

Kriptovaluták elméleti háttere

Ladóczki Bence

docens

Távközlési és Mesterséges Intelligencia Tanszék (TMIT)

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar (VIK)

ladoczki.bence@vik.bme.hu

2026. október 14.

Tartalomjegyzék

1	Bitcoin	2
1.1	A Bitcoin rendszer összefoglalása	2
1.2	Biztonsági modell	2
1.3	Gazdasági ösztönzők	2
2	PoW bányászat	3
2.1	Hardver és energiafogyasztás	3
2.2	Reorganizáció és MEV	3
2.3	Önző bányászat	4
2.3.1	A selfish mining stratégia	4
2.3.2	A modell és a valószínűségi analízis	4
2.3.3	A Markov-lánc egyenletrendszere	5
2.3.4	Javasolt ellenintézkedés	5
2.3.5	Propagációs késleltetés hatása	6
2.4	A közlegelők tragédiája	6
2.5	Blockchain DDoS támadás	7
2.5.1	A támadás mechanizmusa	7
2.5.2	Miért működik a támadás?	7
2.5.3	Formális modell	8
2.5.4	Védekezés	9
2.6	Cryptojacking	9
2.6.1	A támadás módszere	9
2.6.2	Detekció	10
2.6.3	Elrejtési technikák	10

1 Bitcoin

A Bitcoin [1] az előző fejezetekben tárgyalt építőelemek szintézise: Chaum digitális pénz ötlete, Lamport elosztott konszenzusa, a bizánci hibatűrés, és Back Hashcash proof-of-work mechanizmusa. Satoshi Nakamoto 2008-ban egyesítette ezeket egy központi hatóság nélküli elektronikus pénzrendszerrel, amelynek első tranzakciói 2009 januárjában történtek.

1.1 A Bitcoin rendszer összefoglalása

A rendszer működését a korábbi fejezetekben már részletesen tárgyaltuk (digitális aláírásokkal megvalósított tranzakciók a 13. fejezetben, proof-of-work a 14. és 17. fejezetben). Itt a Bitcoin specifikus tervezési döntéseit foglaljuk össze:

- A tulajdonjogot ECDSA digitális aláírások (secp256k1 görbén) igazolják.
- A dupla-költés problémáját elosztott főkönyvvel oldja meg: a leghosszabb lánc (amelyhez a legtöbb proof-of-work tartozik) a kanonikus történet.
- A blokkok SHA-256 hash-sel vannak láncolva, a bányászok Hashcash-szerű részleges hash-ütközést keresnek.
- A nehézséget 2016 blokkonként (≈ 2 hét) állítják, hogy a blokkidő ≈ 10 perc maradjon.
- Minimális strukturális követelmények: bárki szabadon csatlakozhat és kiléphet a hálózatról, nincs regisztráció, nincs engedélyezés.

1.2 Biztonsági modell

A Bitcoin biztonsági garanciája a következő: ha a becsületes csomópontok rendelkeznek a számítási kapacitás többségével, a támadó esélye, hogy utólag felülírja a tranzakciós történetet, exponenciálisan csökken a megerősítések (confirmations) számával. Nakamoto megmutatta, hogy ha a támadó a hashrate q hányadával rendelkezik (ahol $q < 0.5$), akkor annak a valószínűsége, hogy z blokknyi hátrányt behozzon:

$$P(\text{sikeres támadás}) = \left(\frac{q}{1-q} \right)^z$$

Ez exponenciálisan csökken z -vel, ami indokolja a gyakorlatban használt 6 megerősítéses szabályt.

1.3 Gazdasági ösztönzők

A bányászok motivációja kettős: (1) blokk-jutalom (block reward) – kezdetben 50 BTC, amely minden 210 000 blokk (≈ 4 év) után feleződik (halving), és (2) tranzakciós díjak. 2024 áprilisa óta (a negyedik halving után) a blokk-jutalom 3,125 BTC; a következő feleződés várhatóan 2028 tavaszán történik, amikor 1,5625 BTC-re csökken. A feleződési mechanizmus biztosítja, hogy összesen legfeljebb 21 millió BTC kerüljön forgalomba, ami deflációs pénzpolitikát eredményez. Hosszú távon, ahogy a blokk-jutalom nullához közelít, a tranzakciós díjak válnak a bányászok fő bevételi forrásává.

