden obtindre les claus privades de *wallets* (https://es.bitcoin.it/wiki/Pywallet) a partir d'haver aconseguir el fitxer .dat del client Bitcoin-Qt

També contem amb atacs a la xarxa com el anomenat ***sybil attack*** *que* consisteix en crear tants clients a la xarxa de bitcoin que és molt possible que un usuari tingui com a *peers* a la xarxa només clients fraudulents dels que l'atacant ha creat. Aquests clients podrien tirar de la xarxa al atacat, fer que només els blocs que l'atacant crea siguin els únics que reenvia i per tant l'atacat només rebrà *blocks* fraudulents, forçar que existeixi doble despesa .... La xarxa de Bitcoin tracta de fer més difícil aquest tipus d'atac utilitzant diferents tècniques, entre altres fent que cada client només connecti amb una IP per cada rang de direccions amb màscara /16 [34]

Un clàssic dels atacs a la xarxa serien els atacs de denegació de servici (DoS). Enviar molta informació a un node de manera que no pugui processar tota la informació i per tant tampoc les transaccions que s'estan enviant en el moment de l'atac. En aquest sentit els clients són els que implementen mecanismes de protecció contra aquests atacs així com mecanismes del propi protocol de bitcoin, com restringir la mida d'un *block* a 1MB o restringir el nombre de signatures que una transacció d'entrada (input) necessita verificar per a ser validada.

Com que els bitcoin es generen basant-se en un *timestamp* que també indica quin *block* ha estat creat abans o després, existeixen una sèrie d'atacs a la xarxa que es poden realitzar en base a la **manipulació de certs límits** que la xarxa accepta o inclús **manipulant els servidors NTP** on es sincronitzen els nodes. Tots els nodes tenen internament un comptador que representa el temps de xarxa. Aquest temps està basat en el temps medi dels *peers* del node, un temps que és indicat amb la indicació de la versió del node quant un *peer* connecta. Si el temps calculat excedeix de 70 minuts del temps local del node s'utilitza el temps del node com a mecanisme de seguretat. Algunes de les precaucions que utilitza la xarxa, per exemple, són no acceptar *blocks* amb una marca de temps superior a dues hores del temps actual de la xarxa, rebutjar també *blocks* amb un *timestamp* inferior al temps medi dels últims 11 *blocks*. En general la finestra d'un atac és de 140 minuts

(70 minuts per dalt del temps de la xarxa + 70 minuts per baix del temps de la xarxa). [35]

El 14 d'abril de 2014 es va fer púbic una vulnerabilitat de seguretat de OpenSSL coneguda com a Heartbleed. Aquesta vulnerabilitat va afectar al software wallet Bitcoin Core versió 0.9.0. Amb un atac que utilitzi aquesta vulnerabilitat es pot obtenir un bolcat de informació de la memòria del ordinador vulnerable. Per aquest motiu van treure la versió 0.9.1 d'aquest wallet. També la versió 4.1.1 de Android és vulnerable a aquesta vulnerabilitat pel que es recomana utilitzar al menys Android 4.1.2 amb la versió 3.45 del wallet per a aquest sistema operatiu mòbil. [36] El codi CVE d'aquesta vulnerabilitat per a més detalls es pot consultar a CVE-2014-0160

A la web de bitcoin.org podem trobar una recopilació d'alertes de seguretat que han anat afectant a bitcoin i on s'anuncien les noves alertes:

https://bitcoin.org/en/alerts

I al *wiki* de bitcoin també hi ha un llistat dels codis CVS de les vulnerabilitats de seguretat que afecten d'una o altra forma el ecosistema de bitcoin:

https://en.bitcoin.it/wiki/Common_Vulnerabilities_and_Exposures

Altres tipus d'atacs no directament dirigits a la xarxa de bitcoin són els que tenen com a objectiu els diferents serveis de wallets al núvol. Aquests serveis guarden els diners i actuen com a bancs. Per fer aquests serveis és necessari donar al proveïdor del servei les nostres claus privades per a gestionar la nostra cartera i les nostres transaccions. El gran problema d'aquests serveis és que si perden les nostres credencials o simplement si la seguretat es veu compromesa, possiblement els nostres diners desapareixeran, bé perquè algú hagi aconseguit les nostres claus privades i es fa una transacció directament a altres comptes seus o bé perquè no hem fet còpies de seguretat de les claus privades i no podem recuperar els diners que teníem per no poder signar les noves transaccions amb aquesta clau. A més, dedicar temps a planificar i organitzar un atac a un d'aquests serveis resulta prou rentable perquè la quantitat de diners que es pot obtindre d'un atac d'aquestes característiques pot ser molt elevat. També els atacs a empreses dedicades al trading o exchange de bitcoins amb altres monedes poden ser objectiu d'un atac per la gran quantitat de diners que mouen i la quantitat de transaccions que hi gestionen. Aquestes empreses sempre poden tindre un problema o un forat de seguretat que s'aprofiti per un atacant per obtenir benefici. Un dels que més repercussió a nivell mundial ha tingut va ser el de MtGox.com el passat 10 de febrer. Aleshores, aquest proveïdor de canvi de moneda japonès va notificar que havia perdut uns 750.000 BTC d'usuaris i uns 100.000 BTC propis. La pèrdua, segons ells [37], va ser deguda a una vulnerabilitat, coneguda com *transaction malleability*, que fa que una persona pugui treure diners i després modificar l'identificar de la transacció perquè sembli que mai ha ocorregut, tornant a demanar els diners altre cop. [38] L'explicació que donà MtGox va ser que aquest era un problema del protocol de bitcoin però sembla que aquesta vulnerabilitat ja era coneguda i que era responsabilitat dels programadors de serveis d'intercanvi

de moneda qui havien de ser conscients del problema i prendre precaucions per que no s'explotés. Els actius han estat congelats en vistes a que puguin ser utilitzats per pagar els deutes, per tant l'empresa no pot continuar treballant fins que no quedi tot investigat. El 16 d'abril es va designar un administrador judicial provisional sota el procediment d'investigació al que està la empresa sotmesa on ja s'anuncia la possibilitat, prou segura, de declarar l'empresa en fallida. [39] Algunes publicacions apunten a que no totes les pèrdues que ha tingut MtGox són per aquest tipus d'atac, sobre tot després que aparegueren uns 200kBTC a un wallet perdut amb un format antic al sistema. [40] També va sofrir un atac paregut el *marketplace* (principalment de continguts il·legals) anomenat *Silk Road 2*, on van desaparèixer uns 4500BTC [41].

Xarxa de proves testnet

Existeix una xarxa paral·lela a la principal de bitcoin anomenada *testnet*. Aquesta xarxa funciona amb el mateix protocol de bitcoin que la xarxa principal però amb una cadena de blocs diferent. En aquesta xarxa la dificultat mínima 1.0 correspon a una dificultat 0.5 de la xarxa principal i a més si no s'ha generat un *block* en 20 minuts automàticament la dificultat fa un reset i torna a ser la mínima, per tant la producció de bitcoins d'aquesta xarxa pràcticament no té valor. De fet és molt fàcil iniciar-se en com es treballa amb els bitcoins i obtindre alguns bitcoins de test gratuïts.

Hem fet algunes proves amb un client bitcoin per a Android (testnet3 de Andreas Schildbach versió 3.45-test). Per obtindre alguns bitcoins de mostra podem visitar les següents webs:

http://faucet.xeno-genesis.com (donen 0.1 bitcoins de prova)
http://tpfaucet.appspot.com (1btc de prova)

He fet una nova cartera dins el *wallet* per a Android amb l'adreça

mxm76DN4ELyz7s7iKz8oYfxBAxmm8nSYCM

Demanant bitcoins tenim aquesta primera transacció en la que ens donen 1btc del *testnet*

http://blockexplorer.com/testnet/t/2P5jx4Uy2L

fem després un altra petició per a rebre 0.1 BTC amb la següent transacció:

http://blockexplorer.com/testnet/t/4nZu4ZdpEf

Ara decidim tornar el 0.1BTC que ens ha donat xeno-gènesis i veiem que tornem el 0.1 BTC al conte que ens ha donat la transacció però la fem abans que la transacció de 0.1BTC inicial haja estat validada incorporant-la a un *block*. Això fa que s'utilitzi la primera transacció de 1btc com a entrada (input) amb una eixida de 0.1btc per

al destinatari i un canvi que torna a la nostra adreça però descomptant les comissions que s'han cobrat per aquesta transacció que son de 0.0001 btc

http://blockexplorer.com/testnet/t/6mwBQ6FAyM

La *testnet* per tant és de molta utilitat si volem fer proves de transaccions o per a provar *scripts* o *wallets* que programem sense problemes a perdre diners de veritat.

Bitcoin Trading

Com tota moneda, bitcoin porta al darrere un comerç al qual s'accepten bitcoins com a mitjà de pagament. Existeixen una gran quantitat de serveis, pràcticament tots *online*, que accepten bitcoins, des de programari passant per serveis psicològics, llibres, còmics o electrònica de consum [42]. Però una de les parts important són els serveis financers i més concretament els serveis de canvi de moneda (*exchange*). Aquest és el principal objectiu del *bitcoin exchange* a la xarxa i la forma més habitual de fer *trading* amb bitcoins.

