Bitcoin :

A Peer-to-Peer Electronic Cash System

Un sistèma de pagament electronic par-a-par

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Traduccion occitane de bitcoin.org/bitcoin.pdf
per Joan-Lucì Labòrda

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone

Resumit . Ua version d'un sistèma de pagament purament par-a-par que permeteré pagaments en linha dirèctes d'ua partida a l'aute shens passar per ua institucion financèra. Las signaturas digitaus que horneishen ua partida de la solucion, mes los principaus beneficis que son perduts si un tèrç de hidança hè enqüèra besonh entà evitar los dobles-pagaments. Que prepausam ua solucion au problèma deu doble-despensa en utilizant un hialat par-a-par. Lo hialat horodate las transaccions en los hachant en ua cadena que contunha de pròvas-de-tribalh, formant un enregistrament de dadas qui pòt pas estar cambiat shens aver a tornar har la pròva-de-tribalh. La cadena mei longa non solament sèrv de pròva per testimoniatge de la sequéncia deus eveniments, mes que pròva qu'ei gessida deu mei gran grop de poténcia CPU. Tant que la majoritat de la poténcia CPU ei controlada per nods non participant a ua ataca deu hialat, qu'engendraràn la mei longa cadena e que suberpassaràn los atacants . Lo hialat en eth medish qu'exigeish ua estructura minimau . Los messatges que son difusats au mélher e los nods que pòden quitar e rejúnher lo hialat au lor grat, en acceptant la mei longa cadena de pròva-de-tribalh creada en la lor abséncia.

1. **Introduction**

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

1. **Introduccion**

Lo comèrci sus Internet que n'ei vienut a pausar quasi exclusivament sus las institucions financèras agint com tèrç de hidança entà tractar los pagaments electronics. Mentre que lo sistèma fonciona pro plan entà la màger part de las transaccions, que sòfre de flaquèrs inerents au modèle de hidança . Las transaccions totaument irreversiblas ne son pas reaument possiblas, pr'amor que las institucions financèras pòden pas evitar los conflictes de mediacion. La còst de la mediacion qu'aumenta los còsts de transaccion, en limitant lo montant minimum de la transaccion e copant atau la possibilitat de transaccions correntas a petit montant. En mei, qu'i a un còst mei important dens la pèrta de la capacitat a har pagaments irreversibles entaus servicis irreversibles. Dab la possibilitat de reversibilitat, la necessitat de la hidança que s'espandeish . Los comerçants que's deven mesfisar deus lors clients, e los avejar en los demandant mei d'informacion dont aurén pas besonh en procedint autament. Un certan percentatge de frauda qu'ei acceptat com inevitable. Aqueths còsts e incertituds dens los pagaments que pòden estar evitats per la preséncia e l'argent fisics, mes nat mecanisme n'existeish pas entà har pagaments per ua canau de comunicacion shens un tèrç de hidança.

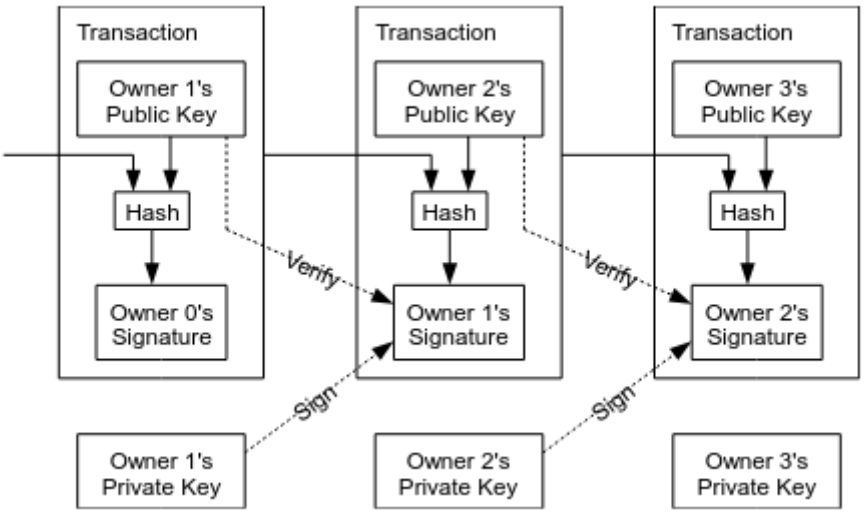
Lo besonh qu'ei d'aver un sistèma de pagament electronic basat sus ua pròva criptografica en plaça de la hidança, permetent a duas partidas volontàrias de realizar enter eras de las transaccions shens lo besonh d'un tèrç de hidança. transaccions calculatoirement incomòdas a invertir que protegirén los venedors de la frauda, e deus mecanismes abituaus de depaus que poirén estar aisidament implementats entà protegir los crompadors. Dens aqueth papèr, que prepausam ua solucion au problèma de la dobla-despensa en utilizant un servidor de horodatage distribuït par-a-par entà engendrar peu calcul la pròva de la cronologia de las transaccions. Lo sistèma qu'ei segur tant que los nods aunèstes contròlan collectivament mei de poténcia CPU qui la de cadun deus grops de nods d'atacants cooperants.