2 PoW bányászat

2.1 Hardver és energiafogyasztás

A Bitcoin bányászat hardvere jelentős fejlődésen ment keresztül: CPU-ról (2009) GPU-ra (2010), majd FPGA-ra (2011) és végül ASIC-okra (2013–). A mai bányászatot dedikált ASIC chippek dominálják (pl. Bitmain Antminer sorozat), amelyek kizárólag SHA-256 hash számítására vannak optimalizálva. A Bitcoin hálózat összesített hashrate-je 2024-ben meghaladta a 600 EH/s-t (exahash per másodperc).

A PoW bányászat energiafogyasztása jelentős: a Cambridge Bitcoin Electricity Consumption Index szerint a hálózat éves fogyasztása 100–150 TWh közé tehető, ami egyes kisebb országok teljes villamosenergia-felhasználásával összemérhető. A legnagyobb bányász-farmok oda települnek, ahol olcsó az energia (pl. Kína volt domináns 2021 előtt, azóta az USA, Kazahsztán és Skandinávia). A Bitmain piaci dominanciája (a bányász-hardver gyártásban) centralizációs aggályt vet fel: ha egyetlen gyártó kontrollálja a hardver-ellátási láncot, az befolyásolhatja az egész hálózatot.

A PoS (proof-of-stake) rendszerek ezzel szemben nagyságrendekkel kevesebb energiát fogyasztanak (az Ethereum Merge után a hálózat energiafogyasztása $\approx 99,95\%$ -kal csökkent), mivel nincs szükség tömeges hash-számításra.

2.2 Reorganizáció és MEV

A PoW fő problémája a véglegesség hiánya. Ha két bányász nagyjából egyszerre talál blokkot, a hálózat rövid időre kettéválik (fork): az egyik fele az egyik blokkot folytatja, a másik fele a másikat. A hálózati késleltetések miatt ez természetes jelenség. Végül az egyik ág hosszabbá válik, a másik eldobásra kerül (orphan block) – de nehéz előre megjósolni, melyik.

Ezt ki is lehet használni: ha egy támadó elegendő hashrate-tel rendelkezik, szándékosan építhet alternatív ágot (reorganizáció, „reorg”), amelyben a saját tranzakciói másképp rendeződnek. Kétféle reorg létezik:

- Célzott reorg: a támadó tudja, melyik tranzakciót akarja eltávolítani vagy átrendezni.
- Vak reorg: a támadó egyszerűen alternatív blokkokat épít anélkül, hogy specifikus tranzakcióra célozna, de a saját tranzakciós sorrendjét optimalizálja.

A reorganizáció különösen veszélyes a Maximal Extractable Value (MEV) kontextusában. Ha van decentralizált tőzsde (DEX) a blokkláncon, a tranzakciók sorrendje pénzben mérhető értéket hordoz. Egy támadó, aki hatással van a reorg-okra, a következő módon profitálhat:

- Frontrunning: ha a támadó meglát egy nagy értékű váltási tranzakciót a mempool-ban, a saját tranzakcióját elé helyezi.
- Szendvics támadás (sandwich attack): a támadó a célpont tranzakciója elé és mögé helyezi a sajátjait – előtte vásárol (felhajtva az árat), utána elad (a megemelt áron).
- MEV-extrakció reorg-gal: a támadó lemásolja egy blokk MEV-tartalmát, reorg-olja az eredeti blokkot, és a saját blokkjában hajtja végre ugyanazokat a nyereséges tranzakciókat.

A blokkok véglegességének lassítása tehát komoly probléma: kiszámíthatatlanná és lassúvá teszi a rendszert. Védekezni kell a rosszindulatú reorgok ellen – ez az egyik fő motiváció a PoS-ra való átállásra, ahol a finality (véglegesség) kriptográfiailag garantált és nem kell megerősítésekre várni.