El canvi de bitcoins bàsicament és un mercat d'oferta i demanda de moneda. És el que a les monedes tradicionals es coneix com a mercat de divises o *Forex (Foreign Exchange)* [43]. Això no és res de nou, des que existeixen monedes s'ha fet negoci amb l'intercanvi de moneda obtenint **una comissió pel servei ofert**. El que marca bitcoin és que el seu preu té prou fluctuació i variacions el que fa que es puguin aconseguir beneficis comprant quant tenen un preu baix i vendre quan estan a un canvi més alt. El preu de la moneda varia en funció de la demanda que hi hagi de gent que vol comprar bitcoins en moneda estrangera i la gent que vol vendre bitcoins per tornar a tindre moneda de curs legal en algun país. La moneda es canvia als portals que fan *bitcoin exchange.* Cada portal gestiona la demanda i la oferta de moneda i per tant els tipus de canvi són diferents entre els diferents portals. Els bitcoins, durant els primers anys, tenien un tipus de canvi pràcticament inexistent. Cap persona canviava diners per bitcoins. Quant es feia, era a preu molt reduït, del voltant del cèntim de dòlar per bitcoin (al 2010). Conforme la moneda es va fent coneguda i cada volta es pot fer ús d'aquesta en més serveis, comença a haver-hi una forta demanda de bitcoins però poca oferta, el que fa que el preu vagi en augment fins el punt més alt que ha conegut el canvi, quasi 1200 dòlars per bitcoin, el 27 de novembre de 2013, repetint-se aquest preu en gener de 2014. Però aquest preu va durar molt poc per a després tornar a uns 800-900 dòlars [44]

Imatge 10 Evolució del preu de canvi Bitcoin - US Dollar

Clar que la referència s'ha de prendre a nivell global perquè, com s'ha comentat, cada empresa dedicada al canvi de moneda porta un valor de canvi diferent. A la gràfica anterior es pot veure l'evolució del preu del bitcoin a nivell global des del seu origen fins avui dia. Una ullada al detall dels últims mesos ens dona una idea de la volatilitat d'aquest mercat i dels canvis tan grans que hi ha a les transaccions i als tipus de canvi

Imatge 11 Detall del canvi al moment de màxim valor dels bitcoins

Aquets mercats són una mica com els mercats d'accions, a més de la oferta i la demanda hi intervenen altres factors com pot ser la credibilitat de la moneda i la seguretat que dóna als inversors. En aquest cas, alguns problemes com els atacs a empreses de canvi de moneda, robatoris de *wallets* o possibles errors al protocol de bitcoin fan que la gent deixi de veure el bitcoin com una inversió on deixar els diners i vulguin eixir. Al mateix temps, en vistes de aquests problemes, la gent no vol comprar i per tant el preu de la moneda cau molt ràpidament.

Fins al seu tancament en febrer de 2014, MtGox.com era el servei de canvi amb més activitat de tota la xarxa de bitcoins i al mateix temps el més antic. Actualment existeixen moltes altres com bitstamp.net, bitfinex.com, BTC-e o BTC China (en ordre de volum de diners mogut). Als canvis en euros qui més volum de diners mou és el servei de Kraken seguit de bitcoin.de. Un lloc interessant per donar una ullada a les diferents cotitzacions de bitcoin a diferents *traders* és Bitcoin Charts en la secció de Markets.

Segona part - Anàlisi de dades

En aquesta segona part del treball anem a recollir una sèrie de dades sobre les transaccions de bitcoins i analitzar-les. A partir de les dades i del seu anàlisi podem posar percentatges i nombres a algunes de les característiques teòriques i sobre tot analitzar el mecanisme de comissions i el seu impacte en les transaccions.

Entorn de treball

Per fer aquesta anàlisi hem fet ús d'un ordinador MacBook Pro de 2.6GHz Core i7 amb 16 GB de RAM, sistema operatiu MacOS X 10.9.2 amb una sèrie de software necessari per fer l'anàlisi:

Apache 2.2.26
PHP 5.4.25
MySQL 5.6.17
phpMyAdmin 4.1.14
Python 2.7.5
MySQL Connector for Python 1.1.6
Sublime Text 2.0.2 (Per editar codi Python)
Splunk 6.1

API de blockchain.info

La web blockchain.info ofereix un API o interfície de programació d'aplicacions (https://blockchain.info/es/api/blockchain_api) que permet fer una consulta que torna les dades en format JSON. Aquest format es caracteritza per tindre una estructura de parelles atribut-valor [45]. Més concretament hem usat la crida de tipus *http://blockchain.info/rawblock/$block_index* que torna les dades de un block específic donant el hash o l'índex del block que volem obtenir. Al següent exemple veiem les dades del block i la primera transacció, la de generació de block).

https://blockchain.info/rawblock/00000000000000068bd79bfdf4ea1912177912ca2d66c76c720582760584e1b


```
{
        "hash":"00000000000000068bd79bfdf4ea1912177912ca2d66c76c720582760584e1b",
        "ver":2,
        "prev_block":"00000000000000051e2cf1ba058ba59e15738f0136f676f08df85edf890e01b",
        "mrkl_root":"9b5032253373cd0d1de4122565088f491c3880ae01855aaaced9ced87b856094",
        "time":1398478333,
        "bits":419470732,
  "fee":1800218,
  "nonce":3183937182,
        "n_tx":128,
        "size":53447,
        "block_index":401223,
        "main_chain":true,
        "height":297731,
        "received_time":1398478333,
        "relayed_by":"85.25.195.79",

        "tx":[{"time":1398478333,"inputs":[{}],"vout_sz":2,"relayed_by":"85.25.195.79","hash":"954a8e5146796003c814
3f43ed46a41850056834c0095f8e52c8a463f59ed607","vin_sz":1,"tx_index":55285831,"ver":1,"out":[{"n":0,"value":1,"addr"
```

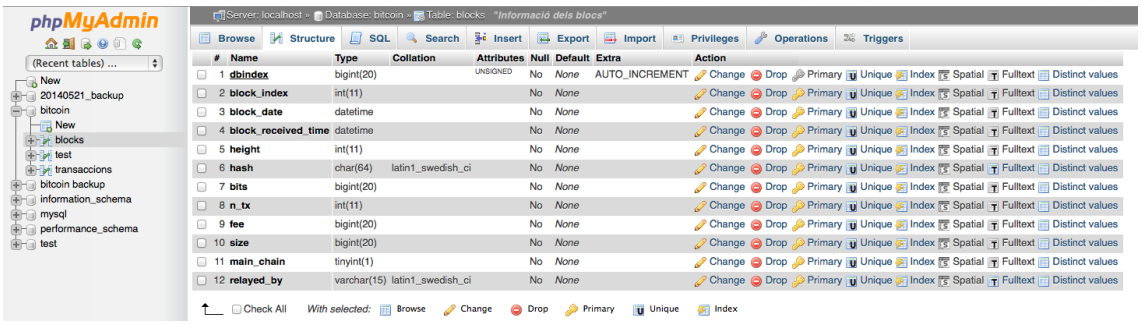
:"1BGbGFBhsXYq6kTyjSC9AHRe1dhe76tD6i","tx_index":55285831,"spent":true,"type":0,"script":"76a91470a419336ae604ddf73e4f8199f1d03b3c2f260b88ac"},{"n":1,"value":2501800217,"addr":"1A73ExsM2doRwTLp82rv5U36QHbBFmHD1X","tx_index":55285831,"spent":false,"type":0,"script":"76a91463dd7f90e949f8e354415233e51d3392d4d8f55288ac"}],"size":138}, ...

Scripts utilitzats i estructura de la base de dades

Una forma prou senzilla de processar les dades en format JSON és amb Python. Aquest llenguatge té eines per llegir d'una URL i si és JSON crear un objecte que es pot recórrer com si fos un array, accedint directament als valors de forma indexada.

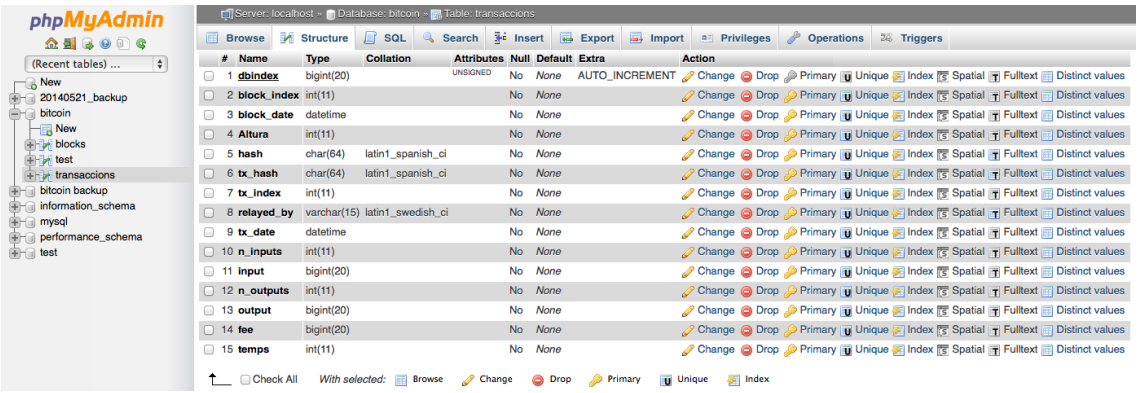
Abans de realitzar l'script necessitem una base de dades per emmagatzemar la informació. Aquesta base de dades l'hem feta amb MySQL amb l'ajuda de phpMyAdmin que ens facilita l'edició de les característiques d'aquesta base de dades. Hem creat una nova base de dades anomenada bitcoin amb dues taules, una anomenada transaccions i un altra anomenada blocks.

La taula block contindrà informació sobre els blocks i les seues característiques, com les comissions totals que ha generat, el nombre de transaccions que hi inclou o la data en la que s'ha generat. També informació sobre qui ha retransmès el block, que ens donarà informació sobre qui ha generat el block. En aquest cas és molt probable que qui retransmet el block sigui qui l'ha generat, perquè segurament blockchain.info està connectat als principals pools de generació de blocks de la xarxa Bitcoin.



Imatge 12 Estructura de la taula blocks de la base de dades MySQL vista amb phpMyAdmin

Pel que fa la taula de transaccions, obtindrem informació sobre detalls interessants d'aquestes, però sense parar en detallar cada input i cada output, sinó només la quantitat total de inputs i outputs que hi tenim a cada block, els diners d'entrada i els d'eixida i el valor de la comissió que s'ha pagat per a cada transacció. També guardem la data en la que la transacció s'ha produït i la del block, amb un camp addicional amb el temps en segons que la transacció ha trigat en incloure's a un block (la data del block menys la data de la transacció). Hem guardat també algunes dades sobre qui ha retransmès la transacció, encara que, en les transaccions, aquesta informació és poc informativa en tant que és informació del peer que l'ha retransmès.



