Eötvös Loránd Tudományegyetem

Természettudományi kar

Nagy nemprímek verifikációja interaktív protokollok segítségével

Horváth Áron

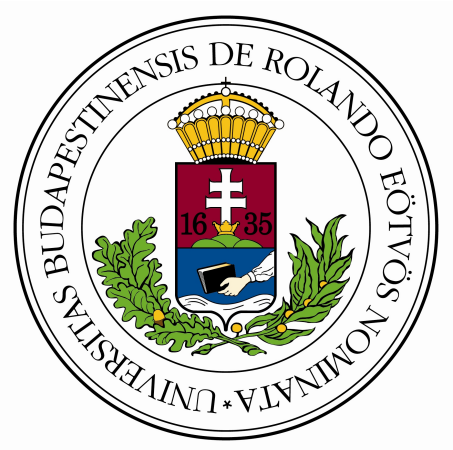
Szakdolgozat

matematika BSc

alkalmazott matematikus szakirány

Témavezető: Kutas Péter

Belső konzulens: Zábrádi Gergely



Budapest, 2024

Tartalomjegyzék

Bevezetés

Az embereket már több ezer éve foglalkoztatja, hogy amikor küldünk valakinek egy üzenetet, akkor hogyan tudnánk bizonyítani neki, hogy azt tényleg mi küldtünk és nem pedig valaki más, aki csak át akarja őt verni.

A mai világban az internet elterjedésével, nagyobb szükség van arra, hogy ezt meg tudjuk tenni, mint valaha. Például ha kapunk egy e-mailt a bankfiókunk kapcsán, akkor szeretnénk tudni, hogy azt tényleg a bank küldte-e, vagy valaki más, aki csak úgy tesz, mintha a bank lenne. Manapság az e-mail-fiókok már automatikusan szűrik, hogy melyik levél megbízható és melyik az, ami csak egy átverés. De hogyan tudják ezt megtenni?

Erre lehetne az a válasz, hogy a banknak van egy publikusan látható, nagyon nehezen megoldható számítási feladata, aminek a megoldásának helyességét gyorsan le lehet ellenőrizni (például egy NP-teljes probléma) és melynek megoldását csak ő ismeri. Ekkor ha elküldi nekünk ezt, mi le tudjuk tesztelni, hogy tényleg helyes-e és ezzel a bank igazolni tudta magát. Ezzel viszont az a probléma, hogy az ellenőrzés után már mi is ismernénk a megoldást és ezzel meg tudnánk téveszteni más embereket. A banknak valahogyan úgy kellene igazolnia magát, hogy közben nem ad át nekünk olyan információt, amivel utána mi át tudnánk verni másokat.

A szakdolgozatom első fejezetében be fogom vezetni az úgynevezett zéróismeretes bizonyítás fogalmát, ami az előbb elmített problémára nyújt megoldást.

Érdekes módon nemrégiben egy teljesen új területen alkamaztak interaktív protokollos technikákat, méghozzá a nagy prímszámok verifikálásában [7]. Több nagyobb prímkereső projekt létezik, melyek egyre nagyobb és nagyobb prímeket keresnek. Ezeknek komoly az erőforrásigényük, mert a sokmillió jegyű számok között szeretnének prímeket találni. Komoly kellemetlenség egy újabb prímszám megtalálásánál, hogy az új eredményt valaki másnak validálnia kell és ennek a validálási folyamatnak is óriásiak az erőforrás igényei. Erre ad egy megoldást a [7] cikk, azáltal hogy a számítási költségek enyhe növelésével

eléri, hogy a validáció sokkal gyorsabb legyen. A használt módszereknek a zéróismeretes bizonyítások mellett köze van az úgynevezett VDF-ekhez is. Ezek olyan függvények, amelyeket nem lehet gyorsan kiszámolni, de gyorsan le lehet ellenőrizni a helyességüket. A VDF-ek legfontosabb alkalmazási területe a blokkláncok világa. Ezek segítségével ki lehet iktatni a bányászatot, amely hatalmas áramköltségének komoly hatása van a klímaváltozásra.

A második fejezetben arra fogok eljárásokat mutatni, hogy hogyan lehet különféle speciális alakú számokról eldönteni, hogy prímek-e, a 2.2-es szakasz végén kitérve egy egyéni eredményemre annak kapcsán, hogy lehet-e általánosítani az egyik ilyen prímtesztet.

A harmadik fejezetben mutatni fogok egy protokollt arra, hogy a második fejezetben szereplő tételek segítségével hogyan lehet gyorsan leellenőrizni egy VDF használatával azt, hogy egy szám tényleg összetett, ha valaki azt állítja nekünk, hogy ő hosszú számolás után bizonyította, hogy igen.

A negyedik fejezetben pedig egy enyéni eredményemet fogom bemutatni, ami a harmadik fejezetben szereplő verifikációs eljárások átalakítása más alakú számokra.

1. fejezet

Interaktív protokollok

Egy levél küldése során gyakran van szükség arra, hogy bizonyítsuk valaki számára, hogy rendelkezünk egy információval. Mi ezt viszont anélkül szeretnénk megtenni, hogy elárulnánk ezt az információt annak, akinek bizonyítani próbáljuk, hogy rendelkezünk vele.

Ehhez kapcsolódóan lett bevezetve az úgynevezett "zéróismeretes bizonyítás", vagy röviden *ZKP* fogalma:

1.0.1. Definíció (informális)**.** *Azokat az interaktív protokollokat nevezzük ZKP-knek, melyekkel úgy bizonyítjuk, hogy rendelkezünk valamilyen információval, hogy erről az információról semmi olyat nem árulunk el, amiből az polinom időben visszafejthető lenne.*

A *ZKP* fogalmát csak később fogjuk precízen definiálni.

1.1 A zéróismeretes bizonyítások rövid története

[11] Titkosítással már nagyon régóta foglalkoznak az emberek. A Caesar-titkosítás a legelső ismert titkosítási eljárás, mely a Krisztus előtti 1. századból származik. Ez a módszer abból áll, hogy a titkosítandó szöveget betűnként eltoljuk valahány indexszel a latin ábécében és azt a szöveget küldjük tovább. Ezt a fogadó fél a betűk visszatolása után el tudja olvasni.

Az első ZKP ötlete ennél jóval később, 1985-ben merült fel, 3 MIT-n lévő kutató, Shafi Goldwasser, Silvio Micali és Charles Rackoff által. 1993-ban két másik kutatóval, Babai Lászlóval és Shlomo Morannal együtt Gödel-díjat nyertek ezért a munkájukért.

[8]1987-ben Russell Impagliazzo és Moti Yung bizonyította a feltörhetetlenség feltételezésével, hogy bármi, ami bizonyítható interaktívan, zéróismeretesen is bizonyítható. Másként megfogalmazva: CZK = IP.

[5]1999-ben Feige, Lapidot és Shamir vezették be a tanú megkülönböztethetetlenségének a fogalmát, mely a ZKP-k egy gyakran használt változata.

2012 januárjában Nir Bitansky, Ran Canetti, Alessandro Chiesa, és Erin Tromer kifejlesztették az úgynevezett zk-SNARK nem-interaktív protokollt, mely egy hatékony utat biztosított a ZKP-ok gyakorlati alkalmazására. Ez mind a mai napig egy gyakran használt séma, mivel a bizonyítás és az ellenőrzés is gyorsan történik, a bizonyíték mérete nagyon kicsi és nem igényel közvetlen kommunikációt a bizonyító és az ellenőrző között.

2018-ban két másik nem-interaktív protokoll is ki lett fejlesztve. A úgynevezett "bulletproof"-ok[4] és a zk-STARK[2], melynek előnye az eddigi nem-interaktív protokollokhoz képest az, hogy nincs szüksége megbízható beállításra.

1.2 A ZKP definíciója és egyszerű példák

A ZKP definíciója

1.2.1. Definíció. *[6]Egy L nyelvhez **interaktív bizonyítási rendszernek** nevezünk egy protokollt két véletlen interaktív géphez, melyeket **bizonyítónak** és **ellenőrzőnek** nevezünk akkor, ha:*

- *Mindkét gépnek van hozzáférése az input szalaghoz.*
- *A két gép tud egymásnak üzeneteket küldeni egy-egy kommunikációs szalaggal.*
- *Mindkét gép csak a saját szalagjait, az input szalagot és a kommunikációs szalagokat látja.*
- *Az ellenőrző lépésszáma a közös input méretében polinomiálisan korlátozott és utána elfogad, vagy elutasít állapotban áll meg. (A bizonyító lépésszáma nincs korlátozva.)*
- *Ha az ellenőrző az előre meghatározott V programját futtatja, akkor az alábbi két feltétel teljesül:*

– ***Teljesség**, azaz ha a bizonyító futtatja az előre meghatározott P programját, akkor minden $c > 0$ konstansra és elég nagy $x \in L$-re, az ellenőrző elfogadja a közös x inputot legalább $1 - |x|^{-c}$ valószínűséggel.*

– ***Megbízhatóság**, azaz minden bizonyító által futtatott P^* programra, minden $c > 0$ konstansra és elég nagy méretű $x \notin L$-re az ellenőrző elutasítja x-et legalább $1 - |x|^{-c}$ valószínűséggel.*

1.2.2. Definíció. *[6]Egy L nyelvhez tartozó interaktív bizonyítási rendszert **zéróismeretes**nek, vagy az angol nevéből (zero-knowledge proof) röviden **ZKP**-nek nevezünk, ha minden polinomiális futásidejű V^* véletlen géphez létezik egy olyan véletlen polinomiális futásidejű M_{V^*} algoritmus, ami egy x inputon készít egy $M_{V^*(x)}$ valószínűségi eloszlást úgy, hogy $\{M_{V^*(x)}\}_{x\in L}$ és $\{\langle P(x), V^*(x)\rangle\}_{x\in L}$ polinomiális időben megkülönböztethetetlenek.*

A gráfizomorfizmus-probléma

1.2.3. Definíció. *Két egyszerű gráfot, G_1-et és G_2-t, akkor nevezünk **izomorf**nak, ha létezik olyan f bijekció a csúcsai között, hogy G_1-ben bármely v_1 és v_2 csúcspár között pontosan akkor megy él, amikor G_2-ben $f(v_1)$ és $f(v_2)$ között. Ha ez teljesül, akkor f-et **izomorifzmus**nak nevezzük.*

1.2.4. Definíció. *A **gráfizomorfizmus-probléma** az az eldöntési probléma, mely során szeretnénk két gráfról algoritmikusan eldönteni, hogy izomorfok-e.*

Innentől fel fogjuk tenni, hogy nem lehet hatékonyan megoldani a gráfizomorfizmus-problémát.

Tegyük fel, hogy Alice egy bank dolgozója és azáltal szeretné igazolni ezt Bobnak, hogy készít két, egymással izomorf gráfot, G_1-et és G_2-t, melyek között csak ő ismeri az f-fel jelölt izomorfizmust (azaz, hogy melyik G_1-beli csúcsnak melyik G_2-beli a párja) és ezt a két gráfot nyilvánosan láthatóvá teszi. Ekkor, ahhoz, hogy ő igazolni tudja, hogy valóban a bank egy alkalmazottja, meg kell győznie Bobot arról, hogy tényleg ismer a két gráf között egy izomorfizmust.

A protokoll leírása:

1. Alice készít egy G_1^* gráfot, mely izomorf a nyilvánosan is látható G_1 gráffal és ezt a gráfot megmutatja Bobnak. A G_1 és G_1^* közti izomorfiát jelöljük g-vel. Mivel az izomorfság egy tranzitív reláció, az általa konstruált gráf G_2-vel is izomorf lesz.

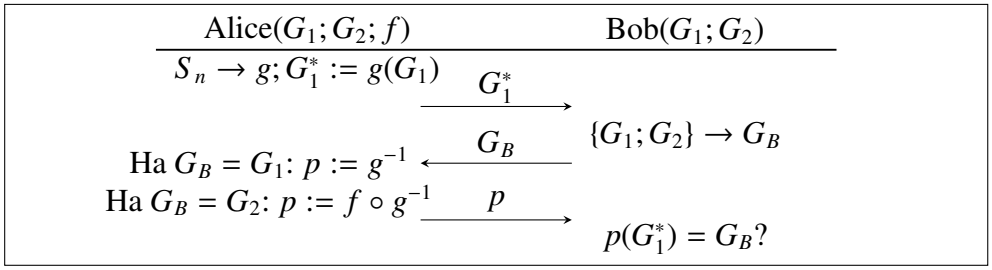
2. Bob kiválasztja egyenlő valószínűséggel az egyiket a G_1 és G_2 gráfok közül (a választása az ábrán G_B-vel van jelölve) és azt kéri Alice-től, hogy mutassa meg, hogy G_1^* tényleg izomorf az általa választott gráffal.

3. Ha Bob G_1-et választotta, Alice megmutatja neki a g^{-1} permutációt (amit g ismeretében csúcsszámban lineáris időben ki tud számolni); Ha pedig G_2-t választja, megmutatja neki $f \circ g^{-1}$-et, ami f-ből és g-ből szintén csúcsszámban lineáris időben kiszámítható. (Az Alice által kiszámolt izomorfizmus az ábrán p-vel van jelölve.)

4. Bob leellenőrzi, hogy az Alice által küldött izomorfizmus tényleg helyes-e (úgy, hogy minden G_1-beli $v_1; v_2$ csúcspárra leellenőrzi, hogy v_1 és v_2 között tényleg pontosan akkor fut él, ha $f(v_1)$ és $f(v_2)$ között is, ez csúcsszámban négyzetes időben megtehető).

5. Ezután Alice konstruál egy G_2^* gráfot is és ezzel megismétlik az 1.-től 4. pontokban leírt lépéseket. Majd egy G_3^* gráffal is... (Ezt sokszor megcsinálják, mondjuk 100-szor).

6. Amennyiben Alice Bob minden kérdésére helyesen felel, meggyőzi őt arról, hogy tényleg ismeri az izomorfizmust G_1 és G_2 között.



1. Figura: ZKP a gráfizomorfizmus-problémára

1.2.5. Megjegyzés. *Alice-nek nem szabad soha kétszer ugyanazt a G^* gráfot adnia Bobnak a folyamat során, hiszen ha Bob a ezen két alkalom egyikén G_1-et, a másikon pedig G_2 választaná, akkor Alice megmutatná neki az izomorfizmust G^* és G_1, valamint G^* és G_2 között is, amiből Bob már ki tudná számolni az izomorfizmust G_1 és G_2 között is, tönkretéve a protokoll zéróismeretességét.*

A protokoll biztonsága: Tegyük fel, hogy Eve, aki nem a bank dolgozója, meg akarja győzni Bobot arról, hogy ő az, mivel szeretné megszerezni Bob összes pénzét. Eve G_1 és

G_2 közül az egyikkel tud izomorf G_1^* gráfot készíteni és ezt mutatni Bobnak (jelöljük az általa választott gráfot G_E-vel, a másikat G_{E^*}-gal, a G_E és G_1^* között lévő izomorfizmust pedig f_1-gyel).

1. eset: Bob valamilyen i-re az i. Eve által konstruált gráfra, a másik gráfot választja mint, amire Eve G_E-t készítette:

Eve ekkor el fog bukni ezen a teszten, hiszen ha tudna egy g_i izomorfizmust találni az általa konstuált G_i^* és a Bob által választott G_{E^*} gráf között, akkor $g_i \circ f_i$-t is ki tudná számolni, ami egy izomorfizmus G_E és G_{E^*}, azaz G_1 és G_2 között. Ez viszont azt jelentené, hogy Eve ezzel megoldaná a gráfizomorfizmus-problémát, amiről feltettük, hogy nem megoldható hatékonyan.

2. eset: Bob minden lépésben ugyanazt a gráfot választja mint Eve:

Ekkor Eve át fog menni ezen a teszten, hiszen az f_i-k ismeretében ki tudja számolni minden esetben f_i^{-1}-et.

Annak a valószínűsége, hogy ez megtörténik 1 teszt esetén $\frac{1}{2}$. Viszont ha ezt a tesztet 100-szor elvégzik, akkor mindössze $\frac{1}{2^{100}}$ valószínűséggel fogja tudni Eve meggyőzni Bobot arról, hogy ismer izomorfizmust a két gráf között, ami körülbelül $8 \cdot 10^{-29}\%$. Ez egy elhanyagolhatóan kis valószínűség. De amennyiben ez a valószínűség még mindig nem lenne elég, 100 helyett 200-szor is elvégezhetnék ugyanezt a tesztet.

A protokoll zéróismeretes: Alice a protokoll folyamán Bobnak nem ad át más adatot, minthogy mutat néhány gráfot és ezekről bizonyítja izomorfok G_1 és G_2 közül pontosan az egyikkel. Ez viszont egy olyan dolog, amit egy külső szemlélő, Chuck, a G_1 és G_2 közti izomorfizmus ismerete nélkül is meg tudna tenni. Tegyük fel, hogy Chuck mindig előre tudja, hogy Bob melyik gráfot fogja választani G_1 és G_2 közül. Ekkor azáltal, hogy mindig a Bob által választandó gráfhoz készít izomorf gráfokat, meg tudja győzni Bobot arról, hogy ő tényleg ismer izomorfizmust G_1 és G_2 között. Tehát mivel az izomorfizmus ismerete nélkül is szimulálható a protokoll, így nem tartalmazhat információátadást.