2.3 Önző bányászat

2013-ban Eyal és Sirer a Cornell Egyetemről publikálták a *Majority is not Enough: Bitcoin Mining is Vulnerable* [2] című tanulmányt, amely megmutatta, hogy a Bitcoin bányászati protokoll nem tartalmaz megfelelő pénzügyi ösztönzöt: ha racionális bányászokat feltételezünk, létezik egy nem protokoll-követő stratégia (selfish mining), amely nagyobb hozamot biztosít, mint a becsületes bányászat.

2.3.1 A selfish mining stratégia

A stratégia lényege: amikor az önző bányász (vagy pool) megtalál egy blokkot, nem publikálja azonnal, hanem titokban folytatja a bányászatot a saját privát ágán. A becsületes bányászok ezalatt a nyilvános láncon dolgoznak, feleslegesen pazarolva a számítási kapacitásukat. A megfelelő időpillanatban az önző bányász felfedi a privát ágát, és ha az hosszabb, mint a nyilvános lánc, a hálózat elfogadja (a leghosszabb lánc szabálya miatt), a becsületes bányászok addigi munkáját eldobva.

2.3.2 A modell és a valószínűségi analízis

Legyen α az önző pool hashrate-aránya ($0 < \alpha < 0.5$), és $(1 - \alpha)$ a becsületes bányászoké. Definiáljuk γ -t mint annak a valószínűségét, hogy egy fork esetén a hálózat becsületes része az önző bányász blokkján kezd dolgozni (azaz milyen gyorsan terjed az önző bányász blokkja).

A rendszer Markov-láncként modellezhető, ahol az állapot a privát ág előnye blokkokban mérve. Az átmeneti valószínűségek:

- $0 \rightarrow 1$ állapot: az önző bányász talál blokkot (α valószínűséggel), titokban tartja.
- $1 \rightarrow 0$ állapot: a becsületes bányászok utolérnek, verseny alakul ki – γ valószínűséggel az önző bányász nyer.
- $k \rightarrow k + 1$: az önző bányász tovább növeli az előnyét.
- $k \rightarrow k - 1$: a becsületes bányászok közelítenek.

A stacionárius eloszlásból kiszámolható az önző pool relatív hozama. A fő eredmények:

- A selfish mining hozama szuperlineárisan nő a bányászati kapacitással (tehát az önző pool vonzza az újabb csatlakozókat).
- Ha $\gamma = 0$ (a hálózat mindig a becsületes ágat preferálja fork esetén), a stratégia $\alpha > 1/3$ felett válik profitábilissá.
- Ha $\gamma = 1$ (az önző bányász blokkja mindig gyorsabban terjed), a stratégia bármely $\alpha > 0$ mellett profitábilis.

- A jelenlegi Bitcoin implementációban γ közel 1-hez lehet, mert a csomópontok az először kapott blokkot továbbítják – és az önző bányász gyorsan terjesztheti a sajátját.

Ez azt jelenti, hogy a korábban gondolt 51%-os küszöb nem helyes: az önző bányászat már $\approx 25\text{--}33\%$ hashrate mellett is profitábilis.

2.3.3 A Markov-lánc egyenletrendszer

Jelölje p_k annak a valószínűségét, hogy az önző pool k blokknyi privát előnnyel rendelkezik, és $p_{0'}$ a versenyhelyzet (fork) valószínűségét. Az állapotvalószínűségek egyenletrendszere:

$$\begin{cases} \alpha p_0 = (1 - \alpha)p_1 + (1 - \alpha)p_2 \\ p_{0'} = (1 - \alpha)p_1 \\ \alpha p_1 = (1 - \alpha)p_2 \\ \forall k \geq 2 : \alpha p_k = (1 - \alpha)p_{k+1} \\ \sum_{k=0}^{\infty} p_k + p_{0'} = 1 \end{cases} \quad (1)$$