Imatge 13 Estructura de la taula de transaccions de la base de dades MySQL vista amb phpMyAdmin

L'script que hem realitzat té un cos principal i una funció. La funció és qui realitza el treball d'analitzar el JSON i d'incloure les dades que ens interessen a la base de dades. L'estructura del JSON que torna l'API de Blockchain.info pot tractar-se en forma d'*array* que es pot recórrer en forma de bucle. Cada transacció és un índex dins un array, dins de cada transacció poden accedir a cada input o a cada output fent un recorregut també de bucle (hi tenim un array de transaccions i dins de cada transacció un altre array de inputs i un altre de outputs). Per exemple si volem accedir a una transacció en concret fem:

data['tx'][index]

suposem que data és on hem llegit el JSON llegit de l'API de blockchain.ifo, index seria l'identificador de la transacció començant en cero. si volguérem accedir al output cero de la transacció 10 ho faríem d'aquesta forma

data['tx'][**10**]['out'][0]

i si d'aquest output volem el seu valor només hauríem d'accedir al seu atribut "value"

data['tx'][**10**]['out'][**0**]['value']

Per tant, és senzill automatitzar l'accés als atributs del JSON, obtindre la informació i operar amb ella.

Una volta obtinguts els valors que volem simplement construïm la sentència SQL que volem fer contra la base de dades, que serà un ***insert*** directament a la taula corresponent de la base de dades.

El que fa senzill automatitzar el recorregut dels blocks és que tenim informació sobre el bloc anterior. Com sabem, els blocs formen una cadena enllaçada, per això cada bloc fa referència al bloc anterior i aquesta informació podem obtenir-la del mateix JSON amb una consulta com ***data['prev_block']*** i amb aquesta informació podem tornar a cridar la funció amb un nou block per analitzar de forma iterativa. Només haurem de posar un límit al programa principal *main* per a que pari en arribar a determinat nombre d'iteracions.

Una volta executat l'script i creades les bases de dades tenim algunes dades que és interessant conèixer:

Taula	Nº de Entrades	Espai usat a la BD
transaccions	1415233	398,1MB
blocks	3669	1,5MB

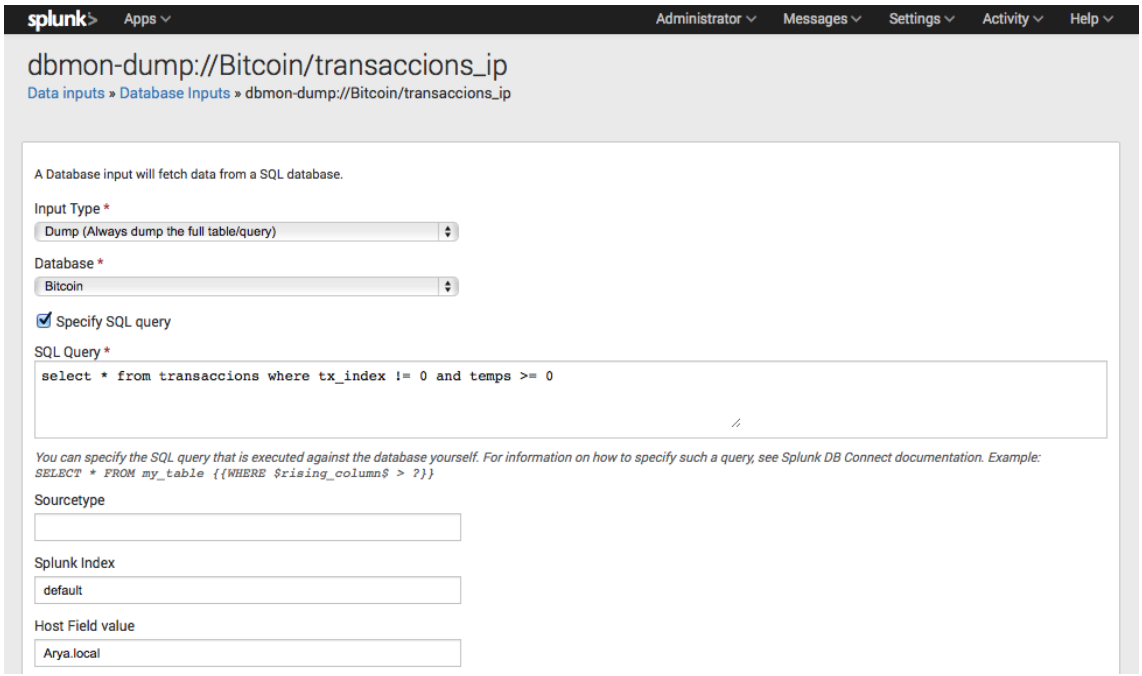
Taula 4 Dades arreplegades a la Base de Dades

Anàlisi de les dades amb Splunk

Splunk és un analitzador de dades i correlador d'esdeveniments. Permet importar dades de logs de sistema o de qualsevol base de dades i a partir d'aquestes dades realitzar informes i gràfiques personalitzades amb les dades. Està especialitzada en el que s'anomena "big data" o amb l'anàlisi de grans quantitats d'informació. Té aplicacions específiques per a seguretat informàtica, correlació d'esdeveniments estudiant diferents fonts de logs de sistema o de diferents orígens (sistemes, servidors web, equips de xarxa ....).

Per al nostre cas splunk obtindrà les dades de la base de dades mySQL que hem creat però una mica filtrada. Hi ha transaccions com les de generació que no utilitzarem a l'anàlisi perquè realment aquestes no paguen comissions. Tampoc farem us de les transaccions amb temps negatius. Com que la xarxa és distribuïda i cada usuari fa ús d'un rellotge diferent, quant es fan les transaccions el usuari que la fa marca l'hora local a la transacció. Com hem vist a la primera part, el sistema permet una desviació de fins a 70 minuts per sota del temps de la xarxa i uns altres 70 per dalt. Això vol dir que tindrem transaccions amb temps d'inclusió al block negatius. Aquestes transaccions amb temps negatius també les llevarem de les estadístiques i ens centrarem només a les transaccions amb temps positius. Per tant a l'hora d'obtenir les dades de la base de dades amb splunk farem dues consultes, la primera per obtenir dades de les transaccions accedint a la taula transaccions i fent ús d'aquesta consulta SQL:

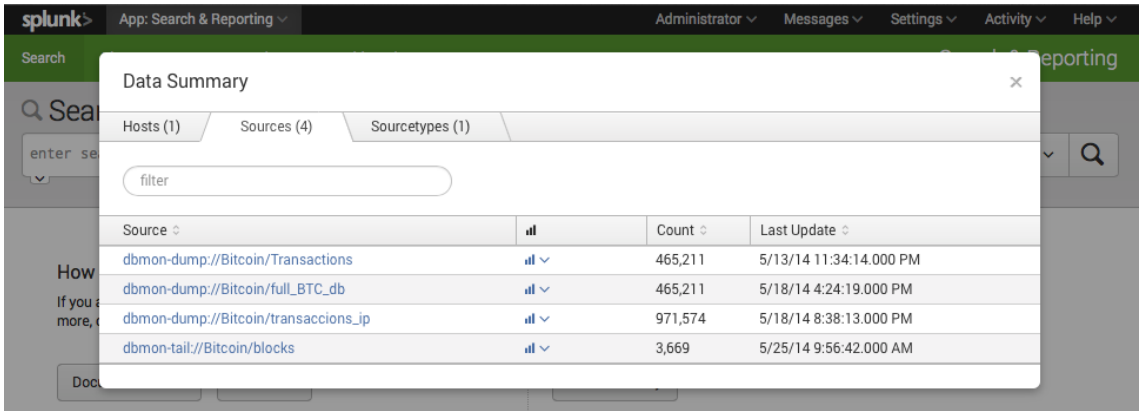
```
select * from transaccions where tx_index != 0 and temps >= 0
```



Imatge 14 Configuració de importació de dades de transaccions amb Splunk

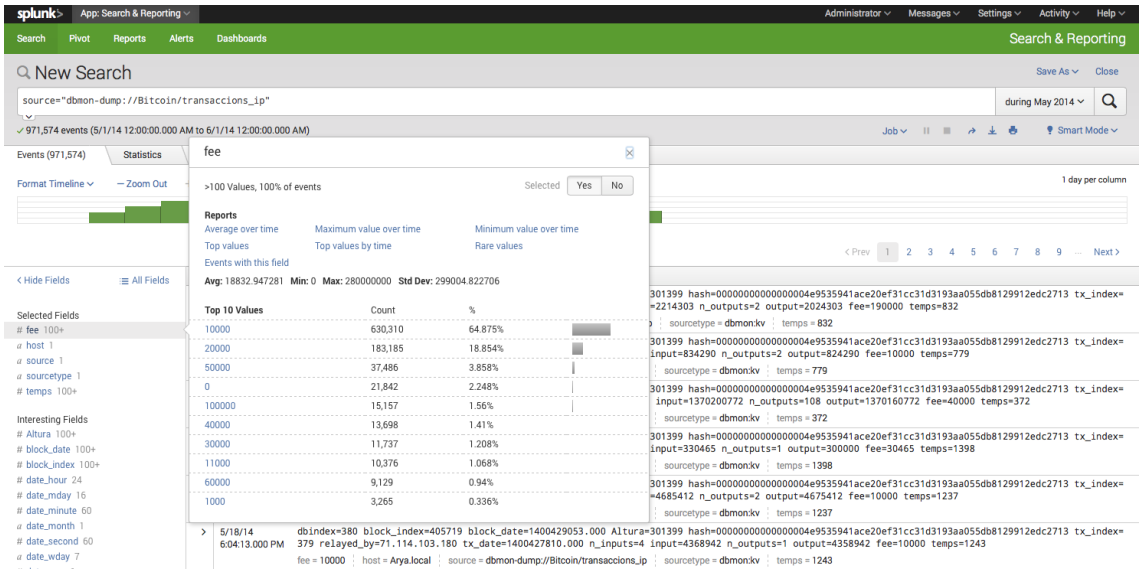
Amb la taula dels blocks no farem una consulta SQL i importarem tota la base de dades perquè ens interessen totes les entrades.

Una volta splunk ha llegit totes les dades podem fer estadístiques basant-nos en aquestes dades, amb una sintaxi pròpia que no resulta molt complicada de entendre.