A diszkrét logaritmus probléma

A korábban leírt ZKP-t a gráfizomorfizmus-problémára nem szokták használni gyakorlatban, mivel ismert rá kvázi-polinomiális algoritmus[1], ami annak veszélyét rejti magában, hogy valaki képes belátható időn belül találni egy izomorfizmust G_1 és G_2 között, ezzel tönkretéve a rendszert. Most viszont fogunk mutatni egy olyan eljárást, mely

ténylegesen alkalmazva van gyakorlatban is és melynek feltörésére nem ismert hatékony módszer.

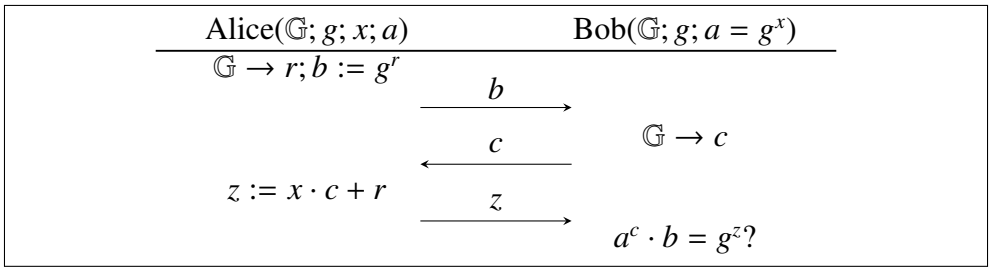
1.2.6. Definíció. *Legyen $\mathbb{Z}_p$ egy ciklikus csoport, legyen $g \in \mathbb{Z}_p$ és a pedig legyen $\mathbb{Z}_p$ g által generált részcsoporjának egy eleme. Ekkor az a elem g-es alapú* ***diszkrét logaritmus****ának nevezzük azt az $x \in \mathbb{Z}^+$ számot, melyre $g^x = g \cdot g \cdot g \cdot \ldots \cdot g = a$.*

1.2.7. Definíció. *A* ***diszkrét logaritmus probléma*** *az a számítási probléma, mely során szeretnénk kiszámolni egy $\mathbb{Z}_p$ csoportban lévő a elem g-es alapú diszkrét logaritmusát.*

Tegyük fel, hogy Alice az által szeretné bizonytani Bobnak, hogy egy megbízható ember, hogy ismeri a-nak a g-es alapú diszkrét logaritmusát egy $\mathbb{G}$ csoportban. Bob $\mathbb{G}$-t, g-t és és a-t ismeri. Ezt Alice az **Schnorr protokoll**nak nevezett ZKP segítségével az alábbi módon tudja megtenni:

A Schnorr protokoll leírása:[15]

1. Alice generál egy random r egész számot és elküldi Bobnak $b = g^r$ értékét.
2. Bob elküld Alice-nek egy általa generált random c pozitív egész számot.
3. Alice kiszámítja $z = x \cdot c + r$ értékét és elküldi Bobnak.
4. Bob leellenőrzi, hogy $a^c \cdot b = g^z$ fennáll-e. Ha igen, akkor *elfogadja* Alice bizonyítékát, ha nem, akkor *elutasítja*.



2. Figura: A Schnorr-protokoll

1.2.8. Állítás. *Alice mindig meg tudja győzni Bobot arról, hogy ő tényleg ismeri a értékét.*

Állítás 1.2.8 bizonyítása: Alakítsuk át $a^c \cdot g^z$ értékét!

$$a^c \cdot b = (g^x)^c \cdot g^r = g^{xc} \cdot g^r = g^{x \cdot c + r} = g^z. \tag{1.1}$$

Tehát, ha Alice ismeri x-et, akkor a Bob által leellenőrzött állítás igaz, vagyis elfogadja Alice bizonyítékát. □

1.2.9. Megjegyzés. *Alice-nek nem szabad soha kétszer ugyanazt az r random számot generálnia, hiszen ekkor g^r értéke is ugyanannyi lenne, amit Bobnak ad. Ekkor Bob, észrevéve az azonosságot, az általa generált c_1 és c_2 számok, illetve a rájuk válaszként kapott z_1 és z_2 ismeretében ki tudná számolni x értékét, tönkretéve a protokoll zéróismeretességét, az alábbi módon:*

Bob tudja, hogy $z_1 = x \cdot c_1 + r$ és $z_2 = x \cdot c_2 + r$. Ezeket összerakva: $z_2 - z_1 = x \cdot (c_2 - c_1)$, tehát $x = \frac{z_2 - z_1}{c_2 - c_1}$ (ezt az értéket Bob könnyen meg tudná határozni, hiszen $\mathbb{Z}_p$-ben lehet hatékonyan inverzet számolni).

1.2.10. Megjegyzés. *Egy $g \in \mathbb{G}$ elem rangját innentől $o_{\mathbb{G}}(g)$-vel fogjuk jelölni.*

A protokoll biztonsága:

Tegyük fel, hogy Eve meg akarja győzni Bob-ot arról, hogy ő ismeri x értékét, úgy, hogy valójában nem teszi. A Bob által ellenőrzött állítás pontosan akkor teljesül, ha $xc + r \equiv z \ (mod\ o_{\mathbb{Z}_p}(g))$, ahol $o_{\mathbb{Z}_p}(g)$ egy nagy szám.

Eve-nek nem érdemes csalnia azzal, hogy úgy küld egy olyan b értéket, amire nem ismeri r-et, hiszen ekkor még nem ismeri még a Bob által generált c számot, így az a^c értéke egy uniform véletlen eleme $\mathbb{Z}_p$-nek számára, ebből kifolyólag bármely fix b-re $a^c \cdot b$ is. Azaz bármely b-re ugyanakkora a valószínűsége annak, hogy meg tudja határozni azt a z-t, amire $a^c \cdot b = g^z$. Tehát nincs oka arra, hogy b-t úgy válassza, hogy ne ismerje azt az r-et, amire $g^r = b$.

Eve ismeri c értékét, hiszen Bob elküldte neki és az előző bekezdés alapján ismeri r-et is, hiszen azt ő generálta. Tehát ha Eve nem elhanyagolható valószínűséggel ki tudná számolni z értékét (azaz a protokoll nem lenne biztonságos), akkor abból a $z - r$-ként meg tudná kapni xc értékét is arra az általa ismert c-re, amit Bob generált, amiből meg tudná határozni x-et is. Viszont korábban feltettük, hogy Eve nem tudja kiszámolni x-et, ezzel ellenomdásra jutva abból a feltételezésből, hogy a protokoll nem biztonságos.

A protokoll zéróismeretes: Alice a protokoll folyamán Bobnak nem ad át más adatot, mint $b = g^r$ értéke és $z = x \cdot c + r$ értéke. Ez viszont egy olyan dolog, amit egy külső szemlélő, Chuck, x ismerete nélkül is meg tudna tenni. Tegyük fel, hogy Chuck előre tudja, hogy Bob milyen c-t fog választani. Ekkor Chuck tud egy véletlenszerű z-t választani és

arra kiszámolni g^z értékét. Mivel a publikus és feltettük, hogy Chuck ismeri c-t, ezért a^c-t is ki tudja számolni. Chuck azt is tudja, hogy Bob azt fogja leellenőrizni, hogy $a^c \cdot b = g^z$ fennáll-e. Ebből a^c-t és g^z-t ki tudja számolni és azokból b-t is, mint $g^z \cdot a^{-c}$ (ezt azért tudja kiszámolni, mivel egy ciklikus csoportbeli elem inverzének meghatározására ismert hatékony algoritmus). Ekkor, ha Chuck az általa random generált z-t és az abból kiszámolt b-t küldi el Bobnak az algoritmus során, akkor Bob el fogja fogadni Chuck bizonyítását, habár Chuck nem ismeri x értékét. Tehát mivel x ismerete nélkül is szimulálható a protokoll, így nem tartalmazhat információátadást.

1.3 Igazolható késleltetési függvények

Az igazolható késleltetési függvények definiálása

1.3.1. Definíció. *(informális)* ***Igazolható késleltetési függvény****nek, vagy röviden VDF-nek nevezünk egy olyan Felépít-Értékel-Igazol algoritmushármast, melyből Felépít egy λ biztonsági paraméterből és egy t késleltetési együtthatóból meghatároz két publikus kulcsot; Értékel t időben kiszámolja az x inputhoz tartozó y outputot és készít egy rövid π bizonyítékot ennek helyességére; Igazol pedig leellenőrzi π és a publikus kulcsok segítségével, hogy az x inputhoz tartozó output tényleg y. Igazol futásidejének alacsonynak kell lennie (lehetőleg t méretében polinomiálisnak), soha nem tévedhet, ha az x-hez tartozó output y és csak nagyon kis valószínűséggel szabad tévednie, ha nem az. x-ből y-ből kiszámolható t szekvenciális lépésben. Továbbá t-nél lényegesen kevesebb lépésből π ismerete nélkül polinom mennyiségű processzorral nem lehet megkülönböztetni y-t egy random outputtól.*

1.3.2. Definíció. *[3]* ***Igazolható késleltetési függvény****nek vagy az angol nevéből (verifiable delay function), röviden* ***VDF****-nek nevezünk egy (Felépít, Értékel, Igazol) algoritmushármast a következőek szerint:*

- ***Felépít***$(\lambda; t) \rightarrow (ek; vk)$ *egy randomizált algoritmus, ami vesz egy λ bizontsági paramétert és egy kívánt t rejtvénynehézséget és elkészít egy publikus paraméterpárt, ek-t (az értékelési kulcsot) és vk-t (az igazolási kulcsot). Felépít λ-ban polinomiális. Kommunkáció során ek és vk megad egy X input- és egy Y output-teret. Továbbá feltételezzük, hogy X könnyen mintavételezhető.*

- ***Értékel(****ek*; *x*) → (*y*; *π**)*** *veszi az x ∈ X inputot és ebből elkészít egy y ∈ Y outputot és egy π bizonyítékot. π kiszámításához használhat random biteket, de y-hoz nem. Az összes Felépít(λ*; *t) által generált (ek*; *vk)-ra és x ∈ X-re, Értékel(ek*; *x)-nek t párhuzamos időben kell futnia poli(log(t)*; *λ) processzor esetén.*

- ***Igazol(****vk*; *x*; *y*; *π*) → **{Igen;Nem}** *egy determinisztikus algoritmus, ami vesz egy x inputot, y outputot és egy π bizonyítékot, és kiírja, hogy "Igen", vagy azt, hogy "Nem". Igazol-nak log(t)-ben és λ-ban polinomiális időben kell tudni futnia, továbbá teljesítenie kell a helyesség, megbízhatóság, szekvenciálisság hármast.*

1.3.3. Definíció. *Egy VDF akkor* ***helyes****, ha minden λ és t paraméterre, Értékel(λ*; *t) → (ek*; *vk)-ra és x ∈ X-re, abból, hogy Értékel(ek*; *x) → (y*; *π) következik, hogy Igazol(vk*; *x*; *y*; *π) = Igen.*

1.3.4. Definíció. *Egy VDF akkor* ***megbízható****, ha minden olyan A algoritmusra, aminek O(poli(t*; *λ)) a futásideje:*

$$\Pr\left[\begin{array}{l|l} Igazol(vk;x;y;\pi) = Igen & Felépít(\lambda;t) \to (ek;vk) \\ y \neq Értékel(ek;x), & A(\lambda;ek;vk;t) \to (x;y;\pi) \end{array}\right] \leq elh(\lambda),$$

ahol elh(λ) egy olyan függvény, amely nagy λ-khoz elhanyagolhatóan kis számokat rendel. (informálisan: riktán fogad el x-hez nem tartozó y outputot)

1.3.5. Definíció. *Definiáljuk a* ***szekvenciális játék****-ot (*A_0 *és* A_1*) ellenség által végzett algoritmusokra úgy, mint az alábbi algoritmusok egymásutánja:*

- *Értékel(λ*; *t) → (ek*; *vk)* \\ *Kiválasztunk egy random (ek*; *vk) párt.*
- $A_0(\lambda; ek; vk; t) \to L.$ \\ A_0 *előfeldolgozza vk-t és ek-t.*
- $X \to x$ \\ *Kiválasztunk egy random x inputot.*
- $A_1(L; ek; vk; x) \to y_A$ \\ A_1 *kiszámol egy* y_A *outputot.*

Azt mondjuk, hogy $(A_0; A_1)$ ***megnyeri a szekvenciális játékot****, ha Értékel(ek*; *x) =* $(y_A; \pi)$ *valamilyen π-re.*

1.3.6. Definíció. *Egy VDF akkor* $(p; \sigma)$***-szekvenciális****, valamilyen p(t) és σ(t) függvényekre, ha nincs olyan* $(A_0; A_1)$ *randomizált algoritmuspár, amire*

- A_0 *futásideje polinomiális t-ben és* λ*-ban.*
- A_1 *futásideje* $\sigma(t)$ *legfeljebb* $p(t)$ *processzorral*
- $(A_0; A_1)$ *meg tudja nyeni a szekvenciális játékot több, mint elh*(λ) *valószínűséggel.*

Egy példa VDF-re

[17] Legyen $\mathbb{G}$ egy olyan csoport, melyenk az elemszáma nem ismert, t egy pozitív egész szám, az értékelési és az igazolási kulcs pedig a csoport valamilyen leírása, ami elárulja, hogy hogyan lehet benne gyorsan két elemet összeszorozni.

Az értékelés és az igazolás az alábbi módon történjenek:

1. Alice (az értékelő) t egymás utáni négyzetre emeléssel kiszámítja g^{2^t} értékét $\mathbb{G}$-ben, ezzel y-t kapva.
2. Alice elküldi Bobnak (az igazolónak) y értékét, aki valamilyen k-ra kiválaszt véletlenszerűen egy l prímet az első 2^{2k} prímszám közül és visszaküldi l-et Alice-nek.
3. Alice kiszámítja $\pi := g^{\lfloor \frac{2^t}{l} \rfloor}$ értékét és visszaküldi Bobnak (ennek kiszámítását Alice hatékonyan meg tudja tenni a g^{2^i} értékek ismeretében, melyeket már kiszámolt korábban).
4. Bob kiszámítja $r := 2^t \pmod{l}$-et és leellenőrzi, hogy $\pi^l \cdot g^r = y$ fennáll-e. Ha igen, akkor *elfogadja* Alice bizonyítását, ha nem, akkor pedig *elutasítja*.

Ez tényleg egy VDF, hiszen Alice (az értékelő) t időben tudja csak kiszámolni y értékét; π ismeretében Bob (az igazoló) tényleg le tudja ellenőrizni $poli(log(t); \lambda)$ időben $g^{2^t} = y$ igazságtartalmát úgy, hogyha az állítás igaz, biztosan nem hibázik (tehát a helyesség teljesül); Ha $g^{2^t} = y$ hamis, akkor csak nagyon kis valószínűséggel fogadja el Bob az állítást (azaz a megbízhatóság teljesül, ezt nem bizonyítjuk); Továbbá az RSA-csoport leírásából λ-ból és t-ből g^{2^l} értékének kiszámolására nem ismert hatékony algoritmus, ez l négyzetremelést igényel $|\mathbb{G}|$ ismerete nélkül (tehát a szekvenciálisság is teljesül).

1.3.7. Megjegyzés. *Olyan* $\mathbb{G}$ *csoportot többféleképpen is lehet készíteni, melynek az elemszáma nem ismert, de mégis tudunk benne számolni.*

Egy lehetőség az az, hogy egy RSA-csoport készítünk. Ennek az a hatránya, hogy szükség van hozzá egy megbízható félre, aki elkészíti nekünk $\mathbb{G}$*-t és ezáltal ő ismeri az elemszámát (ami által képessé válik arra, hogy hatékonyan ki tudja számolni* g^{2^t} *értékét).*

Egy másik lehetőség az úgynevezett osztálycsoport-építés. Ehhez nincs szükség megbízható félre.

2. fejezet

Speciális prímek és prímtesztjeik

2.1 Lucas-Lehmer prímteszt

A Lucas-Lehmer prímteszt az a prímteszt, melynek segítségével a ma ismert 10 legnagyobb prím közül 8-at megtaláltak. A teszt nagyon speciális alakú számokra vonatkozik, emiatt mindössze 51 olyan prím ismert, melynek prímsége ezzel bizonyítható, szóval ezen számokat, a közismertségük miatt, a titkosításban nem alkalmazzák. A legnagyobb ismert prím, melyet ezzel az algoritmussal találtak, a $2^{82589933} - 1$, ami egyben a jelenleg ismert legnagyobb prímszám is.

2.1.1. Definíció. *Egy $M_p = 2^p - 1$ alakú számot **Mersenne-prímnek** nevezünk, ha p és M_p is prím.*

2.1.2. Megjegyzés. *Ha p nem egy prím, akkor M_p sem lehet az, hiszen ha $p = ab$, ahol $a; b > 1$, akkor $M_p = 2^{ab} - 1 = (2^a)^b - 1^b$, amiből ki lehet emelni $2^a - 1$-et, ami $a > 1$ szintén nagyobb 1-nél és $b > 1$ miatt nem is egyenlő M_p-vel.*

2.1.3. Tétel (Lucas-Lehmer-prímteszt)**.** *Legyen p egy 2-nél nagyobb prím és s_n egy pozitív egészekből álló sorozat úgy, hogy $s_0 = 4$ és minden $i > 0$-ra $s_i = s_{i-1}^2 - 2$. Ekkor M_p pontosan akkor prím, ha $M_p | s_{p-2}$.*

A Lucas-Lehmer algoritmus helyességének bizonyítása

Tétel 2.1.3 bizonyítása: Először azt fogjuk megmutatni, hogy $s_p = (2+\sqrt{3})^{2^p} + (2-\sqrt{3})^{2^p}$. (Innentől $(2+\sqrt{3})$-at ω-val, $(2-\sqrt{3})$-at pedig $\overline{\omega}$-val fogjuk jelölni és többször is használjuk majd, hogy $\omega \cdot \overline{\omega} = 1$.) Ez $p = 0$-ra teljesül, innentől pedig indukcióval tudjuk igazolni.