A rekurzióból ($\alpha p_k = (1 - \alpha)p_{k+1}$) következik, hogy $p_k = \left(\frac{\alpha}{1-\alpha}\right)^{k-1} p_1$ minden $k \geq 2$ -re. A normalizálási feltételből megoldva:

$$p_1 = \frac{\alpha - 2\alpha^2}{2\alpha^3 - 4\alpha^2 + 1}, \quad p_0 = \frac{p_1}{\alpha}, \quad p_{0'} = (1 - \alpha)p_1 \quad (2)$$

A pool és a becsületes bányászok bevétele az egyes esetekből (a fork kimenetelétől függően):

$$r_{\text{pool}} = p_{0'} \cdot \alpha \cdot 2 + p_{0'} \cdot \gamma(1 - \alpha) \cdot 1 + p_2 \cdot (1 - \alpha) \cdot 2 + \sum_{k>2} p_k(1 - \alpha) \cdot 1 \quad (3)$$

$$r_{\text{others}} = p_{0'} \cdot \gamma(1 - \alpha) \cdot 1 + p_{0'} \cdot (1 - \gamma)(1 - \alpha) \cdot 2 + p_0 \cdot (1 - \alpha) \cdot 1 \quad (4)$$

A pool relatív bevétele:

$$R_{\text{pool}} = \frac{r_{\text{pool}}}{r_{\text{pool}} + r_{\text{others}}} = \frac{\alpha(1 - \alpha)^2(4\alpha + \gamma(1 - 2\alpha)) - \alpha^3}{1 - \alpha(1 + (2 - \alpha)\alpha)} \quad (5)$$

A selfish mining profitábilis (azaz $R_{\text{pool}} > \alpha$), ha:

$$\frac{1 - \gamma}{3 - 2\gamma} < \alpha < \frac{1}{2} \quad (6)$$

Tehát $\gamma = 0$ esetén a küszöb $\alpha > 1/3$, $\gamma = 1$ esetén $\alpha > 0$ (bármilyen kis pool profitál), és $\gamma = 0.5$ esetén $\alpha > 1/4$.

2.3.4 Javasolt ellenintézkedés

Eyal és Sirer módosítást javasoltak a protokollra: ha egy csomópont két egymásnak ellentmondó blokkot kap, ne csak az elsőt továbbítsa, hanem mindkettőt. Ezáltal a becsületes bányászok véletlenszerűen ($\gamma = 0.5$) oszlanak meg a két ág között, ami a profitabilitási küszöböt $\alpha = 1/4$ -re emeli. Ez nem oldja meg teljesen a problémát, de jelentősen nehezíti az önző bányászatot.

2.3.5 Propagációs késleltetés hatása

Göbel, Keeler, Krzesinski és Taylor 2015-ös tanulmánya [3] egy kétdimenziós folytonos idejű Markov-lánccal modellezi a selfish mining-ot, ahol az állapot (k, ℓ) : a pool k blokkot épített a fork-pont után, a közösség ℓ -et. A pool blokktalálási rátája λ_1 , a közösségé λ_2 (ahol $\lambda_1 < \lambda_2$), és a pool-közösség közötti kommunikáció exponenciális μ paraméterrel.

Becsületes bányászat esetén az átmeneti ráták:

$$q((k, \ell), (k+1, \ell)) = \lambda_1, \quad q((k, \ell), (k, \ell+1)) = \lambda_2 \quad (7)$$

$$q((k, \ell), (0, 0)) = \mu \quad \text{ha } k \neq \ell \quad (8)$$

Selfish mining esetén a pool nem publikálja azonnal a blokkjait. Az átmeneti ráták:

$$q((k, \ell), (k+1, \ell)) = \lambda_1, \quad q((k, \ell), (k, \ell+1)) = \lambda_2 \quad (9)$$

$$q((k, \ell), (0, 0)) = \mu \quad \text{ha } k < \ell \text{ (a közösség nyer)} \quad (10)$$

$$q((k, k-1), (0, 0)) = \mu \quad \text{ha } k \geq 2 \text{ (a pool „bevált” az előnyéből)} \quad (11)$$