Imatge 15 Llistat de fonts de dades amb les que podem treballar amb Splunk (importades prèviament)

Encara que farem ús directament de les cerques al servidor de splunk, es poden realitzar operacions molt més complexes fent ús d'arxius XML que aporten molta més potència. Podem seleccionar diferents fonts de dades encara que nosaltres no cal que fem correlacions entre diferents fonts i les cerques les farem individualment en cada font d'informació



Imatge 16 Visió general de les dades amb Splunk i estadístiques del camp fee

Una volta incorporades les dades a splunk amb els filtres que hem indicat tenim una quantitat de dades per fer les gràfiques i obtindre les estadístiques que es reflecteixen en la següent taula:

Font de dades	Transaccions/blocks
transaccions_ip	971574 transaccions
blocks	3669 blocks

Taula 5 Dades per al anàlisi

Amb totes aquestes eines ja podem començar a analitzar les dades.

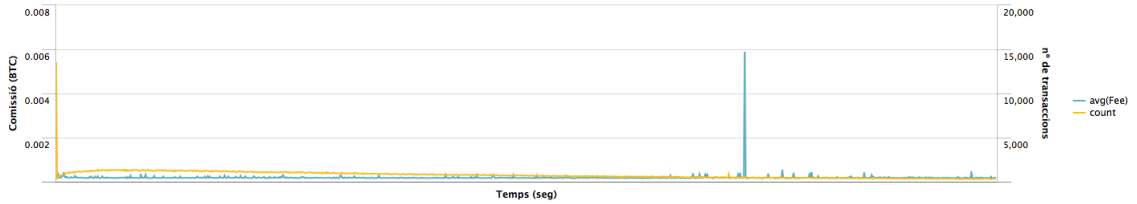
Prioritat en les transaccions amb fees

Ens podem plantejar algunes qüestions per anar analitzant les dades que hem arreplegat.

En primer termini fem una ullada a com queda una gràfica amb quasi totes les transaccions reflectides. És prou difícil fer-ho perquè hi ha més de 900.000 transaccions.

En aquesta gràfica pràcticament no s'aprecia res, encara que majorment es veu que hi ha més transaccions que triguen poc temps que les que triguen més. A més, per la gran quantitat de transaccions, només es mostren les primeres mil transaccions.

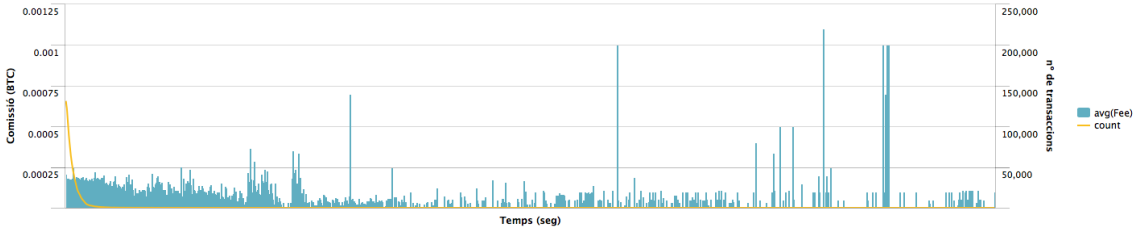
Splunk: source="dbmon-dump://Bitcoin/transaccions_ip"| eval Fee=fee/100000000 | chart avg(Fee) count by temps



Imatge 17 Comissió mitjana en funció del temps (truncada a mil transaccions)

Com que el temps entre blocks actualment és d'uns 564 segons (https://blockexplorer.com/q/interval), quasi 10 minuts, el que podem fer és agrupar les transaccions en la gràfica en intervals de temps. D'aquesta manera reduïm la quantitat d'entrades de l'eix X i també tenim una visió més clara tant del temps de les transaccions com de la comissió mitja que tenen aquestes transaccions.

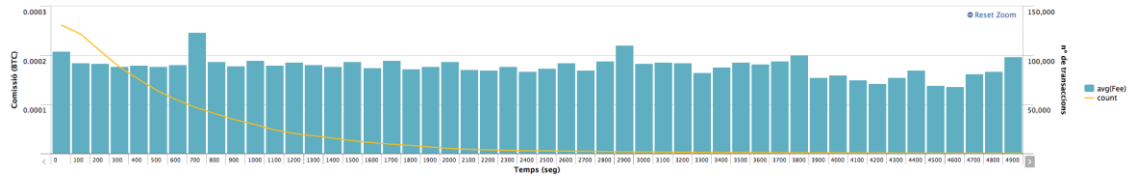
Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" | eval Fee=fee/100000000 | bucket temps span=100s | chart avg(Fee) count by temps



Imatge 18 Comissions mitjanes en funció del temps agrupades en blocs de 100 segons

El problema amb aquestes gràfiques és que a nivell general hi ha tants valors que no podem veure correctament els valors de l'eix X, pel que podem fer un zoom tant al principi com al final que ens aclarirà les dimensions de la gràfica, en cara que ja veiem una cosa clara, conforme avancem a la dreta a l'eix X veiem més transaccions amb valor cero.

Aquest és un zoom del principi de la gràfica on s'aprecia la ràpida caiguda del nombre de transaccions (la major part de les transaccions queden en aquesta part de la gràfica:

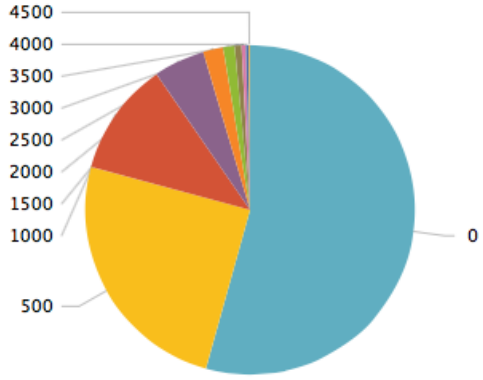


Imatge 19 - Zoom a l'inici de la gràfica de comissions mitjanes en funció del temps

Veiem que el primer grup de transaccions, el que va un temps de 0 a 100 segons, és també el més nombrós de tots amb un total de 131.256 transaccions amb una comissió mitjana de 0,000209 BTC. Ja en aquesta gràfica veiem que quasi tots els grups de transaccions paguen una comissió mitjana al voltant de 0.0002 BTC.

De fet, el 53,7% de les transaccions que hem analitzat triguen entre 0 i 500 segons en incloure's a un block, com es veu a la següent gràfica:

Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" | bucket temps span=500s | top temps



Imatge 20 Temps de transacció més comuns

Temps	Transaccions	Percentatge
0	522418	53,77%
500	239123	24,61%
1000	107621	11,08%
1500	48198	4,96%
2000	18984	1,95%
2500	10800	1,11%
3000	6463	0,66%

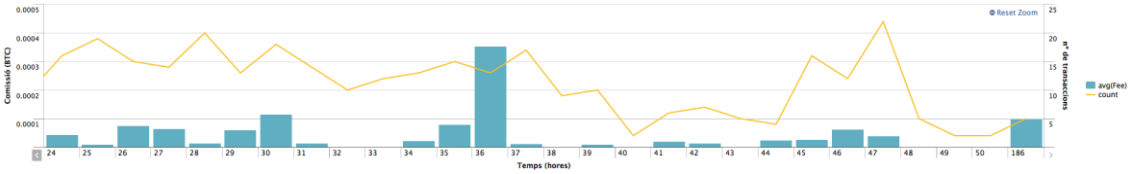
Taula 6 Temps de transacció més comuns

Això vol dir que la majoria de les transaccions es validaran en el següent block, mentre que un 76% s'inclouran en un temps màxim de dos blocks.

Si fem una ullada a les transaccions que més temps han trigat veiem aquest detall interessant:

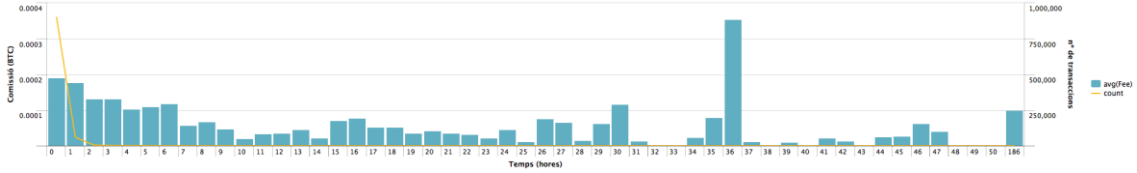
Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" | eval Fee=fee/100000000 | eval Temps=round(temps/3600) | chart avg(Fee) count by Temps

Com a curiositat, la transacció que més temps ha trigat en incloure's a un block ha sigut de 186 hores (més de 7 dies) i ha pagat una comissió per aquesta transacció de 0,0001 BTC. En aquesta gràfica veiem transaccions que han trigat més de 24 hores des que la transacció es va generar fins que es va incloure a un block.



Imatge 21 Comissió mitjana per transacció amb temps superior a 24 hores

Però el que més ens sorprèn és que hi ha transaccions que paguen comissions que han trigat fins a set dies a ser incloses a un block. Es clar que hi ha transaccions que no donaran cap comissió, però pareix que predominen les que paguen alguna comissió.
Podem veure aquesta gràfica del temps en incloure's als blocs per hores per a totes les dades



Imatge 22 Comissió mitjana per transacció per hores

Veiem que hi han transaccions amb comissions altes que han trigat fins a 186 hores en incloure's a un block. Aquestes dades ens porten a concloure que no pareix que hi hagi una relació directa entre pagar una comissió i trigar menys temps en ser inclosos als blocks, encara que sí es veu clarament que les transaccions que triguen menys sí han pagat comissió.