2. **Transactions**

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

2. **Transaccions**

Que defineishem ua pèça (de moneda) electronica com ua cadena de signaturas electronicas. Cada proprietari transferiment la pèça au seguent en signant lo talhucatge, de la transaccion precedenta, de la clau publica deu proprietari vienent e en ajustant tot aquerò a la fin de la pèça. Un beneficiari que pòt verificar las signaturas entà verificar la cadena de proprietat.



The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.

Lo problèma evidentament qu'ei que lo beneficiari pòt pas verificar qu'un deus proprietaris n'a pas despensat dus còps la pèça. Ua solucion comuna qu'ei d'introdusir ua autoritat de hidança, o emetedor de moneda, qui verifica cada transaccion tà çò de la dobla-despensa. Après cada transaccion, la pèça que deu estar renviada a l'emetedor de moneda entà eméter ua pèça navèra, e solas las pèças emetudas per l'emetedor que son famosas non despensadas dus còps. Lo problèma dab aquera solucion qu'ei que lo destin de tot lo sistèma monetari depend de la societat qui emet la moneda, dab cada transaccion davant passar per era, juste com ua banca.

We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

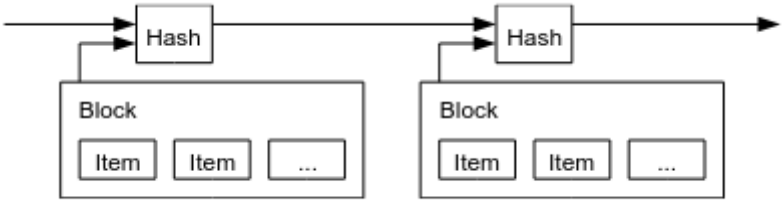
Qu'avem besonh d'un mejan entau beneficiari de saber que los precedents proprietaris an pas signat de transaccions precedentas. Entà las nostas fins, la transaccion efectuada lo mei de d'òra qu'ei la qui compta, atau que prestam pas atencion a las temptativas seguentas de dobla-despensa. Lo sol mejan entà confirmar l'abséncia d'ua transaccion qu'ei d'estar au corrent de totas las transaccions. Dens lo modèle d'un emetedor centrau de moneda, aqueste qu'èra au corrent de totas las transaccions e que decidiva qui arribava en prumèr. Entà complir parièra tasca shens un tèrç de hidança, las transaccions que deven estar anonciadas publicament [1], e qu'avem besonh d'un sistèma entaus participants entà s'acordar sus ua istòria unica de l'ordi dens lo quau estón recebudas. Lo beneficiari qu'a besonh de la pròva qui au moment de cada transaccion, la majoritat deus nods qu'èra d'acòrd qui èra la prumèra recebuda.

3. **Timestamp Server**

The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

3. **Servidor de horodatage**

La solucion qui prepausam que comence dab un servidor de horodatage . Un servidor de horodatage que fonciona en prenent la mèrca numerica d'un blòc d'items a horodater e a la publicar largament, tau com dens un jornau o un fòrum sus Internet [2-5]. Lo horodate que pròva que las dadas an devut existir suu pic deu horodatage, evidentament, entà poder obtiéner la lor mèrca numerica. Cada horodate qu'includeish lo horodate precedenta dens la soa mèrca, formant ua cadena, dab cada navèra horodate ahortint aqueras la precedint.



4. **Proof-of-Work**

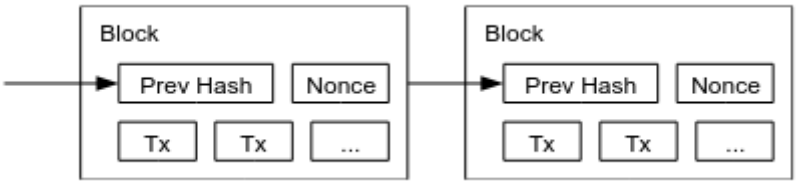
To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-
of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.