A stacionárius eloszlás egyenletei ($\ell > k$ esetén):

$$\pi(k, \ell)(\lambda_1 + \lambda_2 + \mu) = \pi(k-1, \ell)\lambda_1 \cdot \mathbf{1}_{k>0} + \pi(k, \ell-1)\lambda_2 \cdot \mathbf{1}_{\ell>0} \quad (12)$$

Numerikus példa a cikkből: $\lambda_1 = 0.6/h$, $\lambda_2 = 5.4/h$ (a pool a hashrate 10%-ával rendelkezik), $\mu = 720/h$ (5 másodperces átlagos propagációs késleltetés). A stacionárius valószínűségek becslétes bányászat esetén:

$$\pi(0, 0) = 0.9917, \quad \pi(1, 0) = \pi(0, 1) \approx 0.004$$

Selfish mining esetén az inkonzisztens állapotok valószínűsége jelentősen megnő:

$$\pi(0, 0) = 0.8177, \quad \pi(1, 0) = 0.0818, \quad \pi(0, 1) = 0.0121$$

A fő következtetés: az orphan blokkok rátája becslétes bányászat mellett $\approx 0.5/\text{nap}$, selfish mining mellett $\approx 3.7/\text{nap}$ – ez a különbség detektálható, és diagnosztikai eszközként használható a selfish mining jelenlétének kimutatására a hálózatban.

2.4 A közlegelők tragédiája

A közlegelők tragédiája (tragedy of the commons) egy közgazdasági modell, amely megmutatja, hogyan vezet az egyéni racionális viselkedés a közös erőforrás tönkretételéhez. Ez a jelenség közvetlenül releváns a PoW bányászat centralizációs tendenciáihoz.

Tekintsük a következő példát. Egy angol közlegelő 10 tehenet tud optimálisan eltartani. Kezdetben 10 gazda van, mindegyik 1 tehenet tart, és minden tehén 10 liter tejet ad naponta. Az egyensúlyi állapotban tehát minden gazda 10 liter tejet kap.

Az egyik gazda felismeri, hogy ha kihajt egy második tehenet, a legelő összerhelése 11 tehenre nő, a tejhozam tehenenként 9 literre csökken, de ő két tehénnel $2 \times 9 = 18$ litert kap az eredeti 10 helyett. Számára tehát megéri. Más gazdák is felismerik ezt, és ők is kihajtanak egy-egy plusz tehenet. A folyamat előrehaladtával a tejhozam tovább csökken:

Tehenek száma	Liter/tehen	Aki 2 tehenet tart	Aki 1 tehenet tart
10 (optimális)	10	–	10
11	9	18	9
14	8	16	8
16	5	10	5
18	2	4	2

A dezertálás problémája: ha egy gazda visszavonja a második tehenét a közjó érdekében, hátrányba kerül a többiekkel szemben (kevesebb tejet kap, miközben a legelő alig javul). Egyénileg tehát racionális tovább használni az erőforrást, de ha mindenki ezt teszi, mindenki rosszabbul jár.

A blokkláncokra alkalmazva: a bányász-poolok méretének növekedése hasonló dinamikát követ. Egy nagyobb pool kisebb varianciájú (kiszámíthatóbb) jövedelmet ad, tehát az egyéni bányász számára racionális csatlakozni a legnagyobb pool-hoz. De ha mindenki ezt teszi, egyetlen pool dominálja a hálózatot, és a decentralizáltság – ami az egész rendszer biztonságát adja – megszűnik. Az egyéni érdekek tehát a közjó (decentralizáció, biztonság) ellen hatnak, és az optimális erőforráskezelés megköveteli a közös szabályok (protokoll) betartását.

2.5 Blockchain DDoS támadás

A BDoS (Blockchain Denial of Service) támadást Michael Mirkin és társai írták le [4]. A támadás célja nem a tranzakciók manipulálása, hanem a blokklánc teljes leállítása – úgy, hogy a becsületes bányászok számára gazdaságilag racionálissá teszi a bányászat abbahagyását.

