Bitcoin: Een Peer-to-Peer Elektronisch Cash Systeem

Satoshi Nakamoto

satoshin@gmx.com

www.bitcoin.org

Nederlandse vertaling: 14tE9fUuvSgZ7NyoMJBn2PKH4ApngWPQUG

Abstract. Een volledig peer-to-peer versie van elektronisch geld zou toestaan om online betalingen direct te versturen van de ene partij naar de andere zonder langs een financiële instantie te gaan. Digitale handtekeningen zorgen voor een deel van de oplossing, maar de grootste voordelen gaan verloren als er nog steeds een vertrouwde derde partij vereist is om dubbele uitgaves te voorkomen. Wij stellen een oplossing voor dit dubbele uitgave probleem via het gebruik van een peer-to-peer netwerk. Dit netwerk plaatst tijdstempels op transacties door ze te hashen in een continue keten van op hashes gebaseerde bewijs-van-werk (proof of work). De langste keten dient hierbij niet louter als bewijs van de volgorde van waargenomen gebeurtenissen, maar bewijst dat deze kwam van de grootste poule van CPU kracht. Zo lang de meerderheid van die CPU kracht gecontroleerd wordt door knooppunten in het netwerk die niet samenwerken om dit netwerk aan te vallen, zullen zij de langste keten genereren en de aanvallers te snel af zijn. Het netwerk zelf vergt een minimale structuur. Boodschappen worden verstuurd op een best-effort-basis, en knooppunten kunnen zich op het netwerk aansluiten en verlaten op gelijk welk moment, zolang ze de langste proof-of-work keten aanvaarden als bewijs van wat er gebeurd is tijdens hun afwezigheid.

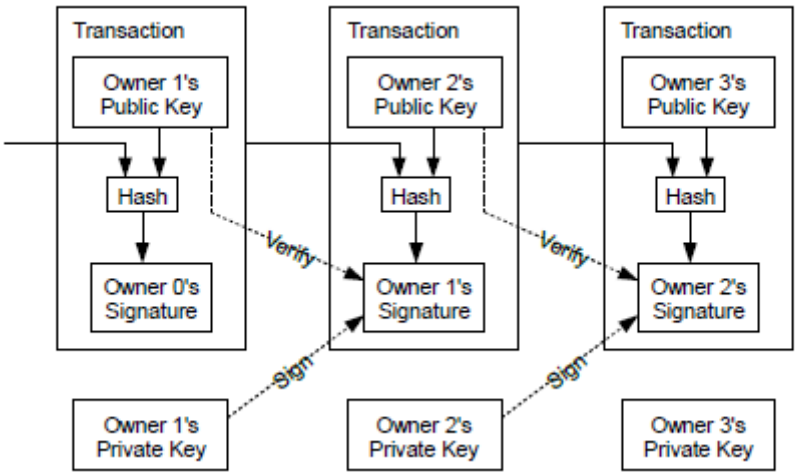
1. Introductie

Handel op het internet vertrouwt vandaag de dag bijna uitsluitend op financiële instellingen die dienst doen als vertrouwde derde partijen bij het verwerken van elektronische betalingen. Terwijl dit systeem goed genoeg werkt voor de meeste transacties, lijdt het nog steeds onder de inherente zwakke plekken van het op vertrouwen gebaseerde model. Compleet onomkeerbare transacties zijn niet echt mogelijk, sinds de financiële instituties het niet kunnen vermijden om te bemiddelen bij disputen. De kostprijs van deze tussenkomst verhoogt transactiekosten, beperkt de minimum praktische transactiegrootte en de mogelijkheid om kleine losse transacties te plegen. Daarnaast is er een bredere kost bij het verliezen van het maken van onomkeerbare betalingen en onomkeerbare diensten. Met de mogelijkheid voor een terugboeking, vergroot de nood voor de vertrouwensfactor. Verkopers moeten op hun hoede zijn voor hun klanten, en hen lastigvallen voor meer informatie dan ze anders zouden nodig hebben. Een zeker percentage aan fraude wordt aanvaard als onvermijdelijk. Deze kosten en onzekerheden bij betalingen kunnen worden vermeden wanneer iemand in persoon fysieke valuta gebruikt, maar er bestaat geen enkel mechanisme om betalingen te maken via een communicatiekanaal zonder een vertrouwde partij.