Tegyük fel, hogy i-re igaz, ekkor

$$s_{i+1} = s_i^2 - 2 = \left(\omega^{2^p} + \overline{\omega}^{2^p}\right)^2 - 2 = \left(\omega^{2^p}\right)^2 + 2 \cdot (\omega \cdot \overline{\omega})^{2^p} + \left(\overline{\omega}^{2^p}\right)^2 - 2 = \\ = \omega^{2^{p+1}} + \overline{\omega}^{2^{p+1}} + 2 \cdot 1^{2^p} - 2 = \omega^{2^{p+1}} + \overline{\omega}^{2^{p+1}}. \tag{2.1}$$

Ez pont az, amit be akartunk látni.

Szükségesség:[13] Mivel $p > 2$, $2^p - 1 \equiv 7 \ (mod\ 8)$ és mivel páratlan, $2^p - 1 \equiv 1 \ (mod\ 3)$. Ekkor a kvadratius reciprocitást használva: $\left(\frac{3}{M_p}\right) = -\left(\frac{M_p}{3}\right) = -\left(\frac{1}{3}\right) = -1$, tehát a 3 egy kvadratikus nemmaradék modulo M_p. Ekkor az Euler-kritériumot használva azt kapjuk, hogy $3^{\frac{M_n-1}{2}} \equiv -1 \ (mod\ M_p)$. Továbbá mivel $\left(\frac{2}{M_p}\right) = \left(\frac{2}{7}\right) = 1$, $2^{\frac{M_n-1}{2}} \equiv 1 \ (mod\ M_p)$ is teljesül. Összesítve:

$$24^{\frac{M_p-1}{2}} \equiv \left(2^{\frac{M_p-1}{2}}\right)^3 \cdot 3^{\frac{M_p-1}{2}} \equiv 1 \cdot -1 \equiv -1 \ (mod\ M_p). \tag{2.2}$$

Legyen X az a gyűrű, amit a $\mathbb{Z}_{M_p}\left[\sqrt{3}\right]$ gyűrűbővítéssel kapunk (azaz egy olyan gyűrű melyben $a + b\sqrt{3}$ alakú elemek vannak, ahol a és b is $\mathbb{Z}_{M_p}$ egy eleme). Mivel $p > 2$, ennek $\omega, \overline{\omega}$ és $2\sqrt{3}$ is eleme. Ekkor, (a véges testek fölötti binomiális tétel, a Kis-Fermat tétel és az Euler-kritérium miatt) X-ben:

$$(6 + 2\sqrt{3})^{M_p} = 6^{M_p} \cdot 2^{M_p} \cdot \sqrt{3}^{M_p} = 6 + 2 \cdot 3^{\frac{M_p-1}{2}} \cdot \sqrt{3} = 6 - 2\sqrt{3}. \tag{2.3}$$

Könnyen ellenőrizhető, hogy $\omega = \frac{\left(6-2\sqrt{3}\right)^2}{24}$, tehát X-ben 2.2-t és 2.3-at használva:

$$\omega^{\frac{M_p+1}{2}} = \frac{\left(6 + 2\sqrt{3}\right)^{M_p+1}}{24^{\frac{M_p+1}{2}}} = \frac{\left(6 + 2\sqrt{3}\right)\left(6 + 2\sqrt{3}\right)^{M_p}}{24 \cdot 24^{\frac{M_p-1}{2}}} = \frac{\left(6 + 2\sqrt{3}\right)\left(6 - 2\sqrt{3}\right)}{-24} = -1. \tag{2.4}$$

Azaz X-ben (kihasználva, hogy $\omega \cdot \overline{\omega} = 1$):

$$0 = \omega^{\frac{M_p+1}{2}} \cdot \overline{\omega}^{\frac{M_p+1}{4}} + \overline{\omega}^{\frac{M_p+1}{4}} = \omega^{\frac{M_p+1}{4}} + \overline{\omega}^{\frac{M_p+1}{4}} = \omega^{\frac{2^p}{4}} + \overline{\omega}^{\frac{2^p}{4}} = \omega^{2^{p-2}} + \overline{\omega}^{2^{p-2}} = s_{p-2}. \tag{2.5}$$

X-ben egy x elem pontosan akkor volt 0, ha $\mathbb{Z}_{M_p}$-ben is, ahol pedig akkor, ha $M_n | x$. Ezzel tehát azt kaptuk, hogy $M_p | s_{p-2}$. Ez pedig pont az, amit be akartunk látni.

Elégségesség:[18] Tegyük fel, hogy $M_p | s_{p-2}$, ekkor valamilyen k egész számra $\omega^{2^{p-2}} + \overline{\omega}^{2^{p-2}} = k \cdot M_p$. Ezt $\omega^{2^{p-2}}$-nel szorozva azt kapjuk, hogy

$$\omega^{2^{p-1}} = k \cdot M_p \cdot \omega^{2^{p-2}} - (\omega \cdot \overline{\omega})^{2^{p-2}} = k \cdot M_p \cdot \omega^{2^{p-2}} - 1. \tag{2.6}$$

Legyen q a legkisebb prímosztója M_p-nek. Tegyük fel, hogy M_p összetett, ekkor $q \leq \sqrt{M_p}$. Legyen Y az a gyűrű, amit a $\mathbb{Z}_q\left[\sqrt{3}\right]$ gyűrűbővítéssel kapunk (azaz egy olyan gyűrű melyben $a + b\sqrt{3}$ alakú elemek vannak, ahol a és b is $\mathbb{Z}_q$ egy eleme. Könnyen végiggondolható, hogy ez tényleg egy gyűrű és az elemszáma q^2). M_p páratlan, tehát $q > 2$, azaz ω és $\overline{\omega}$ is eleme ennek a gyűrűnek. Y elemszáma q^2. Mivel $q | M_p$, Y-ban fennáll, hogy $k \cdot M_p \cdot \omega^{2^{p-2}} = 0$, így 2.6 miatt $\omega^{2^{p-1}} = -1$ szintén teljesülni fog Y-ban.

Legyen Y^* az a halmaz, ami Y invertálható elemeiből áll, (mivel $\omega \cdot \overline{\omega} = 1$, ω és $\overline{\omega}$ is Y^* eleme lesz, viszont a 0 nem). Y^* egy csoportot alkot az Y-beli szorzásra nézve. $o_{Y^*}(\omega) | 2^p$, mivel $\omega^{2^p} = \left(\omega^{2^{p-1}}\right)^2 = (-1)^2 = 1$. Viszont tudjuk, hogy $\omega^{2^{p-1}} \neq 1$, tehát $o_{Y^*}(\omega)$ csak 2^p lehet.

Mivel a $0 \notin Y^*$, így $|Y^*| < |Y| = q^2$. Továbbá a Lagrange-tétel miatt 2^p osztója Y^* elemszámának, azaz nem nagyobb nála. Összefoglalva:

$$2^p \leq |Y^*| < q^2 \leq M_p = 2^p - 1 < 2^p. \tag{2.7}$$

Ezzel ellentmondásra jutottunk abból, hogy M_p összetett, ezzel bizonyítva a tételt. □

Páros tökéletes számok

2.1.4. Definíció. *Azon n pozitív egész számokat, melyek osztóinak összege pontosan* $2n$*,* ***tökéletes szám****oknak nevezzük.*

[16]Bizonyított, hogy pontosan azok a páros számok tökéletesek, melyek előállnak $M_p \cdot \frac{M_p+1}{2}$ alakban, ahol M_p egy Mersenne-prím, úgyhogy a Lucas-Lehmer algoritmus ezen számok keresésére is használható.

Páratlan tökéletes számokat jelenleg nem ismerünk, de az sem bizonyított, hogy nem léteznek.

2.2 A Proth-tétel

2.2.1. Definíció. *Egy olyan $P = k \cdot 2^n + 1$ alakú számot, ahol k páratlan és kisebb, mint 2^n,* ***Proth-számnak*** *nevezünk.*

2.2.2. Definíció. *Egy olyan Proth-számot, mely egyben prím is,* ***Proth-prímnek*** *nevezünk.*

2.2.3. Tétel (Proth-tétel)**.** *Egy Proth-szám pontosan akkor prím, ha létezik olyan a egész melyre $a^{\frac{P-1}{2}} \equiv -1 \ (mod\ P)$. Sőt, ha P prím, akkor ha minden olyan a-ra, amire a $\left(\frac{a}{P}\right)$ Jacobi-szimbólum értéke -1, fennáll ez az összefüggés.*

A Proth-tétel bizonyítása

A Proth-tétel bizonyításához az alábbi lemmát fogjuk használni:

2.2.4. Lemma (Pocklington-kritérium)**.** *[9] Legyen $N - 1 = A \cdot B$, ahol $A > B$, továbbá legyenek A különböző prímosztói $p_1; p_2; \ldots; p_k$. Ekkor N pontosan akkor prím, ha minden $1 \leq i \leq k$-ra létezik olyan a_{p_i} egész, hogy $a_{p_i}^{N-1} \equiv 1 \ (mod\ N)$ és $gcd(a_{p_i}^{\frac{N-1}{p_i}} - 1; N) = 0$ (ahol $gcd(a; b)$ a és b legnagyobb közös osztóját jelöli).*

Lemma 2.2.4 bizonyítása: Szükségesség: Ha N egy prím, akkor a_{p_i}-t úgy választva, hogy egy $\mathbb{Z}_N$-beli g generátorelemre $a_{p_i} \equiv g \ (mod\ p_i)$, $a_{p_i}^{N-1} \equiv 1 \ (mod\ N)$ teljesülni fog minden i-re, továbbá $\frac{N-1}{p_i} < N - 1$ miatt $a_{p_i}^{\frac{N-1}{p_i}} \equiv g^{\frac{N-1}{p_i}} \not\equiv 1 \ (mod\ N)$, azaz $a_{p_i}^{\frac{N-1}{p_i}} - 1 \not\equiv 0 \ (mod\ N)$, amiből N prímsége miatt következik, hogy $gcd(a_{p_i}^{\frac{N-1}{p_i}} - 1; N) = 0$.

Elégségesség: Tegyük fel, hogy N összetett ezen feltételek mellett. Legyen q a legkisebb prímosztója N-nek és legyen α_i minden $1 \leq i \leq k$-ra a legnagyobb olyan kitevő, amire $p_i^{\alpha_i} | N$. Továbbá legyen

$$b_i \equiv a_{p_i}^{\frac{N-1}{p_i^{\alpha_i}}} \ (mod\ q). \tag{2.8}$$

Mindkét oldalt a $p_i^{\alpha_i}$-edikre emelve azt kapjuk, hogy

$$b_i^{p_i^{\alpha_i}} \equiv a_{p_i}^{N-1} \equiv 1 \ (mod\ q). \tag{2.9}$$

Ha 2.8-ban $p_i^{\alpha_i - 1}$-edikre emelünk akkor pedig azt, hogy

$$b_i^{p_i^{\alpha_i - 1}} \equiv a_{p_i}^{\frac{N-1}{p_i}} \not\equiv 1 \ (mod\ q). \tag{2.10}$$

(Az utóbbi ekvivalencia azért nem teljesül, mivel ha $a_{p_i}^{\frac{N-1}{p_i}}$ kongruens volna 1-gyel ($mod\ q$), akkor q osztaná $a_{p_i}^{\frac{N-1}{p_i}} - 1$-et és N-et is, ellentmondva annak, hogy $gcd(a_{p_i}^{\frac{N-1}{p_i}} - 1; N) = 0$.)

Mivel q az N összetett szám legkisebb prímosztója volt, $q < \sqrt{N}$, továbbá $A > B$-ből, $A \cdot B = N - 1$-ből és abból, hogy $N > 1$:

$$A \geq \sqrt{N-1} > \sqrt{N} - 1. \tag{2.11}$$

Összefoglalva:

$$\sqrt{N} < \sqrt{N-1} + 1 \leq A + 1 \leq q \leq \sqrt{N} \tag{2.12}$$

Ezzel ellentmondásra jutottunk, bizonyítva N prímségét. □

Tétel 2.2.3 bizonyítása. Írjunk 2.2.4-ben A helyére 2^n-t, B helyére pedig k-t. Ekkor mivel A egyetlen prímosztója a 2, azt kapjuk, hogy P prímsége ekivalens azzal, hogy létezik olyan a egész, amire $a^{P-1} \equiv 1\ (mod\ P)$ és $gcd(a^{\frac{P-1}{2}} - 1; P) = 1$.

Ha P prím akkor a tétel állítása az Euler-kritériumból triviálisan következik.

Tegyük fel, hogy P összetett és valamilyen a-ra $a^{\frac{P-1}{2}} \equiv -1\ (mod\ P)$, ekkor ezt négyzetre emelve azt kapjuk, hogy

$$a^{P-1} \equiv 1\ (mod\ P), \tag{2.13}$$

továbbá

$$a^{\frac{P-1}{2}} - 1 \equiv -2\ (mod\ P). \tag{2.14}$$

Mivel P páratlan, ezért $gcd(-2; P) = 1$.

Összefoglalva, ha volna olyan a, amire $a^{\frac{P-1}{2}} \equiv -1\ (mod\ P)$, akkor P teljesítené a Pocklington-kritériumot, azaz nem lehetne összetett.

□

2.2.5. Megjegyzés. *A legnagyobb ismert Proth-prím jelenleg a* $10223 \cdot 2^{31172165} + 1$*, mely a megtalálásának idején a* 7*. legnagyobb ismert prím és egyben a legnagyobb ismert nem-Mersenne prím is volt.*

Fermat-prímek

2.2.6. Definíció. $s \in \mathbb{N}$ *esetén a* $2^{2^s} + 1$ *alakú számokat* ***Fermat-számoknak*** *nevezzük.*

2.2.7. Definíció. *Azokat a Fermat-számokat, melyek egyben prímek is,* ***Fermat-prímeknek*** *nevezzük.*

2.2.8. Állítás. *Minden $s \in \mathbb{Z}^+$ esetén a $\left(\frac{3}{F_s}\right)$ Jacobi-szimbólum értéke -1.*

Állítás 2.2.8 bizonyítása: Mivel $2^s > 1$, ezért 2^{2^s} osztható 4-gyel, tehát $F_s \equiv 1 \ (mod\ 4)$. Ekkor a kvadratikus reciprocitás miatt $\left(\frac{3}{F_s}\right) = \left(\frac{F_s}{3}\right)$. Mivel $F_s - 1$ a 2-nek egy páros kitevős hatványa, ezért a 4-nek is hatványa, azaz 4^l alakú. Ekkor:

$$F_s \equiv 4^l + 1 \equiv 1^l + 1 \equiv 2 \ (mod\ 3) \tag{2.15}$$

Tehát $\left(\frac{F_s}{3}\right) = \left(\frac{2}{3}\right) = -1$. Éppen ezt akatuk bizonyítani. □

A Proth-tételt $k = 1$ és $n = 2^s$ esetére felírva pont a Fermat-számokat kapjuk, és azt az állítást, hogy $F_s = 2^{2^s} + 1$ pontosan akkor prím, hogyha minden olyan a-ra, amire az $\left(\frac{a}{F_s}\right)$ Jacobi-szimbólum értéke -1, fennáll az, hogy $a^{2^{2^s-1}} \equiv -1 \ (mod\ F_s)$.

[14] Tehát 2.2.8 értelmében F_s pontosan akkor prím, ha $3^{2^{2^s-1}} \equiv -1 \ (mod\ F_s)$. Ezt az állítást szokták **Pépin-teszt**nek nevezni.

Jelenleg 5 Fermat-prím ismert: az $F_0 = 3$, az $F_1 = 5$, az $F_2 = 17$, az $F_3 = 257$, és az $F_4 = 65537$. A tesztet F_{32}-ig végezték el eddig, az a sejtés, hogy, 65537-nél nagyobb Fermat-prím nem is létezik, de ez nem bizonyított.

A Proth-tétel egy lehetséges általánosítása

A Proth-tételt olvasva felmerülhet bennünk az a kérdés, hogy a tétel akkor is igaz marad-e, ha a elhagyjuk belőle a $k < 2^n$ feltételt. Vagy ha általánosan nem is, akkor legalább az, hogy speciális n-ekre igaz lesz-e.