4. **Pròva-de-tribalh**

Entà implementar un servidor de horodatage distribuït en par-a-par, qu'avem besonh d'ua pròva-de-tribalh similar a la d'Adam Back “Hashcash” [6], meilèu que d'un jornau o de publicacion sus un fòrum Internet. La pròva-de-tribalh qu'implica la recèrca d'ua valor qui un còp hachat, tau com dab lo SHA-256, que balha
ua mèrca numerica començant per un nombre balhat de bits a zèro. Lo tribalh mejan demandat qu'ei exponenciau en foncion deu nombre de bits a zèro exigits e que pòt estar verificat en executant un talhucatge unic.

Entau noste hialat de horodatage, qu'implementam la pròva-de-tribalh per incrémentation d'ua valor d'ajustament dens lo blòc dinc a trobar ua valor qui balha ua mèrca dab lo nombre de zèros requerits. Un còp qui la carga CPU ei estada despensada entà contentar la pròva-de-tribalh, lo blòc ne pòt pas mei estar cambiat shens tornar har lo tribalh. Per'mor que los blòcs son chaînés après lo blòc considerat, lo tribalh entà cambiar lo blòc que deuré includir de tornar har tots los blòcs posteriors.



The proof-of-work also solves the problem of determining representation in majority decision
making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority
decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.

To compensate for increasing hardware speed and varying interest in running nodes over time,
the proof-of-work difficulty is determined by a moving average targeting an average number of
blocks per hour. If they're generated too fast, the difficulty increases

La pròva-de-tribalh que resòlv tanben lo problèma de la definicion deu procediment de decision majoritària. Si la majoritat èra basada sus ua-adreça-IP-un-vòte, que poiré estar fraudada per quau que sia capabla d'atribuïr hèra d'IPs. La pròva-de-tribalh qu'ei per esséncia ua-CPU-un-vòte. La decision majoritària qu'ei representada per la cadena mei longa, qui a la mei grana pròva-de-tribalh investida. Si ua majoritat de la poténcia CPU ei controlada per nods aunèstes, la cadena aunèsta que creisherà la mei lèu e que despassarà totas autas cadenas en competicion . Entà modificar un blòc passat, un atacant qu'auré a tornar har la pròva-de-tribalh deu blòc e de tots los blòcs après eth, e ad aqueth moment atrapar e suberpassar lo tribalh deus nods aunèstes. Que mostraram mei tard que la probabilitat de sostien d'un atacant mei lent diminuesca dab l'ajustèr deus blòcs subsequents.

Entà compensar l'aumentacion de la velocitat deu materiau e modificar l'interès de l'usatge deus nods au briu deu temps, la dificultat de la pròva-de-tribalh qu'ei determinada per ua mejana mobila ciblant un nombre mejan de blòcs calculats per òra. Si son engendrats tròp lèu, la dificultat qu'aumenta

5. **Network**

The steps to run the network are as follows:

1) New transactions are broadcast to all nodes.
2) Each node collects new transactions into a block.
3) Each node works on finding a difficult proof-of-work for its block.
4) When a node finds a proof-of-work, it broadcasts the block to all nodes.
5) Nodes accept the block only if all transactions in it are valid and not already spent.
6) Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some
nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proof-of-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one

New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

5. **Hialat**

Las estapas entà har foncionar lo hialat que son com segueish :

1) Las transaccions navèras que son difusadas a tots los nods .
2) Cada nod qu'amassa las transaccions navèras dens un blòc .
3) Cada nod que tribalha entà trobar ua pròva-de-tribalh dificile entau son blòc.
4) Quan un nod tròba ua pròva-de-tribalh, que difusa lo blòc a tots los nods .
5) Los nods qu'accèptan lo blòc solament si totas las transaccions son validas e pas dejà despensadas.
6) Los nods qu'exprimeishen la lor acceptacion deu blòc en tribalhant a crear lo blòc vienent de la cadena, en utilizant la mèrca numerica deu blòc acceptat com la mèrca precedenta.