Transacción Ver información de una transacción de Bitcoin

61a7a00590ab9c3a86b6252dd2a63f798e6088fb9bcb5c583a860c323331c1db

37Yi2imJ2crpRvKquf3UDsbE6xNdcNjs9T → 3NpCPwnaws2gNKtxFmbmFHFFTUPquSHvrd 0.0397 BTC

0.0397 BTC

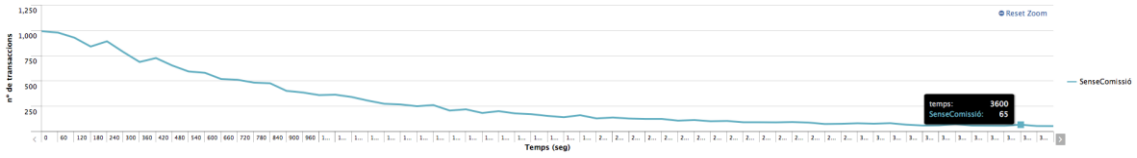
Resumen	
Tamaño	435 (bytes)
Hora de Recepción	2014-05-08 00:32:29
Incluídas en el Bloque	300897 (2014-05-15 18:55:53 +11,183 minutos)
Confirmaciones	2484 Confirmaciones
Retransmitido por la IP	5.10.170.94 (whois)
Visualizar	Ver Gráfico de Árbol

Entradas y Salidas	
Entrada total	0.0398 BTC
Salida Total	0.0397 BTC
Comisiones	0.0001 BTC
Estimado de BTCs transaccionados	0.0397 BTC
Scripts	Mostrar scripts y Coinbase

Imatge 23 Detall d'una transacció generada el dia 8/5/14 i inclosa a un block el 15/5/14

Fem una nova gràfica mostrant el temps que triguen les transaccions sense comissió durant la primera hora:

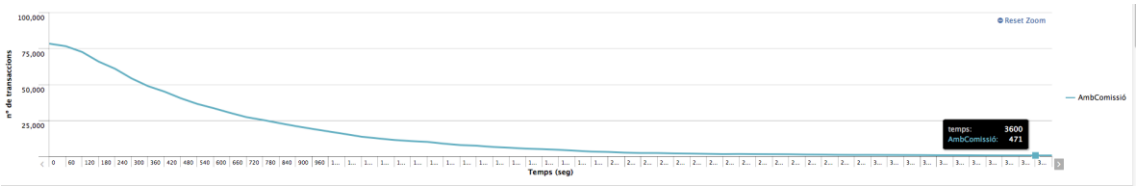
Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" | search fee=0 | bucket temps span=60s | stats count as SenseComissió by temps | sort temps ASC



Imatge 24 Transaccions sense comissió en funció del temps

Com veiem, quasi totes les transaccions mantenen un mateix patró. Tant les operacions amb comissió com les que no tenen comissió es validen en poc temps en termes generals.

Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" | search fee>0 | bucket temps span=60s | stats count as AmbComissió by temps | sort temps ASC



Imatge 25 Transaccions amb comissió en funció del temps

El que es veu és que hi ha una gran quantitat de transaccions que paguen comissions però també aquestes es distribueixen al llarg del temps.

Una de les característiques de les comissions és que no són obligatòries, poden no pagar-se, però el sistema penalitza aquestes transaccions si superen una mida en bits. Aquesta mida es relaciona directament amb el nombre de inputs i de outputs de cada transacció. Quant tenim més inputs/outputs la transacció ocupa més espai al bloc i per tant per al miner serà més difícil tractar aquesta informació. Per tant anem a analitzar si el temps que triga una transacció en incloure's al bloc es relaciona amb el nombre de inputs/outputs de cada transacció.

En les taules següents veiem que la quantitat de inputs més habitual quant no es paga comissió és de 1 en tant que als outputs el més normal es que hagin dos, el de pagament i el canvi que es retorna a qui origina la transacció.

Nº Inputs	Transaccions	Percentatge
1	16716	76.53%
2	2451	11.22%
3	1276	5.84%
4	568	2.60%
5	276	1.26%
6	213	0.97%
7	55	0.25%
8	40	0.18%
144	20	0.09%

Taula 7 inputs més habituals

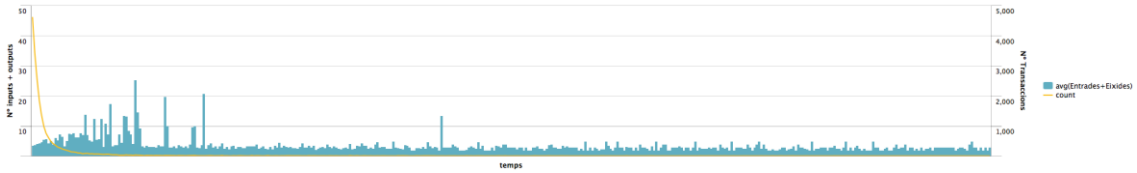
Nº Outputs	Transaccions	Percentatge
2	17103	78.30%
1	4166	19.07%
3	168	0.76%
4	152	0.69%
5	62	0.28%
6	43	0.19%
7	41	0.18%
8	22	0.10%
41	13	0.05%

Taula 8 outputs mes habituals

Potser resulta curiós que el 19,07% de les transaccions tenen només un output. Això és difícil en una transacció comercial perquè, com sabem, els inputs han de ser pel mateix valor complet que ens van arribar. Aquestes transaccions segurament són persones que fan una transferència des d'una adreça bitcoin cap un altra també seua per centralitzar el seus diners en només una adreça en lloc de tindre repartit tots els bitcoins per diferents adreces. De fet si fem el mateix anàlisi amb transaccions que paguen comissió només un 8% tenen un output. Pareix lògic que si volem passar diners d'una adreça nostra a un altra no vulguem pagar comissions.

Si representem les transaccions amb comissió cero en una gràfica indicant el nombre de entrades/eixides en funció de la comissió que han pagat tenim una distribució com aquesta (agrupant el temps en marges de 300 segons):

Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" fee=0 | bucket temps span=300s | eval Entrades+Eixides=(n_outputs + n_inputs) | chart avg(Entrades+Eixides), count by temps

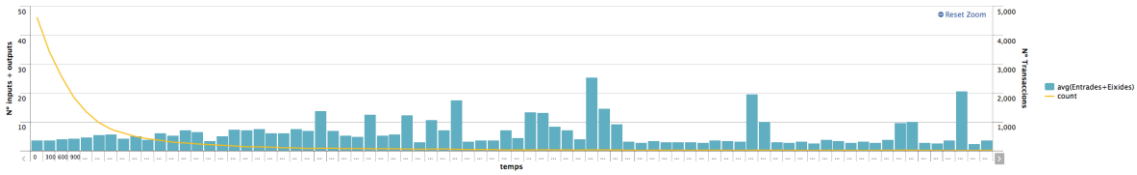


Imatge 26 mitjana de la suma de entrades i eixides en funció del temps en incloure una transacció sense comissió en un block

En aquesta gràfica general, la mitjana de inputs/outputs per a un temps entre 0-300 segons és de 3,66 (amb 4625 transaccions), el que és normal, perquè tindrem habitualment un input i dos outputs com hem vist. El que veiem principalment és que quant més triga una transacció de comissió cero en incloure's a un bloc, conforme anem a la dreta a l'eix de les X, no tenim transaccions amb una mitjana de inputs/outputs elevat sinó que també tenen normalment al voltant de tres inputs/outputs.

El que concloem d'aquesta anàlisi és que pareix que el temps que una transacció que no paga comissions triga en ser inclosa a un bloc té poc a veure amb el nombre de inputs/outputs als que fa referència la transacció. Encara que té poc a veure, sí veiem que les transaccions amb més quantitat de inputs/outputs són allunyades de ser incloses en poc temps, lloc que ocupen les transaccions amb menys entrades (parlant sempre de transaccions que no paguen comissions).

En aquesta gràfica veiem que les transaccions amb més inputs/outputs no són incloses en poc temps i es confirmen més tard.

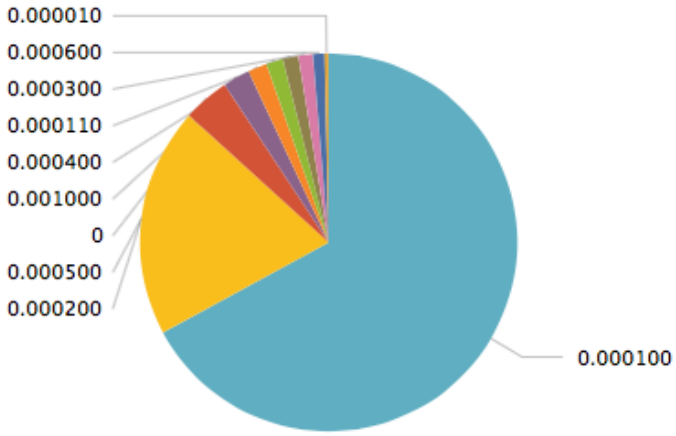


Imatge 27 zoom de detall del nombre de inputs + outputs en funció del temps de ser incloses a un block

Quina és la comissió més comuna en les transaccions?

Si analitzem quins són els valors de comissió més habitual tenim que el top 10 de les comissions més utilitzades en aquesta taula:

Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" | eval Fee=fee/100000000 | top 10 Fee



Imatge 28 Comissions més habituals a les transaccions

Comissió (BTC)	Transaccions	Percentatge
0.0001	630921	64.93%
0.0002	184385	18.97%
0.0005	37540	3.86%
0	21842	2.24%
0.0010	15161	1.56%
0.0004	13724	1.41%
0.00011	12486	1.28%
0.0003	11763	1.21%
0.0006	9137	0.94%
0.00001	3270	0.33%

Taula 9 Comissions més habituals

La comissió més estesa es la de 0,1 mBTC que és la comissió per defecte per a una transacció amb una mida de 1000 bytes segons l'estàndard [20], en tant que una transacció típica te una mida de 500 bytes.

Aquesta és una llista de les wallets més usades (basada en la quantitat de descarregues web)

	Client	downloads	tipus
1	Bitcoin-Qt	1.300.000	full
2	Blockchain	212.000	web
3	Bitcoin Wallet for Android	75.000	mobile
4	Bitcoin by MtGox Mobile	50.000	mobile
5	MultiBit	-	light
6	Electrum	-	light
7	Easywallet.org	-	web
8	BitcoinX	-	-
9	Coinbase	115.000	-
10	Freecoin	-	-

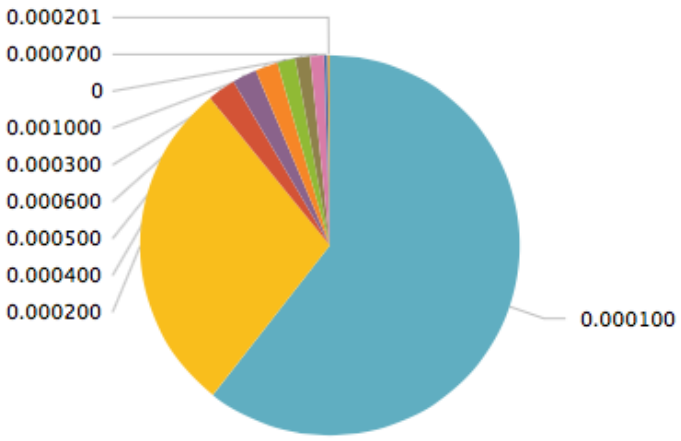
Taula 10 Wallets de bitcoin més utilitzats

https://en.bitcoin.it/wiki/Bitcoin_Ladder#Top_clients

En aquest sentit un dels wallets més utilitzats, el bitcoin-Qt, que en la seua versió 0.9.0 ha reduït a 0.01mBTC la comissió mínima per kilobyte però manté la de 0.1mBTC/Kb durant períodes de gran quantitat de transaccions per assegurar que la transacció s'inclou a un block i és confirmada el més ràpid possible [46].

