Önálló laboratórium beszámoló

Crypto currency megoldások elemzése

Készítette:

Bura Pál

Témavezetők:
Csapodi Márton és Szabó Áron

Pázmány Péter Katolikus Egyetem
Információs Technológiai és Bionikai Kar

Budapest, 2014

Tartalomjegyzék

1. fejezet

Bevezetés és motiváció

A jelenlegi pénzvilágban használt pénznek három főbb tulajdonsága van. Először is fizikailag létezik, kézbe lehet venni és gazdát cserélhet. Másodszor, a készpénz úgy van megtervezve, hogy nehéz legyen hamisítani, harmadszor pedig, a készpénz névtelen, azaz nem tudni, hogy ki mikor mennyit költött el belőle. [1] A digitális pénz olyan fizetőeszköz, amelynek fizikai valója nincsen, csak az internet világában létezik, mégis alternatívát próbál nyújtani a jelenlegi pénzvilág működésére. Anonimitását és a hamisítás megnehezítését különböző kriptográfiai módszerekkel biztosítják.

A kriptopénzek kialakításának legfőbb motivációja a decentralizáltság és a tranzakciók gazdaságosabbá tétele. A hagyományos pénzvilágban lebonyolított pénzküldéshez ugyanis szükség van egy harmadik, központi félre, amely a feladó és a címzett között kapocsként működik. Ezt a szerepet a bankok látják el. A gond ezzel ott van, hogy a tranzakció két valódi résztvevője meg kell, hogy bízzon egy harmadik félben. Ezen probléma orvoslására született meg Satoshi Nakamoto koncepciója, amely azóta rengeteg továbbfejlesztett kriptopénz alapját is alkotja.

A dolgozat legfőbb célja a különböző modelleken alapuló crypto currency megoldások bemutatása és összehasonlítása, megvizsgálva, hogy mik voltak a tervezésnél figyelembe vett legfontosabb szempontok.

A dolgozat az alábbi felosztást követi:

- **2. fejezet (Bitcoin)**: felvázolja az első és legelterjedtebb kriptopénz architektúráját és bemutatja az őt érintő kritikákat és gyengeségeket;
- **3. fejezet (Peercoin)**: ismerteti az első olyan digitális pénzt, amely először használta a *proof-of-stake* és *proof-of-work* hibridet;
- **4. fejezet (Egyéb kriptopénzek)**: bemutatja pár kisebb kriptopénz tulajdonságait, kiemelve a számottevőbb különbségeket;
- **5. fejezet (Kriptopénzek összehasonlítása)**: összehasonlítja a jelenlegi három legnagyobb kriptopénzt;
- **6. fejezet (Összefoglalás)**: összefoglalja a dolgozatban leírtakat.

2. fejezet

Bitcoin

A Bitcoin egy nyílt forráskódú peer-to-peer pénz, amelynek használata különböző kliensprogramok segítségével történik. A rendszert egy teljesen decentralizált hálózat alkotja. Ez azt jelenti, hogy a pénz átutalásánál nincsen szükség egy köztes harmadik félre, amelynek szerepét a normál pénzvilágban a bankok látják el.

A rendszer az összes létrejövő tranzakciót egy blokkokból álló láncban tárolja el, amelynek módosításához a hálózat teljes számítási kapacitásának legalább a felével rendelkezni kell. Ezen kívül a Bitcoin protokoll a ma ismert legmodernebb kriptográfiai eljárásokat alkalmazza az adatok biztonságban tartásáért.

Mivel ez a rendszer senkihez nem tartozik (a normál pénzvilág rendszereivel szemben, ahol pl. az internetbankok is valamilyen bankhoz tartoznak) így a tranzakciók elfogadását, érvényesítését és a bitcoinok előállítását is maga a hálózat végzi. Az összes végbement tranzakció elérhető a blokkláncból, így minden egyes bitcoin útja a kezdetekig visszakövethető. Ezzel szemben viszont a felhasználók címei titkosak, semmit nem mondó karakter sorozatok. Az anonimitást [2] ráadásul az is növeli, hogy a rendszer azt támogatja, hogy a felhasználók minden egyes tranzakciónál új címet használjanak. Új bitcoinok az egyes blokkok generálásakor kerülnek a hálózatba, átlagosan tíz percenként, egészen addig, amíg a teljes bitcoin készlet el nem éri a 21 milliót. Ezt úgy érik el, hogy 4 évente feleződik a blokkok generálásért járó bitcoin jutalom, amely kezdetben 50-re volt állítva. [3]

A Bitcoin rendszer tranzakciói sokkal gyorsabban végbemennek, mint a hagyományos pénzvilág tranzakciói, viszont a visszaigazolás csak akkor érkezik meg, ha a tranzakció ténylegesen végbement. Innentől kezdve azonban visszavonhatatlan lesz. A következő alfejezetben található a rendszer részeinek és protokolljának részletes leírása.

2.1. Architektúra

A Bitcoinhoz kapcsolódó legalapvetőbb fogalmak a tranzakció, a blokk, a blokklánc és a bányászás. A tranzakciók blokkokba rendeződnek, amelyek egy lánccá állnak össze. A bányászó felhasználók a bányászás folyamatával fogják össze a blokkokká a tranzakciókat és fűzik hozzá a lánc végéhez. Az egyes részek külön alfejezetben kerülnek kifejtésre.

2.1.1. Tranzakciók

Egy elektronikus érme digitális aláírások láncolataként határozható meg. A tranzakció pedig egy aláírt adatrészként, amelyek blokkokba rendeződnek és a teljes hálózat számára elérhetőek.

A bankoknál a számlák a publikusak és a tranzakciók a titkosak, míg a Bitcoin esetében a cím van elrejtve, a tranzakciók pedig elérhetőek az összes felhasználó számára. Az interneten bármikor visszanézhetjük az összes eddig végbement tranzakciót a *Genesis block* (a legelső blokk) óta.

Egy tranzakció az alábbi struktúrát követi.

2.1. táblázat. Tranzakció struktúra

Mező	**Méret**
Version no.	4 byte
In-counter	1-9 byte
List-of-inputs	
Out-counter	1-9 byte
List-of-outputs	
Lock time	4 byte

A *Version no.* a verziószám, amely jelenleg 1. *In-counter* és az *Out-counter* a bemenetek és a kimenetek számát foglalja magába. Mindkettő egy változó hosszúságú integer, amely az értéktől függően más méretet vesz fel (<0xFD: 1 byte, <=oxFFFF: 3 byte, <=0xFFFFFFFF: 5 byte, különben: 9 byte). A *Lock_time* azt jelöli, hogy egy tranzakció mikor lesz végleges. *List of inputs* és a *List of outputs* a bemenetek és kimenetek összessége, mivel nem csak egy bemenetű és kimenetű tranzakciók lehetnek.