Los nods que considèran tostemps la cadena mei longa com la cadena valida e que contunhan a tribalhar entà l'espandir. Si dus nods difusan duas versions diferentas deu blòc vienent simultanèament, los autes nods que pòden recéber l'ua o l'aute en prumèr. Dens aqueth cas, que tribalhan sus la prumèra qui an recebuda, mes que sauvan l'auta branca se per cas vairé mei longa. Lo ligam que serà romput quan la pròva vienenta-de-tribalh ei trobada e ua branca que vad mei longa ; los nods qui èran a tribalhar sus las autas brancas que commutaràn alavetz sus la mei longa.

Las difusions de las transaccions navèras n'an pas besonh d'aténher necessàriament tots los nods. Tant qu'atenhen hèra nods, que seràn integradas dens un blòc abans longtemps. Las difusions de blòcs que son tanben tolerantas a las pèrtas de messatges. Si un nod recep pas un blòc, qu'ac demandarà quan recébia lo blòc vienent e que realizarà que'u ne manque un.

6. **Incentive**

By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.

The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.

The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth

6. **Prima de resultat**

Per convencion, la prumèra transaccion dens un blòc qu'ei ua transaccion especiau qui comença per ua navèra pèça detienuda peu creator deu blòc. Aquò qu'ajusta ua incitacion entaus nods a suportar lo hialat, e que horneish un mejan iniciau de méter pèças en circulacion puishqu'i a pas autoritat centrau d'emission de moneda entad ac har. L'ajustèr estable d'un montant constant de navèras pèças qu'ei analògue aus cercaires d'aur despensant de las ressorsas entà ajustar aur en circulacion. Dens lo noste cas, que s'ageish de temps CPU e d'electricitat qui son despensats.

La prima de resultat que pòt tanben estar finançada per despensas de transaccion. Si la valor sortida d'ua transaccion ei inferiora a la soa valor d'entrada, la diferéncia que constitueish las despensas de transaccion qui son ajustats a la prima de resultat deu blòc contienent la transaccion. Un còp metut circulacion un nombre predeterminat de pèças, la prima de resultat que's pòt convertir totaument en fresc de transaccion e estar totaument non inflacionista.

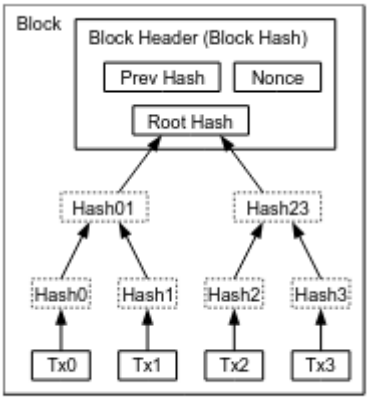
La prima de resultat que pòt ajudar a encoratjar los nods a demorar aunèstes. Si un atacant cobés èra capable d'amassar mei de poténcia CPU qui los nods aunèstes, qu'auré a causir enter escrocar las gents en recuperant fraudulosament los sons pagaments, o, engendrar pèças navèras. Que deuré trobar mei profieitós entà jogar dens las règlas, aquestas ac favorizant en l'auherint mei de navèras pèças qui tota la rèsta deu monde amassat, qui de sapar lo sistèma e la validitat de la soa pròpia fortuna.

7. **Reclaiming Disk Space**

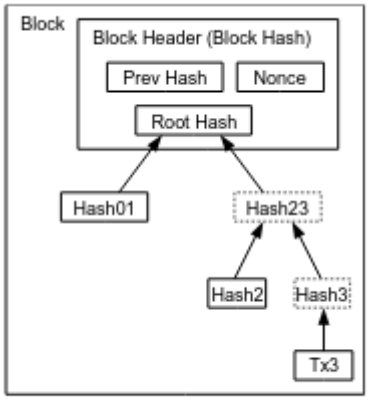
Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block's hash,
transactions are hashed in a Merkle Tree [7][2][5], with only the root included in the block's hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.

7. **Demanda d'espaci disc**

Un còp qui la darrèra transaccion d'ua pèça ei enterrada au devath de pro de blòcs, las transaccions de despensas la precedint que pòden estar getadas entà sauvar espaci disc. Entà facilitar aquò shens copar la mèrca numerica deu blòc, las transaccions que son hachadas dens un arbo de Merkel [7][2][5], dab solament la rasic inclusa dens la mèrca numerica deu blòc. Los ancians blòcs que pòden alavetz estar compactats per ronhatge de las brancas de l'arbo. Las mèrcas interioras de l'arbo n'an pas besonh d'estar estocadas.



Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory

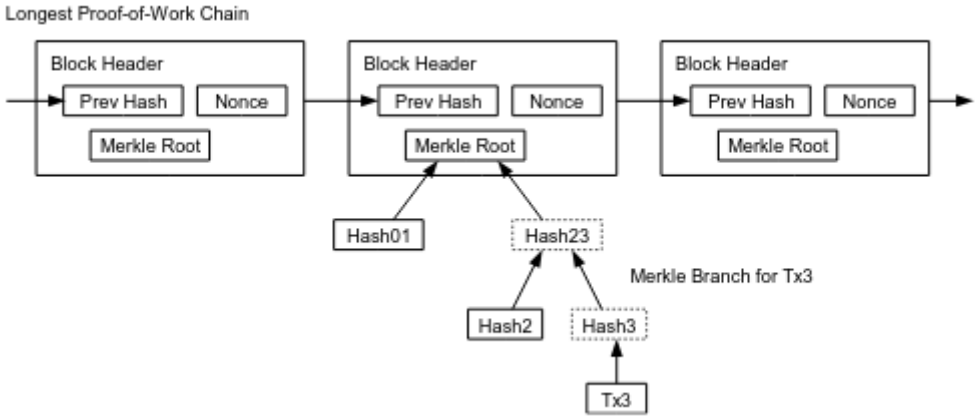
Un qu'encaborreish de blòc shens transaccion que deuré estar a l'entorn d'80 octets. Si supausam los blòcs engendrats cada dètz minutas, 80 octets * 6 * 24 * 365 = 4,2 MOctets per an. Dab los ordenadors tipicament venuts dab 2 GOctets de RAM en 2008, e la lei de Moore predisent ua creishença correnta de 1,2 GOctets per an, l'estocatge ne deuré pas estar un problèma quan los encaborriràs de blòcs que deven estar gardats en memòria.

8. **Simplified Payment Verification**

It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's timestamped in. He can't check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.

8. **Verificacion de pagament simplificada**

Qu'ei possibla de verificar pagaments shens har foncionar un nod complet deu hialat. Un utilizator qu'a solament besonh de gardar ua copia de las qu'encaborreishes de blòc de la mei longa cadena assegurada per la pròva-de-tribalh, qui pòt obtiéner en requêtant los nods deu hialat dinc a que sia convençut qu'a la mei longa cadena e qu'obtiénia la branca de Merkel ligant la transaccion au blòc lo horodatant. Ne pòt pas verificar la transaccion entà eth medish, mes en la ligant a ua plaça dens la cadena, que pòt véder que lo hialat l'a acceptada, e los blòcs ajustats après qu'ac confirmen



As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker's fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user's software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification

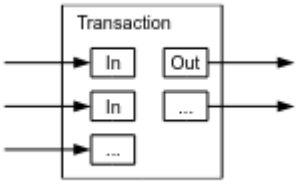
Com tala, la verificacion qu'ei hidabla tant que los nods aunèstes contròlan lo hialat, mes qu'ei mei vulnerable si lo hialat ei espotit per la poténcia d'un atacant. Tant que los nods pòden verificar las transaccions entà eths medishs, lo metòde de verificacion simplificada que pòt estar enganada per las transaccions fabricadas d'un atacant tant que l'atacant contunha a suberpassar lo hialat. Ua estrategia entà se protegir contra aquò que seré d'acceptar alèrtas deus nods deu hialat qui detèctan un blòc invalide, provocant lo telecargament deu blòc complet e de las transaccions suspecciosas peu logiciau utilizator entà confirmar la divergéncia. Las enterpresas qui receben sovent deus pagaments que voleràn probablament har foncionar los lors pròpis nods entà ua securitat mei independenta e ua verificacion mei rapida.

9. **Combining and Splitting Value**

Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.

9. **Combinason e separacion de las valors**

Maugrat que sia possible de manipular las pèças individuaument, que seré drin comòde de har ua transaccion separada entà cada centime dens un transferiment. Entà autorizar las valors a estar divididas o combinadas, las transaccions que contienen entradas e sortidas multiplas . Normaument qu'i aurà sia ua entrada unica provienent d'ua mei gròssa e precedenta transaccion o mantua entrada combinant mei petits montants e au mensh duas sortidas : ua entau pagament, e ua entau tornat de la moneda, se s'escad entau pagador.



It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history.