Els wallets de blockchain.info inclouen una comissió per defecte de 0,5 mBTC si es fa ús de l'API per fer una transacció [47] encara que fent una anàlisi de les transaccions amb origen el mateix blockchain (amb una IP d'origen de loopback si utilitzem l'API de blockchain per obtenir informació de les transaccions) tenim que també utilitza de forma habitual les transaccions de 0.1mBTC

Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" relayed_by=127.0.0.1 | eval Fee=fee/100000000 | top 10 Fee



Imatge 29 Comissions més habituals amb origen blockchain.info

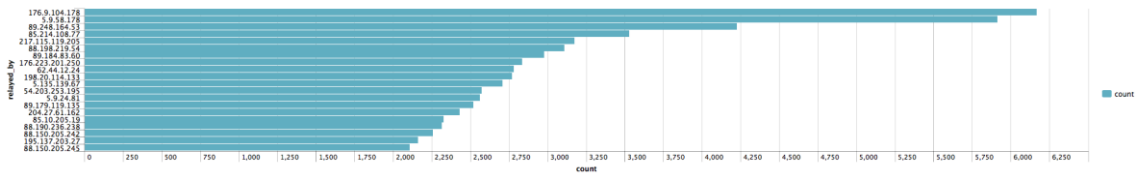
Comissió (BTC)	Transaccions	Percentatge
0.0001	243321	58.80%
0.0002	114633	27.70%
0.0004	9656	2.33%
0.0005	8379	2.02%
0.0006	7714	1.86%
0.0003	6191	1.49%
0.0010	5053	1.22%
0	4713	1.12%
0.0007	1163	0.28%
0.000201	811	0.19%

Taula 11 Comissions habituals a blockchain.info

Hi ha adreces IP amb moltes transaccions retransmeses?

Amb una segona font de dades, que ja hem usat per a revisar les comissions a la pregunta anterior, podem fer una revisió de les adreces IP des de les que arriben les transaccions i obtenir quines són les que més transaccions retransmeten, encara que eliminarem les transaccions de 127.0.0.1 que són les més nombroses perquè lògicament són originades pel sistema on s'originen les dades que hem arreplegat. Podríem dir que aquesta és una llista dels 20 peers de blockchain que més transaccions retransmeten.

Splunk: source="dbmon-dump://Bitcoin/transaccions_ip" relayed_by != 127.0.0.1 | top limit=20 relayed_by



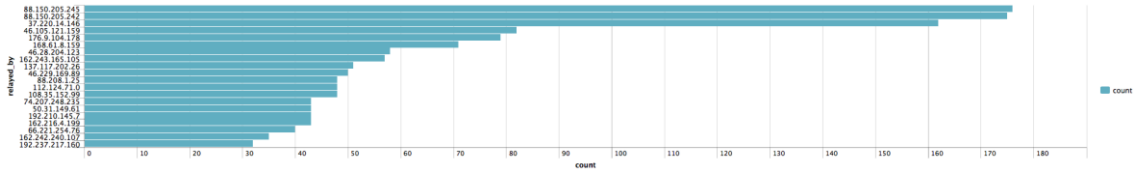
Imatge 30 adreces IP que més transaccions retransmeten

La primera adreça, amb 6170 transaccions, pertany a bitminter.com, un proveïdor dedicat a fer mineria de bitcoins. La segona, 5.9.58.178, correspon a *fast.zaeda.net*, el domini es d'un programador que ha fet contribucions a wallets com bitcoin 0.9.

Darrere d'aquesta adreça hi ha un node bitcoin. Aquesta anàlisi el que ens informa, per dalt de tot, són els peers més actius del node que s'està analitzant.

Però, **qui retransmet més transaccions també retransmet blocs ?** en aquest cas podem fer també una cerca de qui retransmet els blocs.

Splunk: source="dbmon-tail://Bitcoin/blocks"| top limit=20 relayed_by



Imatge 31 adreces IP que més blocks retransmeten

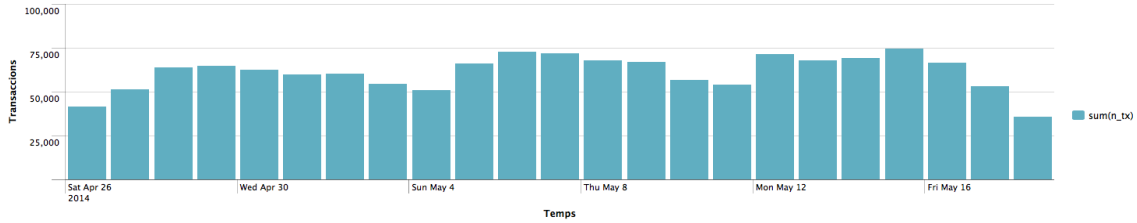
En aquest cas és la IP 88.150.205.245 que correspon amb una de les adreces de uk1.ghash.io (http://totalhash.com/network/ip:88.150.205.245). També la segona IP de la llista correspon a uk.ghash.io. Aquest és un dels pool de mineria de bitcoin més grans que hi ha. Pareix lògic que els pools més grans de miners siguin els que més blocks generen a la xarxa.

Quina és la quantitat de transaccions incloses als blocs durant el temps?

A més de fer estadístiques de les transaccions podem anar a fer estadístiques directament dels blocs que agrupen les transaccions i veure algunes característiques que podem treure analitzant les dades.

Podem veure la quantitat de transaccions que s'inclouen als blocs durant un dia:

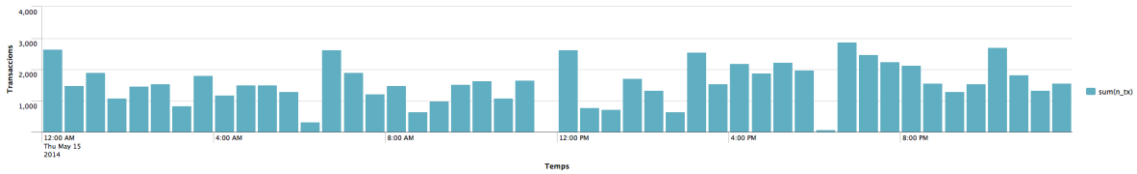
Splunk: source="dbmon-tail://Bitcoin/blocks" | timechart sum(n_tx)



Imatge 32 transaccions totals per dia

Veiem que als caps de setmana hi ha una quantitat de transaccions menor, com possiblement correspondria a les transaccions de negocis normals. Això pareix curiós en tant que els bitcoins són una moneda digital i les transaccions són electròniques, les tendes virtuals i sempre obertes, però els hàbits de la gent continuen reflectint-se inclús a les transaccions virtuals.

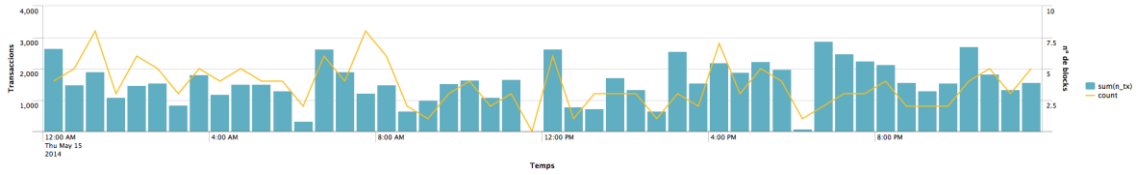
Podem mirar en detall un dia concret (el 15 de maig per exemple):



Imatge 33 detall del nombre de transaccions del dia 15 de maig de 2014

Pareix curiós que hi ha períodes de mitja hora als que quasi no tenim transaccions. Aquests períodes són moments en els que no s'hi han generat blocs i per tant no s'han validat transaccions. Ho veiem millor en aquesta gràfica, igual que l'anterior però amb informació sobre el nombre de blocs generats:

Splunk: source="dbmon-tail://Bitcoin/blocks" | timechart sum(n_tx) count



Imatge 34 nombre de transaccions i nombre de blocks en funció del temps del dia 15 de maig de 2014

Entre les 11:30 GMT (serien les 9:30) i les 12 en la gràfica es marca que no s'hi han generat blocs. És curiós que amb una generació aproximada de un block cada 500 segons tinguem aquests buits de blocs de tan en tant. En aquest cas entre aquests dos blocs hi ha una diferència de mitja hora:

Altura	Hash	Data (GMT)
300841	00000000000000001fc85b6e3869f1149da1c8b6369c2b187a5635432bb830a2	2014-05-15 11:29:38
300842	00000000000000001b80f75bf6517cf0cfc6f0e6d7ba5d1738e06bfc65979a6b	2014-05-15 12:03:10

Taula 12 Blocks consecutius generats amb més de mitja hora de diferència

El que sí que veiem és que després de mitja hora amb pocs blocs hi ha un altra mitja hora en la qual es validen moltes transaccions encara que no necessàriament amb molts blocs (cada bloc pareix tindre moltes més transaccions); com entre les 18:30 i les 19 hores del dia 15 que en el període anterior només hi ha un bloc amb poques transaccions i en la següent mitja hora tenim dos blocs que validen 2870 transaccions.

Aquests períodes (i d'altres) són els que faran que al final existeixi una desviació entre la previsió de creació de blocks i la realitat, el que farà que la dificultat s'ajusti en un futur i es minimitzin aquests espais buits de temps sense blocks.