Egy tranzakció felépítése egy valós példán látható a legjobban. Példaként itt van a 147777. blokk [4] egyik tranzakciója, annyi változtatással, hogy az átláthatóság kedvéért a hashek első 7 számjegyét jelenítem csak meg és sorszámokat adok az egészhez a könnyebb hivatkozhatóság végett.

```
{
  "hash":"b47c370...",
  "ver":1,
  "vin_sz":3,
  "vout_sz":1,
  "lock_time":0,
  "size":584,
  "in":[
    {
      "prev_out":{
        "hash":"e3129e1...",
        "n":0
      },
      "scriptSig":"3045022... 0440168..."
    },
    {
      "prev_out":{
        "hash":"1de5529...",
        "n":0
      },
      "scriptSig":"3045022... 042b2e4..."
    },
    {
      "prev_out":{
        "hash":"9375b6a...",
        "n":0
      },
      "scriptSig":"3045022... 0401aa6..."
    }
  ],
  "out":[
    {
      "value":"0.02000000",
      "scriptPubKey":"OP_DUP OP_HASH160 7f67b89... OP_EQUALVERIFY OP_CHECKSIG"
    }
  ]
}
```


2.1. ábra. Egy tranzakció felépítése

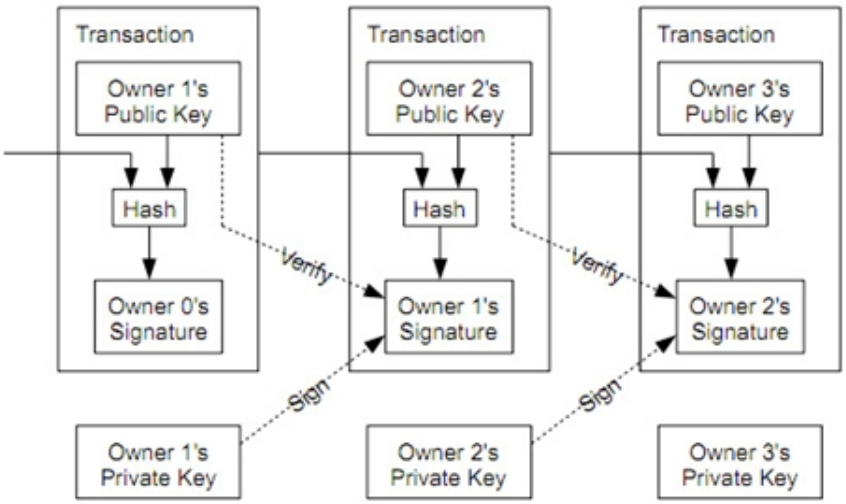
- **2. sor**: A tranzakciót azonosító hash.
- **3. sor**: A Bitcoin protokol verziószáma, amely jelenleg az 1.
- **4. sor**: Inputok száma, jelen esetben 3.
- **5. sor**: Outputok száma, ebben a tranzakcióban 1.
- **6. sor**: *Lock_ time* annak beállítására, hogy a tranzakció mikor lesz végleges. A 0 azt jelenti, hogy a tranzakció azonnal véglegesítésre kerül.
- **7. sor**: A tranzakció mérete byte-okban.
- **8. sor – 30. sor**: A három input külön-külön blokkban:

- **10. sor – 12. sor**: Azt mutatja meg, hogy ezt a bemenetet az „e3129e1” kezdetű tranzakció n. (jelen esetben nulladik) kimenetéből vesszük.
- **14. sor**: Itt az aláírás található, majd egy szóközt követően a bitcoinokat küldő felhasználó publikus kulcsa.
- **16. sor – 22. sor**: A második bemenet, az előzőhöz teljesen hasonlóan értelmezhető.
- **23. sor – 29. sor**: A harmadik bemenet.

- **31. sor – 36. sor**: Az outputok külön-külön blokkban, jelen esetben csak egy:
  - **33. sor**: A kimenet értéke, ebben a példában 0.02 bitcoin.
  - **34. sor**: Ez egy olyan kifejezés, amely a Bitcoin scripting nyelvében van megírva. A lényegi információ ebből a sorból a „7f67b89” kezdetű hash, amely a tranzakció címzettjét jelöli.

Látható, hogy az inputok bitcoin értékét nem tartalmazza a tranzakció. Ezt természetesen vissza lehet keresni a korábbi tranzakciók outputjai között. Egy standard tranzakcióban az inputok összege legalább akkora kell legyen, mint az outputok összege (ez alól természetesen kivételt képez a *Genesis block* és a *coinbase* tranzakciók, amikor is bitcoint adunk hozzá a hálózathoz). Abban az esetben, amikor az inputok összege nagyobb, mint az outputoké, a differencia azt a bányászt illeti, aki hitelesítette a tranzakciót magába foglaló blokkot.

Elsőre érdekesnek tűnhet az, hogy több input és több output van, de ennek köszönhetően tudjuk a „visszajárókat” kezelni. Tegyük fel, hogy 1.5 bitcoint akarok küldeni valakinek. Ezt megtehetem úgy, hogy felhasználom egy korábbi tranzakcióból kapott 2 bitcoinomat. Természetesen én nem akarom az egészet elküldeni, csak 1.5-öt belőle. Itt jön képbe a többszörös kimenet lehetősége, ahol is 1.5 bitcoint elküldök a címzettnek, 0.5-öt pedig az én egyik saját címemre.



2.2. ábra. Tranzakciók lánca [3]

Egy digitális érme úgy kerülhet máshoz, hogy az aktuális tulaj aláírja az előző tranzakció (amelyben hozzá került az érme) hashét és a címzett nyilvános kulcsát a saját titkos kulcsával. Ezek után az új tulajdonos az aláírások megvizsgálásával leellenőrizheti a tranzakciók sorozatát. Ez a módszer arra viszont még nem ad biztosítékot, hogy a feladó nem költötte-e el többször a küldött pénzt. Erre lesz megoldás az úgy nevezett *proof-of-work*, amelyről a későbbiekben lesz szó. [5]

2.1.2. Blokk

Minden elfogadásra váró blokk tartalmazza a frissen végbement tranzakciókat és egy referenciát az előző elfogadott blokkra. Egy új blokk akkor kerülhet be a hálózatba, ha megoldotta a kitűzött matematikai feladatot. A bányászás alapvetően az a folyamat, amikor a felhasználó elsőként próbálja megtalálni a megoldást az adott blokk feladatára. Megoldást találni nagyon nehéz, de a hálózat többi tagjának számára könnyű egy eredményről leellenőrizni, hogy az helyes-e. A matematikai problémára több megoldás is lehet, de a blokk érvényesítéséhez elég csak az egyiket megtalálni.

Mivel a blokkgenerálásért jutalom jár, ezért minden blokkban benne van egy cím is, amelyre azt küldeni kell. Kezdetben 50 BTC járt egy generált blokkért, de ez 210000 blokkonként feleződik.

A tranzakciók szétküldésre kerülnek a hálózatban, így minden bányász hozzáadja az új tranzakciókat ahhoz a blokkhoz, amely megoldásán éppen dolgozik.

A matematikai probléma nehézsége a hálózathoz igazodik úgy, hogy óránként átlagosan hat új blokk generálódjon. 2016 blokkonként (nagyjából két hét) a hálózat megnézi, hogy mennyi idő alatt generálódott le ennyi blokk és összehasonlítja a célértékkel, majd ehhez optimalizálja a matematikai feladat nehézségét.

Egy blokk struktúrája a lentebb látható módon néz ki.

2.2. táblázat. Blokk struktúra

Mező	Méret
Magic no.	Blocksize
Blockheader	80 byte
Version	4 byte
hashPrevBlock	32 byte
hashMerkleRoot	32 byte
Time	4 byte
Bits	4 byte
Nonce	4 byte
Transaction counter	1-9 byte
Transactions	

A *Magic no.* azt mutatja meg, hogy melyik hálózatról van szó. Ez egy olyan stringnek lett választva, amely nem valószínű, hogy normál adatban előfordul. A karakterek ritkán használt ASCII karakterek, UTF-8-ként nem érvényesek. A Bitcoin hálózat esetében a *Magic no.* a 00xD9B4BEF9, a testnet *Magic* értéke pedig a 0xDAB5BFFA.