2.2.9. Tétel. *Minden $n > 0$ egészre végtelen sok olyan k páratlan szám van, amire $P = k \cdot 2^n + 1$ összetett és melyre létezik olyan a szám, hogy $a^{\frac{P-1}{2}} \equiv -1 \ (mod\ P)$.*

2.2.10. Lemma. *Legyen p egy olyan prímszám, melyre $p \equiv 1 \ (mod\ n)$ valamilyen n-re. Ekkor $\mathbb{Z}_p$-ben létezik primitív n-edik egységgyök.*

Lemma 2.2.10 bizonyítása: Legyen $p = a \cdot n + 1$. Mivel p prím, emiatt a maradékrend-szerében létezik egy g-vel jelölt generátorelem (azaz primitív $p - 1$-edik egységgyök). Ekkor $g^{\frac{p-1}{n}}$ egy primitív n-edik egységgyök lesz, hiszen az n-edik hatványa tényleg 1 és ha lenne egy $\frac{p-1}{n}$-nél kisebb l szám, amire $(g^l)^n \equiv g^{ln} \equiv 1 \ (mod\ p)$, akkor g-nek lenne egy

$p-1 = a \cdot n$-nél kisebb kitevős h hatványa, ami kongruens lenne 1-gyel modulo p (egész pontosan $h := l \cdot n$), ellentmodva g generátorelemségének. □

Tétel 2.2.9 bizonyítása: Legyen $P = p \cdot q$, ahol p és q olyan prímek, melyekre $p \equiv 1\ (mod\ 2^{n+1})$; $q \equiv 2^n + 1\ (mod\ 2^{n+1})$.
A Dirichlet-tétel alapján minden n-re végtelen sok olyan (p;q) számpár létezik, mely teljesíti ezeket a feltételeket.

Ekkor 2.2.10 miatt $\mathbb{Z}_p$-ben és $\mathbb{Z}_q$-ban léteznek primitív 2^n-edik egységgyökök. Jelöljük ezeket rendre $\epsilon_{2^n}^p$-vel és $\epsilon_{2^n}^q$-val.

Tekintsük az

$$\begin{aligned} a &\equiv \epsilon_{2^n}^p\ (mod\ p) \\ a &\equiv \epsilon_{2^n}^q\ (mod\ q) \end{aligned} \tag{2.16}$$

szimultán kongruenciarendszert.

Ekkor mivel $p \cdot q \equiv 2^n + 1\ (mod\ 2^{n+1})$ és mivel $\epsilon_{2^n}^p$ hatványai 2^n hosszú ciklust alkotnak:

$$a^{\frac{P-1}{2}} = a^{\frac{p \cdot q - 1}{2}} \equiv a^{\frac{(2^n+1)-1}{2}} = a^{2^{n-1}}\ (mod\ p) \tag{2.17}$$

Hasonlóan:

$$a^{\frac{P-1}{2}} = a^{\frac{p \cdot q - 1}{2}} \equiv a^{\frac{(2^n+1)-1}{2}} = a^{2^{n-1}}\ (mod\ q) \tag{2.18}$$

Mivel a 2^n-edik primitív egységgyök volt, $a^{2^{n-1}}$ csak egy primitív második egységgyök lehet, amiből mindössze 1 létezik modulo p és modulo q is, ez pedig a -1.

Tehát:

$$\begin{aligned} a^{\frac{P-1}{2}} &\equiv -1\ (mod\ p) \\ a^{\frac{P-1}{2}} &\equiv -1\ (mod\ q) \end{aligned} \tag{2.19}$$

A kínai maradéktétel értelmében pontosan 1 olyan maradék van modulo $p \cdot q = P$, ami kongruens -1-gyel modulo p és modulo q is. Ez pedig az $a^{\frac{P-1}{2}} \equiv -1\ (mod\ P)$. Tehát minden n-re mutattunk végtelen sok olyan k-t, hogy $k \cdot 2^n + 1$ összetett, mégis létezik olyan a, hogy $a^{\frac{P-1}{2}} \equiv -1\ (mod\ P)$, ezzel bizonyítva a tételt. □

3. fejezet

Óriási nemprímek verifikációja

3.1 Proof of Exponentiation

Tegyük fel, hogy Botond kiszámolta egy adott T pozitív egész számra ismételt négyzetre emeléssel, hogy egy $\mathbb{G}$ csoportban lévő $x; y$ elemekre fennáll a $x^{2^T} = y$ egyenlőség, ahol $T = 1$, vagy páros. Mi szeretnénk leellenőrizni, hogy Botond igazat mondott-e. Ezt a "Pietrzak's Proof of Exponentiation" (innentől PPoE) nevű algoritmussal tudjuk megtenni.

Ez az algoritmus egy olyan VDF, melynek a λ biztonsági paramétere egy páros természetes szám.

Az algoritmus leírása és futásideje

Algoritmus (PPoE):[7]

Bemenet: $(x; y; T; \mathbb{G})$

Ellenőrizendő állítás: $x^{2^T} = y$.

Ha $T = 1$:

Ellenőrizzük le, hogy $x^2 = y$ fennáll-e. Ha igen, akkor *elfogadjuk* az állítást, ha nem, akkor *elutasítjuk*.

Ha $T \neq 1$:

Megkérjük Botondot, hogy mondja el nekünk $v := x^{2^{\frac{T}{2}}}$ értékét (melyet elvileg már kiszámolt korábban).

Ha $v \notin \mathbb{G}$, akkor *elutasítjuk* az állítást.

Ha $v \in \mathbb{G}$:

Generálunk uniform véletlen egy r egész számot az $1; 2; 3; \ldots; 2^\lambda - 1$ számok közül.

Kiszámítjuk $x' := x^r \cdot v$ és $y' := y \cdot v^r$ értékeket.

Ha $\frac{T}{2}$ páros, vagy 1, akkor elvégezzük a $PPoE(x'; y'; \frac{T}{2}; \mathbb{G})$ algoritmust; Ha nem, akkor pedig a $PPoE(x'; y'^2; \frac{T}{2} + 1; \mathbb{G})$ algoritmust.

3.1.1. Megjegyzés. *Az algoritmus futásideje $O(\lambda \cdot log(T))$, hiszen egy köre során az input* 3. *tagja mindig nagyjából a felére csökken. Ezenkívül egy adott kör során $T = 1$ esetén elvégzünk egy szorzást, $T > 1$ esetén pedig generálunk λ random bitet, két számot az r-edik hatványra emelünk, ahol r λ bitből áll és további konstans mennyiségű műveletet végzünk még.*

PPoE teljessége

3.1.2. Tétel. *Ha $x^{2^T} = y$ $\mathbb{G}$-ben, akkor PPoE mindig el fogja fogadni az állítást.*

Tétel 3.1.2 bizonyítása: A bizonyítás során minden lépés értelmes, hiszen kizárólag olyan T-t osztunk 2-vel, amiről megvizsgáltuk előtte, hogy 1-nél nagyobb páros szám és az algoritmust is kirázólag úgy hívjuk meg, hogy a 3. inputján egy páros szám, 1, vagy egy páratlan számnál eggyel nagyobb érték (azaz egy másik páros szám) szerepel. A bizonyítás további részében azt fogjuk megmutatni, hogyha az eredeti állítás igaz volt, akkor az algoritmus során meghívott $PPoE$-nek adott állítás is igaz lesz.

1. eset: Ha $T = 1$, akkor nyilvánvalóan helyes az algoritmus.

2. eset: Ha $T \neq 1$, akkor $\frac{T}{2}$ biztosan egész, tehát v értéke értelmes lesz.

1. aleset: Ha $\frac{T}{2}$ páros, vagy 1 akkor az algoritmust $x^r \cdot v$-re és $y \cdot v^r$-re fogjuk felírni. Ekkor az, hogy $x'^{2^{\frac{T}{2}}} = y'$, azt jelenti, hogy a bal oldalon:

$$(x^r \cdot v)^{2^{\frac{T}{2}}} = \left(x^r \cdot x^{2^{\frac{T}{2}}}\right)^{2^{\frac{T}{2}}} = (x^r)^{2^{\frac{T}{2}}} \cdot x^{2^{\frac{T}{2}} \cdot 2^{\frac{T}{2}}} = x^{r \cdot 2^{\frac{T}{2}}} \cdot x^{2^T}. \tag{3.1}$$

A jobb oldalon pedig:

$$y \cdot v^r = y \cdot \left(x^{2^{\frac{T}{2}}}\right)^r = y \cdot x^{r \cdot 2^{\frac{T}{2}}}. \tag{3.2}$$

Tehát, ha $x^{2^T} = y$, akkor az egyenlőség itt is fennáll, hiszen mindkét oldal ugyanazzal a számmal van szorozva ehhez képest.

2. aleset: Ha $\frac{T}{2}$ páratlan és nem 1, akkor $\frac{T}{2} + 1$ páros. Ekkor az algoritmust $x^r \cdot v$-re és $(y \cdot v^r)^2$-re fogjuk felírni. Ekkor az, hogy $x'^{2^{\frac{T}{2}+1}} = y'^2$, azt jelenti, hogy a jobb oldalon:

$$(x^r \cdot v)^{2^{\frac{T}{2}+1}} = (x^r)^{2^{\frac{T}{2}+1}} \cdot \left(x^{2^{\frac{T}{2}}}\right)^{2^{\frac{T}{2}+1}} = x^{r\cdot 2^{\frac{T}{2}+1}} \cdot x^{2^{\frac{T}{2}}\cdot 2^{\frac{T}{2}+1}} = x^{r\cdot 2^{\frac{T}{2}+1}} \cdot x^{2^{T+1}} \tag{3.3}$$

A jobb oldalon pedig:

$$(y \cdot v^r)^2 = y^2 \cdot \left(\left(x^{2^{\frac{T}{2}}}\right)^r\right)^2 = y^2 \cdot x^{r\cdot 2\cdot 2^{\frac{T}{2}}} = y^2 \cdot x^{r\cdot 2^{\frac{T}{2}+1}}. \tag{3.4}$$

Tehát, ha $x^{2^T} = y$, akkor az egyenlőség itt is fennáll (hiszen ennek a négyzete szerepel mindkét oldalon ugyanazzal a számmal szorozva).

□

PPoE megbízhatósága

3.1.3. Definíció. *Egy* $(x; y\alpha; T; \mathbb{G})$ *négyest* α***-hazugnak*** *és benne* $\alpha \neq 1$*-et* ***rossz elem****nek nevezzük akkor, ha* $x^{2^T} = y$*, de PPoE elfogadja az* $(x; y\alpha; T; \mathbb{G})$ *állítást.*

3.1.4. Megjegyzés. *Mivel* $\mathbb{G}$ *egy csoport, ezért minden eleme felírható fix* y *érték mellett* $y\alpha$ *alakban alkalmas* α *választásával.*

A célunk az lenne, hogy adjunk egy jó felső korlátot arra, hogy *PPoE* elfogad egy hamis állítást abban az esetben, hogy $(x; y; T; \mathbb{G})$ helyett $(x; v; T; \mathbb{G})$ lenne az igaz valamilyen nagyon kis rangú v-re. Ehhez az alábbi lemmát fogjuk használni:

3.1.5. Lemma. *[7] Legyen* $(x; y\alpha; T; \mathbb{G})$ *egy* α*-hazug állítás valamilyen* $\alpha \in \mathbb{G}$*-re. Legyen* μ *egy tetszőleges* $\mathbb{G}$*-beli elem,* p^e *egy olyan prímhatvány, ami osztja* $o_{\mathbb{G}}(\alpha)$*-t,* r *pedig uniform véletlen szám a* $0; 1; 2; \ldots; 2^\lambda$ *halmazból. Továbbá tegyük fel, hogy* $(x^r\mu; \mu^r y\alpha; \frac{T}{2}; \mathbb{G})$ *egy* β*-hazug állítás valamilyen* $\beta \in \mathbb{G}$*-re. Ekkor bármely* $l \leq e$ *számra, annak a valószínűsége, hogy* p^{e-l+1} *nem osztja* β *rendjét legfeljebb* $\frac{1}{p^l}$ *uniform véletlenül választott* r *mellett.*

Lemma 3.1.5 bizonyítása: Legyen γ az a $\mathbb{G}$-beli elem, melyre $\mu = \gamma \cdot x^{2^{\frac{T}{2}}}$. Ekkor $(x^r\mu; \mu^r y\alpha; \frac{T}{2}; \mathbb{G})$-be behelyettesítve a bal oldalon:

$$(x^r \cdot \mu)^{2^{\frac{T}{2}}} = x^{r\cdot 2^{\frac{T}{2}}} \cdot \left(\gamma \cdot x^{2^{\frac{T}{2}}}\right)^{2^{\frac{T}{2}}} = x^{r\cdot 2^{\frac{T}{2}}} \cdot x^{2^T} \cdot \gamma^{2^{\frac{T}{2}}} \tag{3.5}$$

A jobb oldalon pedig:

$$\mu^r \cdot y\alpha = \left(\gamma \cdot x^{2^{\frac{T}{2}}}\right)^r \cdot \alpha \cdot y = x^{r\cdot 2^{\frac{T}{2}}} \cdot \gamma^r \cdot \alpha \cdot y. \tag{3.6}$$

A két oldalt összevetve:

$$x^{2^T} = y \cdot \alpha\gamma^{r-2^{\frac{T}{2}}} \tag{3.7}$$

Tehát, ha az eredeti állítás α-hazug volt és $\beta = \alpha\gamma^{r-2^{\frac{T}{2}}}$, akkor az új állítás β-hazug.

Mi r egyenletesen random választása mellett a $Pr\left[\beta^{p^{e-l}\cdot s} = 1\right]$ valószínűséget szeretnénk megbecsülni, ahol s egy tetszőleges p-vel nem osztható pozitív egész szám (hiszen ebből következik, hogy p^{e-l+1} nem osztja β rangját).

$$Pr\left[\beta^{p^{e-l}\cdot s} = 1\right] = Pr\left[\left(\alpha\gamma^{r-2^{\frac{T}{2}}}\right)^{p^{e-l}\cdot s} = 1\right] = Pr\left[\gamma^{\left(r-2^{\frac{T}{2}}\right)\cdot p^{e-l}\cdot s} = \alpha^{-p^{e-l}\cdot s}\right] \tag{3.8}$$

Legyen $d := o_{\mathbb{G}}(\gamma)$. Mivel γ hatványai d-esével ciklizálnak, emiatt bármely értéket legfeljebb $\frac{1}{d} + \frac{1}{2^\lambda}$ valószínűséggel vesznek fel (a $\frac{1}{2^\lambda}$-os tag azért jelenik meg, mivel ha $d \nmid 2^\lambda$, akkor a hatvány néhány értéket eggyel többször vesz fel). Ezt γ helyett $\gamma^{p^{e-l}\cdot s}$-re felírva:

$$\begin{aligned} Pr\left[\gamma^{\left(r-2^{\frac{T}{2}}\right)\cdot p^{e-l}\cdot s} = \alpha^{-p^{e-l}\cdot s}\right] &\leq \frac{1}{o_{\mathbb{G}}(\gamma^{p^{e-l}\cdot s})} + \frac{1}{2^\lambda} = \frac{1}{\frac{d}{gcd(d;p^{e-l}\cdot s)}} + \frac{1}{2^\lambda} = \\ &= \frac{gcd(d;p^{e-l}\cdot s)}{d} + \frac{1}{2^\lambda} \end{aligned} \tag{3.9}$$

Tegyük fel, hogy $r - 2^{\frac{T}{2}}$ helyére m-et írva, a 3.9 elején felírt egyenlőség fennáll. Ekkor:

$$o_{\mathbb{G}}(\alpha^{-p^{e-l}\cdot s}) = o_{\mathbb{G}}(\gamma^{p^{e-l}\cdot sm}) = \frac{d}{gcd(d;p^{e-l}\cdot s)}. \tag{3.10}$$

Átrendezve (az utolsó egyenlőtlenség azért áll fenn, mivel $p^e \mid o_{\mathbb{G}}(\alpha)$ és $l \leq e$):

$$d = o_{\mathbb{G}}(\alpha^{-p^{e-l}\cdot s}) \cdot gcd(d;p^{e-l}\cdot s\cdot m) = o_{\mathbb{G}}(\alpha^{p^{e-l}\cdot s}) \cdot gcd(d;p^{e-l}\cdot s\cdot m) \geq p^l \cdot gcd(d;p^{e-l}\cdot s\cdot m). \tag{3.11}$$

A 3.8-ben és 3.9-ben kiszámolt egyenlőtlenségbe 3.11-et helyettesítve:

$$Pr\left[\beta^{p^{e-l}\cdot s} = 1\right] \leq \frac{gcd(d;p^{e-l}\cdot s)}{p^l \cdot gcd(d;p^{e-l}\cdot s\cdot m)} + \frac{1}{2^\lambda} \leq \frac{gcd(d;p^{e-l}\cdot s)}{p^l \cdot gcd(d;p^{e-l}\cdot s)} + \frac{1}{2^\lambda} = \frac{1}{p^l} + \frac{1}{2^\lambda}. \tag{3.12}$$

Mivel $\frac{1}{p^l}$ jellemzően sokkal nagyobb $\frac{1}{2^\lambda}$-nál, $\frac{1}{p^l} + \frac{1}{2^\lambda} \approx \frac{1}{p^l}$. Ezzel bebizonyítottuk a lemmát.