2.5.1 A támadás mechanizmusa

A támadó a következőt teszi: amikor talál egy blokkot, nem a teljes blokkot publikálja, hanem csak a blokk fejlécét (header). Ez a fejléc bizonyítja, hogy a támadó megoldotta a PoW feladatot, tehát érvényes blokkja van – de a tartalmát nem fedi fel. A becsületes bányászok ezután döntés előtt állnak:

- Ha folytatják a bányászatot a saját blokkjukon, és a támadó később benyújtja a teljes blokkot, az ő munkájuk elvész (orphan).
- Ha abbahagyják a bányászatot ebben a körben, biztosan nem veszítenek energiaköltséget – de nem is keresnek.

Az egyénileg racionális döntés az abbahagyás: ha meghallom, hogy valaki talált blokkot, biztosan jobban járok, ha nem pazarlom az áramot felesleges versenyre. Ez ismét a közlegelő tragédiája: mindenki motivált a közjó (a blokklánc működése) ellenére változtatni a stratégiáján.

2.5.2 Miért működik a támadás?

A domináns stratégia elemzése Markov-lánc alapú. Összehasonlítjuk a bányász hasznosságát két esetben: (1) megáll a fejléc publikálása után, (2) folytatja a bányászatot. A döntés a következő paraméterektől függ:

- A bányászat energiaköltsége (mennyibe kerül az áram az adott lokáción).

- A bányász hashrate-aránya (milyen hatékonysággal bányászik).
- A blokk-jutalom aktuális értéke (a halving ciklusok miatt ez periodikusan csökken).

2021-es számadatok mellett: 1 dollár áramköltségre várhatóan 1,47 dollár bevétel jut. Ebben a szűk margóban a fejléc-publikálás elegendő ahhoz, hogy a folytatás várható haszna negatívvá váljon – a bányász racionálisan leáll.

Tovább súlyosbítja a helyzetet, hogy a bányász nem csak leállhat, hanem átválthat egy másik láncra (pl. Bitcoin Cash), amely hasonló algoritmussal bányászható. Ez még nagyobb motivációt ad arra, hogy ne vegyen részt az adott körben. A támadó költsége minimális: csupán egyetlen PoW megoldás kiszámítása szükséges, és a blokk tartalmát soha nem kell benyújtania.

2.5.3 Formális modell

Jelölje λ a blokk-találási rátát, K a blokk-jutalmat, c_i a bányász i költségét időegységenként, α_A a támadó hashrate-arányát, α_{B^*} a becsületes bányászokét. A becsületes bányász hasznosságfüggvénye és profitabilitási faktora:

$$U_i^b \triangleq \lambda K - c_i, \quad \omega_i^b \triangleq \frac{\lambda K}{c_i} \quad (13)$$

Az U_i^b a bányász nettó haszna időegységenként támadás nélkül: a várható bevétel (λK , a blokk-találási ráta szorozva a jutalommal) mínusz az energiaköltség. Az ω_i^b a bevétel-költség arány: ha $\omega_i^b > 1$, a bányászat nyereséges; ha $\omega_i^b = 1$, nullszaldós.

A rendszer három állapotban lehet: (0) a becsületes közösség talált blokkot (normál működés), (1) a támadó publikálta a fejlécet (a bányászok döntés előtt állnak), (2) a támadó korábban publikált fejlécet és most a közösség is talált blokkot (versenyhelyzet). Az állapotvalószínűségek az S stratégia mellett:

$$p_0^S = \frac{\alpha_{B^*}(S)}{\alpha_A \alpha_{B^*}(S) + \alpha_A + \alpha_{B^*}(S)}, \quad p_1^S = \frac{\alpha_A}{\alpha_A \alpha_{B^*}(S) + \alpha_A + \alpha_{B^*}(S)} \quad (14)$$

$$p_2^S = \frac{\alpha_A \cdot \alpha_{B^*}(S)}{\alpha_A \alpha_{B^*}(S) + \alpha_A + \alpha_{B^*}(S)} \quad (15)$$

ahol $\alpha_{B^*}(S) = \alpha_{B^*} + \alpha_i$ ha $S = S_{\text{mine}}$ (a bányász folytatja, tehát hozzáadja a kapacitását), és $\alpha_{B^*}(S) = \alpha_{B^*}$ ha $S = S_{\text{stop}}$ (a bányász megáll). Figyeljük meg, hogy p_1^S (a támadási állapot valószínűsége) csökken, ha több becsületes bányász dolgozik – tehát a megállás növeli a támadás sikerességét.