A *Blocksize* a blokk mérete byte-okban, a *Transaction counter* a tranzakciók számát jelöli. A *Transactions* pedig a tranzakciók nem üres listája.

Ezeken kívül a blokk részét képezi a *Blockheader*, amelynek további mezői vannak:

- **Version**: A verziószám, amely akkor változik, ha frissül a szoftver és az új verziószámot ad.
- **hashPrevBlock**: Egy 256 bites hash, amely az előző blokk header-éből képződött. Akkor változik, ha új blokk jön létre.
- **hashMerkleRoot**: Egy egy szintén 256 bites hash, amely a blokkban található összes tranzakció alapján generálódik. Akkor változik, ha egy tranzakciót elfogadnak.
- **Time**: A jelenlegi idő másodpercekben 1970-01-01T00:00 UTC óta.
- **Bits**: A jelenlegi target érték, amely akkor változik, ha a nehézség változtatásra kerül.
- **Nonce**: 32 bites szám, amely folyamatosan inkrementálódik, minden egyes hash kipróbálása után. Ha ez a szám túlcsordul akkor a *coinbase*-ben megadott *Extra nonce* kerül használatra.

A *hashMerkleRoot* nélkül az összes bányász ugyanazokat a hasheket generálná sorban, így mindig a legnagyobb számítási teljesítményű gép nyerne, mert ő találná meg először a *targetnek* megfelelő hasht.

A Merkle fa egy hash fa, amely a Bitcoin esetében a következő példán szemléltetett módon néz ki.

Legyen 3 tranzakciónk, T1, T2 és T3, az ezekből képzett hashek pedig h1 = hash(T1), h2 = hash(T2), h3 = hash(T3). Ha páratlan tranzakciónk van, akkor az utolsó tranzakciót kétszer vesszük, hogy páros számú hashünk legyen, ezért vesszük még a h4 = hash(T4)-et is. Ezt követően számítjuk a h5 = hash(concat(h1, h2) és h6 = hash(concat(h3, h4) értékeket, majd pedig a h7 = hash(concat(h5, h6) hasht. Az egész folyamatban a hash(T) = sha256(sha256(T)), mivel a Bitcoin dupla hashelést alkalmaz. A végén a h7 lesz a Merkle root, amely magában foglalja az összes tranzakciót.

A blokkok első tranzakciója azonban mindig egy úgy nevezett *coinbase* tranzakció, amely a blokkot generáló felhasználó egyik egyedi Bitcoin címét tartalmazza, így a tranzakciókból képzett hash mindig más lesz, azaz gyakorlatilag garantálva van, hogy a bányászok

különböző hasheket generáljanak. Egy blokkban mindig csak egy *coinbase* tranzakció található. Minden generált hash nyerési esélye azonos a hálózatban.

A tranzakcióhoz hasonlóan egy blokk felépítése is egy valós példán látható a legjobban. Jelen példa a *Genesis block*-ot követő első blokkot [4] mutatja. Ebben az esetben is lerövidítettem a hasheket és számmal láttam el a sorokat.


```
1  {
2    "hash":"0000000...",
3    "ver":1,
4    "prev_block":"0000000...",
5    "mrkl_root":"0e3e235...",
6    "time":1231469665,
7    "bits":486604799,
8    "nonce":2573394689,
9    "n_tx":1,
10   "size":215,
11   "tx":[
12     {
13       "hash":"0e3e235...",
14       "ver":1,
15       "vin_sz":1,
16       "vout_sz":1,
17       "lock_time":0,
18       "size":134,
19       "in":[
20         {
21           "prev_out":{
22             "hash":"0000000...",
23             "n":4294967295
24           },
25           "coinbase":"04ffff001d0104"
26         }
27       ],
28       "out":[
29         {
30           "value":"50.00000000",
31           "scriptPubKey":"0496b53... OP_CHECKSIG"
32         }
33       ]
34     }
35   ],
36   "mrkl_tree":[
37     "0e3e235..."
38   ]
39 }
```


2.3. ábra. Egy blokk felépítése

- **2. sor**: A tranzakciót azonosító hash.
- **3. sor**: A Bitcoin protokol verziószáma, amely jelenleg az 1.

- **4. sor**: Az előző blokk hashe, jelen esetben a *Genesis block* referenciája (A 2. és 4. sorban található két hash nem egyezik meg, de az első hét karakterük pont ugyanaz).
- **5. sor**: *Merkle root*. A blokkban található összes tranzakció alapján generált hash fa rootja.
- **6. sor**: 1231469665 másodperc telt el 1970-01-01T00:00 UTC óta, tehát 2009.01.09. 02:54:25-kor keletkezett ez a blokk.
- **7. sor**: A *target* értéke, azaz, hogy milyen nehéz legyen a matematikai feladat. Jelen esetben ez 486604799 (0x1d00ffff). A szám kompakt formában található, ebből kerül levezetésre a nehézség.
- **8. sor**: A *nonce* érték, amely minden egyes hash kipróbálásánál inkrementálódni fog 0-tól kezdve 2573394689-ig.
- **9. sor**: A tranzakciók száma a blokkban, ebben a példában 1.
- **10. sor**: A blokk mérete byte-okban.
- **11. sor – 35. sor**: Tranzakciók felsorolása és kifejtése. Ebben az esetben csak egy tranzakció van. A paraméterek a korábban leírt példa tranzakcióhoz hasonlóan értendők. Ebben az esetben annyi érdekesség van, hogy *scriptSig* helyett *coinbase* található. Ez azt jelenti, hogy a tranzakció bemenetének típusa *Generation*, nem pedig *Address*, tehát bitcoin kerül be a hálózatba, amely a blokkot érvényesítő bányászt illeti.
- **36. sor – 37. sor**: A Merkle fa, az értéke megegyezik az 5. sorban megadott *Merkle root* értékével.

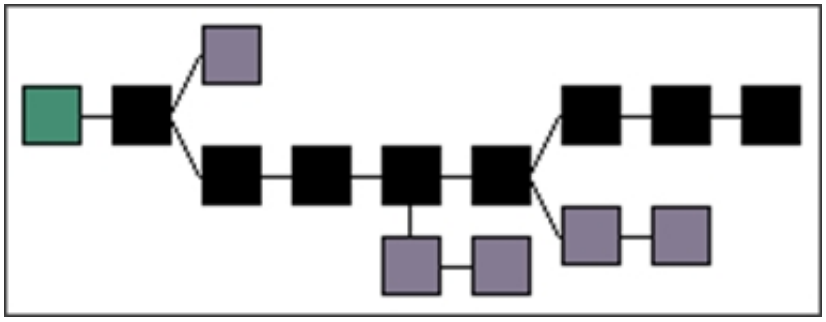
2.1.3. Blokklánc

Az így képzett blokkok egy láncot képeznek, amely minden felhasználó gépén megtalálható. Ebből a láncból ki lehet nyerni, hogy egyes felhasználók milyen összeggel rendelkeztek bármely időpillanatban.

A legelső blokk a *Genesis block*, ez a lánc eleje, az egyetlen olyan blokk, amely nem tartalmazza egy korábbi blokk hashét. Egyetlen tranzakció volt benne, amelyben 50 BTC került jóváírásra az 1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa című felhasználó számára.