Er is nood aan een elektronisch betalingssysteem gebaseerd op cryptografisch bewijs in de plaats van vertrouwen, dat aan twee daartoe bereide partijen toestaat rechtstreeks met elkaar te handelen zonder de nood aan een vertrouwde derde partij. Transacties die computationeel onpraktisch zijn om terug te boeken zouden verkopers beschermen van fraude, en routine escrows of derdenrekeningen zouden makkelijk kunnen worden geïmplementeerd om kopers te beschermen. In deze paper, stellen we een oplossing voor die het dubbele uitgave probleem aankaart via een peer-to-peer verdeelde tijdstempel server om rekenkundig bewijs te leveren van de chronologische volgorde van transacties. Dit systeem is veilig zolang eerlijke knooppunten collectief meer CPU kracht leveren dan die van een collaborerende groep van aanvallende nodes.

2. Transacties

We definiëren een elektronische munt als een keten van digitale handtekeningen. Elke eigenaar transfereert de munt naar de volgende door een digitale handtekening te plaatsen op een hash van de voorgaande transactie en de publieke sleutel van de volgende eigenaar en deze toe te voegen aan het einde van de munt. Een te betalen persoon kan deze handtekeningen verifiëren om de keten van eigenaarschap te verifiëren.



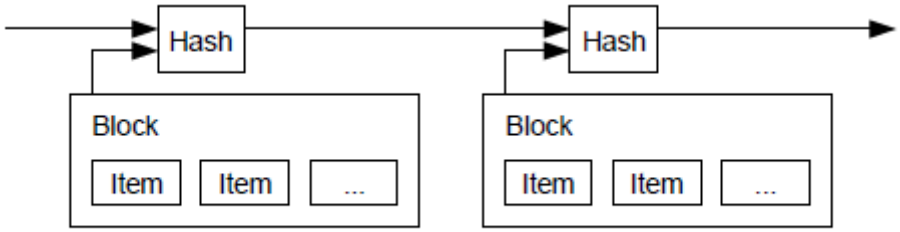
Het probleem is natuurlijk dat de te betalen persoon niet kan verifiëren dat een van de eigenaars de munt niet twee keer heeft uitgegeven. Een veel voorkomende oplossing is de introductie van een vertrouwde centrale autoriteit, of munter, die elke transactie controleert op dubbele uitgaves. Na elke transactie, dient de munt terug te keren naar de munterij, om een nieuwe munt uit te giften, en enkel munten die zo rechtstreeks in circulatie gebracht zijn kunnen vertrouwd worden om niet dubbel uitgegeven te worden. Het probleem met deze oplossing is dat het lot van het hele monetaire systeem afhangt van het bedrijf achter deze muntslag, omdat elke transactie langs hen moet gaan, net als bij een bank.

We hebben een manier nodig voor de te betalen persoon om te weten dat de vorige eigenaars geen eerdere transacties had getekend. Voor onze doeleindes, de vroegste transactie is degene die telt, zodat we ons dan geen zorgen meer maken om latere pogingen tot een dubbele uitgave. De enige manier om de afwezigheid van transacties te bevestigen is op de hoogte zijn van alle transacties. In dit munterij gebaseerde model, was de munterij bewust van elke transacties om te beslissen welke er het eerste arriveerde. Om dit te verwezenlijken zonder gebruik van een vertrouwde partij, moeten transacties plaatsvinden in de volgorde van ontvangst. De ontvanger heeft bewijs nodig dat op het tijdstip van elke transactie, de meerderheid van de knooppunten akkoord ging dat het de eerst ontvangen transactie was.