□

3.1.6. Tétel. *[7] Legyen $(x; y\alpha; T; \mathbb{G})$ egy α-hazug állítás valamilyen $\alpha \in \mathbb{G}$-re. Továbbá legyen e egy olyan természetes szám melyre $2^e \mid o_{\mathbb{G}}(\alpha)$. Ekkor annak a valószínűsége, hogy valamilyen e-nél kisebb l természetes számra 2^{e-l} nem osztja egy rossz elem rangját a PPoE egy köre után, legfeljebb $\frac{1}{2^l}$.*

Tétel 3.1.6 bizonyítása: 3.1.5-ben megmutattuk, hogy akkor, ha *PPoE* az $(x'; y'; \frac{T}{2}; \mathbb{G}) = (x^r \cdot v; v^r \cdot y; \frac{T}{2}; \mathbb{G})$ állítást vizsgálja meg legközelebb, akkor ez a valószínűség legfeljebb $\frac{1}{2^{l+1}} < \frac{1}{2^l}$ (mivel $2^{e-l} = 2^{e-(l+1)+1}$). Ha a $(x'; y'^2; \frac{T}{2} + 1; \mathbb{G}) = (x^r \cdot v; v^{2r} \cdot y^2; \frac{T}{2} + 1; \mathbb{G})$ állítást, akkor a korábbi rossz elem egyszer négyzetre eleműdik, ami a felére csökkeni a rangját, azaz legfeljebb kétszeresére növeli a valószínűségét az előző esethez képest annak, hogy $\frac{1}{2^{e-l}}$ nem osztja ezt az értéket, ezzel $2 \cdot \frac{1}{2^{l+1}} = \frac{1}{2^l}$ felső korlátot adva. Mivel mindkét esetben legfeljebb $\frac{1}{2^l}$ volt a valószínűség, így általánosan is. Ezzel bizonyítottuk a tételt. □

3.2 Az ellenőrző protokoll

Tegyük fel, hogy Botond egy $k \cdot 2^n + 1$ alakú P Proth-szám esetén kiszámolta egy olyan a-ra $a^{\frac{P-1}{2}}$ értékét, amire az $\left(\frac{a}{P}\right)$ Jacobi-szimbólum értéke -1 és $-\mu$-t kapott (ahol $\mu \neq 1$ és n páratlan), ezzel 2.2.3 alapján bizonyítva μ összetettségét. Mi le szeretnénk ellenőrizni, hogy ennek a hatványnak tényleg $-\mu$-e az értéke, vagy Botond csak át akart minket verni. Ezt az alábbi, λ páros biztonsági paraméterrel rendelkező VDF-fel tudjuk megtenni:

A protokoll leírása és futásideje

Algoritmus:[7]

Bemenet: $(n; k; x; \mu)$

Ellenőrizendő állítás: $x^{\frac{P-1}{2}} = x^{k \cdot 2^{n-1}} \equiv -\mu \ (mod\ P)$, ahol $P = k \cdot 2^n + 1; k\ \mu \not\equiv 1 \ (mod\ P)$ és $\left(\frac{x}{P}\right) = -1$.

Ha $\mu \equiv 1 \ (mod\ P)$, akkor az állítást *elutasítjuk*.

Számoljuk ki az $\left(\frac{P}{x}\right)$ Jacobi-szimbólum értékét!

Ha $\left(\frac{P}{x}\right) = 0$, akkor $gcd(P; x) \neq 1$, szóval P tényleg összetett, azaz az állítást *elfogadjuk*.

Ha $\left(\frac{P}{x}\right) = 1$, akkor az állítást *elutasítjuk*.

Ha $\left(\frac{P}{x}\right) = -1$, akkor:

  Legyen $\mu_1 := \mu^k$.

  Ha $\mu_1 = 1$, akkor:

    Legyen $d := o_P(\mu)$ és $\alpha \equiv 2^{-n} \ (mod\ d)$.

    Ha $x^k \equiv \mu^{2\alpha} \ (mod\ d)$, akkor az állítást *elfogadjuk*, különben *elutasítjuk*.

  Ha $\mu_1 \neq 1$, akkor:

    Legyen $\mu_2 := \mu_1^{2^{\lambda \cdot log_2(n)}}$

Ha $\mu_2 \neq 1$:

Végezzük el a $PPoE(x^k; -\mu; n-1; \mathbb{Z}_P)$ algoritmust. Ha ez elfogadja a megkapott állítást, akkor mi is *elfogadjuk* az általunk kapottat. Ha elutasítja, akkor mi is *elutasítjuk*.

Ha $\mu_2 = 1$:

Számoltassuk ki Botonddal $x^{k \cdot 2^{n-1-\lambda \cdot log_2(n)}}$ értékét! (Ezt gyorsan ki tudja számolni, hiszen $x^{k \cdot 2^{n-1}}$ értékét már állítólag egyszer kiszámolta). A számolás során kapott erdményét jelöljük y-nal. Ezután végezzük el a $PPoE(x^k; y; n-1-\lambda \cdot log_2(n); \mathbb{Z}_P)$ algoritmust.

Ha $PPoE(x^k; y; n-1-\lambda \cdot log_2(n); \mathbb{Z}_P)$ elutasít, akkor mi is *elutasítjuk* az állítást.

Ha $PPoE(x^k; y; n-1-\lambda \cdot log_2(n); \mathbb{Z}_P)$ elfogad:

Ellenőrizzük le, hogy $y^{2^{\lambda \cdot log_2(n)}} = -\mu$ teljesül-e. Ha igen, akkor *elfogadjuk* az állítást. Ha nem, akkor *elutasítjuk*.

3.2.1. Megjegyzés. *Az algoritmus futásideje $O(log(k) + \lambda \cdot log(n))$, hiszen az elején a Jacobi-szimbólum kiszámolása $O(log(n))$ lépés a kvadtraikus reciprocitást használva. Utána $O(log(k))$ lépés μ_1 kiszámolása és $\mu_1 = 1$ esetén az ellenőrzés is. További $O(log(n))$ lépés μ_2 kiszámolása és 3.1.1 alapján a PPoE-ök elvégzése is.*

A protokoll teljessége

3.2.2. Állítás. *[7] Ha $(n; k; x; \mu)$ igaz, akkor a protokoll el fogja fogadni a bemenetet.*

Állítás 3.2.2 bizonyítása: Az elején a protokoll leteszteli, hogy $\mu \not\equiv 1 \pmod P$ fennáll-e, ha nem, akkor a bemenet nem lesz elfogadva, ha igen, akkor pedig ez a feltétel automatikusan teljesül, tehát innentől már nem kell vele foglalkozni. Ezután a protokoll kiszámolja $\left(\frac{P}{X}\right)$ értékét. Mivel $P = k \cdot 2^n + 1$ és $n > 1$, emiatt $P \equiv 1 \pmod 4$, tehát a kvadratikus reciprocitás szerint $\left(\frac{P}{x}\right) = \left(\frac{x}{P}\right)$. Az algoritmus pontosan akkor fog ezután továbbfutni, ha $\left(\frac{P}{x}\right) = -1$, ami szintén része volt az ellenőrizendő állításnak. Ha a protokoll továbbjutott az eddigieken, akkor márcsak azt kell ellenőrizne, hogy $x^{k \cdot 2^{n-1}} = \mu$ fennáll-e.

1. eset: $\mu_1 = 1$:

Ilyenkor a protokoll akkor fogad el, ha $x^k = \mu^{2\alpha}$. Ha a bemenet igaz, akkor ezt átalakítva azt kapjuk, hogy:

$$\mu^{2\alpha} = \mu^{2 \cdot 2^{-n}} = (\mu^2)^{2^{-n}} = ((-\mu)^2)^{2^{-n}} = \left(x^{k \cdot 2^{n-1}}\right)^{2 \cdot 2^{-n}} = x^k. \tag{3.13}$$

Tehát, ha a bemenet igaz, akkor ebben az esetben tényleg el fogja fogadni azt a protokoll.

2. eset: $\mu_1 \neq 1 \neq \mu_2$**:**

Ilyenkor a protokoll elvégzi a $PPoE(x^k; -\mu; n-1; \mathbb{Z}_p)$ algoritmust.

Amit $PPoE$ ezáltal ellenőriz az az, hogy $\left(x^k\right)^{2^{n-1}} = y$ fennáll-e, ami pont az az állítás, amit ellenőrizni akarunk és mivel $PPoE$ mindig elfogadja az igaz állítást, a protokoll szintén el fogja fogadni ebben a lépésben.

3. eset: $\mu_1 \neq 1 = \mu_2$**:**

Ilyenkor $\left(x^k\right)^{2^{n-1-\lambda \cdot log_2(n)}} = x^{k \cdot 2^{n-1-\lambda \cdot log_2(n)}} = y$. Amit ilyenkor leellenőrzünk az az, hogy $y^{2^{\lambda \cdot log_2(n)}} = -\mu$ fennáll-e. Ha ez az állítás igaz volt, akkor:

$$-\mu = y^{2^{\lambda \cdot log_2(n)}} = \left(x^{k \cdot 2^{n-1-\lambda \cdot log_2(n)}}\right)^{2^{\lambda \cdot log_2(n)}} = x^{k \cdot 2^{n-1-\lambda \cdot log_2(n)+\lambda \cdot log_2(n)}} = x^{k \cdot 2^{n-1}}. \tag{3.14}$$

Tehát a protokoll itt pontosan akkor fogja elfogadni a bemenetet, ha az ellenőrizendő állítás igaz volt.

Tehát mindhárom esetben el volt fogadva az igaz bemenet. Ezzel az állítást bizonyítottuk. □

A protokoll megbízhatósága

3.2.3. Állítás. *[7] Annak a valószínűsége, hogy a protokoll hibásan elfogadja a bemenetet (azaz* $x^{k \cdot 2^{n-1}} = -1$*, a protokoll viszont mégis elfogadja a bemenetet), legfeljebb* $2^{-\lambda+2} \cdot log_2(n)$*.*

Állítás 3.2.3 bizonyítása: A Jacobi-szimbólum kiszámolását hiba nélkül meg tudjuk csinálni a kvadratikus reciprocitást használva, tehát csak a további lépések során keletkezhet hiba. Ezekről külön-külön fogjuk bizonyítani, hogy kisebb valószínűséggel hibáznak, mint $2^{-\lambda+2} \cdot log_2(n)$.

1. eset: $\mu_1 = 1$:

Csak akkor van probléma, ha az algoritmus úgy fogadja el a bizonyítékot, hogy $x^{k \cdot 2^{n-1}} \equiv -1 \ (mod\ P)$. Tegyük fel, hogy P prím. Mivel $\mu_1 = \mu^k = 1$, ezért $d = o_P(\mu) | k$, tehát k páratlanságából kifolyólag d is páratlan.

Mivel P prím, 2.2.3 miatt $\left(x^k\right)^{2^{n-1}} \equiv -1 \ (mod\ P)$, tehát $o_{\mathbb{Z}_P}(x^k) = 2^n$. Másrészt viszont tudjuk, hogy $o_{\mathbb{Z}_P}(\mu^2) = d$ (hiszen ha μ rangja páratlan, akkor $o_{\mathbb{Z}_P}(\mu) = o_{\mathbb{Z}_P}(\mu^2)$), szóval

$o_{\mathbb{Z}_P}(\mu^{2\alpha})|d$, azaz szintén egy páratlan szám. Ezeket összerakva, ha P prím, $\mu^{2\alpha}$ rangja más, mint x^k-é, tehát $x^k \neq \mu^{2\alpha}$.

Ezzel bizonyítottuk, hogy ebben a lépésben sem keletkezhet hiba.

2. eset: $\mu_1 \neq 1 \neq \mu_2$:

Ilyenkor a protokoll elvégzi a $PPoE(x^k; -\mu; n-1; \mathbb{Z}_p)$ algoritmust.

Ekkor ha P prím, akkor a Kis-Fermat-tételből $\mu^{k\cdot 2^n} \equiv 1 \ (mod\ P)$, de $1 \not\equiv \mu_2 \equiv \mu_1^{2^{\lambda\cdot log_2(n)}} \equiv$ $\equiv \mu^{k\cdot 2^{\lambda\cdot log_2(n)}} \ (mod\ P)$. Ezt a kettőt összerakva: $2^{\lambda\cdot log_2(n)}|o_{\mathbb{Z}_P}(\mu)$.

Mivel $o_{\mathbb{Z}_P}(1) = 1$, ezért ahhoz, hogy a protokoll hibásan elfogadja a bemenetet, egy rossz elem rangjának PPoE egy lépése során átlagosan 2^λ-os tényezővel kellene csökkennie, hiszen kezdetben ez a rang legalább $2^{\lambda\cdot log_2(n)}$ és PPoE annyi kört tesz, amennyi a bemenete 3. elemének logaritmusának felső egészrésze, azaz ebben az esetben $log_2(n)$-et (hiszen a 3. tényező legfeljebb 1 eltéréssel mindig a felére csökken $PPoE$ egy köre során).

Speciálisan legalább egy körben kellene a rossz elem rangjának egy 2^λ-os tényezővel csökkennie. 3.1.6 alapján 1 kör esetén ennek a valószínűsége legfeljebb $2^{-\lambda+2}$, azaz $log_2(n)$ kör esetén legfeljebb $2^{-\lambda+2} \cdot log_2(n)$.

Ezzel erre az esetre is bizonyítottuk az állításban megadott korlátot a hiba valószínűségére.

3. eset: $\mu_1 \neq 1 = \mu_2$:

Ha ez az eset áll fenn, akkor pontosan akkor fog az algoritmus hibásan elfogani egy bizonyítékot, ha $y^{2^{\lambda\cdot log_2(n)}} = -\mu$ (mivel ez pontosan ki tudjuk számolni) és $PPoE(x^k; y; n - 1 - \lambda \cdot log_2(n); \mathbb{Z}_P)$ hibásan elfogad.

Ez a probléma akkor következhet be, ha $PPoE$ olyan α-ra α-hazug, amire $-\mu = y^{2^{\lambda\cdot log_2(n)}} = (y'\alpha)^{2^{\lambda\cdot log_2(n)}} = y'^{2^{\lambda\cdot log_2(n)}} \cdot \alpha^{2^{\lambda\cdot log_2(n)}}$, továbbá $y'^{2^{\lambda\cdot log_2(n)}} = -1$, ahol y' jelöli $x^{k\cdot 2^{n-1-\lambda\cdot log_2(n)}}$ helyes eredményét $\mathbb{Z}_P$-ben (ha az utóbbi feltétel nem teljesül, akkor P 2.2.3 értelmében nem prím, szóval nem hibázunk az összetettség verifikációja során).

Ezeket összerakva akkor lehet csak probléma, ha $-\alpha^{2^{\lambda\cdot log_2(n)}} = -\mu$, azaz

$$\alpha^{2^{\lambda\cdot log_2(n)}} = \mu. \tag{3.15}$$

Könnyen végiggondolható, hogy általános $\mathbb{G}$ csoportban teljesül az, hogy tetszőleges $a \in \mathbb{G}$ elemre és $i \in \mathbb{Z}^+$-ra

$$o_{\mathbb{G}}\left(a^{2^i}\right) = \frac{o_{\mathbb{G}}(a)}{gcd\left(2^i; o_{\mathbb{G}}(a)\right)}. \tag{3.16}$$

Másként: α^{2^i} rangja $o_{\mathbb{G}}(\alpha)$ legnagyobb páratlan osztója, ha $2^i \nmid o_{\mathbb{G}}(\alpha)$ és $\frac{o_{\mathbb{G}}(\alpha)}{2^i}$, ha $2^i \mid o_{\mathbb{G}}(\alpha)$.

Mivel $o_{\mathbb{Z}_p}(\mu)$ páros (hiszen különben $\mu_1 = 1$ teljesült volna már korábban), 3.15 miatt $o_{\mathbb{Z}_p}(\alpha^{2^{\lambda \cdot log_2(n)}})$ is az, tehát 3.16 miatt $2^{\lambda \cdot log_2(n)} \mid o_{\mathbb{G}}(\alpha)$.

Tehát egy rossz elem rangja legalább $2^{\lambda \cdot log_2(n)}$ $PPoE(x^k; y; n - 1 - \lambda \cdot log_2(n); \mathbb{Z}_P)$-ben. Innentől a 2. esettel megegyező módon bizonyíthatjuk, hogy a hiba valószínűsége legfeljebb $2^{-\lambda+2} \cdot log_2(n)$.