Innentől kezdve az összes blokk már egy korábbi blokkhoz kapcsolódik hozzá és annak hashére hivatkozik. A hálózat mindig a leghosszabb láncot veszi figyelembe. Létrejöhet ugyanis olyan eset, amikor két különböző felhasználó közel azonos időben oldja meg a matematikai problémát, így két blokk (legyen ez A és B) egyszerre kerül be a láncba, azaz a blokkláncban elágazás jön létre. Erre az a megoldás, hogy a hálózat nem tesz semmit és mindenki folytatja a bányászást. A felhasználók egy része az A blokk ágán folytatja a munkát, a maradék pedig a másikon. Egy idő után egy új blokk kerül érvényesítésre, például a B ágon, így az az ág hosszabb lesz, mint a másik. Ekkor az összes

felhasználó áttér az immár hosszabb B láncra és az A ágat figyelmen kívül hagyják. Az A blokkban található érvényes, de még nem elfogadott tranzakciók viszont újra bekerülnek a feldolgozásra váró tranzakciók sorába, így egy másik blokk majd magába foglalja őket. Természetesen az A blokkot érvényesítő felhasználó ekkor nem kapja meg a jutalmat.



2.4. ábra. Blokklánc [6]

A fenti ábrán egy példalánc látható. Zöld jelöli a *Genesis blokkot*, a fekete a leghosszabb használatban lévő láncot, a szürke pedig az olyan nem használt blokkokat, amelyek az előbb felvázolt példa esetében keletkezhetnek, és nem képezik részét a blokkláncnak.

2.1.4. Bányászás

A munkabizonyíték egy olyan adat, amelyet nehéz úgy előállítani, hogy teljesítsen egy bizonyos előre megszabott feltételt, viszont ha megvan az adat, akkor könnyű leellenőrizni, hogy valóban teljesíti-e azt. A Bitcoin esetében erre azért van szükségünk, hogy egy bitcoint ne lehessen többször elkölteni. Vegyük ugyanis azt az esetet, amikor A felhasználó át akarja verni B-t és C-t, azzal, hogy mindkettőnek ugyanazt a bitcoint küldi el, így kétszer költi el azt. Lehetne az a protokoll, hogy B ilyenkor szétküldi A „üzenetét" a teljes hálózatnak, majd megvárja, amíg más felhasználók visszajeleznek, hogy ez a bitcoin valóban A-hoz tartozik, és csak utána fogadja el azt, így C-nek már úgy jelez vissza a hálózat, hogy az a bitcoin már nem A-hoz tartozik. Ekkor azonban A csinálhatná azt, hogy valamilyen módszerrel különálló felhasználókat generál (több százmilliót) és átveszi a hálózat irányítását. Ekkor amikor B és C megkérdi a hálózattól, hogy a bitcoin A-hoz tartozik-e, akkor A generált felhasználói mindkét esetben igennel válaszolnak, így átverve B-t és C-t.

Ennek elkerülése végett került bevezetésre a *proof-of-work*, amelynek köszönhetően nem a felhasználók száma számít, hanem a számítási teljesítmény. Két előnye is van, egyrészt nehézzé teszi egy tranzakció érvényesítését, másrészt jutalmat is ad ezért, így motiválva a felhasználókat. A felhasználó tehát nem tudja kétszer elkölteni a bitcoinját, mert ahhoz óriási számítási teljesítményre lenne szüksége.

A munkabizonyíték működése is egy példán érthető meg legjobban. Legyen az alap stringünk a „This is the base string”. A célunk az, hogy egy olyan SHA-256 hasht találjunk ehhez, amelynek az első két karaktere 0. A stringhez hozzáfűzünk még egy *nonce* értéket, hogy a string változzon, majd ez minden hash kipróbálása után inkrementálódik.

hash(„This is the base string0”) = 040b0e0da1334eaf04944b79e32d197e39064239ec5b044ea53f8181b811324d
hash(„This is the base string1”) = d1abef6d0ed82a7ca17c1ccc7fe3bd5de11c13778b1be47eed1f505bb79e41fc
...
hash(„This is the base string32”) = 00a64756ee58d8e2048e2538f3f8e1ec2915cf620f96238a7ee68d4321af7b94

32 iteráció után kaptuk meg az első olyan értéket, amelynek első két karaktere 0. A feladat nehezíthető, ha pl. azt várjuk el, hogy az első 4 karakter legyen 0. Ekkor 131549 iterációra van szükségünk. Ha azt szeretnénk, hogy az első 8 is 0 legyen, akkor még sokkal hosszabb számításokra van szükségünk. A jelen példát százmillióig futtatva sem talált eredményt.

Tegyük fel most például, hogy D felhasználó talál egy olyan értéket, amely teljesíti a megszabott feltételeket. Ekkor szétküldi a tranzakciókból álló blokkot a hálózatban és hozzáfűzi a megtalált *nonce* értéket. A többi felhasználó egy pillanat alatt le tudja ellenőrizni, hogy a *nonce* helyes értéket ad-e, azaz D tényleg elvégezte-e a munkát. Ha igen, akkor az új blokk a hozzákerül a lánchoz.

Jogosan felmerülő kérdés, hogy egy felhasználó miért szánná a számítási teljesítményét a tranzakciók érvényesítésére. Erre lett kitalálva a jutalom rendszer. Az a felhasználó, akinek sikerül érvényesíteni egy blokkot 50 BTC-t kap kezdetben. Ez az érték 210000 érvényesített blokkonként feleződik, ami azt jelenti, hogy 2140-re ez az érték 10^{-8} érték alá fog csökkeni, amely a Bitcoin legkisebb egysége. Ekkorra bitcoinok száma eléri a maximumát, így nem fog több bekerülni a rendszerbe. A tranzakciók érvényesítése viszont továbbra is megéri majd, hiszen a lehetőség van tranzakciós díj mellékelésére, amely a bányászt illeti majd, aki segített az érvényesítésben. Kezdetben a tranzakciós díjak rendre nullák voltak, de minél kisebb lesz az érvényesítések díja, a tranzakciós díjak annál jobban felértékelődnek.

A Bitcoin nem a korábban példaként felhozott *proof-of-work-öt* használja, azaz nem olyan hasheket keres, amelyeknek bizonyos számú karaktere 0 az elején. Olyan értékek oldják meg a feladatot, amelyek egy bizonyos *target* értéknél kisebbek, vagy egyenlőek. Ez a *target* úgy van szabályozva, hogy a feladat nehézségének köszönhetően nagyjából 10 percenként keletkezzen új blokk.

A tranzakciók blokkba rendezését és a matematikai feladat megoldásának folyamatát nevezzük bányászásnak. Az egészet a bitcoin és arany közötti párhuzammal a legkönnyebb megérteni. Az arany esetében szó szerinti bányászás történik és az így befektetett fizikai munka gyümölcse az arany. A Bitcoin esetében a befektetett munka a számítási teljesítmény és az áram, a jutalom pedig a bitcoin. Ugyanúgy ahogy az arany esetében, itt is megtörténhet az, hogy valaki hiába bányászik, nem talál semmit, azaz hiába számítja a hasheket, nem talál olyat, amely megfelelő lenne.

2.2. Kritika és gyengeségek

Annak ellenére, hogy a Bitcoin egy jól átgondolt rendszer, természetesen több kritika is érte már. Ebben az alfejezetben a fontosabb gyengeségek és bírálatok kerülnek kifejtésre.

2.2.1. Defláció kérdése

A központosított gazdaságokban a pénzt egy központi jegybank felügyeli és bocsátja ki, figyelembe véve a gazdát cserélő áruk értékét és forgalmának növekedését, hogy azokkal nagyjából stabil áron lehessen kereskedni [7]. Ezzel szemben egy teljesen decentralizált rendszerben nincsen semmilyen központi szerv, hanem egy peer-to-peer hálózat csomópontjai hozzák létre a pénzt, egy előre meghatározott algoritmus szerint, előre megadott ütemben és mértékben.