A bányász hasznosságfüggvénye a két stratégia mellett:

$$U_i^{S_{\text{stop}}} = \left(p_0^{S_{\text{stop}}} + p_2^{S_{\text{stop}}} \right) \lambda K - \left(1 - p_1^{S_{\text{stop}}} \right) c_i \quad (16)$$

Ha megáll: csak akkor keres, ha normál állapotban (0) vagy versenyhelyzetben (2) van (mert ilyenkor az ő korábbi blokkja érvényes marad), de a költsége is csak akkor merül fel, amikor ténylegesen bányászik (azaz nem az (1) állapotban).

$$U_i^{S_{\text{mine}}} = \left(p_0^{S_{\text{mine}}} + p_2^{S_{\text{mine}}} + (1 - \gamma)(1 - \alpha_A)p_1^{S_{\text{mine}}} \right) \lambda K - c_i \quad (17)$$

Ha folytatja: az (1) állapotban is dolgozik, és $(1 - \gamma)(1 - \alpha_A)$ valószínűséggel az ő blokkja nyer a fork-ban (a γ paraméter itt is a támadó blokkterjedési előnyét jelöli). Viszont a költsége folyamatos (c_i minden időegységben).

A megállás domináns stratégia, ha $U_i^{S_{\text{stop}}} > U_i^{S_{\text{mine}}}$. A normalizált különbség:

$$D(\alpha_{B^*}) = \left(p_0^{S_{\text{stop}}} + p_2^{S_{\text{stop}}} - p_0^{S_{\text{mine}}} - p_2^{S_{\text{mine}}} - (1 - \gamma)(1 - \alpha_A)p_1^{S_{\text{mine}}} \right) \omega_i^b + p_1^{S_{\text{stop}}} \quad (18)$$

Ebből a megállás feltétele a profitabilitási faktorra (ha $D > 0$, vagyis a megállás jobb):

$$\omega_i^b < Q(\alpha_{B^*}) \triangleq \frac{p_1^{S_{\text{stop}}}}{p_0^{S_{\text{mine}}} + p_2^{S_{\text{mine}}} + (1 - \gamma)(1 - \alpha_A)p_1^{S_{\text{mine}}} - p_0^{S_{\text{stop}}} - p_2^{S_{\text{stop}}}} \quad (19)$$

Ez a Q küszöbérték: ha a bányász profitabilitási faktora ez alatt van, racionálisan leáll. Minél kisebb a Q , annál nehezebb a támadás (mert csak nagyon alacsony margójú bányászokat tud leállítani).

A legszűkebb korlát (amikor minden becsületes bányász leáll, azaz $\alpha_{B^*} = 1 - \alpha_A - \alpha_i$):

$$\omega_i^b < \frac{\alpha_A + \alpha_i + \alpha_A \alpha_i}{\alpha_i + \alpha_A \alpha_i + (1 - \gamma)\alpha_A(1 - \alpha_A)} \quad (20)$$

Ez azt jelenti, hogy ha a bányász profitabilitási faktora ($\omega_i^b = \lambda K / c_i$) ez alá a korlát alá esik, racionálisan leáll. A 2021-es Bitcoin hálózatban $\omega_i^b \approx 1.47$, ami azt jelenti, hogy a bányászok bevétele mindössze 47%-kal haladja meg a költségeiket – ez a szűk margó teret ad a BDoS támadásnak, mert a Q küszöb viszonylag magas értékeknél is teljesülhet.