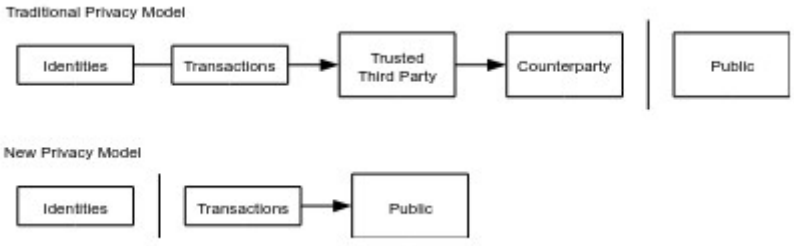
Que deu estar notat que la disseminacion, on ua transaccion depend de mantua transaccion, e qu'aqueras transaccions dependen de hòrt mei, n'ei pas un problèma ací . N'i a pas jamei lo besonh d'ua copia completa e autonòma de l'istòria de las transaccions

10. **Privacy**

The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were

10. **Vita privada**

Lo modèle bancari tradicionau que realiza un nivèu de proteccion de la vita privada en limitant l'accès a las informacions a las personas concernidas e au tèrç de hidança. La necessitat d'anonciar totas las transaccions publicament qu'escarta aqueth metòde, mes la proteccion de la vita privada que pòt enqüèra estar assegurada en rompent lo flux d'informacion a un aute endret : en gardant las claus publicas anonimas. Lo public que pòt véder que quauqu'un qu'ei arsec d'enviar un montant a quauqu'un mei, mes shens informacion ligant la transaccion a quauqu'un. Açò qu'ei similar au nivèu d'informacion remetut per la borsa, on las òras e montants deus escambis, lo “quasernet d'ordis”, qu'ei publica, mes shens díser qui son las partidas



As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

Com parahuec addicionau, un parelh navèth de claus que poiré estar utilizada entà cada transaccion entà los gardar non ligadas a un proprietari comun. Totun, la ligason qu'ei inevitabla dab las transaccions multi-entradas, qui revèlan necessàriament que las lors entradas èran detienudas per un medish proprietari. Lo risc qu'ei que si lo proprietari d'ua clau ei revelat, las ligasons que pòden revelar autas transaccions qui an apartienut au medish proprietari

11. **Calculations**

We consider the scenario of an attacker trying to generate an alternate chain faster than the honest chain. Even if this is accomplished, it does not throw the system open to arbitrary changes, such as creating value out of thin air or taking money that never belonged to the attacker. Nodes are not going to accept an invalid transaction as payment, and honest nodes will never accept a block containing them. An attacker can only try to change one of his own transactions to take back money he recently spent.

The race between the honest chain and an attacker chain can be characterized as a Binomial Random Walk. The success event is the honest chain being extended by one block, increasing its lead by +1, and the failure event is the attacker's chain being extended by one block, reducing the gap by -1.

The probability of an attacker catching up from a given deficit is analogous to a Gambler's Ruin problem. Suppose a gambler with unlimited credit starts at a deficit and plays potentially an infinite number of trials to try to reach breakeven. We can calculate the probability he ever reaches breakeven, or that an attacker ever catches up with the honest chain, as follows [8]:

p = probability an honest node finds the next block

q = probability the attacker finds the next block

qz = probability the attacker will ever catch up from z blocks behind

$$q_z = \begin{cases} 1 & \text{if } p \le q \\ (q/p)^z & \text{if } p > q \end{cases}$$

Identities Transactions Trusted
Third Party Counterparty Public
Identities Transactions Public
New Privacy Model
Traditional Privacy Model

Given our assumption that p > q, the probability drops exponentially as the number of blocks the attacker has to catch up with increases. With the odds against him, if he doesn't make a lucky lunge forward early on, his chances become

11. **Calculs**

Que consideram lo scenari d'un atacant ensajant d'engendrar ua cadena alternativa mei lèu que la cadena aunèsta. Quan ua tau tasca serà complida, n'obreish pas la pòrta a tots los cambiaments arbitraris, taus com crear moneda èx nihilo o préner argent qui a pas jamei apartienut a l'atacant. Los nods ne van pas acceptar ua transaccion invalida com pagament, e los nods aunèstes n'acceptaràn pas jamei un blòc los contienent. Un atacant que pòt solament ensajar de cambiar ua de las soas pròpias transaccions entà recuperar l'argent qui a despensat recentament.