A Bitcoin rendszer úgy lett kitalálva, hogy a teljes pénzkészlet véges legyen. Az első évben 2625000 bitcoin került a hálózatba, ahogy az ezt követő három évben is. Ezután ez az összeg feleződött, így az ezután következő négy évben (2016-ig) négyszer 1312500 bitcoin keletkezik. Az egész folyamat így megy majd 2140-ig, amikor is a teljes bitcoin készlet eléri a 21 milliót és efelé már nem megy. Annak ellenére, hogy ennél több soha nem lesz, a részleges tartalékra épülő bankrendszer elterjedésével ennél jóval több bitcoin kerülhet majd használatba.

Az defláció kérdése a Bitcoin megjelenése óta folyamatosan terítéken van, és azóta sincs egyetértés arról, hogy jó dolog-e. Egyesek azt mondják, hogy a fix pénzkészlet miatt deflációs spirál jöhet létre. Ez azt jelenti, hogy a csökkenő árak pénzfelhalmozásra ösztönöznek, így egyre kevesebb pénz marad a piacon, tehát az árak tovább csökkennek. Ez a hagyományos bankrendszerek esetében adósságtörlesztésbe tereli a pénzt, amely először lelassítja, majd teljesen megöli a gazdaságot.

A Bitcoin mellett állók erre azt mondják, hogy ez csak a hagyományos pénzeket érinti, így a Bitcoin rendszerére ez nem igaz. A bitcoin esetében csak akkor lesz defláció, ha maga a hálózat nő és a gazdaság fejlődik. Véleményük szerint a deflációt azért tartják rossznak a mai világban, mert mindig váratlanul jön, szemben a Bitcoin rendszerével, ahol mindenki tudhatja előre, hogy a pénze az idő múlásával többet fog érni. Ugyanakkor a fix pénzkészlet miatti folyamatos értéknövekedés miatt a bitcoin árfolyam teljesen képlékeny, amely a normál kereskedésnél hátrányt jelent, hiszen egy átlagos adás-vételnél a felek nem spekulálni akarnak, csak termékeket eladni, vagy venni. [8]

A pénzfelhalmozás miatt történő folyamatos defláció egyre nehezebbé tenné az újonnan bekapcsolódó felhasználók helyzetét, amelyre egyes vélemények szerint egy konkurens alternatív rendszer lehetne a megoldás, amelyben a rendelkezésre álló érmék száma tranzakciók összértékével arányosan növekedne. A Bitcoin alapjait hordozó Peercoin például már egy fix 1%-os inflációval lett megtervezve, ahol a pénzkészlet sem fix.

Pénzfelhalmozás esetén létrejöhet olyan eset is, hogy egy felhasználó nagy mennyiségű bitcoint halmoz fel (legyen ez 10000 BTC), ugyanakkor a defláció miatt már egy bitcoin

is rengeteget (legyen 10000 dollár) ér (ez magában nem probléma, mivel a rendszer úgy lett megtervezve, hogy egy bitcoin 10^8 részre osztható). Ekkor, ha ez a felhasználó egyszerre elkölti ezt az összeget, az igencsak megbolygathatja a gazdaságot és az árfolyamot. Ez az eset azonban a mostani hagyományos gazdaságban is megtörténhet abban az esetben, ha egy gazdag ország úgy dönt, hogy eladja teljes dollár készletét. [9] [10]

2.2.2. Nagy számítási teljesítmény egy kézben

Ha egy támadó a teljes hálózat számítási teljesítményének legalább 50%-át birtokolja, akkor átveheti az irányítást a teljes rendszer felett, amíg ez a feltétel teljesül. Ez alatt azt kell érteni, hogy valamilyen szinten rendelkezhet a tranzakciók felett. Egyrészt egy bitcoint többször is elkölthet, másrészt pedig eldöntheti az egyes tranzakciókról, hogy elfogadásra kerüljenek-e és megakadályozhatja, hogy a felhasználók érvényes blokkokat bányásszanak. Ezek ellenére viszont a támadó nem módosíthatja mások tranzakcióit, nem változtathatja meg a bányászásért kapott jutalmat, nem hozhat létre bitcoinokat a semmiből és nem költhet el olyan pénzt, amely nem tartozott hozzá. [11]

Nyilvánvalóan hasonló jellegű támadások lehetnek akkor is, ha például csak 40%-t birtokolja a felhasználó a teljes számítási teljesítménynek, de így kisebb az esélye arra, hogy a próbálkozása sikeres legyen.

Annak ellenére, hogy ez a sebezhetőség jelen van, nem valószínű, hogy valaki is megpróbálna élni vele, mivel az óriási számítási teljesítmény szükséglet ellenére nem ad akkora hatalmat a hálózat felett.

A Peercoin rendszerében erre is van egy megoldás. Míg a Bitcoin rendszer ilyen jellegű megtámadásához a tejes számítási teljesítmény 51%-át kell birtokolni, addig a PPCoin esetében ezzel szemben az összes PPCoin 51%-át, amelynek ára jelentősen nagyobb, mint adott esetben egy olyan nagy számítási kapacitású rendszer kiépítése, amely a hálózat teljesítményének több mint a felét kiteszi. Arról nem is beszélve, hogy ha a támadó az összes PPCoin több, mint 50%-át birtokolja, akkor neki egyáltalán nem lenne célja az, hogy ezek után a hálózatot megtámadja, hiszen akkor maga és a saját pénze ellen játszik.

2.2.3. Selfish mining

A Bitcoin hálózat megjelenése óta több bányásztársulás alakult, amelynek lényege, hogy a társulásba belépő bányászok együtt dolgoznak a blokkok érvényesítéséért, a jutalmat pedig a beleadott számítási teljesítmény alapján elosztják.

A *selfish mining* egy olyan támadás, amely a bányászási folyamatot támadja és ennek résztvevői több jutalmat kapnak, mint amennyi méltányos lenne egy normál bányásztársulásban.

A támadásban résztvevő felhasználók ugyanúgy egy bányásztársulást alkotnak, mint a többiek és ugyanúgy is bányásznak. A különbség csak az, ők titokban tartják, amikor hitelesítenek egy blokkot és csak bizonyos esetekben teszik publikussá.

Az algoritmus a következőképpen működik. Legyen a normális blokklánc a publikus ág, a *selfish miner-ek* titokban tartott ága pedig a privát ág. Ekkor a bányászás során a következő esetek történhetnek meg:

- A publikus ág hosszabb, mint a privát ág: Ekkor mivel a *selfish pool* teljes számítási kapacitása jóval kisebb, mint a többieké, esélytelen a saját privát águkat érvényesíteni, így átveszik ők is a publikus ágat.
- A *selfish pool* hitelesít egy blokkot, így előnyben van: Ők ezt a blokkot titokban tartják, innentől pedig két eshetőség van:
  - A többiek találnak egy blokkot, így megszüntetve a *selfish pool* előnyét: A *selfish pool* ekkor azonnal publikálja az általa korábban talált blokkot, a tagjai pedig ezt a láncot fogják folytatni. A többiek a Bitcoin algoritmusnak megfelelően választanak a kettő közül. Ha a *selfish pool* láncára épül hamarabb a következő blokk, akkor a *selfish pool* és a többiek is egy blokknyi nyereséget kapnak, ha a többiekére, akkor a *selfish pool* nem kap semmit.
  - A *selfish pool* talál egy újabb blokkot, ezzel kettőre növelve az előnyét: A *selfish pool* folytatja a bányászást és mindig publikálnak egy blokkot, amint a többiek is találnak egyet. Mivel a *selfish pool* számítási teljesítménye jóval kisebb, ezért egy idő után biztosan feljönnek a többiek, de amikor már csak egy az előny, akkor publikálják a teljes láncukat ezzel megszerezve minden nyereséget.

