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. november 3.

1 KZG-commitmentek és az L2-rollupok

A korábbi, *Skálázás: Layer-2 megoldások, rollupok és az EIP-4844* alfejezetben láttuk, hogy a rollupok sok tranzakció adatát kötegekben teszik közzé az Ethereum alaprétegén. A KZG (Kate–Zaverucha–Goldberg) eljárás egy rövid, kriptográfiai commitmentet készít egy polinomhoz, majd rövid bizonyítékkal igazolja a polinom egy adott helyen felvett értékét. Az alábbi leírás az eredeti, 2010-es polinom-commitment-konstrukció lényeges lépéseit követi [1].

A bizonyítandó állítás. Legyen $\mathbb{F}_r$ egy r elemű véges test, és legyen

$$p(X) = \sum_{j=0}^{d} a_j X^j \in \mathbb{F}_r[X] \tag{1}$$

egy legfeljebb d fokú polinom. A polinomhoz tartozó commitment azt teszi lehetővé, hogy az azt készítő fél egy rövid C értéket tegyen közzé, majd később megadjon egy z helyen felvett $y = p(z)$ értéket és egy rövid bizonyítékot. Az ellenőrzőnek nem kell megkapnia a teljes polinomot: azt ellenőrzi, hogy a közölt érték összhangban van-e a korábban közzétett commitmenttel.

Nyilvános paraméterek és commitment. A rendszer előkészítésekor egy megbízható vagy több résztvevő által végrehajtott ceremónia titkos $\tau \in \mathbb{F}_r$ értéket választ. A nyilvános paraméterek tartalmazzák a τ hatványaihoz tartozó görbepontokat, de magát τ értékét nem:

$$\mathrm{SRS} = \big([1]_1, [\tau]_1, \ldots, [\tau^d]_1; [1]_2, [\tau]_2\big). \tag{2}$$

Itt a szögletes zárójel a megfelelő csoport generátorával való skalárszorzást jelöli; G_1 és G_2 egy párosításra alkalmas elliptikusgörbe-csoport, az e bilineáris párosítás pedig $G_1 \times G_2$-ből egy G_T csoportba képez.

A commitmentet készítő fél a polinom együtthatóit a nyilvános paraméterekkel összesíti:

$$\boxed{C = [p(\tau)]_1 = \sum_{j=0}^{d} a_j [\tau^j]_1.} \tag{3}$$

Az eredmény egyetlen görbepont, mérete nem függ a polinom fokszámától. A titkos τ-t a ceremónia résztvevőinek meg kell semmisíteniük: ha valaki később ismeri ezt az értéket, hamis commitmenteket és értékbizonyítékokat készíthet.

Bizonyíték egy polinomértékre. Legyen z az a hely, ahol az értéket fel akarjuk nyitni, és legyen $y = p(z)$ a közölt érték. A polinomok maradéktételéből következik, hogy $p(X) - p(z)$ osztható $X - z$-vel. Ezért létezik egy $q_z(X)$ hányadospolinom, amelyre

$$p(X) - y = (X - z) q_z(X), \qquad q_z(X) = \frac{p(X) - y}{X - z}. \tag{4}$$

A bizonyíték ennek a hányadospolinomnak a rövid commitmentje:

$$\boxed{\pi = [q_z(\tau)]_1.} \tag{5}$$

Ellenőrzés párosítással. Az ellenőrző megnézi, hogy az alábbi egyenlőség teljesül-e:

$$\boxed{e\big(C - y[1]_1, [1]_2\big) \stackrel{?}{=} e\big(\pi, [\tau]_2 - z[1]_2\big).} \tag{6}$$

Ennek helyessége közvetlenül adódik a párosítás bilinearitásából és a polinomazonosságból:

$$\begin{aligned} e\big(C - y[1]_1, [1]_2\big) &= e\big([p(\tau) - y]_1, [1]_2\big), && (7)\\ e\big(\pi, [\tau]_2 - z[1]_2\big) &= e\big([q_z(\tau)]_1, [\tau - z]_2\big) && (8)\\ &= e\big([(\tau - z) q_z(\tau)]_1, [1]_2\big) && (9)\\ &= e\big([p(\tau) - y]_1, [1]_2\big). && (10) \end{aligned}$$

Így az ellenőrző azt tanúsítja, hogy a nyilvános commitmenthez tartozó polinom valóban y értéket vesz fel z-nél. A bizonyíték és a commitment is egy-egy görbepont, tehát a méretük a polinom fokszámától független; a bizonyíték előállítása és az ellenőrzés számítási költsége ettől még nem nulla.