2.5.4 Védekezés

Ha a becsületes bányászok hosszú távon gondolkodnak, elméletileg dönthetnek úgy, hogy a támadás ellenére is folytatják a bányászatot – de ez egyénileg nem racionális, csak a kollektív érdeket szolgálja. A probléma tehát alapvetően játékelméleti: a Nash-egyensúly [5] a bányászat leállítása, ami a támadó célját szolgálja. Lehetséges védekezési mechanizmusok közé tartozik a fejléc-nélküli bányászat (ahol a megtalált blokk bizonyítéka csak a teljes blokk lehet) és a protokoll szintű módosítások, amelyek büntetik az üres fejléc-publikálást.

2.6 Cryptojacking

A cryptojacking a felhasználók számítási erőforrásainak jogosulatlan felhasználása kriptovaluta-bányászatra. A *How You Get Shot in the Back* [6] című 2018-as tanulmány részletesen feltérképezte a jelenséget.

2.6.1 A támadás módszere

A támadók JavaScript kódot rejtenek weboldalakba, amely a látogatók böngészőjében fut és kriptovalutát bányászik (jellemzően Monero-t, amely CPU-barát PoW algoritmust használ). A felhasználó nem telepít semmit – elég meglátogatni az érintett weboldalt, és a böngészőablak nyitva tartása alatt a gép dolgozik a támadónak.

A tanulmány adatai alapján havonta körülbelül 10 millió felhasználót érint a probléma. A szerzők 2770 különböző bányász-szkript mintát detektáltak 853 936 weboldalon. A becsült napi extra energiafogyasztás 278 000 kWh, amellyel a támadók napi $\approx 59\,000$ USD bevételt generálnak – miközben az áramköltséget teljes egészében a felhasználók (és a környezet) fizetik.

2.6.2 Detekció

A bányász-szkriptek detektálása a viselkedésük alapján lehetséges: a hash-számítás periodikus stack-hívásokat generál, amelyek mintázata egyértelműen megkülönböztethető a normál webes JavaScript-tevékenységtől. A periodikus CPU-terhelés és a jellegzetes WebAssembly/asm.js hívásláncok jól azonosíthatók.

2.6.3 Elrejtési technikák

A támadók számos módszert alkalmaznak a detekció elkerülésére:

- CPU-használat korlátozása (pl. 50%-ra, hogy a felhasználó ne vegye észre a lassulást).
- A kód obfuszkálása (minifikálás, változónevek véletlenszerűsítése, dinamikus betöltés).
- A payload elrejtése legitim JavaScript fájlokban (pl. egy online könyvesbolt `bookstore.js` fájljába ágyazva, amely a látogató gépén 32 szálon, 100% CPU-használattal indítja a bányászatot).

Hivatkozások

- [1] S. Nakamoto, *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. <https://bitcoin.org/bitcoin.pdf>
- [2] I. Eyal, E. G. Sirer, *Majority is not Enough: Bitcoin Mining is Vulnerable*, Financial Cryptography and Data Security, Lecture Notes in Computer Science, vol. 8437, Springer, pp. 436–454, 2014.
- [3] R. Göbel, P. Keeler, A. E. Krzesinski, P. G. Taylor, *Bitcoin Blockchain Dynamics: The Selfish-Mine Strategy in the Presence of Propagation Delay*, Performance Evaluation, vol. 104, pp. 23–41, 2016.
- [4] M. Mirkin, Y. Ji, J. Pang, A. Klages-Mundt, I. Eyal, A. Juels, *BDoS: Blockchain Denial-of-Service*, Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 601–619, 2020.
- [5] J. Nash, *Equilibrium Points in N-Person Games*, Proceedings of the National Academy of Sciences, 36(1):48–49, 1950.
- [6] R. K. Konoth, E. Vineti, V. Moonsamy, M. Lindorfer, C. Kruegel, H. Bos, G. Vigna, *How You Get Shot in the Back: A Systematical Study about Cryptojacking in the Real World*, Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS), pp. 1701–1713, 2018.