Conclusions

Ja fa 6 anys que Bitcoin existeix i s'ha convertit en la moneda digital més utilitzada. Durant aquest temps les bases de Bitcoin s'han mantingut molt robustes a nivell tècnic. Utilitzant criptografia com a base per a la seguretat s'ha garantit a nivell tècnic una qualitat que certifica les transaccions i fa el sistema segur inclús a futur. El concepte de cadena de blocks per a mantenir l'històric de les transaccions i evitar la doble despesa és molt bo perquè garanteix la fiabilitat de les transaccions ja realitzades i evita la doble despesa. També és molt interessant el fet que es tingués en compte l'evolució de la tecnologia i la velocitat de computació a l'hora de generar nous bitcoins amb un mecanisme com la dificultat ajustable per tal de mantindre la creació de nous bitcoins amb una proporció controlada, encara que decreixent, al llarg dels anys. A més, aquest mecanisme s'autoajusta en funció del temps que els miners triguen en crear nous blocks, el que fa encara més descentralitzat el sistema, que es regula en aquest punt de forma autònoma. La descentralització és, potser, un dels aspectes de bitcoin per a mi més interessants, juntament amb l'anonimat, en tant que elimina la figura d'un banc central que pugui prendre decisions sobre la moneda en si mateixa, creant un ecosistema de serveis al voltant d'aquesta economia. Però potser aquesta manca d'entitat centralitzada que miri pel futur de la moneda també fa que sigui més feble a atacs de tipus econòmic i menys predictible. L'anonimat de les transaccions des del meu punt de vista potser faci que existeixin moviments de diners "en negre" i que sigui molt difícil el seu seguiment, encara que no impossible, en tant que la xarxa coneix l'adreça IP de qui realitza una transacció i es podria fer un seguiment de l'origen d'una transacció. La part tècnica de la descentralització corresponent al funcionament en forma de xarxa peer-to-peer fa que el sistema sigui molt estable front a caigudes de nodes, creant una xarxa molt gran de nodes que sostenen el sistema pel mateix fet de estar-hi connectats.

A nivell de seguretat de les transaccions i de robustesa del sistema, és un entorn ben pensat i que funciona. Com qualsevol sistema monetari es basa en la confiança dels seus usuaris, igual que la moneda dels diferents països, i és aquesta confiança i l'ús efectiu de realitzar transaccions amb la moneda, la qui atorga validesa a la moneda. Avui dia hi ha una amplia acceptació de l'ús dels bitcoins i poc a poc hi ha més comerços que accepten aquesta moneda. Un punt en contra és la fluctuació que tenen els bitcoins en referència al tipus de canvi amb altres monedes. Això fa que en alguns moments sigui difícil entrar en l'economia de bitcoin i que els preus de les tendes online siguin a voltes molt diferents en bitcoins o en altres monedes com l'euro. Això fa que les tendes hagin d'actualitzar els seus preus amb prou freqüència si volen tindre preus competitius en bitcoins (o no perdre diners, que també podria ser). També fa que les comissions que es paguin s'hagin d'ajustar al valor econòmic de la moneda. Si finalment es produeix inflació dels preus o deflació, el valor de la comissió potser sigui massa elevat o massa insignificant i per tant poc interessant per als miners.

El sistema monetari té data de caducitat en quant a creació de moneda (21 milions de BTC). Com hem vist, el fet que cada volta sigui menys interessant crear blocks

per la recompensa de crear-ho farà a la llarga que es paguin més comissions per cada transacció. Això penso que a futur farà que la gent tracti de no pagar moltes comissions i potser un intercanvi acabi allargant-se molt en el temps, una experiència contraria al que es tracta d'obtenir amb les transaccions electròniques, l'immediatesa.

Pel que fa a l'anàlisi de les dades, pareix que el nivell teòric del funcionament de bitcoin es reflecteix al funcionament normal encara que no pareix que hi hagi una relació directa entre el pagament de les comissions i el temps que una transacció triga en incloure's als blocks. Pareix, en la teoria, que una transacció que pagui més comissió serà molt més interessant per als miners i tractaran de incloure-la a un dels seus blocks però per altra banda pareix que hi ha transaccions amb comissió que triguen molt en ser incloses. Al mateix temps, podríem pensar que les transaccions que no paguen comissions serien de les últimes en ser incloses a un block, però tampoc pareix que sigui així. El temps que una transacció triga en ser oficialitzada normalment és de dos blocks, però després hi ha algunes que triguen molt més, encara que són prou aïllades. El temps de propagació a la xarxa peer-to-peer pareix que tampoc té res a veure perquè aquesta propagació de la transacció és pràcticament immediata. Hi ha miners o pools de mineria que també donen servei de wallets al núvol i són, per tant, també origen de moltes transaccions. En aquest sentit pareix una pràctica possible que els miners d'aquest servei de wallet al núvol tractin d'incloure abans les seues transaccions que les de la resta de xarxa, d'aquesta forma també incentiven que els usuaris tinguin wallets al seu sistema i facin les transaccions amb ells. Hem trobat transaccions que han trigat fins a 7 dies en ser incloses a un block pagant comissió. Aquest, encara que aïllat, és un comportament realment difícil d'entendre quant existeix un mecanisme que augmenta el nivell de prioritat de transaccions conforme són més antigues. Això, amb els resultats de les dades que hem obtingut, em fa pensar que la implementació de la prioritat de les transaccions no és del tot fiable o, al menys, aquesta prioritat és tractada de forma particular pels miners, que apliquen el seu propi algorisme de priorització, segurament tenint en compte el que marca el protocol de bitcoin però afegint les seues pròpies prioritats, potser prioritzar transaccions amb wallets específics o amb origen concret com ara els seus propis sistemes de pagament. Si aquesta és l'evolució del sistema a l'hora de prioritzar els pagaments, els usuaris tractaran de fer ús dels serveis de empreses que ofereixen gestionar els diners de tercers a canvi de fer més ràpides les seues transaccions, el que farà que molts dels diners quedin en mans de poques empreses que seran les que tinguin major capacitat de computació. Afortunadament com hem vist també hi ha límits en quant a acaparar capacitat de computació, al menys autoimposats, perquè si una organització arribés a controlar més del 50% de la capacitat total de computació podria alterar les transaccions, penalitzant transaccions d'altres o desfent les transaccions que ell mateix fa. Aquest és un punt important a tenir en compte conforme els pools de miners creixen i espere que l'autoregulació de la xarxa eviti a futur.

Una part interessant d'analitzar i que no hem fet en aquesta memòria es analitzar la quantitat de bitcoins que s'han perdut al llarg del temps. Potser resulti prou complex fer una anàlisi d'aquest tipus perquè no podem saber a priori qui ha perdut les claus privades d'una adreça. L'estudi portaria a analitzar la quantitat de

bitcoins creats (fàcilment calculable en funció dels blocs creats i del pagament de la recompensa per la generació del block) i tractar d'obtenir les adreces que fa molt de temps que no tenen moviments però amb valor positiu de bitcoins. Aquesta anàlisi no podria ser mai 100% fiable però ens donaria una idea de la quantitat de bitcoins perduts i que no són recuperables (destrucció de moneda). Finalment el sistema tindrà un total de 21 milions de bitcoins menys els que s'hagin perdut o no siguin accessibles (pèrdua dels wallets per no fer còpia de seguretat, pèrdua de les claus privades .....). Algunes dades apunten a que en 2013 un 35% dels bitcoins existents no s'havien gastat des de 2011, el que podrien ser bitcoins perduts o simplement gent estalviant diners .... [48]

Bibliografía

[1] "Wikipedia: Moneda," 2014. [Online]. Available: http://es.wikipedia.org/wiki/Moneda. [Accessed 20 Abril 2014].

[2] "Viquipèdia: Seca," 2014. [Online]. Available: http://ca.wikipedia.org/wiki/Seca. [Accessed 20 Abril 2014].

[3] "Wikipedia: Papel moneda," 2014. [Online]. Available: http://es.wikipedia.org/wiki/Papel_moneda. [Accessed 20 Abril 2014].

[4] "Wikipedia: Patron oro," 2014. [Online]. Available: http://es.wikipedia.org/wiki/Patr%C3%B3n_oro. [Accessed 20 Abril 2014].

[5] "Wikipedia: Dinero fiduciario," 2014. [Online]. Available: http://es.wikipedia.org/wiki/Dinero_fiduciario. [Accessed 20 Abril 2014].

[6] "Wikipedia: Dinero electrónico," 2014. [Online]. Available: http://es.wikipedia.org/wiki/Dinero_electr%C3%B3nico. [Accessed 20 Abril 2014].

[7] "Coinmarket CAP," 2014. [Online]. Available: https://http//coinmarketcap.com/. [Accessed Maig 2014].

[8] "Ripple: Guide," 2014. [Online]. Available: https://ripple.com/guide/. [Accessed 20 Abril 2014].

[9] "Wikipedia: Litecoin," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Litecoin. [Accessed 20 Abril 2014].

[10] "Wikipedia: Peercoin," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Peercoin. [Accessed 20 Abril 2014].

[11] "Wikipedia: Dogecoin," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Dogecoin. [Accessed 21 Abril 2014].

[12] "Wikipedia: Mastercoin," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Mastercoin. [Accessed 20 Abril 2014].

[13] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf. [Accessed 12 Abril 2014].

[14] "Wikipedia: Elliptic Curve Digital Signature Algorithm," 2014. [Online]. Available: http://en.wikipedia.org/wiki/Elliptic_Curve_DSA. [Accessed 12 Abril 2014].

[15] "Bitcoin Wiki: Technical background of version 1 Bitcoin addresses," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Technical_background_of_Bitcoin_addresses. [Accessed 12 Abril 2014].

[16] "Coindesk: How to Make a Paper Wallet for Bitcoin," 2014. [Online]. Available: http://www.coindesk.com/information/paper-wallet-tutorial/. [Accessed 13 Abril 2014].

[17] "Bitcoin Wiki: Transactions," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Transactions. [Accessed 13 Abril 2014].

[18] "Bitcoin Wiki: Script," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Script. [Accessed 13 Abril 2014].

[19] "Bitcoin Wiki: Units," 2014, [Online]. Available:

https://en.bitcoin.it/wiki/Units. [Accessed 14 Abril 2014].

[20] "Bitcoin Wiki: Transaction fees," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Transaction_fees. [Accessed 14 Abril 2014].