Kapcsolat az EIP-4844 blobjaival. Az EIP-4844 olcsóbb módot ad a rollupoknak arra, hogy tranzakciós adatot tegyenek közzé az Ethereumon. A tranzakciók sorrendjét kijelölő fél összegyűjti az L2-tranzakciókat, és köteget készít belőlük. A köteg adatait tömörített bájtsorozattá kódolja, majd egy vagy több blobba rendezi. Egy blob 4096, egyenként 32 bájtos mezőelemet tartalmaz. A blobot a konszenzusréteg csomópontjai terjesztik; tartalmát az Ethereum virtuális gépe (EVM) nem tudja közvetlenül kiolvasni.

A blobhoz egy rövid KZG-commitment készül. Ezt úgy lehet elképzelni, mint egy adatcsomaghoz tartozó, rögzített digitális azonosítót: később ellenőrizhető vele, hogy egy bemutatott adatérték valóban ugyanabból a csomagból származik-e, anélkül, hogy a teljes csomagot az EVM-nek át kellene adni.

A blob 4096 elemét a véges test elemeiként kezeljük. Ezek a blobhoz rögzített mintapontokon felvett polinomértékek; belőlük egyértelműen meghatározható egy,

4096-nál kisebb fokú polinom. A commitment ennek a polinomnak egy rövid, 48 bájtos görbepont-ábrázolása: a polinom együtthatóit a nyilvános KZG-paraméterekkel összesítve számítják ki. Így a teljes polinom helyett elég ezt az egy pontot közzétenni. A commitment önmagában nem árulja el a blob adatait. Az Ethereum specifikációja a blobot mezőelemek sorozataként, a commitmentet pedig az ezek által meghatározott polinomhoz tartozó pontként írja le [2].

A bizonyítást benyújtó fél – például a rollupot működtető fél vagy egy ellenőrzést végző résztvevő – azt állítja, hogy a polinom a z helyen y értéket ad, és mellékel egy rövid KZG-bizonyítékot. Az EIP-4844 az EVM-ben, a 0x0A címen egy beépített, előre lefordított pontérték-ellenőrző függvényt (precompile) határoz meg. Ez ellenőrzi a bizonyítékot és azt, hogy a commitment az L1-tranzakcióban rögzített azonosítóhoz tartozik [2].

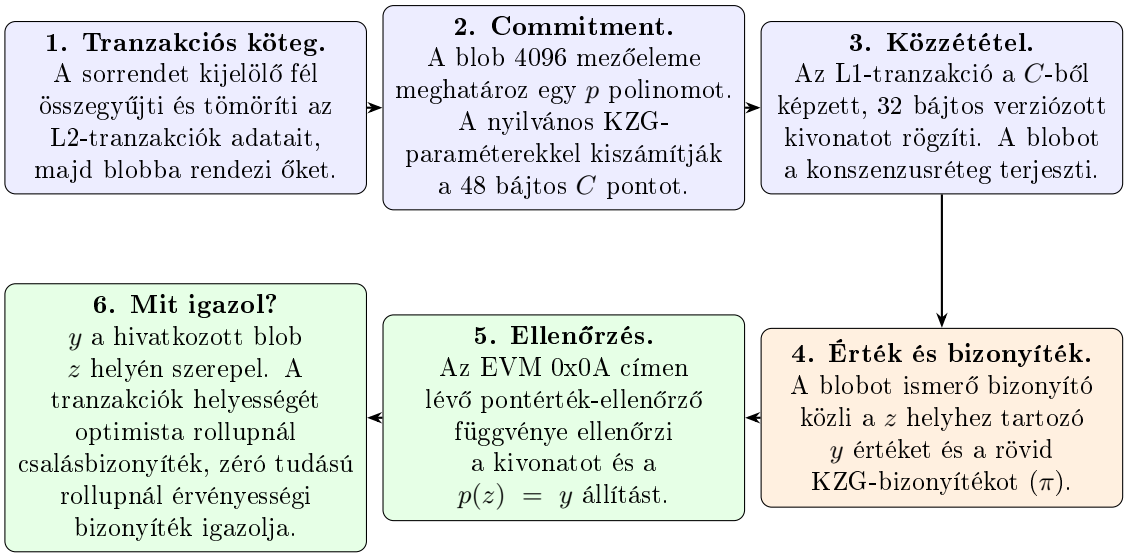
Ez az azonosító a commitmentből készített 32 bájtos, verzióval ellátott kivonat (versioned hash). Kiszámításakor a commitment SHA-256 kivonatának első bájtját az EIP-4844 a KZG-hez tartozó 0x01 verzióbájtra cseréli. A blob tranzakció ezt a rövid azonosítót tartalmazza; a commitment és a blob adatai a konszenzusrétegen jutnak el a csomópontokhoz. Az ellenőrző függvény összeveti a kapott commitmentből számított azonosítót a tranzakcióban rögzítettel, majd ellenőrzi a $p(z) = y$ állítást. Így meggyőződik arról, hogy a bemutatott adatérték az Ethereumon hivatkozott blobhoz tartozik, anélkül, hogy kiolvasná a teljes blobot [2].