La corsa enter la cadena aunèsta e la cadena de l'atacant que pòt estar caracterizada per un Camin Aleatòri Binomiau. L'eveniment d'escaduda qu'ei que la cadena aunèsta ei espandida per un blòc, aumentant la soa avança de +1 e un eveniment de mauescaduda qu'ei l'extension de la cadena de l'atacant redusint l'escart de -1.

La probabilitat qui un atacant ganha en partint d'un retard iniciau qu'ei analògue au problèma de la roeina deu jogador. Que supausatz un jogador, dispausant d'un crèdit illimitat qui parteish dab un retard e que jòga potenciaument unnombre infinit d'ensais entà ensajar d'aténher lo punt d'equilibri. Que podem calcular la probabilitat qui aténhia un dia l'equilibri, o qu'un atacant atrapa un dia la cadena aunèsta, com segueish [8] :

p = probabilitat qui un nod aunèste tròba lo blòc vienent.

q = probabilitat qui l'atacant tròba lo blòc vienent.

qz = probabilitat qui l'atacant atrapa un dia en partint dab z blòcs de retard.

$$q_z = \begin{cases} 1 & si\ p \leqslant q \\ \left(\frac{q}{p}\right)^z & si\ p > q \end{cases}$$

Modèle tradicionau de las dadas privadas
Identitats Transaccions Tèrç
de hidança Contrapartida Public
PublicTransactionsIdentités
Nouveau que modèla dadas privadas

Estant balhada la nosta ipotèsi qui p > q, la probabilitat que tomba de faiçon exponenciau dab l'aumentacion deu nombre de blòcs qui l'atacant que deu atrapar. Dab l'astre contra eth, si hè pas un abans astruc autanlèu la debuta, los

vanishingly small as he falls further behind.

We now consider how long the recipient of a new transaction needs to wait before being sufficiently certain the sender can't change the transaction. We assume the sender is an attacker who wants to make the recipient believe he paid him for a while, then switch it to pay back to himself after some time has passed. The receiver will be alerted when that happens, but the sender hopes it will be too late.

The receiver generates a new key pair and gives the public key to the sender shortly before signing. This prevents the sender from preparing a chain of blocks ahead of time by working on it continuously until he is lucky enough to get far enough ahead, then executing the transaction at that moment. Once the transaction is sent, the dishonest sender starts working in secret on a parallel chain containing an alternate version of his transaction.

The recipient waits until the transaction has been added to a block and z blocks have been linked after it. He doesn't know the exact amount of progress the attacker has made, but assuming the honest blocks took the average expected time per block, the attacker's potential progress will be a Poisson distribution with expected value:

sons astres que s'apriman, tant mei que recula.

Que consideram adara quant de temps, lo destinatari d'ua transaccion que deu esperar entà èste pro certan qui l'expeditor la pòt pas mei cambiar. Que supausam que l'expeditor ei un atacant qui vòu har créder entà un temps au destinatari qui l'a pagat, puish, qu'inverteish la transaccion entà se remborsar après periòde. Lo destinatari que serà alertat quan aquò qu'arribarà, mes l'expeditor qu'espèra que serà tròp tard.

Lo destinatari qu'engendra un parelh navèth de claus e que balha la clau publica a l'expeditor juste abans la signatura. Açò qu'evita que l'expeditor prepara ua cadena de blòcs a l'avança en tribalhant dessús a contunhar dinc a que l'astre lo permétia d'èste pro en avança, e d'executar alavetz la transaccion. Un còp la transaccion enviada, l'expeditor mauaunèste que comença de tribalhar en secret sus ua cadena parallèla contienent ua version alternativa de la soa transaccion.

Lo destinatari qu'espèra dinc a que la transaccion sia aponduda a un blòc e z blòcs que son estats ligats après . Ne coneish pas l'avançament exacte qui l'atacant a realizat, mes en supausant que los blòcs aunèstes an pres lo temps mejan esperat per blòc, l'avançament potenciau de l'atacant que seguirà ua lei de Poisson dab la valor esperada :

$$\lambda = z\frac{q}{p}$$

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

Entà obtiéner la probabilitat que l'atacant posca enqüèra atrapar, que multiplicam la densitat de la lei de Poisson entà cada montant d'avançament que poiré aver, per la probabilitat qu'aja atrapat despuish aqueth punt :

$$\sum_{k=0}^{\infty} \frac{\lambda^k e^{-\lambda}}{k!} \cdot \begin{cases} (q/p)^{(z-k)} & \text{if } k \le z \\ 1 & \text{if } k > z \end{cases}$$

Rearranging to avoid summing the infinite tail of the distribution…

En réarrangeant entà evitar de somar la seria infinida de la distribucion :

$$1 - \sum_{k=0}^{z} \frac{\lambda^k e^{-\lambda}}{k!} \left(1 - (q/p)^{(z-k)}\right)$$

Converting to C code...