Ezzel mindhárom esetre beláttuk, hogy legfeljebb $2^{-\lambda+2} \cdot log_2(n)$ a hamis bemenet elfogadásának valószínűsége, ezzel bizonyítva az állítást. □

4. fejezet

Prímszámok $k \cdot 2^n \cdot 3^m + 1$ alakban

4.1 A Proth-tétel átalakítása $k \cdot 2^n \cdot 3^m + 1$ alakú számokra

4.1.1. Definíció. *Nevezzük egy m pozitív egész és p prímszám kubikus szimbólumának azt a* $\left[\frac{m}{p}\right]_{\mathbb{Z}}$*-vel jelölt függvényt, ami* 1*, ha m kubikus maradék modulo p;* -1*, ha kubikus nemmaradék modulo p; és* 0*, ha p|m.*

4.1.2. Definíció. *Definiáljuk egy m pozitív egész és n összetett szám* $\left[\frac{m}{n}\right]_{\mathbb{Z}}$*-nel jelölt kubikus szimbólumát az alábbi módon:*

$$\left[\frac{m}{n}\right]_{\mathbb{Z}} := \left[\frac{m}{p_1}\right]_{\mathbb{Z}} \cdot \left[\frac{m}{p_2}\right]_{\mathbb{Z}} \cdot \ldots \cdot \left[\frac{m}{p_k}\right]_{\mathbb{Z}}, \tag{4.1}$$

ahol $p_1, p_2, \ldots p_k$ *a multipicitással vett prímosztói n-nek.*

4.1.3. Tétel (Proth-tétel $k \cdot 2^n \cdot 3^m + 1$ alakú számokra)**.** *Legyen* $A = k \cdot 2^n \cdot 3^m + 1$*, ahol* $k < 2^n \cdot 3^m$ *egész,* $2; 3 \nmid k$*, illetve* $n; m > 0$ *egészek. Ekkor A pontosan akkor prím, ha minden olyan a-ra, amire a* $\left(\frac{a}{A}\right)$ *Jacobi- és* $\left[\frac{a}{A}\right]_{\mathbb{Z}}$ *kubikus szimbólum értéke egyszerre* -1*, fennállnak az alábbiak:*

- $a^{\frac{A-1}{6}} \equiv \epsilon_6 \ (mod\ A)$
- $gcd(\epsilon_6 + 1; A) = 1$

Ahol ϵ_6 *egy primitív 6. egységgyököt jelöl modulo A (mindkét helyen ugyanazt a primitív 6. egységgyököt).*

Tétel 4.1.3 bizonyítása: Szükségesség: Ha A egy prímszám, akkor abból, hogy 6-tal osztva 1 maradékot ad, 2.2.10 miatt következik az, hogy létezik primitív 6. egységgyök modulo A. Mivel a ekkor kvadratikus nemmaradék, így $a^{\frac{A-1}{2}} \equiv -1 \ (mod\ A)$, továbbá mivel a egy kubikus nemmaradék, valamilyen ϵ_3 primitív 3. egységgyökre $a^{\frac{A-1}{3}} \equiv \epsilon_3 \ (mod\ A)$. Így $a^{\frac{A-1}{6}} \equiv a^{\frac{A-1}{2}-\frac{A-1}{3}} \equiv \frac{a^{\frac{A-1}{2}}}{a^{\frac{A-1}{3}}} \equiv \frac{-1}{\epsilon_3} \equiv \frac{(\epsilon_6)^3}{(\epsilon_6)^2} \equiv \epsilon_6 \ (mod\ A)$. Tehát az első feltétel teljesül.
Továbbá mivel a -1 nem lehet primitív 6. egységgyök (ugyanis a négyzete 1): $A \nmid a^{\frac{A-1}{6}} + 1$, azaz A prímségéből kifolyólag $gcd(\epsilon_6; A) = 1$. Tehát a második feltétel is teljesül.

Elégségesség: Használjuk 2.2.4-et! A-ra azt mondja, hogy pontosan akkor prím, ha létezik olyan a_2 egész, amire $a_2^{A-1} \equiv 1 \ (mod\ A)$ és $gcd\left(a_2^{\frac{A-1}{2}} - 1; A\right) = 1$, továbbá egy olyan a_3 egész, melyre $a_3^{A-1} \equiv 1 \ (mod\ A)$ és $gcd\left(a_3^{\frac{A-1}{3}} - 1; A\right) = 1$. Azt fogjuk bizonyítani, hogy ez következik a tételben szereplő állításból.

Ha a_2 és a_3 helyére is a-t írjuk, akkor az az $a_2^{A-1} \equiv 1 \ (mod\ A)$ és az $a_3^{A-1} \equiv 1 \ (mod\ A)$ feltételeket teljesíti, mivel $a^{A-1} \equiv \left(a^{\frac{A-1}{6}}\right)^6 \equiv (\epsilon_6)^6 \equiv 1 \ (mod\ A)$.

Az $gcd(a_2^{\frac{A-1}{2}} - 1; A)$ feltétel azért fog teljesülni, mivel a kvadratikus nemmaradék, azaz $a_2^{\frac{A-1}{2}} \equiv -1 \ (mod\ A)$. Tehát $gcd(a_2^{\frac{A-1}{2}} - 1; A) = gcd(-2; A) = gcd(2; A)$, ami A páratlanságából kifolyólag 1.

a egy kubikus nemmaradék, így $a^{\frac{A-1}{3}} \not\equiv 1 \ (mod\ A)$, azaz:

$$\begin{aligned} 1 = gcd(\epsilon_6 + 1; A) = gcd(-1 \cdot (\epsilon_6 + 1); A) = gcd((\epsilon_6)^3 \cdot (\epsilon_6 + 1); A) = \\ = gcd((\epsilon_6)^4 - 1; A) = gcd((\epsilon_3)^2 - 1; A) = gcd\left(\left(a^{\frac{A-1}{3}}\right)^2 - 1; A\right) = \\ = gcd\left(\left(a^{\frac{A-1}{3}} - 1\right) \cdot \left(a^{\frac{A-1}{3}} + 1\right); A\right) = gcd\left(a^{\frac{A-1}{3}} - 1; A\right) \cdot gcd\left(a^{\frac{A-1}{3}} + 1; A\right). \end{aligned} \tag{4.2}$$

Mivel az utolsó két szám szorzata 1 és mindkettő pozitív egész, mindkettőnek 1-nek kell lennie, tehát $gcd(a^{\frac{(A-1)}{3}} - 1; A) = 1$ is teljesül. □

4.1.4. Megjegyzés. *Ez a tételt a $k \cdot 3^m + 1$ alakú számokra is használhatjuk, ahol $k < 3^m$, hiszen ha k páratlan, akkor ez biztosan nem prím. Ha pedig páros, akkor ki tudjuk belőle emelni a legnagyobb 2-hatványt amivel osztható, ezzel egy $k' \cdot 2^n \cdot 3^m$ alakú számot kapva, ahol $2^n \cdot 3^m > 3^m > k = k' \cdot 2^n > k'$, ami pont az a feltétel, ami az előző tétel alkalmazásához szükséges.*

4.2 PPoE $k \cdot 2^n \cdot 3^m + 1$ alakú számokra

Szeretnénk a PPoE algoritmust átalakítani arra az esetre, ha a $\mathbb{G}$ csoportban azt szeretnénk leellenőrizni, hogy $x^{2^T \cdot 3^M} = y$ teljesül-e, ahol T és M is pozitív páros számok, vagy 1-gyel egyenlőek és λ továbbra is egy páros biztonsági paraméter:

Tegyük fel, hogy Botond $x^{2^T \cdot 3^M}$ értékét úgy számolta ki, hogy néha négyzetre és néha köbre emelte az általa előbb kiszámolt értéket, mely kezdetben x (összesen T-szer emelt négyzetre és M-szer köbre) és közben ügyelt arra, hogy "arányosan hatványozzon", azaz minden $1 \leq k \leq T + M$-re a k. hatványozásig nagyjából $T \cdot \frac{k}{T+M}$-szer emelt négyzetre és $M \cdot \frac{k}{T+M}$-szer köbre.

Az algoritmus leírása és futásideje

Algoritmus ($PPoE_2$):

Bemenet: $(x; y; T; M; \mathbb{G})$

Ellenőrizendő állítás: $x^{2^T \cdot 3^M} = y$.

Ha $T = 1$:

Ha $M = 1$:

Ellenőrizzük le, hogy $x^6 = y$ fennáll-e. Ha igen, akkor *elfogadjuk* az állítást, ha nem, akkor *elutasítjuk*.

Ha $M \neq 1$:

Megkérjük Botondot, hogy mondja el nekünk $w := x^{3^{\frac{M}{2}}}$ értékét (melyet a korábban kiszámolt négyzetre és köbre emeléseiből gyorsan ki tud számolni, ha figyelt arra, hogy arányosan hatványozzon).

Ha $w \notin \mathbb{G}$, akkor *elutasítjuk* az állítást.

Ha $w \in \mathbb{G}$:

Generálunk uniform véletlen egy r egész számot a $1; 2; 3; \ldots; 2^\lambda - 1$ halmazból.

Kiszámítjuk az $x^* := x^r \cdot w$ és az $y^* := y \cdot w^{2r}$ értékeket.

Ha $\frac{M}{2}$ páros, vagy 1, akkor elvégezzük a $PPoE_2(x^*; y^*; 1; \frac{M}{2}; \mathbb{G})$ algoritmust; Ha nem, akkor pedig a $PPoE_2(x^*; y^{*3}; 1; \frac{M}{2} + 1; \mathbb{G})$ algoritmust.

Ha $T \neq 1$:

Ha $M = 1$:

Megkérjük Botondot, hogy mondja el nekünk $w := x^{2^{\frac{T}{2}}}$ értékét (melyet a korábban kiszámolt négyzetre és köbre emeléseiből gyorsan ki tud számolni, ha figyelt arra, hogy arányosan hatványozzon).

Ha $w \notin \mathbb{G}$, akkor *elutasítjuk* az állítást.

Ha $w \in \mathbb{G}$:

Generálunk uniform véletlen egy r egész számot a $1; 2; 3; \ldots; 2^\lambda - 1$ halmazból.

Kiszámítjuk az $x^* := x^r \cdot w$ és az $y^* := y \cdot w^{3r}$ értékeket.

Ha $\frac{T}{2}$ páros, vagy 1, akkor elvégezzük a $PPoE_2(x^*; y^*; \frac{T}{2}; 1; \mathbb{G})$ algoritmust; Ha nem, akkor pedig a $PPoE_2(x^*; y^{*2}; \frac{T}{2} + 1; 1; \mathbb{G})$ algoritmust.

Ha $M \neq 1$

Megkérjük Botondot, hogy mondja el nekünk $v := x^{2^{\frac{T}{2}} \cdot 3^{\frac{M}{2}}}$ értékét (melyet a korábban kiszámolt négyzetre és köbre emeléseiből gyorsan ki tud számolni, ha figyelt arra, hogy arányosan hatványozzon).

Ha $v \notin \mathbb{G}$, akkor *elutasítjuk* az állítást.

Ha $v \in \mathbb{G}$:

Generálunk uniform véletlen egy r egész számot az $1; 2; 3; \ldots; 2^\lambda - 1$ halmazból.

Kiszámítjuk az $x' := x^r \cdot v$ és az $y' := y \cdot v^r$ értékeket.

Ha $\frac{T}{2}$ páros, vagy 1, és $\frac{M}{2}$ is:

Elvégezzük a $PPoE_2(x'; y'; \frac{T}{2}; \frac{M}{2}; \mathbb{G})$ algoritmust;

Ha $\frac{T}{2}$ páros, vagy 1, de $\frac{M}{2}$ egy 1-nél nagyobb páratlan szám:

Elvégezzük a $PPoE_2(x'; y'^3; \frac{T}{2}; \frac{M}{2} + 1; \mathbb{G})$ algoritmust.

Ha $\frac{M}{2}$ páros, vagy 1, de $\frac{T}{2}$ egy 1-nél nagyobb páratlan szám:

Elvégezzük a $PPoE_2(x'; y'^2; \frac{T}{2} + 1; \frac{M}{2}; \mathbb{G})$ algoritmust.

Ha $\frac{T}{2}$ és $\frac{M}{2}$ is 1-nél nagyobb páratlan számok:

Elvégezzük a $PPoE_2(x'; y'^6; \frac{T}{2} + 1; \frac{M}{2} + 1; \mathbb{G})$ algoritmust.

4.2.1. Megjegyzés. *$PPoE_2$ futásideje $O(\lambda \cdot (log(T) + log(M))$, ez 3.1.1-hez hasonlóan igazolható.*

Az algoritmus teljessége

4.2.2. Állítás. *Ha $x^{2^T \cdot 3^M} = y$ $\mathbb{G}$-ben, akkor $PPoE_2$ mindig el fogja fogadni az állítást.*

Állítás 4.2.2 bizonyítása: A bizonyítás során minden lépés értelmes, hiszen kizárólag olyan számokat osztunk 2-vel, melyekről korábban már tudjuk, hogy párosak és az algoritmust

is kirázólag úgy hívjuk meg, hogy a 3. és 4. inputján egy páros szám, 1, vagy egy páratlan számnál eggyel nagyobb érték (azaz egy másik páros szám) szerepel. A bizonyítás további részében azt fogjuk megmutatni, hogyha az eredeti állítás igaz volt, akkor az algoritmus során meghívott $PPoE_2$-nek adott állítás is igaz lesz.

1. eset: Ha $T = M = 1$, akkor nyilvánvalóan helyes az algoritmus.

2. eset: Ha $T = 1 \neq M$, akkor $\frac{M}{2}$ biztosan egész, tehát w értéke értelmes lesz.

1. aleset: $\frac{M}{2}$ páros, vagy 1

Ekkor az algoritmust $(x^r \cdot w; y \cdot w^{2r}; 1; \frac{M}{2}; \mathbb{G})$-re fogjuk felírni. Alakítsuk át a két oldalt, a bal odalon:

$$(x^r \cdot w)^{2 \cdot 3^{\frac{M}{2}}} = \left(x^r \cdot x^{3^{\frac{M}{2}}}\right)^{2 \cdot 3^{\frac{M}{2}}} = (x^r)^{2 \cdot 3^{\frac{M}{2}}} \cdot x^{3^{\frac{M}{2}} \cdot 2 \cdot 3^{\frac{M}{2}}} = x^{2r \cdot 3^{\frac{M}{2}}} \cdot x^{2 \cdot 3^M}. \tag{4.3}$$

A jobb oldalon pedig:

$$y \cdot w^{2r} = y \cdot \left(x^{3^{\frac{M}{2}}}\right)^{2r} = y \cdot x^{2r \cdot 3^{\frac{M}{2}}}. \tag{4.4}$$

Tehát, ha $x^{2 \cdot 3^M} = y$, teljesül, akkor az egyenlőség 4.3 és 4.4 között is fennáll, mivel ugyanazzal számmal vannak szorozva $x^{2 \cdot 3^M} = y$-hoz képest.

2. aleset: $\frac{M}{2}$ páratlan és nem 1

Ekkor az algoritmust $(x^r \cdot w; (y \cdot w^{2r})^3; 1; \frac{M}{2}; \mathbb{G})$-re fogjuk felírni. Alakítsuk át a két oldalt, a bal oldalon:

$$(x^r \cdot w)^{2 \cdot 3^{\frac{M}{2}+1}} = \left(x^r \cdot x^{3^{\frac{M}{2}}}\right)^{2 \cdot 3^{\frac{M}{2}+1}} = (x^r)^{2 \cdot 3^{\frac{M}{2}+1}} \cdot x^{3^{\frac{M}{2}} \cdot 2 \cdot 3^{\frac{M}{2}+1}} = x^{2r \cdot 3^{\frac{M}{2}+1}} \cdot x^{2 \cdot 3^{M+1}}. \tag{4.5}$$

A jobb oldalon pedig:

$$(y \cdot w^{2r})^3 = y^3 \cdot \left(\left(x^{3^{\frac{M}{2}}}\right)^{2r}\right)^3 = y^3 \cdot x^{2r \cdot 3 \cdot 3^{\frac{M}{2}}} = y^3 \cdot x^{2r \cdot 3^{\frac{M}{2}+1}}. \tag{4.6}$$

Tehát, ha $x^{2 \cdot 3^M} = y$, akkor az egyenlőség itt is fennáll (hiszen ennek a köbe szerepel mindkét oldalon ugyanazzal a számmal szorozva).

3. eset: Ha $T \neq 1 = M$:

4.5-höz és 4.6-hoz hasonlóan ez az eset ugyanúgy bizonyítható, mint a 2. eset.

4. eset: Ha $T \neq 1 \neq M$:

Ebben az esetben 4 lehetséges aleset is lenne, de itt csak azt az egyet bizonyítjuk, ha $\frac{T}{2}$ és $\frac{M}{2}$ is 1-nél nagyobb páratlan számok. A másik három 4.7-hez és 4.8-höz hasonlóan végigszámolható.

Ebben az esetben az algoritmust $(x^r \cdot v; (y \cdot v^r)^6; \frac{T}{2} + 1; \frac{M}{2} + 1; \mathbb{G})$-re fogjuk felírni. Alakítsuk át a két oldalt, a bal oldalon:

$$\begin{aligned}(x^r \cdot v)^{2^{\frac{T}{2}+1} \cdot 3^{\frac{M}{2}+1}} = \left(x^r \cdot x^{2^{\frac{T}{2}} \cdot 3^{\frac{M}{2}}}\right)^{2^{\frac{T}{2}+1} \cdot 3^{\frac{M}{2}+1}} = (x^r)^{2^{\frac{T}{2}+1} \cdot 3^{\frac{M}{2}+1}} \cdot x^{2^{\frac{T}{2}} \cdot 2^{\frac{T}{2}+1} \cdot 3^{\frac{M}{2}} \cdot 3^{\frac{M}{2}+1}} = \\ = x^{6r \cdot 2^{\frac{T}{2}} \cdot 3^{\frac{M}{2}}} \cdot x^{6 \cdot 2^T \cdot 3^M}.\end{aligned} \tag{4.7}$$

A jobb oldalon pedig:

$$(y \cdot v^r)^6 = y^6 \cdot \left(\left(x^{2^{\frac{T}{2}} \cdot 3^{\frac{M}{2}}}\right)^r\right)^6 = y^6 \cdot x^{6r \cdot 2^{\frac{T}{2}} \cdot 3^{\frac{M}{2}}}. \tag{4.8}$$

Tehát, ha $x^{2 \cdot 3^M} = y$, akkor az egyenlőség itt is fennáll (hiszen ennek a 6. hatványa szerepel mindkét oldalon ugyanazzal a számmal szorozva). □

Az algoritmus megbízhatósága

4.2.3. Definíció. *Egy* $(x; y\alpha; T; M; \mathbb{G})$ *ötöst 3.1.3-hoz hasonlóan akkor nevezzük* α***-hazugnak*** *és benne* $\alpha \neq 1$*-t* ***rossz elem****nek, ha* $x^{2^T \cdot 3^M} = y$*, de* $PPoE_2$ *elfogadja az* $(x; y\alpha; T; M; \mathbb{G})$ *állítást.*

A célunk az lenne, *PPoE*-höz hasonlóan $PPoE_2$-ről is belássuk, hogy kis valószínűséggel csökken nagy mértékben a rangja. Ehhez az alábbi lemmát fogjuk használni:

4.2.4. Lemma. *Legyen* $(x; y\alpha; T; M; \mathbb{G})$ *egy* α*-hazug állítás valamilyen* $\alpha \in \mathbb{G}$*-re. Legyen* μ *egy tetszőleges* $\mathbb{G}$*-beli elem,* p^e *egy olyan prímhatvány, ami osztja* α $\mathbb{G}$*-beli rendjét,* r *pedig egy tetszőleges szám a* $0; 1; 2; \ldots; 2^\lambda$ *halmazból. Továbbá tegyük fel, hogy* $(x^r\mu; \mu^r y\alpha; \frac{T}{2}; \frac{M}{2}; \mathbb{G})$ *egy* β*-hazug állítás valamilyen* $\beta \in \mathbb{G}$*-re. Ekkor bármely* $l \leq e$ *számra, annak a valószínűsége, hogy* p^{e-l+1} *nem osztja* β *rendjét legfeljebb* $\frac{1}{p^l}$.