Intuíció egy konkrét példával. Tegyük fel, hogy egy rollup egy blobban teszi közzé a tranzakciói adatait, és az Ethereum rögzíti a blobhoz tartozó KZG-commitmentet. Később egy ellenőrzést végző résztvevő meg akarja tudni, mi volt a blob 17. adathelyén. A blobot ismerő bizonyító közli, hogy ott y szerepel, és mellékel egy rövid bizonyítékot. Az ellenőrző az Ethereumon már rögzített commitmentet, a 17-es helyet, az állított értéket és a bizonyítékot használja. Ha az ellenőrzés sikeres, akkor biztos lehet benne, hogy a közölt érték annak a blobnak a 17. adatértéke, amelyhez a commitment készült. Nem kell ehhez az ellenőrzőnek letöltenie és újra átnéznie a teljes blobot.

A hasonlatban a commitment olyan, mint egy lezárt adatcsomaghoz rögzített azonosító, a KZG-bizonyíték pedig olyan igazolás, amellyel egy kiválasztott adatmező ellenőrizhető. Az azonosító önmagában nem mondja meg, mi van a csomagban, és nem is helyettesíti a bizonyítékot: a kettő együtt teszi lehetővé az ellenőrzést. A siker csak azt igazolja, hogy a bemutatott adat a közzétett blobhoz tartozik; azt nem, hogy a rollup minden tranzakciót helyesen számolt ki.

Ez a rollup ellenőrzésében azért hasznos, mert lehetővé teszi egy-egy adatdarab hitelesítését a teljes blob újbóli feldolgozása nélkül. Például egy optimista rollup tranzakcióinak eredményét vitatva a csalásbizonyíték megmutathatja, melyik blobban szerepel a vitatott adat, és KZG-bizonyítékkal igazolhatja, hogy a bemutatott érték valóban annak a blobnak a része. Egy zéró tudású rollup bizonyítéka pedig azt is ellenőrizheti, hogy a számítás során felhasznált adat ahhoz a blobhoz tartozik, amelynek commitmentjét az Ethereumon rögzítették.

Fontos különbség: a KZG azt ellenőrzi, hogy a bemutatott adatérték egyezik-e a közzétett blobhoz tartozó értékkel. Azt önmagában nem dönti el, hogy a rollup a tranzakciókat helyesen hajtotta-e végre. Erről az optimista rollupoknál a csalásbizonyítási eljárás, a zéró tudású rollupoknál pedig az állapotátmenet érvényességi bizonyítéka gondoskodik. Az sem ugyanaz, hogy egy adat hitelességét igazolni tudjuk, mint hogy az adat mindenki számára elérhető: az adat elérhetőségét az Ethereum hálózati szabályai biztosítják, a KZG-bizonyíték pedig az értékek helyességét ellenőrzi.



1. ábra: A KZG-commitment és egy blobbeli érték ellenőrzésének folyamata.

Kapcsolat az adat-elérhetőségi mintavételezéssel. A KZG-commitment önmagában nem teszi elérhetővé a blob teljes adatát, és nem is rejti el azt. A tényleges adatot a hálózatnak kell terjesztenie és ideiglenesen tárolnia. A PeerDAS a blobok kiterjesztett adatait kisebb cellákra bontja; a cellákhoz tartozó KZG- bizonyítékokkal egy csomópont úgy ellenőrizhet mintákat, hogy nem kell a teljes blobot letöltenie. Ez a commitmenthez kötött részadat-ellenőrzés segíti a rollupok számára szükséges adat-elérhetőség skálázását [3].

Lényeg röviden. A KZG tömör kriptográfiai commitmentet ad egy polinomhoz, és rövid bizonyítékot egy polinomérték helyességére. Az EIP-4844 ezt a tulajdonságot használja arra, hogy a rollupok nagy adatkötegeit olcsóbb blobokban rögzítse, miközben a commitmenthez kapcsolódó értékek később ellenőrizhetők. A rollup díjcsökkenését a blob-adatközzététel és annak külön díjpiaca adja; a KZG az adatokhoz tartozó rövid, ellenőrizhető commitmentet biztosítja.

Hivatkozások

[1] A. Kate, G. M. Zaverucha, I. Goldberg, *Constant-Size Commitments to Polynomials and Their Applications*, ASIACRYPT 2010, LNCS 6477, pp. 177–194. `https://www.iacr.org/archive/asiacrypt2010/6477178/6477178.pdf`

[2] Ethereum Improvement Proposals, *EIP-4844: Shard Blob Transactions*. `https://eips.ethereum.org/EIPS/eip-4844`

[3] Ethereum Improvement Proposals, *EIP-7594: PeerDAS – Peer Data Availability Sampling*. `https://eips.ethereum.org/EIPS/eip-7594`