Cryptográfia jegyzőkönyv 2

Blokklánc és tranzakciók

2020.04.04 ITKoin_02.py

Schmidt László

„A jegyzőkönyvben NEM a pontos crypto valuta megvalósítása hanem az ITkoin pytonos megvalósítása van dokumentálva.”

Feladat:

Blokklánc definiálás és felépítése, tranzakció kezelése.

A rendszer működés/menete:

Van egy blokklánc ami egy speciális blokkal indít, a header hash az az az elötte lévő blokkra mutató lenyomat csupa egyes ez indikálja hogy itt kezdődik a blokklánc . Az első blokknál mi 100 csaposhi összeget adunk minden felhasználónak. Felhasználók azok akiknek a lánc indulása elött volt kulcspárjaik, ez nem befolyásolja, hogy később ne lehessenek új felhasználók az az pénztárcák létrehozva. Ezt a műveletet úgy végezzük el, hogy bemenet nélkül a semmiből végzünk tranzakciót a pénztárcák publikus kulcsára, ezeket egyszerűen beletesszük a feldolgozásra váró tranzakciók listályába, és kibányásszuk a blokkot.

Egy blokk felépítése:

Két részből áll fejléc és tranzakció.

A fejlécnek három része van „nonce” erről később, az előző blokk lenyomata, valamint a tranzakció lenyomata. Az previous_block_hash köti a blokkunkat a többihez, ez által ha az előző blokkunkat lenyomatoljuk akkor ezt a lenyomatott kell kapunk ahhoz hogy tudjuk a blokk valósságát bizonyisítani. Ezen tulajdonsága miatt a hash tartalmazza az előző blokkot, az előző blokk pedig az őt megelőzőt, a lánc bármely pontján változtatva a hashfüggvény tulajdonságai miatt a lenyomata változik és a rákövetkező összes blokk is változik. Ezért nem lehet utólag letagadni a tranzakciókat. A tranzakció lenyomata lehetővé teszi hogy a tranzakcióról bebizonyítsuk, hogy igen ezen pénztárcák között történt a tranzakció. A nonce egy egész szám ami bármi lehet, a bányászás nehezítésében játszik szerepet.

A tranzakció. Egy tranzakció áll egy bemeneti listából és egy kimeneti listából, valamint egy tranzakció azonosítóból ami egy lenyomat (txid) a tranzakcióról.

A tranzakció inputja a megelőző tranzakciók outputja kiegészítve az aláírással, az az előző tranzakció lenyomata, elkölthető összeg aláírva és a publikus kulcs az aláírás ellenőrzéséhez.

A tranzakció outputja az input lesz valamint a visszajáró vissza utalása magamnak.

Az utalás menete:

Összeszedjük a saját nem elköltött az az a pénztárcákban lévő csaposhikat , ezeket aláírjuk és betesszük a tranzakció inputjába, és létrehozzuk ehhez a tranzakcióhoz az outputokat, az az amit szeretnénk az összeggel kezdeni. Ha a tranzakciót összeállítottuk akkor betesszük a pending_tranzaction listába hogy a bányászok beletegyék a következő blokkba. A blokk megszületése után válik a tranzakció megtörténté. Az inputhoz szükséges információk a blokkláncban vannak ezért ahhoz hogy összeszedjük őket végig kell olvasni a blokkláncot.

Elindulunk az elején és minden blokk outputjából megnézzük melyik amit nekünk küldtek és ezeket beletesszük egy listába, de ebben a listában olyan is lehet amit már elköltöttünk ezért azokat összehasonlítjuk az inputokkal és ha van egyezés akkor kivesszük a listából. A végén csak az elkölthető mennyiség lesz benne. Ez olyan mint amikor összeszámoljuk az aprópénzünket a pénztárcában. Nekünk ezután nincs más dolgunk mint felhasználó.

Hogy lesz a blokklánc része. A várakozó tranzakciókat a bányászok összegyűjtik és beleteszik egy blokkba ahol meg kell oldaniuk egy feladatot. Ez a feladat a nonce meghatározása. A blokklánc minden eleme ismeri az elötte lévő blokk header_hash ét, ennek meghatározása úgy működik hogy a az egész header-ből készül egy lenyomat. A lenyomat mindig teljesen más lesz ha akár 1 bit értékben is eltér a header. Egy lenyomatot készíteni egyszerű de egy olyan lenyomatot készíteni ami n db 0 val kezdődik az nem triviális, és csak is találgatással lehet megoldani. A nonce érték változtatásával befolyásolható a hash kimenete. A feladat, hogy egy olyan noncet találjunk amit behelyettesítve a headerbe a lenyomata n db 0 val fog kezdődni. Ennek az esély ½^n . Ezt a számítás igényes folyamatot nevezik bányászásnak, a bányászás a „prof of work” a garancia arra hogy a blokk valódi és nem téves/hamis adatokat tartalmaz. Ez ott válik el, hogy egy blokk megtalálása szerencsével is járhat így ha mi hamis blokkokat akarnánk közzétenni akkor innentől kezdve minden blokkot nekünk kéne megtalálni ami hatalmas számítási kapacitást igényelne és emiatt egy idő után elmaradnánk a blokklánccal ekkor viszont mindenki a hosszabb blokklánccal folytatja ami biztosan a valós a másikat meg eldobja. A mi esetünkben az a feladat hogy 4db 0 val kezdődjön, ezt úgy keressük, hogy minden sikertelen próbálkozásra növeljük a nonce értékét 1 el. Amikor megtaláljuk az értéket akkor létrehozzuk a blokkot és hozzátesszük a lánchoz ami a broadcastot szimulálja.

Összegezve a folyamat.

Ismerjük a láncot csupa 1 el kezdődik. Tranzakció elött összeszedjük a pénzünket a láncból, beletesszük az új tranzakció inputjába kiszámoljuk a visszajárót és azt is beletesszük a tranzakcióba. A tranzakciót a pendig listába tesszük és várjuk, hogy a bányászok beletegyék a következő blokkba. A bányászok várják a tranzakciókat és azokat összegyűjtve megkezdik a bányászást ami egy matek feladvány, ki kell találniuk egy számot amitől a lenyomat 4 db 0 val fog kezdődni. Ha megvan akkor hozzátesszük a lánc végéhez.

Eddig a program semmiféle ellenőrzést nem végez, ezért minden művelet végrehajtható akármilyen változókkal. Végezhetünk tranzakciót úgy is hogy a végén mínuszban leszünk. Tehetünk tranzakciót más nevében, és akár input nélkül is. Valamint nem ellenőrizzük a láncot az az a header_hast sem mikor hozzátesszük a lánchoz.

Mit kell ellenőrizni?

A tranzakció létrehozójának és a bányásznak is le kell ellenőriznie a láncot, hogy minden rendben van e vele, és csak akkor tegyenek hozzá ha egyezik e a hash valamint ne egyből tegyék hozzá mert lehet hogy több elágazás is van és a leghosszabbal kell folyatni. A bányásznak le kell ellenőriznie, hogy a tranzakciók helyesek e, az inputok valósak e és lefedik az összeget, alávan e írva és hogy az a személy írta alá akihez tartozik, van e visszautalás. ha kész a blokk vagy még akár közben is le kell ellenőrizni hogy valaki ki bányászta e már a blokkot.