Az algoritmust publikálók arra jutottak, hogy a *selfish pool* jövedelmét az alábbi képlet szemlélteti, ahol γ annak a valószínűsége, hogy a többiek a *selfish pool* blokkját választják a 2/a esetben, α a *selfish pool*, $1-\alpha$ pedig a többiek erejét jelöli:

$$R_{pool} = \frac{r_{pool}}{r_{pool}+r_{others}} = ... = \frac{\alpha(1-\alpha)^2(4\alpha+\gamma(1-2\alpha))-\alpha^3}{1-\alpha(1+(2-\alpha)\alpha}$$

Az elméleti elképzelésük szimulálása után arra a megfigyelésre jutottak, hogy adott γ esetén, α számítási erővel rendelkező *selfish pool* nagyobb jutalmat kap, mint amennyit érdemelne, ha α az alábbi tartományban van (feltéve, hogy α 0 és 0.5 között található):

$$\frac{1-\gamma}{3-2\gamma} < \alpha < \frac{1}{2}$$

Ha a *selfish pool sybil támadást* (kifejtve a későbbi alfejezetben) alkalmaz, azzal elérheti γ növekedését, mivel mire a többiek blokkjáról tudomást szerezne egy csomópont, addigra az már a *selfish pool* blokkján dolgozik.

Megoldásként a protokoll módosítását javasolják. Amikor egy csomópont konkurens ágakat talál, akkor mindkettőt továbbítsa, és egyenletes eloszlás szerint véletlenszerűen válassza ki, hogy melyiken dolgozik majd. Ennek köszönhetően γ 0.5 lesz, a küszöbérték pedig 0.25. Ezen változtatás alkalmazása csökkenti annak az esélyét, hogy a *selfish pool* növelje γ értékét az adattovábbítás irányításával. [12]

2.2.4. Decentralizáltság kérdése

Ez a deflációhoz hasonlóan egy olyan pont, amely megosztja a véleményeket. Alapvetően az egész Bitcoin rendszer megalakulásának az volt a célja, hogy egy nem bizalom alapú pénznemet hozzanak létre, amely mentes minden köztes médiumtól és pénzügyi intézménytől. Ennek köszönhetően a küldött pénz a feladótól egyből a címzetthez kerül, és nem megy át közben semmilyen harmadik félen.

A decentralizáltsággal viszont az is együtt jár, hogy a hagyományos pénzügyi termékekkel ellentétben a Bitcoin esetében nincsen olyan közbenső szervezet, amely jogilag közbeléphetne, ha valakinek ellopják a bitcoinjait. Ha valakinek a bankkártyáját csalással használják, vagy a bankja befuccsol, vannak törvények, amelyek korlátozzák a fogyasztói veszteségeket.

A Bitcoin esetében is volt valamennyire centralizált lépés, amikor *checkpoint* került bele. Ennek a lényege az, hogy a blokklánc lezárásra kerül a *checkpoint-ig* bezárólag, azaz egy esetleges 50%-nál nagyobb számítású teljesítménnyel rendelkező támadás se tudja megváltoztatni az ezen pont előtt történteket. *Checkpoint* viszont csak szoftverfrissítésnél kerülhet a rendszerbe, a Peercoin-nal ellentétben, ahol folyamatos a *checkpoint* képzés. [13]

2.2.5. További gyengeségek

Annak ellenére, hogy a Bitcoin támogatja az anonimitást, az érmék „útvonalát" végignézve a címeket identitásokhoz lehet kötni. Mivel minden tranzakció publikus, ezért ha az egyik tranzakció címe kiderül, hogy kihez tartozik, akkor innen indulva esetleg kitalálható, hogy a többi cím kikhez tartozik.

A *sybil attack* névre hallgató támadás esetében a támadó általa irányított csomópontokkal töltheti meg a hálózatot, amelyeknek számítási teljesítménye nincsen, de a hálózat részét képezik. Ennek következtében a támadó megteheti, hogy mások tranzakcióit és blokkjait nem továbbítja, így levágva őket a hálózatról és kitéve őket például *double-spending* támadásnak. A Bitcoin hálózat alacsony késleltetésű átviteleit könnyen meg lehet akadályozni egy időzített támadással, ha a felhasználó több támadó csomóponthoz is kapcsolódva van.

A Bitcoin hálózat *Denial of Service* (DoS) támadással is sebezhető, hiszen ha nagy mennyiségű adatot küldenek egyszerre egy csomópontnak, akkor annyira le lehet azt foglalni, hogy Bitcoin tranzakciókat ne tudjon feldolgozni. A rendszernek vannak erre beépített védekező mechanizmusai, de a kifinomultabb DoS támadások így is problémát okozhatnak. [11]

3. fejezet

Peercoin

A Peercoin egy olyan kriptopénz, amely a Bitcoin rendszerét fejlesztette tovább, kijavítva egyes gyengeségeit. A hálózat fenntartásához hosszú távon jóval kevesebb energiára van szükség és az 51%-os támadással szemben is védve van.

3.1. Architektúra

A Bitcoin 2008-as megalakulása óta a *proof-of-work* volt az alap koncepció a kriptopénzek tervezésénél. A Peercoin azonban egy *proof-of-work* és *proof-of-stake* hibridet használ a blokkok generálására. A Bitcoin esetében ahogy a hálózat közeledik a 21 millió kibányászott bitcoinhoz, annál kevésbé éri meg a bányászás az ezért kapott jutalom feleződése miatt, így pedig a tranzakciós díjak emelésére van szükség, hogy a teljes hálózat biztonsági szintje megmaradjon és továbbra is nagy legyen a teljes számítási teljesítmény. A Peercoin többek között erre kínál megoldást egy olyan kriptopénzzel, amelynek az idő múlásával egyre kevésbé van szüksége nagy számítási teljesítményhez a hálózat biztonságban tartásáért. [14]

Az rendszer alapvető alkotóelemei a Bitcoinon alapulnak, de szükség van néhány ettől eltérő koncepció áttekintésére.

3.1.1. Coin age

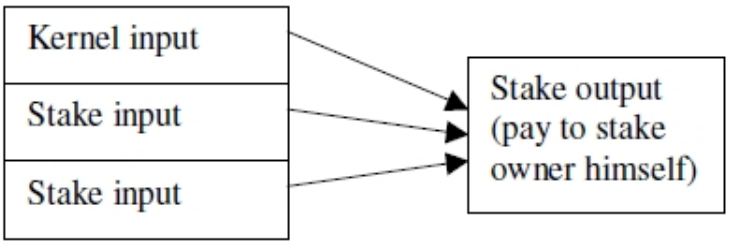
A coin age egy könnyen meghatározható érték, amely már a Bitcoin készítői számára is ismert volt, de ők csak tranzakciók rangsorolására használták. Kiszámításához a rendelkezésre álló pénz mennyiségét kell megszorozni annyival, amennyi ideje az rendelkezésre áll. Ha tehát A 10 peercoint kap B-től és azt 50 napig magánál tartja, akkor az ebből számított *coin age* 10 x 50 = 500. Viszont amint A elkölti ezt a 10 peercoint, vele együtt a *coin age* is eltűnik. A könnyebb meghatározhatóság végett a tranzakciók ki lettek bővítve egy időbélyeg mezővel.