Lemma 4.2.4 bizonyítása: A lemma 3.1.5-gyel szinte teljesen megegyező módon bizonyítható, mindössze azzal a különbséggel, hogy a bizonyítás során $2^{\frac{T}{2}}$ helyett $2^{\frac{T}{2}} \cdot 3^{\frac{M}{2}}$-t írunk minden helyen, a 2^T helyett pedig $2^T \cdot 3^M$-et. □

4.2.5. Megjegyzés. *4.2.4 akkor is igaz, ha* $(x^r\mu; \mu^r y\alpha; \frac{T}{2}; \frac{M}{2}; \mathbb{G})$ *helyett* $(x^r\mu; \mu^r y\alpha; \frac{T}{2}; 1; \mathbb{G})$, *vagy* $(x^r\mu; \mu^r y\alpha; 1; \frac{M}{2}; \mathbb{G})$ *a* β*-hazug állítás, a bizonyítás ezekben az esetekben is teljesen hasonló 3.1.5-höz.*

4.2.6. Tétel. *Legyen* $(x; y\alpha; T; M; \mathbb{G})$ *egy* α*-hazug állítás valamilyen* $\alpha \in \mathbb{G}$*-re. Továbbá legyen e egy olyan természetes szám melyre* $2^e \mid o_{\mathbb{G}}(\alpha)$*, f pedig olyan, amire* $3^f \mid o_{\mathbb{G}}(\alpha)$*. Ekkor annak a valószínűsége, hogy valamilyen e-nél kisebb l természetes számra* 2^{e-l} *nem osztja egy rossz elem rangját a* $PPoE_2$ *egy köre után, legfeljebb* $\frac{1}{2^l}$*. Továbbá annak a valószínűsége, hogy valamilyen* f*-nél kisebb h természetes számra* 3^{f-h} *nem osztja egy rossz elem rangját a* $PPoE_2$ *egy köre után, legfeljebb* $\frac{1}{3^h}$*.*

Tétel 4.2.6 bizonyítása: A tétel 3.1.6-tal szinte megegyező módon bizonyítható 4.2.4 és 4.2.5 felhasználásával, csak nem 2 esetre kell felírnunk a lemmák valamelyikét (aszerint, hogy $PPoE_2$-nek mi lesz a következő köre), hanem 8-ra és mindet a 2-es és 3-as kitevőre is. Közben, használjuk azt, hogy tetszőleges $g \in \mathbb{G}$-re $o_{\mathbb{G}}(g)$ pontosan akkor osztható q^i-vel, ha $o_{\mathbb{G}}(g^p)$ is, ahol p és q egymástól különböző prímek. A bizonyítás során $p = 2$ és $q = 3$. □

4.3 Az ellenőrző protokoll

Ha $A = k \cdot 2^n \cdot 3^m + 1$ (ahol n és m páratlanok, mindketten nagyobbak, mint a logaritmusuk valamilyen λ-szorosa és $2; 3 \nmid k$) alakú számokra szeretnénk átalakítani a korábban látott összetettség-ellenőrző protokollt, arra 4.1.3 alapján kétféle bizonyítékot kaphatunk Botondtól. Az egyik az az, hogy mutat nekünk egy olyan ϵ_6 hatodik egységégyököt, amire $gcd(\epsilon_6 + 1; A) \neq 1$. Ezt könnyen le tudjuk ellenőrizni az Euklideszi-algoritmussal (de ha bármilyen más olyan r számot mutat, amire $gcd(r; A) \neq 1$ és $A \nmid r$, az szintén egy jó bizonyíték A összetettségére). A másik lehetőség az az, hogy egy x-et mutat, amire $\left(\frac{x}{A}\right) = -1$; $\left[\frac{x}{A}\right]_{\mathbb{Z}} = -1$ és $x^{\frac{A-1}{6}} \equiv \mu \ (mod\ A)$ egy olyan μ számra, amire μ nem kongruens egy primitív 6. egységgyökkel modulo A. Ennek ellenőrzésére az alábbi algoritmut használhatjuk:

A protokoll leírása és futásideje

Algoritmus:

Bemenet: $(n; m; k; x; \mu)$

Ellenőrizendő állítás: $x^{k \cdot 2^{n-1} \cdot 3^{m-1}} \equiv -\mu \ (mod\ A)$, ahol $A = k \cdot 2^n \cdot 3^m + 1$; $1 \not\equiv \mu \not\equiv \epsilon_3 \ (mod\ A)$; $\left(\frac{x}{A}\right) = -1$ és $\left[\frac{A}{x}\right]_{\mathbb{Z}} = -1$.

Ha $\mu^3 \equiv 1 \ (mod\ A)$, akkor az állítást elutasítjuk

Ha $\mu^3 \not\equiv 1 \ (mod\ A)$:

Számoljuk ki az $\left(\frac{A}{x}\right)$ Jacobi-szimbólum értékét!

Ha $\left(\frac{A}{x}\right) = 0$, akkor $gcd(A; x) \neq 1$, szóval A tényleg összetett, azaz az állítást *elfogadjuk*.

Ha $\left(\frac{A}{x}\right) = 1$, akkor az állítást *elutasítjuk*.

Ha $\left(\frac{A}{x}\right) = -1$, akkor:

Számoljuk ki az $\left[\frac{x}{A}\right]_{\mathbb{Z}}$ kubikus szimbólum értékét!

Ha $\left[\frac{x}{A}\right]_{\mathbb{Z}} = 1$, akkor az állítást *elutasítjuk*.

Ha $\left[\frac{x}{A}\right]_{\mathbb{Z}} = -1$, akkor:

Legyen $\mu_1 := \mu^k$.

Ha $\mu_1 = 1$, akkor:

Legyen $d := o_P(\mu)$ és $\alpha \equiv 2^{-n} \cdot 3^{-m} \ (mod\ d)$.

Ha $x^k \equiv \mu^{6\alpha} \ (mod\ d)$, akkor az állítást *elfogadjuk*, különben *elutasítjuk*.

Ha $\mu_1 \neq 1$, akkor:

Legyen $\mu_2 := \mu_1^{2^{\lambda \cdot log_2(n)} \cdot 3^{\lambda \cdot log_2(m)}}$

Ha $\mu_2 \neq 1$:

Végezzük el a $PPoE_2(x^k; -\mu; n-1; m-1; \mathbb{Z}_A)$ algoritmust. Ha ez elfogadja a megkapott állítást, akkor mi is *elfogadjuk* az általunk kapottat. Ha elutasítja, akkor mi is *elutasítjuk*.

Ha $\mu_2 = 1$:

Számoltassuk ki Botonddal $x^{k \cdot 2^{n-1-\lambda \cdot log_2(n)} \cdot 3^{m-1-\lambda \cdot log_2(m)}}$ értékét (ezt gyorsan meg tudja tenni, hiszen állítólag $x^{k \cdot 2^{n-1} \cdot 3^{m-1}}$ értékét már kiszámolta). A számolás során kapott erdményét jelöljük y-nal. Ezután végezzük el a $PPoE_2(x^k; y; n-1-\lambda \cdot log_2(n); m-1-\lambda \cdot log_2(m); \mathbb{Z}_A)$ algoritmust.

Ha $PPoE_2$ *elutasít*, vagy $y^{2^{\lambda \cdot log_2(n)} \cdot 3^{\lambda \cdot log_2(m)}} = -\mu$ nem áll fenn, akkor *elutasítjuk* a bizonyítékot, különben *elfogadjuk*.

4.3.1. Megjegyzés. *A protokoll futásideje a kubikus szimbólum kiszámolásán kívül $O(log(k)+\lambda\cdot(log(n)+log(m))$, ez 3.2.1-hez hasonlóan igazolható. A kubikus szimbólum kiszámolására majd később térünk vissza.*

A protokoll teljessége

4.3.2. Állítás. *Ha $(n;m;k;x;\mu)$ igaz, akkor a protokoll el fogja fogadni a bemenetet.*

Állítás 4.3.2 bizonyítása: A verifikáció elején a protokoll leellenőrzi, hogy μ tényleg nem egy harmadik egységgyök-e, ahogyan azt az egyik feltétel kéri. Ezt követően azt ellenőrzi, $\left(\frac{x}{A}\right)$ és $\left[\frac{x}{A}\right]$ tényleg mindketten -1-ek-e, ahogyan azt le kell, hogy ellenőrizzük (mivel $A \equiv 1 \ (mod\ 4)$, $\left(\frac{x}{A}\right) = \left(\frac{A}{x}\right)$). Ha a protokoll eljut odáig, hogy ezek mind teljesülnek, akkor márcsak az $x^{k\cdot 2^{n-1}\cdot 3^{m-1}} \equiv -\mu \ (mod\ A)$ állítás az, amivel probléma lehet.

1. eset: $\mu_1 = 1$:

Ebben az esetben a protokoll akkor fogadja el a bemenetet, ha $x^k = \mu^{6\alpha}$. Ha a bemenet igaz, akkor ezt átalakítva azt kapjuk, hogy:

$$\mu^{6\alpha} = \mu^{6\cdot 2^{-n}\cdot 3^{-m}} = (\mu^{2\cdot 3})^{2^{-n}\cdot 3^{-m}} = ((-\mu)^{2\cdot 3})^{2^{-n}\cdot 3^{-m}} = \left(x^{k\cdot 2^{n-1}\cdot 3^{m-1}}\right)^{2\cdot 2^{-n}\cdot 3^{-m}} = x^k. \tag{4.9}$$

Tehát, ha a bemenet igaz, akkor ebben az esetben tényleg el fogja fogadni azt a protokoll.

2. eset: $\mu_1 \neq 1 \neq \mu_2$:

Ilyenkor a protokoll elvégzi a $PPoE_2(x^k; -\mu; n-1; m-1; \mathbb{Z}_p$ algoritmust.

Amit $PPoE_2$ ezáltal ellenőriz az az, hogy $\left(x^k\right)^{2^{n-1}\cdot 3^{m-1}} = y$ fennáll-e, ami pont az az állítás, amit ellenőrizni akarunk és mivel $PPoE_2$ mindig elfogadja az igaz állítást, a protokoll is el fogja fogadni ebben a lépésben.

3. eset: $\mu_1 \neq 1 = \mu_2$:

Ilyenkor $\left(x^k\right)^{2^{n-1-\lambda\cdot log_2(n)}\cdot 3^{m-1-\lambda\cdot log_2(m)}} = x^{k\cdot 2^{n-1-\lambda\cdot log_2(n)}\cdot 3^{m-1-\lambda\cdot log_2(m)}} = y$. Amit ilyenkor leellenőrzünk az az, hogy $y^{2^{\lambda\cdot log_2(n)}\cdot 3^{\lambda\cdot log_2(n)}} = -\mu$ fennáll-e. Ha ez az állítás igaz volt, akkor:

$$\begin{aligned} -\mu = y^{2^{\lambda\cdot log_2(n)}\cdot 3^{\lambda\cdot log_2(m)}} &= \left(x^{k\cdot 2^{n-1-\lambda\cdot log_2(n)}\cdot 3^{m-1-\lambda\cdot log_2(m)}}\right)^{2^{\lambda\cdot log_2(n)}\cdot 3^{\lambda\cdot log_2(m)}} = \\ &= x^{k\cdot 2^{n-1-\lambda\cdot log_2(n)+\lambda\cdot log_2(n)}\cdot 3^{m-1-\lambda\cdot log_2(m)+\lambda\cdot log_2(m)}} = x^{k\cdot 2^{n-1}\cdot 3^{m-1}}. \end{aligned} \tag{4.10}$$

Tehát a protokoll itt pontosan el fogja fogadni a bemenetet, ha az ellenőrizendő állítás igaz volt, mivel $PPoE_2$ is megteszi.

Tehát mindhárom esetben el volt fogadva az igaz bemenet. Ezzel az állítást bizonyítottuk. □

A protokoll megbízhatósága

4.3.3. Állítás. *Annak a valószínűsége, hogy a protokoll hibásan elfogadja a bemenetet (azaz* $x^{k\cdot 2^{n-1}\cdot 3^{m-1}} = -1$*, a protokoll viszont mégis elfogadja a bemenetet), legfeljebb* $2^{-\lambda+3} \cdot max(log_2(n); log_2(m))$.

Állítás 4.3.3 bizonyítása: Feltételezve, hogy Jacobi-szimbólumot és a kubikus szimbólumot is tudunk pontosan számolni, a protokoll elején nem hibázunk. A protokoll ezután háromféleképpen folytatódhat.

1. eset: $\mu_1 = 1$:

Ebben az esetben $o_{\mathbb{Z}_A}(\mu)$ rangja se 2-vel, se 3-mal nem osztható, hiszen k-ról feltettük, hogy ilyen. A bizonyítás további része 3.2.3 első esetéhez teljesen hasonló, csak 4.1.3-at használva 2.2.3 helyett.

2. eset: $\mu_1 \neq 1 \neq \mu_2$:

Mivel A prím, a Kis-Fermat-tételből $\mu^{k\cdot 2^n\cdot 3^m} \equiv 1\ (mod\ A)$, de $1 \not\equiv \mu_2 \equiv \mu_1^{2^{\lambda\cdot log_2(n)}\cdot 3^{\lambda\cdot log_2(m)}} \equiv$ $\equiv \mu^{k\cdot 2^{\lambda\cdot log_2(n)}\cdot 3^{\lambda\cdot log_2(m)}}$. Ezt a kettőt összerakva $2^{\lambda\cdot log_2(n)} | o_{\mathbb{Z}_A}(\mu)$, vagy $3^{\lambda\cdot log_2(m)} | o_{\mathbb{Z}_A}(\mu)$, azaz $o_{\mathbb{Z}_A}(\mu) \geq$ $min(2^{\lambda\cdot log_2(n)}; 3^{\lambda\cdot log_2(m)})$.

1. aleset: $2^{\lambda\cdot log_2(n)} | o_{\mathbb{Z}_A}(\mu)$ és $o_{\mathbb{Z}_A}(\mu)$ felbontásában a 2 nagyobb kitevőn van, mint a 3:

Ilyenkor 3.2.3-hoz hasonlóan bizonyítható, hogy ahhoz, hogy egy rossz elem rangja elérje a 3-at (azaz a protokoll hibásan elfogadja a bemenetet), $PPoE_2$ egy lépése során egy rossz elem rangjának átlagosan legalább $\sqrt[log_2(n)]{\frac{2^{\lambda\cdot log_2(n)}}{3}}$ részére kellene csökkenie. Speciálisan legalább egyszer ennyivel kellene csökkennie.

$$\sqrt[log_2(n)]{\frac{2^{\lambda\cdot log_2(n)}}{3}} \geq \sqrt[log_2(n)]{2^{\lambda\cdot log_2(n)-2}} \geq 2^{\lambda} - 1 \geq 2^{\lambda-1}, \tag{4.11}$$

tehát legalább 1 körben a rangnak legalább egy $2^{\lambda-1}$-es tényezővel kellene csökkennie. 4.2.6 alapján egy adott kör esetén ennek a valószínűsége legfeljebb $2^{-\lambda+3}$. Tehát $log_2(n)$ kör során legfeljebb $2^{-\lambda+3} \cdot log_2(n) \leq 2^{-\lambda+3} \cdot max(log_2(n); log_2(m))$.

2. aleset: $3^{\lambda\cdot log_2(n)} | o_{\mathbb{Z}_A}(\mu)$ és $o_{\mathbb{Z}_A}(\mu)$ felbontásában a 3 nagyobb kitevőn van, mint a 2:

Ez az előző alesethez teljesen hasonlóan indokolható, a végén azt kapjuk, hogy a becsülni kívánt valószínűség legfeljebb $3^{-\lambda+3} \cdot log_2(m)$, ami szintén kisebb, mint $2^{-\lambda+3} \cdot max(log_2(n); log_2(m))$.