3. Tijdstempel server

De oplossing die wij voorstellen, begint met een tijdstempel server. Een tijdstempel server werkt door het nemen van een hash van een block items die een tijdstempel moeten krijgen en het ruim publiceren van die hash, zoals bijvoorbeeld in een krant of een Usenet bericht [2-5]. De tijdstempel bewijst dat de data bestaan moet hebben op dat moment, natuurlijk, om in de hash te kunnen geraken. Elke

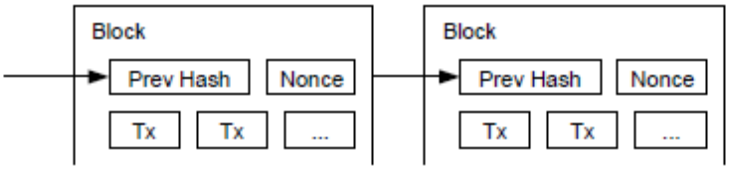
tijdstempel bevat de vorige tijdstempel in zijn hash, en vormt een keten, waarin elke vorige tijdstempel die daarvoor versterkt.



4. Proof-of-Work

Om een gedistribueerde tijdstempel server op een peer-to-per basis te implementeren, zullen we een proof-of-work systeem gelijkaardig aan Adam Blacks Hashcash moeten gebruiken, in de plaats van krantenartikels of Usenet berichten. Deze proof-of-work omvat het scannen naar een waarde die wanneer gehasht, zoals met SHA-256, de hash begint met een getal van nul bits. Het gemiddelde benodigde werk hiervoor is exponentieel in het aantal nul bits vereist en kan geverifieerd worden door het uitvoeren van een enkele hash.

Voor ons tijdstempelnetwerk implementeren we het proof-of-work systeem door het verhogen van een nonce (gelegenheidsgegevens) in het blok tot er een waarde wordt gevonden die de hash in het blok de vereiste nul bits geeft. Eens de CPU inspanning besteed is om te voldoen aan het proof-of-work systeem, kan het blok niet meer veranderd worden zonder al het werk te herdoen. Wanneer later blokken aan elkaar geketend worden, omvat het werk dat nodig is om het huidige blok te veranderen, het opnieuw uitvoeren van het werk van alle blokken die erna komen.



Het proof-of-work systeem lost daarnaast ook het probleem op van het bepalen van vertegenwoordiging in besluitvorming. Wanneer de meerderheid gebaseerd zou zijn op basis van één IP-adres-per-stem, zou dat systeem omzeild kunnen worden door gelijk wie dat meerdere IP's kan toewijzen. Proof-of-work is in essentie één CPU = één stem. De meerderheidsbeslissing is vertegenwoordigd door de langste keten, die de grootste proof-of-work inspanning in zich geïnvesteerd heeft. Wanneer een meerderheid van de CPU kracht onder controle is van eerlijke knooppunten, zal de eerlijke keten het snelst groeien en alle concurrerende ketens voorbijstreven. Om een voorgaande blok aan te passen zou een aanvaller al het werk van de proof-of-work van het blok en alle blokken erna moeten opnieuw doen en dan het werk van de eerlijke knooppunten moeten inhalen en overtreffen. We zullen later aantonen dat de waarschijnlijkheid dat een tragere aanvaller kan inhalen exponentieel verkleint naarmate er meer blokken toegevoegd worden.

Om te compenseren voor een verhoging in hardware snelheid en met de tijd afwisselende interesse om knooppunten te draaien, is de moeilijkheid van het proof-of-work systeem bepaald door een voortschrijdend gemiddelde dat gebaseerd is op het aantal blokken per uur. Wanneer de blokken te snel worden aangemaakt, verhoogt de moeilijkheidsgraad.