A Peercoin protokoll a *coin age* alapján választja ki, hogy melyik az aktuálisan leghosszabb blokklánc. Az egyes tranzakciók által elhasznált *coin age* összeadódik a blokkokban, majd az a lánc számít a leghosszabbnak, amelyben a legnagyobb a teljes elhasznált *coin age*.

3.1.2. Proof-of-stake és minting

A Peercoin is *proof-of-work* rendszerként indult, de az új peercoinok hálózatba kerülésével szép lassan átvált a *proof-of-stake* használatára. A *proof-of-stake* ötlete már a Bitcoin keretein belül is felmerült, de az az elképzelés, hogy ezt blokkok generálására és a biztonság növelésére is használni lehet, csak a Peercoinnál került elő. A *proof-of-work-höz* hasonlóan a *proof-of-stake* hamisítása is igen nehéz.

A Peercoin hibrid rendszerében két féle blokk létezik, a Bitcoinnal analóg *proof-of-work* blokk és a *proof-of-stake* blokk. Bitcoin esetében *coinbase-nek* hívták azt a speciális tranzakciót, amikor új bitcoin kerül a hálózatba. A *proof-of-stake* blokkok esetében is van ilyen tranzakció, amelynek neve *coinstake*. Az ilyen tranzakciós blokkokban a peercoin tulajdonosa fizet magának, ezzel elhasználva a *coin age-ét*, cserébe viszont új blokkot generálhat, a teljes lánc által elfogyasztott *coin age* pedig növekszik. Az ezért kapott jutalom pedig 1%-ra van állítva, azaz gyakorlatilag az elköltött összeg 101%-át kapja meg.



3.1. ábra. Coinstake tranzakció felépítése [14]

A *coinstake* tranzakciók felépítése a fenti ábrán látható. Az első input a *kernel*, amelynek a *proof-of-work-höz* hasonlóan bizonyos feltételt kell teljesítenie, így a *proof-of-stake* blokkok generálása is egy sztochasztikus folyamat, viszont a hashelés korlátos tartományon történik, a *proof-of-work* végtelen tartományával szemben. Ennek köszönhető az, hogy az ilyen blokkok generálása nem igényel nagy számítási teljesítményt. A teljesítendő feltétel minden felhasználó esetében más, szemben a Bitcoinnal, ahol az összes node egy fix *target* értéket kap. Minél nagyobb *coin age-el* rendelkezik egy felhasználó, annál könnyebben teljesíti a hash target protokollt. Ha tehát A 200 coin évet (*coin-year*) halmozott fel és 2 nap alatt generálna egy *kernelt*, akkor B, aki csak 100 coin évvel rendelkezik, 4 nap alatt generálná azt. Tehát egy felhasználó *coinstake* blokk generálásának valószínűsége arányos az általa felhalmozott *coin age* mennyiségével. Aki például a hálózatban rendelkezésre álló peercoin-ok 1%-át birtokolja, az a *proof-of-stake* blokkok 1%-át tudja generálni.

A Bitcoinhoz hasonlóan itt is létezik a bányászás (*mining*), de a *proof-of-stake* blokkok generálását *minting-nek* nevezik. Minden egyes érméhez *coin age* tartozik, a felhasználók *stake-je* pedig ebből számolódik. Egy érme akkor vehet részt a *minting* folyamatában, ha a felhasználó legalább 30 napig tartotta azt. Az érmék „életkora” 90 nap után viszont nem nő tovább.

A Peercoin esetében a *proof-of-work* és *proof-of-stake* blokkok generálásának nehézsége is folyamatosan igazodik a rendszerhez, nem csak kéthetente, mint a Bitcoin protokollban.

3.1.3. Checkpointing

Az esetleges támadások ellen egy úgy nevezett *checkpointing* lett bevezetve, melynek lényege az, hogy egy adott *checkpoint* előtt történt tranzakciók nem lesznek módosíthatóak, addig a pontig bezárólag a blokklánc gyakorlatilag be van fagyasztva. 2010-ben a Bitcoin láncába is bekerült egy *checkpoint*, de a Peercoinnál ez folyamatosan történik, naponta többször. Ez a megoldás centralizált lépésnek mondható, de ezt alkalmazzák, amíg valaki ki nem talál egy elosztott *checkpoint* módszert.

4. fejezet

Egyéb kriptopénzek

A Bitcoin megjelenése óta rengeteg kriptopénz került forgalomba, több-kevesebb sikerrel. Sok közülük a Bitcoin rendszerén alapul, de elsőségét még egyik sem tudta elvenni. A most következő alfejezetekben pár érdekesebb konstrukció kerül bemutatásra.

4.1. Litecoin

A Litecoin is a Bitcoin protokollon alapul, de a bányászáshoz nem kellenek külön arra specializált hardver eszközök. A tranzakciók megerősítése négyszer gyorsabb (átlagosan 2.5 perc), mint a Bitcoin esetében. A proof-of-work-nél a scrypt hash függyvényt használja az SHA-256 helyett. A protokoll szerint összesen 84 millió litecoin kerül majd kibányászásra. A blokkok bányászásának nehézsége ugyanúgy 2016 blokk után kerül igazításra, viszont mivel a Litecoin hálózatban négyszer gyorsabban generálódnak új blokkok, így ez a változás 3.5 naponta történik.

4.2. Namecoin

A Namecoin egy Bitcoin protokollon alapuló elosztott tartománynév-rendszer (DNS), amely domainnevek regisztrálását, adásvételét és frissítését teszi lehetővé. Kialakítását az kezdeményezte, hogy a jelenlegi DNS-rendszer esetében a DNS-szerverek gazdái akármikor letilthatják a hozzájuk tartozó domainneveket. A Namecoin hálózatában ez nem lehetséges, mivel konkrét szerverek nincsenek, így nem rendelkezik fölötte egyetlen központi hatóság sem. A domainnevek digitális aláírásokkal és munkabizonyítékokkal védve vannak.

4.3. Primecoin

A Primecoin fejlesztői olyan protokollt vezettek be, ahol a bányászásnak van egy másodlagos haszna is a hálózat biztonságának biztosítása mellett, méghozzá prímszám láncok keresése. A bányászás nehézsége blokkonként hangolásra kerül, a tranzakciók megerősítése pedig percenként történik. A blokkokért járó jutalom mindig 999 leosztva az aktuális nehézségi érték négyzetével.

4.4. Mastercoin

Az eddig bemutatott kriptopénzekkel szemben a Mastercoin nem csinált magának saját blokkláncot, ehelyett a Bitcoin rendszer már meglévő láncát használja, ennek köszönhetően már a legelejétől hatalmas hashelési teljesítmény védte. Az alap ötlet szerint a Mastercoin egy alkalmazás réteget szolgáltat az alsó Bitcoin rétegre, olyasmi ez, mint pl. a HTTP és a TCP/IP közötti kapcsolat. A mastercoinok nem bányászással keletkeztek, hanem egy egy hónapos támogatószerző folyamat során kerültek a rendszerbe. A támogatók 100 MSC-t kaptak minden egyes beküldött BTC-ért, ezen kívül további bónuszt attól függően, hogy milyen korán tették meg ezt. Így végül összesen 619478.6 MSC került a hálózatba. A Mastercoin rendszer legnagyobb előnye a sok általa bevezethető funkció. A felhasználók pénznemeket definiálhatnak, majd ezek között válthatnak. Lehetőség van fogadást regisztrálni egy másik féllel, hogy egy adott ár egy bizonyos érték alatt, vagy fölött lesz egy jövőbeli időpontban, ill. a rendszer kvázi letéti számlát is kezelni tud.