3. eset: $\mu_1 \neq 1 = \mu_2$:

3.2.3-hoz nagyon hasonlóan bizonyítható, hogy ekkor csak akkor fogadhatja el az algoritmus hibásan a bemenetet, ha $(x^k; y; n - 1 - \lambda \cdot log_2(n); m - 1 - \lambda \cdot log_2(m); \mathbb{Z}_A)$ egy olyan α-ra $\alpha - hazug$, amire

$$\alpha^{2^{\lambda \cdot log_2(n)} \cdot 3^{\lambda \cdot log_2(m)}} = -\mu. \tag{4.12}$$

Továbbá általános $\mathbb{G}$ csoportban:

$$o_{\mathbb{G}}(\alpha^{2^i \cdot 3^j}) = \frac{o_{\mathbb{G}}(\alpha)}{gcd(2^i \cdot 3^j; o_{\mathbb{G}}(\alpha))} \tag{4.13}$$

Ebben az esetben $o_{\mathbb{Z}_A}(\mu)$ páros, vagy 3-mal osztható (hiszen különben $\mu_1 = 1$ már teljesült volna korábban), tehát 4.12 miatt $o_{\mathbb{Z}_A}(\alpha^{2^{\lambda \cdot log_2(n)} \cdot 3^{\lambda \cdot log_2(m)}})$ is osztható 2-vel, vagy 3-mal, azaz 4.13 miatt $2^{\lambda \cdot log_2(n)} | o_{\mathbb{Z}_A}(\alpha)$ és $3^{\lambda \cdot log_2(m)} | o_{\mathbb{Z}_A}(\alpha)$ közül legalább az egyik teljesül.

Ha $2^{\lambda \cdot log_2(n)} | o_{\mathbb{Z}_A}(\alpha)$, akkor 3.2.3 2. esetéhez hasonlóan bizonyítható, hogy a hibás bemenet elfogadásának valószínűsége legfeljebb $2^{-\lambda+2} \cdot log_2(n) < 2^{-\lambda+3} \cdot max(log_2(n); log_2(m))$; ha pedig $3^{\lambda \cdot log_2(m)} | o_{\mathbb{Z}_A}(\alpha)$, akkor legfeljebb $3^{-\lambda+2} \cdot log_2(m) < 2^{-\lambda+3} \cdot max(log_2(n); log_2(m))$.

Ezzel mindhárom esetre beláttuk, hogy legfeljebb $2^{-\lambda+3} \cdot max(log_2(n); log_2(m))$ a hamis bemenet elfogadásának valószínűsége, ezzel bizonyítva az állítást. □

4.4 Kubikus reciprocitás

Az előző szakaszban tárgyalt ellenőrző protokoll igénylni azt, hogy ki tudjuk számolni hatékonyan egy kubikus szimbólum értékét. Sajnos itt falba ütközünk, mivel erre általánosan nem ismert gyors módszer. A protokoll hatékonysága leginkább azon múlik, hogy tudunk-e olyan a kvadratikus nemmaradékokat választani, melyre $\left[\frac{a}{A}\right]_{\mathbb{Z}}$ értékét ki tudjuk számolni, akkor ha A egy prím (hiszen ha A nem egy prím, akkor semmilyen a-re nem fog teljesülni a 4.1.3-ban leírt feltétel, tehát összetett számok esetén nem számít, hogy jól számoljuk-e ki a kubikus szimbólum értékét). Ebben a szakaszban olyan tételeket gyűjtöttem össze, melyek segíthetnek abban, hogy ki tudjuk küszöbölni ezt a problémát, sajnos nekem nem sikerült erre megoldást találnom. Ezen tételek egy része igényli az Eisenstein-egészek ismeretét, szóval először ennek a számhalmaznak a számunkra fontosabb tulajdonságait gyűjtöttem össze.

Az Eisenstein-egészek

4.4.1. Definíció. *A* $\mathbb{Z}[\omega]$ *gyűrű eleimet* ***Eisenstein-egészeknek*** *nevezzük, ahol* $\omega = \frac{-1+i\sqrt{3}}{2}$, *azaz* $\omega^3 = 1$. *Innentől ezt a számhalmazt* $\mathbb{E}$*-vel fogjuk jelölni.*

4.4.2. Tétel. *(bizonyítás nélkül) Az* $\mathbb{E}$ *gyűrű Euklideszi.*

4.4.3. Megjegyzés. $\mathbb{E}$*-ben* 6 *egység van: Az* 1, a -1, ω, ω^2, $-\omega$ *és* $-\omega^2$. *Az* $\mathbb{E}$*-beli egységek halmazát innentől* $\mathbb{E}_\epsilon$*-nal fogjuk jelölni.*

4.4.4. Definíció. *Azon* $\mathbb{E}\backslash\mathbb{E}_\epsilon$*-beli* π *számokat, melyek nem állnak elő két* $\mathbb{E}\backslash\mathbb{E}_\epsilon$*-beli elem szorzataként* ***Eisenstein-prímeknek*** *nevezzük.*

4.4.5. Definíció. *Egy* $\alpha \in \mathbb{E}$*-beli szám* ***konjugáltja*** *az a* $\overline{\alpha}$*-vel jelölt szám, melyre* $Re(\alpha) = Re(\overline{\alpha})$ *és* $Im(\alpha) = -Im(\overline{\alpha})$

4.4.6. Megjegyzés. $\alpha \in \mathbb{E}$ *esetén, ha* $\alpha = a + b\omega$, *ahol* $a; b \in \mathbb{Z}$, *akkor* $\overline{\alpha} = a - b\omega$

4.4.7. Definíció. *Egy* $\alpha \in \mathbb{E}$ *szám* ***normájának*** *nevezzük* $\alpha \cdot \overline{\alpha}$ *értékét. Jele:* $N(\alpha)$.

4.4.8. Megjegyzés. $\alpha \in \mathbb{E}$ *esetén, ha* $\alpha = a - b\omega$, *ahol* $a; b \in \mathbb{Z}$, *akkor* $N(\alpha) = a^2 - ab + b^2$

4.4.9. Tétel. $\alpha; \beta \in \mathbb{E}$ *esetén* $N(\alpha\beta) = N(\alpha)N(\beta)$

4.4.10. Megjegyzés. *4.4.9 könnyen igazolható 4.4.8 segítségével, mindössze egy rövid számolást igényel.*

4.4.11. Tétel. *(bizonyítás nélkül)* $\mathbb{E}$*-ben pontosan azok a számok prímek, melyeket meg lehet úgy szorozni egy egységgel, ezzel* π*-t kapva, hogy az alábbi feltételek egyike teljesüljön rá:*

- π *egész és egy* $3k + 2$ *alakú prím* $\mathbb{Z}$*-n*
- $\pi \cdot \overline{\pi}$ *egész és* $3k + 1$ *alakú prím* $\mathbb{Z}$*-n*
- $\pi = i\sqrt{3}$

Tételek a kubikus reciprocitásról

4.4.12. Definíció. *Egy* π *Eisenstein-prím és egy* $\alpha \in \mathbb{E}$; *szám* $\left[\frac{\alpha}{\pi}\right]_{\mathbb{E}}$*-vel jelölt* ***kubikus szimbólum****ának nevezzük azt a* ϵ *egységet, melyre* $\epsilon \equiv \alpha^{\frac{N(\pi)-1}{3}} \pmod{\pi}$, *ha létezik ilyen* ϵ*; és* $\left[\frac{\alpha}{\pi}\right]_{\mathbb{E}} = 0$, *ha nem létezik ilyen* ϵ.

4.4.13. Tétel. *(bizonyítás nélkül)* $\left[\frac{\alpha}{\pi}\right]_{\mathbb{E}}$ *értéke pontosan akkor* 0*, ha* $\pi \mid \alpha$*.*

4.4.14. Tétel. *(bizonyítás nélkül)* $\left[\frac{\alpha}{\pi}\right]_{\mathbb{E}}$ *értéke pontosan akkor* 1*, ha létezik olyan* $\beta \in \mathbb{E}$*, amire* $\beta^3 \equiv \alpha \ (mod\ \pi)$

4.4.15. Megjegyzés. *4.4.13 és 4.4.14 is könnyen igazolható annak felhasználásával, hogy a Kis-Fermat-tétel Eisenstein-prímekre is teljesül, de ezekkel a bizonyításokkal nem fogunk foglalkozni.*

4.4.16. Tétel. *Legyenek* α *és* β *Eisenstein-egészek,* π *pedig egy Eisenstein-prím, ekkor* $\left[\frac{\alpha\beta}{\pi}\right]_{\mathbb{E}} = \left[\frac{\alpha}{\pi}\right]_{\mathbb{E}} \cdot \left[\frac{\beta}{\pi}\right]_{\mathbb{E}}$

Tétel 4.4.16 bizonyítása. $\left[\frac{\alpha\beta}{\pi}\right]_{\mathbb{E}} = (\alpha\beta)^{\frac{N(\pi)-1}{3}} \ (mod\ \pi) = \alpha^{\frac{N(\pi)-1}{3}} \ (mod\ \pi) \cdot \beta^{\frac{N(\pi)-1}{3}} \ (mod\ \pi) = \left[\frac{\alpha}{\pi}\right]_{\mathbb{E}} \cdot \left[\frac{\beta}{\pi}\right]_{\mathbb{E}}$ □

4.4.17. Tétel. *(bizonyítás nélkül)[10] Ha* π_1 *és* π_2 *Eisenstein-prímek, akkor* $\left[\frac{\pi_1}{\pi_2}\right]_{\mathbb{E}} = \left[\frac{\pi_2}{\pi_1}\right]_{\mathbb{E}}$

4.4.18. Lemma. *Legyen* $p \equiv 2 \ (mod\ 3)$ *egy prímszám és z egy egész szám. Ekkor* $\left[\frac{z}{p}\right]_{\mathbb{Z}} = 1$ *pontosan akkor, ha* $\left[\frac{z}{p}\right]_{\mathbb{E}} = 1$

Lemma 4.4.18 bizonyítása: 4.4.11 alapján p Eisenstein-prím is, továbbá 4.4.14 alapján a $\left[\frac{z}{p}\right]_{\mathbb{E}}$ kubikus szimbólumnak értéke pontosan akkor 1, ha p előállt úgy, mint egy α Eisenstein-egész köbe modulo p. Ekkor α-t valamelyik 3. egységgyökkel szorozva egy z egész számot kapunk és $a = \alpha^3 = \alpha^3 \cdot \epsilon_3^3 = (\alpha \cdot \epsilon_3)^3 = z^3$. Ami definíció szerint azt jelenti, hogy $\left[\frac{z}{p}\right]_{\mathbb{Z}} = 1$. □

4.4.19. Tétel. *Legyenek* p_1 *és* p_2 *egymástól különböző,* $3k + 2$ *alakú prímek. Ekkor* $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}} = \left[\frac{p_2}{p_1}\right]_{\mathbb{Z}}$.

Tétel 4.4.19 bizonyítása: 4.4.18 alapján $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}}$ pontosan akkor 1, ha $\left[\frac{p_1}{p_2}\right]_{\mathbb{E}}$ is, továbbá $\left[\frac{p_2}{p_1}\right]_{\mathbb{Z}}$ pontosan akkor 1, ha $\left[\frac{p_2}{p_1}\right]_{\mathbb{E}}$ is.

Ekkor 4.4.17-t használva azt kapjuk, hogy $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}}$ pontosan akkor 1, ha $\left[\frac{p_2}{p_1}\right]_{\mathbb{Z}}$ is. Mivel p_1 és p_2 relatív prímek, ezért, ha $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}} \neq 1$, akkor az értéke csak -1 lehet, hasonlóan $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}}$-nek is. Tehát akkor $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}}$ és $\left[\frac{p_2}{p_1}\right]_{\mathbb{Z}}$ akkor is egyenlőek, ha $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}} = 1$ és akkor is, ha $\left[\frac{p_1}{p_2}\right]_{\mathbb{Z}} \neq 1$. Ezzel bizonyítottuk a tételt. □

4.4.20. Tétel. *(bizonyítás nélkül)[10] Legyen* p *egy* $3k+1$ *alakú prímszám. Ekkor* $\left[\frac{2}{p}\right]_{\mathbb{Z}} = 1$ *pontosan akkor, ha* p *felírható* $L^2 + 27M^2$ *alakban, ahol* $C; D \in \mathbb{Z}$*. Továbbá, ha felírható, akkor ez a felírás előjelektől eltekintve egyértelmű.*

4.4.21. Tétel. *(bizonyítás nélkül)[10] Legyenek p és q 3-mal osztva 1 maradékot adó, egymástól különböző prímek. Továbbá legyen $4pq = L^2 + 27M^2$ (ez a felírás az előjelektől eltekintve egyértelmű). Ekkor* $\left[\frac{q}{p}\right]_{\mathbb{Z}} = \left[\frac{L}{p}\right]_{\mathbb{Z}} \cdot \left[\frac{L}{q}\right]_{\mathbb{Z}} \cdot \left[\frac{p}{q}\right]_{\mathbb{Z}}$.

4.4.22. Megjegyzés. *4.4.21-ben L és M kiszámolását az úgynevezett Cornacchia-algoritmussal meg tudjuk tenni. [12]*

4.4.19; 4.4.20; 4.4.21 és a Cornacchia-algoritmus segítségével talán található egy gyors algoritmus egy olyan a szám keresésére, hogy $\left[\frac{a}{A}\right]_{\mathbb{Z}}$ egyenlő legyen -1-gyel, ahol $A = k \cdot 2^n \cdot 3^m + 1; k < 2^n \cdot 3^m$ prím, ezzel a 4.3-as szakaszban található ellenőrző protokoll hatékonyan elvégezhetővé téve, de nekem sajnos nem sikerült kiküszöbölnöm ezt a problémát.

Megoldatlan problémák

Ebben a szakdolgozatban bemutattam, hogy hogyan lehet hatékonyan ellenőrizni speciális alakú számok összetettségét, ha valaki erről egy megfelelő bizonyítékot ad nekünk. Az, hogy hogyan lehet hatékony algoritmust készíteni egy Proth-szám prímségének eldöntésére, megoldatlan.

Az is megoldatlan általánosan, hogy mely speciális alakú számokra és hogyan lehet a Proth-számokra mutatott verifikációs algoritmust átalakítani, például lehet-e hasonló protokollt készíteni Mersenne-prímekre.

A 4.4 szakaszban megadott probléma, hogy hogyan lehet megfelelő x értéket találni ahhoz, hogy a 4.3-as szakaszban megadott verifikációs algoritmus során hatékonyan ki tudjuk számolni, $\left[\frac{x}{A}\right]_{\mathbb{Z}}$ értékét A prímsége esetén, szintén nem megoldott.

Irodalomjegyzék

[1] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697, 2016.

[2] Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, and Michael Riabzev. Scalable, transparent, and post-quantum secure computational integrity. *Cryptology ePrint Archive*, 2018.

[3] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. Verifiable delay functions. In *Annual international cryptology conference*, pages 757–788. Springer, 2018.

[4] Benedikt Bünz, Jonathan Bootle, Dan Boneh, Andrew Poelstra, Pieter Wuille, and Greg Maxwell. Bulletproofs: Short proofs for confidential transactions and more. In *2018 IEEE symposium on security and privacy (SP)*, pages 315–334. IEEE, 2018.

[5] Uriel Feige, Dror Lapidot, and Adi Shamir. Multiple noninteractive zero knowledge proofs under general assumptions. *SIAM Journal on computing*, 29(1):1–28, 1999.

[6] Oded Goldreich and Yair Oren. Definitions and properties of zero-knowledge proof systems. *Journal of Cryptology*, 7(1):1–32, 1994.

[7] Charlotte Hoffmann, Pavel Hubáček, Chethan Kamath, and Krzysztof Pietrzak. Certifying giant nonprimes. In *IACR International Conference on Public-Key Cryptography*, pages 530–553. Springer, 2023.

[8] Russell Impagliazzo and Moti Yung. Direct minimum-knowledge computations. In *Conference on the Theory and Application of Cryptographic Techniques*, pages 40–51. Springer, 1987.

[9] Stefan Lance. A survey of primality tests, 2014.

[10] Franz Lemmermeyer. Cubic reciprocity. In *Reciprocity Laws: From Euler to Eisenstein*, pages 209–233. Springer, 2000.

[11] Siqi Liu. Privacy protection revolution: Zero-knowledge proof. In *2022 International Conference on Data Analytics, Computing and Artificial Intelligence (ICDACAI)*, pages 394–397. IEEE, 2022.

[12] François Morain and Jean-Louis Nicolas. On cornacchia's algorithm for solving the diophantine equation u2+ dv2= m. *Projet*, 1000:a1, 1990.

[13] ØJ Rødseth. A note on primality tests for n= h· 2 n- 1. *BIT Numerical Mathematics*, 34(3):451–454, 1994.

[14] Alena Šolcová and Michał Křižek. Fermat and mersenne numbers in pepin's test1. *Demonstratio Mathematica*, 39(4):737–742, 2006.

[15] Antonio Villani. Zero-knowledge proofs and applications.

[16] John Voight. Perfect numbers: An elementary introduction. *University of California, Berkley*, 1998.

[17] Benjamin Wesolowski. Efficient verifiable delay functions. In *Advances in Cryptology–EUROCRYPT 2019: 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, Germany, May 19–23, 2019, Proceedings, Part III 38*, pages 379–407. Springer, 2019.

[18] Jason Wojciechowski. Mersenne primes, an introduction and overview, 2003.