En convertint en còdi C :

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
    double p = 1.0 - q;
    double lambda = z * (q / p);
    double sum = 1.0;
    int i, k;
    for (k = 0; k <= z; k++)
    {
        double poisson = exp(-lambda);
        for (i = 1; i <= k; i++)
        poisson *= lambda / i;
        sum -= poisson * (1 - pow(q / p, z - k));
    }
    return sum;
}
```

Running some results, we can see the probability drop off exponentially with z.

En obtienent quauques resultats, que podem véder que la probabilitat tomba de faiçon exponenciau dab z

```
q=0.1
z=0 P=1.0000000
z=1 P=0.2045873
z=2 P=0.0509779
z=3 P=0.0131722
z=4 P=0.0034552
z=5 P=0.0009137
z=6 P=0.0002428
z=7 P=0.0000647
z=8 P=0.0000173
z=9 P=0.0000046
z=10 P=0.0000012
q=0.3
z=0 P=1.0000000
z=5 P=0.1773523
z=10 P=0.0416605
z=15 P=0.0101008
z=20 P=0.0024804
z=25 P=0.0006132
z=30 P=0.0001522
z=35 P=0.0000379
z=40 P=0.0000095
z=45 P=0.0000024
z=50 P=0.0000006
```

Solving for P less than 0.1%…

P < 0.001
q=0.10 z=5
q=0.15 z=8
q=0.20 z=11
q=0.25 z=15
q=0.30 z=24
q=0.35 z=41
q=0.40 z=89
q=0.45 z=340

12. **Conclusion**

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

12. **Conclusion**

Qu'avem prepausat un sistèma de transaccions electronicas se passant de hidança. Qu'avem començat dab un quadre de foncionament ordinari de pèças de moneda establidas per signaturas electronicas, qui auhereish un contròle poderós de la proprietat, mes qui ei incomplet shens mejan d'evitar la dobla-despensa. Entà resòlver aquò, qu'avem prepausat un hialat par-a-par utilizant la pròva-de-tribalh entà enregistrar ua istòria publica de las transaccions, qui vad lèu impracticabla a un atacant de modificar si los nods aunèstes contròlan la majoritat de la poténcia CPU. Lo hialat qu'ei robuste dens la soa simplicitat non estructurada . Los nods que tribalhan tots amassa dab drin de coordinacion. N'an pas besonh d'estar identificats, puishque los messatges son pas expedits cap tà destinacions particularas e qu'an solament besonh d'estar liurats au mélher. Los nods que pòden quitar e rejúnher lo hialat au lor grat, en acceptant com pròva la cadena de pròva-de-tribalh de çò qui s'ei passat en la lor abséncia. Que vòtan dab la lor poténcia CPU, exprimint la lor acceptacion deus blòcs valides en tribalhant a los espandir e a regetar los blòcs invalides en refusant de tribalhar dessús. Totas las règlas e primas de resultat necessaris que pòden estar impausadas dab aqueth mecanisme de consensus.

References

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In 20th Symposium on Information Theory in the Benelux, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In Journal of Cryptology, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In Sequences II: Methods in Communication, Security and Computer Science, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In Proceedings of the 4th ACM Conference on Computer and Communications Security, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure,", 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957

Nòtas de traduccion

Aquera traduccion qu'ei estada realizada dab lo mei gran suenh mes ne pretend pas de nada manièra estar un substitut perfèit deu document originau en anglés de Satoshi Nakamoto bitcoin.org/bitcoin.pdf . Aqueth document que deu donc èste utilizat dab la prudéncia qui s'impausa e que convien d'utilizar los documents originaus entà realizar tota òbra derivada.

Doble-spending :	Doble-despensa
Peer to peer :	par-a-par
Proof-of-work :	Pròva-de-tribalh
Best effort :	Au mélher
Coin :	Pèça
Time stamp :	Horodate
Time stamping	Horodatage
Hash :	Mèrca numerica, talhucatge
IP : Internet Protocol (address) :	Adreça Internet