4.5. Zerocoin

A Zerocoin a Bitcoin protokoll egy kibővítése, amely magasabb szintű anonimitást biztosít. A Bitcoin hálózat esetében különböző adatbányász módszerekkel meg lehet határozni, hogy egyes címek mely identitásokhoz tartoznak. Mivel minden tranzakció elmentésre kerül a blokkláncban, azok teljes útja visszakövethető. Ha egy felhasználó például az interneten közzé teszi a saját Bitcoin címét, ezt felhasználva már sok dologra következtetni lehet. A Zerocoin erre ad megoldást az úgy nevezett pénzmosodáival.

5. fejezet

Kriptopénzek összehasonlítása

Rengeteg kriptopénz van jelenleg is forgalomban, és nagyrészt a Bitcoin alapjain nyugszanak. Ebben a fejezetben a három jelenlegi legnagyobb koncepció táblázatos összehasonlítása látható. A kibányászott készlet, a piaci tőke és az érték mezők a táblázatban a 2014.04.27. napi állást mutatják.

5.1. táblázat. A három legnagyobb kriptopénz összehasonlítása

	Bitcoin	**Litecoin**	**Peercoin**
Megjelenés	2009.01.03.	2011.10.07.	2012.08.12.
Egység	BTC	LTC	PPC
Érték	$ 455.39	$ 10.6	$ 2.22
Piaci tőke	$ 5,783,101,141	$ 295,150,916	$ 47,455,416
Kibányászott készlet	12,699,350 BTC	27,841,204 LTC	21,352,809 PPC
Teljes pénzkészlet	21 millió	84 millió	korlátlan
Legkisebb egyég	0.00000001 (satoshi)	0.00000001 (spark)	0.00000001
Hash algoritmus	SHA-256	Scrypt	SHA-256
Bányászás	proof-of-work	proof-of-work	proof-of-work proof-of-stake hibrid
Jutalom	kezdetben 50 BTC blokkonként, 4 évente feleződik	kezdetben 50 LTC blokkonként, 4 évente feleződik	proof-of-work: feleződik, proof-of-stake: évi 1%
Nehézség hangolása	2016 blokkonként 14 nap	2016 blokkonként 3.5 nap	bányászok száma alapján folyamatos
Tranzakciós díj	nincs	nincs	0.001 PPC

Értelemszerűen mindhárom pénz sok azonosságot hordoz, mivel ugyanazon az alapon nyugszanak, de egyes fontos pontokban eltérnek. A Bitcoin és a Litecoin egyaránt *proof-of-work*-öt használ a blokkok generálására és a hálózat biztonságban tartása miatt, a Peercoin ugyanakkor egy *proof-of-work* és *proof-of-stake* hibridet, amelynek köszönhetően jóval nagyobb biztonságot ad az 51%-os támadások ellen. A hash algoritmus alapján viszont a Peercoin hasonlít jobban a Bitcoinra, mivel az SHA-256 van használatban, a Litecoin Scryptjével szemben. A bányászásért kapott jutalom a Bitcoin és Litecoin esetében teljesen azonos, a Peercoinnál ez már azért is más kell, hogy legyen, mert két féle blokktípus létezik. A bányászás nehézségének finomítása az első két esetben azonosan (habár a Litecoin rendszerében ez hamarabb történik meg, mivel gyorsabb a blokkgenerálás), meghatározott blokkszám után történik, míg a Peercoinnál ez folyamatosabb. Meghatározó különbség az is, hogy az első két kriptopénz teljes pénzkészlete korlátos, és tranzakciós díj sincsen.

6. fejezet

Összefoglalás

A kriptopénz egy olyan matematikai és kriptográfiai megoldásokon alapuló digitális pénz, amelynek elsődleges célja a tranzakciók gazdaságosabbá tétele és a köztes harmadik fél kizárása.

Ezen dolgozat betekintést nyújt a kriptopénzek világába, ismerteti a jelenlegi legmodernebb koncepciókat és elmagyarázza, hogy a tervezésüknél milyen szempontok voltak figyelembe véve. Az első crypto currency koncepciója részletesen ki van fejtve, a rendszer architektúrájától kezdve, az őt érintő gyengeségekig. A Bitcoin mellett részletezve lett a Peercoin, amely ugyan nem szerzett még akkora hírnevet, de az eltérő koncepciója miatt fontos volt jobban kitérni rá. Ezen két megoldáson kívül több kisebb kriptopénz is érintésre került.

Látható, hogy a 2009-ben megjelent Bitcoin óriási innovációt vitt a hagyományos pénzvilágba és azóta is számtalan hasonló koncepció került implementálásra. Az ideális kriptopénz matematikailag bonyolult kell, hogy legyen, mégis könnyen megérthető az egyszerű felhasználók számára. Fontos a decentralizált tervezés, megfelelő biztonság és felhasználói védelem mellett. Meg kell őrizni az anonimitást az adócsalás és más törvényellenes cselekedet lehetőségének elkerülése közben.

A lehetőségek tárháza igen nagy, egy majdani minden szempontból kiforrott kriptopénz komoly vetélytársa lehet a hagyományos alapokon nyugvó pénzrendszernek.

Irodalomjegyzék

[1] Kriptopénz és fizetés – fizikai pénzből digitálisba [Online]. Available: http://bitcoin.hu/kriptopenz-es-fizetes-fizikai-penzbol-digitalisba/

[2] Reid, F. and Harrigan, M., "An Analysis of Anonymity in the Bitcoin System," in Privacy, Security, Risk and Trust (PASSAT), 2011 IEEE Third International Conference on and 2011 IEEE Third International Confernece on Social Computing (SocialCom), Oct. 9-11, 2011, pp. 1318 - 1326.

[3] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," Nov. 1, 2008.

[4] Bitcoin Block Explorer [Online]. Available: http://blockexplorer.com/

[5] Dorit Ron and Adi Shamir, "Quantitative Analysis of the Full Bitcoin Transaction Graph," in Financial Cryptography and Data Security, 2013, pp. 6-24.

[6] Bitcoin Block chain [Online]. Available: https://en.bitcoin.it/wiki/Block_chain

[7] Bitcoin Controlled supply [Online]. Available: https://en.bitcoin.it/wiki/Controlled_supply

[8] Spekulációról és felhalmozásról [Online]. Available: http://bitcoin.hu/spekulaciorol-es-felhalmozasrol/

[9] Deflációs spirál [Online]. Available: http://bitcoin.hu/bevezeto/gazdasagi-kerdesek/deflacios-spiral

[10] Bitcoin Deflationary spiral [Online]. Available: https://en.bitcoin.it/wiki/Deflationary_spiral

[11] Bitcoin Weaknesses [Online]. Available: https://en.bitcoin.it/wiki/Weaknesses

[12] Ittay Eyal and Emin Gün Sirer, "Majority is not Enough: Bitcoin Mining is Vulnerable," Department of Computer Science, Cornell University, Ithaca, Nov. 15, 2013.

[13] Ben Laurie, "Decentralised Currencies Are Probably Impossible But Let's At Least Make Them Efficient," Jul. 5, 2011.

[14] Sunny King and Scott Nadal, "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake," Aug. 19, 2012.