[21] "Bitcoin Transaction Fees Explained," 12 Febrer 2014. [Online]. Available: http://bitcoinfees.com/. [Accessed 14 Abril 2014].

[22] "Bitcoin Wiki: Blocks," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Blocks. [Accessed 18 Abril 2014].

[23] "Bitcoin Wiki: Block hashing algorithm," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Block_hashing_algorithm. [Accessed 16 Abril 2014].

[24] "Wikipedia: Cryptographic nonce," 2013. [Online]. Available: http://en.wikipedia.org/wiki/Cryptographic_nonce. [Accessed 16 Abril 2014].

[25] "Bitcoin Wiki: Target," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Target. [Accessed 16 Abril 2014].

[26] "Bitcoin Wiki: Difficulty," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Difficulty. [Accessed 16 Abril 2014].

[27] "Bitcoin Wiki: Block Chain," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Block_chain. [Accessed 18 Abril 2014].

[28] "Bitcoin Wiki: Scalability," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Scalability. [Accessed 18 Abril 2014].

[29] "Bitcoin Wiki: Proof of work," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Proof_of_work. [Accessed 18 Abril 2014].

[30] "Bitcoin Wiki: Mining," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Mining. [Accessed 18 Abril 2014].

[31] "Bitcoin Wiki: Pooled mining," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Pooled_mining. [Accessed 18 Abril 2014].

[32] "Bitcoin Wiki: Comparison of mining pools," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Comparison_of_mining_pools. [Accessed 18 Abril 2014].

[33] "Bitcoin wiki: Double-spending," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Double-spending. [Accessed 18 Abril 2014].

[34] "Bitcoin Wiki: Weaknesses," 2014. [Online]. Available: https://en.bitcoin.it/wiki/Weaknesses. [Accessed 18 Abril 2014].

[35] "Timejacking Bitcoin," 25 Maig 2011. [Online]. Available: http://culubas.blogspot.com.es/2011/05/timejacking-bitcoin_802.html. [Accessed 18 Abril 2014].

[36] "bitcoin.org: heartbleed," 11 Abril 2014. [Online]. Available: https://bitcoin.org/en/alert/2014-04-11-heartbleed. [Accessed 11 Abril 2014].

[37] "MtGox.com," 28 Febrer 2014. [Online]. Available: https://www.mtgox.com/img/pdf/20140228-announcement_eng.pdf. [Accessed 19 Abril 2014].

[38] "The Telegraph," 26 Febrer 2014. [Online]. Available: http://www.telegraph.co.uk/technology/news/10662749/MtGox-and-Bitcoin-where-has-251m-gone.html. [Accessed 19 Abril 2014].

[39] “MtGox,” 16 Abril 2014. [Online]. Available: https://www.mtgox.com/img/pdf/20140416_002_announce_en.pdf. [Accessed 19 Abril 2014].

[40] “Crypto Coins News,” 27 Març 2014. [Online]. Available: http://www.cryptocoinsnews.com/news/malleability-bankrupt-mt-gox/2014/03/27. [Accessed 19 Abril 2014].

[41] “Cryptocoins News,” 13 Febrer 2014. [Online]. Available: http://www.cryptocoinsnews.com/news/silk-road-2-0-has-been-hacked-and-at-least-4673-btc-stolen-operator-says-centralized-escrow-system-cant-ever-work/2014/02/13. [Accessed 19 Abril 2014].

[42] “Bitcoin Wiki: Trade,” 2014. [Online]. Available: https://en.bitcoin.it/wiki/Trade. [Accessed 19 Abril 2014].

[43] “Wikipedia: Mercado de divisas,” 2014. [Online]. Available: http://es.wikipedia.org/wiki/Mercado_de_divisas. [Accessed 19 Abril 2014].

[44] “Wikipedia: History of Bitcoin - Prices and value history,” 2014. [Online]. Available: http://en.wikipedia.org/wiki/History_of_Bitcoin#Prices_and_value_history. [Accessed 19 Abril 2014].

[45] “Wikipedia: JSON,” 2014. [Online]. Available: http://en.wikipedia.org/wiki/JSON. [Accessed 1 Maig 2014].

[46] Bitcoin, “Bitcoin 0.9.0,” 2014. [Online]. Available: https://bitcoin.org/bin/0.9.0/README.txt. [Accessed Maig].

[47] “blockchain.info Wallet API,” 2014. [Online]. Available: https://blockchain.info/es/api/blockchain_wallet_api. [Accessed Maig 2014].

[48] L. Orsini, «What Happens To Lost Bitcoins?,» 13 Gener 2014. [En línia]. Available: http://readwrite.com/2014/01/13/what-happens-to-lost-bitcoins#awesm=~oFXePLXoqmNmog. [Últim accés: 1 Juny 2014].

Aplicatius utilitzats:

http://www.splunk.com
http://www.apache.org
http://dev.mysql.com/downloads/mysql/
http://dev.mysql.com/downloads/connector/python/
http://www.phpmyadmin.net
http://www.sublimetext.com

La base de dades utilitzada per als anàlisis es pot descarregar d'aquesta URL:
https://www.dropbox.com/s/xbjtsg1wbwxm5se/bitcoin.sql.zip

Són 99 MB de fitxer SQL comprimit que regeneraria les dades utilitzades.

Annex 1 – Script en Python

L'script utilitzat recull a partir de l'últim block generat 100 blocks enrere de la cadena principal de blocks de bitcoin. Si s'especifica un hash o índex al executar l'script començarà a obtenir les dades a partir d'aquest block.

Als comentaris del codi no s'han usat signes de puntuació perquè a l'hora de executar el codi pot donar errors per la codificació del arxiu.

ús de l'script:

Si volem iniciar amb l'últim block generat utilitzem sense paràmetres

>python bitcoin.py

si volem començar a partir d'un hash específic indiquem el valor del hash

>python bitcoin.py <hash>

El codi es el següent, encara que també es pot descarregar d'aquesta URL: https://www.dropbox.com/s/ivzpoo43rtxd7t5/bitcoin.py

```
#!/usr/bin/python
import urllib, json, time, sys
import mysql.connector

def obtindre_block(block_index):

    # Guardarem els valors en llistes per despres poder utilitzar els valors.
    in_tx=[]
    out_tx=[]
    fee=[]
    temps=[]
    conndb = mysql.connector.connect(user='bitcoin', database='bitcoin') #fem
la connexio amb la DB
    cursor = conndb.cursor() # fem un cursor per a insertar les dades a la DB

    data = json.loads(urllib.urlopen("http://blockchain.info/rawblock/" +
block_index).read())  # Descarreguem el bloc

    # Obtenim la data del block en format llegible

    block_date = time.strftime("%Y/%m/%d %H:%M:%S",
time.localtime(int(data['time'])))
    block_received_time = time.strftime("%Y/%m/%d %H:%M:%S",
time.localtime(int(data['received_time'])))

    for t in range(len(data["tx"])):    # recorrem el bloc, la variable t
recorre cada trasaccio
        in_tx_temp = 0  # inicialitzem el sumatori del valor dels inputs de la
transaccio t
        out_tx_temp = 0 # inicialitzem el sumatori del valor dels outputs de
la transaccio t
        fee_temp = 0
        temps_temp = 0
        i=0 # variable per a recorrer els inputs
        j=0 # variable per a recorrer els outputs
        for i in range(len(data['tx'][t]['inputs'])):
            if(t!=0):
                in_tx_temp=in_tx_temp +
data['tx'][t]['inputs'][i]['prev_out']['value'] # sumem al valor de input el
```

```
nou valor per a cada input
        in_tx.append(in_tx_temp)
        for j in range(len(data['tx'][t]['out'])):
            out_tx_temp = out_tx_temp + data['tx'][t]['out'][j]['value']  #
sumem els outputs
        out_tx.append(out_tx_temp)
         if(t==0):
            fee_temp = out_tx_temp
        else:
            fee_temp = in_tx_temp - out_tx_temp
        fee.append(fee_temp)
        temps_temp = data['time'] - data['tx'][t]['time']
        temps.append(temps_temp) # Temps en segons que triga la transaccio en
fer-se efectiva (temps de bloc - temps de tx)

        tx_date = time.strftime("%Y/%m/%d %H:%M:%S",
time.localtime(int(data['tx'][t]['time'])))

        # Construim les dades que introduim a la DB

        add_tx = ("INSERT INTO transaccions "
                "(block_index, block_date, altura, hash, tx_hash, tx_index,
relayed_by, n_inputs, input, n_outputs, output, tx_date, fee, temps) "
                "VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s,
%s)")

        data_tx = (data['block_index'], block_date, data['height'],
data['hash'], data['tx'][t]['hash'], t, data['tx'][t]['relayed_by'],
len(data['tx'][t]['inputs']), in_tx[t], len(data['tx'][t]['out']), out_tx[t],
tx_date, fee[t], temps[t])
        cursor.execute(add_tx, data_tx)

    # Una volta hem fet totes les tx del block enviem les dades a la DB i
tamquem el cursor i la connexio

    add_block = ("INSERT INTO blocks "
            "(block_index, block_date, block_received_time, height, hash,
bits, n_tx, fee, size, main_chain, relayed_by) "
            "VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s, %s)")

    data_block = (data['block_index'], block_date, block_received_time,
data['height'], data['hash'], data['bits'], data['n_tx'], data['fee'],
data['size'], data['main_chain'], data['relayed_by'])
    cursor.execute(add_block, data_block)

    conndb.commit()
    cursor.close()
    conndb.close()
    return data['prev_block']  # Tornem el hash del bloc anterior al actual

# Cos principal del programa

if (len(sys.argv)) < 2:
    latest_block =
json.loads(urllib.urlopen("http://blockchain.info/latestblock").read())
    block_index=str(latest_block["block_index"]) # Obtenim el index del ultim
bloc generat
else:
    if (len(sys.argv[1])) != 64:
        print "El hash es incorrecte"
        exit()
    else:
        block_index = sys.argv[1]
print "Block_index \t Altura \t Hash \t Tx_Index \t input \t output \t fee \t
temps"
z = 0

if
```

```
while z < 100: #obtenim els 100 primers blocks de la cadena
    block_index = obtindre_block(block_index)
    z += 1
```

Isidro Pastor Jordà
València , 2014