Hogyan snapszerozzunk telefonon?

SZAKDOLGOZAT

Készítette: Bradák Tímea

Matematika BSc - tanári szakirány

Témavezető: dr. Szabó Csaba, egyetemi tanár
ELTE TTK, Algebra és Számelmélet Tanszék

Eötvös Loránd Tudományegyetem
Természettudományi Kar
Budapest, 2012

Tartalomjegyzék

0.1. Interaktív bizonyítás 2
0.1.1. Nullaismeretű bizonyítások 4
0.1.2. Alkalmazásai 5
0.2. Érmefeldobás telefonon 7
0.2.1. Használt számelméleti háttér 7
0.2.2. Érmefeldobás telefonon 12
0.2.3. Négyzetgyök ismeretének bizonyítása 15
0.2.4. Faktorizáció ismeretének bizonyítása 18
0.3. Hogyan snapszerozzunk telefonon 21
0.3.1. Kártyapakli kiosztása két személy között 22
0.3.2. Néhány gyanús kártya 24

0.1. Interaktív bizonyítás

A több résztvevő által végzett együttes számítás, az *interakció* központi szerepet játszik az informatika több területén. Az interaktív bizonyítások fogalma a 80-as évek közepén alakult ki, ez az informatika sok területén előrelépést eredményezett. Az *interaktív bizonyítás* számítástudományi fogalma lényegében egy időben két egymástól független kutatás eredményeként született meg. Az egyik kriptográfia hátterű: Goldwasser, Micali és Rackoff [8] a biztonságos információközlés egy problémáját vizsgálva eljutottak a nullaismeretű bizonyítás fogalmáig.
A másik megközelítés Babai Lászlótól származik, aki mátrixcsoportokkal kapcsolatos algoritmikus kérdések tanulmányozására dolgozta ki az Arthur- Merlin-játékokat[1]. A két munkából kibontakozott elmélet ma is egyike a legfontosabb számításelméleti területeknek. Kiemelkedő eredményükért Babai, Goldwasser, Micali és Rackoff elnyerték a számításelmélet legmagasabb szakmai kitüntetését, a Gödel-díjat.
Egy eldöntendő probléma tekinthető úgy is mint egy formális nyelv. A probléma példányait elkódoljuk a megfelelő ábécé feletti szavakban. Ezek után magát a problémát azonosítjuk azzal a formális nyelvvel, mely azokat a szavakat tartalmazza, melyek a probléma "igen" példányait kódolják, vagyis azokat a példányokat melyekre a problémát eldöntendő algoritmus "igen" választ ad.
Az így kapott formális nyelvet általában ugyanúgy nevezzük, mint magát a problémát (például L).
Az ***Arthur-Merlin-játékokban*** két szereplőnk van, Arthur király és Merlin, a varázsló, akik egymással kommunikálnak. Arthur türelmetlen uralkodó, csak polinom idejű működésre képes: egy x bemenettel maximum $|x|^c$ ideig tud foglalkozni. Ezzel szemben Merlinnek varázsereje van: nincs korlátja a számítási képességeinek. Bármely L nyelv és $x \in I^*$ szó esetén egyetlen lépésben el tudja dönteni, hogy $x \in L$ igaz-e. A két fél együttműködésének, interakciójának célja a következő: adott egy L nyelv és egy $x \in I^*$ szó; Arthur szeretne meggyőződni (bizonyosságot szerezni) arról,hogy $x \in L$ igaz-e. Ennek érdekében számításokat végezhet, és kérdéseket tehet fel Merlinnek. Merlin minden kérdésre válaszol. A kérdésekre kapott válaszok és saját számításai alapján dönt arról, hogy az $x \in L$ állítást igaznak tartja-e. A folyamat során feltesszük,hogy az igen döntést elfogadásnak, a nem döntést elutasításnak nevezzük. Feltesszük itt, hogy a helyes bizonyítást elfogadja. Az egész információcserére és a számításokra együttesen csak legfeljebb $|x|^c$ idő van, ahol $c > 0$ csak L-től függő állandó. A protokollal szemben két alapvető követelményünk van a teljesség, és a helyesség:

- Teljesség: Ha $x \in L$, akkor erről Merlin meg tudja győzni Arthurt, vagyis

végül Arthur elfogadja x-et.

- Helyesség: Ha $x \notin L$, akkor Merlin nem tudja meggyőzni Arthurt arról, hogy $x \in L$ (még akkor sem, ha csal). Ekkor Arthur elutasítja x-et.

Az interaktív bizonyítási rendszerben két játékos van.

- Hitelesítő személy H, aki egy állításról szeretné eldönteni, hogy igaz vagy hamis. (Bonyolultságelméleti definíciókkal egy adott L nyelv esetén egy x szóról akarja eldönteni $x \in L$ teljesül-e.) A hitelesítő polinomiális idejű véletlenített algoritmusokat használhat és kérdést tehet fel a bizonyítónak.

- Bizonyító B, válaszol a hitelesítőnek. A számítási képességeinek nincs korlátja, bármit ki tud számolni. De lehet rosszindulatú bizonyító is, aki hazudik a kérdésekre, vagy a bizonyításban, és a csalást H nem tudja észrevenni polinomiális időben (mert pl. exponenciális idejű a bizonyító bizonyítása, és a hitelesítő csak polinomiális ideig tud számolni).

Az ***interaktív bizonyítási rendszerek*** az Arthur−Merlin−játékok változatai. Jelentéktelen különbség a terminológiában az, hogy Merlin a ˝bizonyító˝, Arthur pedig a ˝hitelesítő˝, és a kommunikáció nem játékként zajlik, hanem protokoll formájában. Első pillantásra a két modell között az a jelentős különbség, hogy Artúr véletlen bitje nyilvánosan − és elsősorban Merlin számára − ismert, ezzel ellentétben a hitelesítő véletlen bitje az interaktív bizonyítási rendszerekben titkos. Goldwasser és Sipser megmutatták azonban, hogy valójában lényegtelen az, hogy a véletlen bit titkos, vagy nyilvános [7]. Tehát az Arthur−Merlin−játékok és az interaktív bizonyítási rendszerek egymással ekvivalensek.
Az L eldöntendő kérdés, az IP bonyolultságelméleti osztályba tartozik (interaktív polinom idejű osztály), ha $\exists H$ hitelesítő és B bizonyító, úgy hogy ha:

- igaz a válasz ($x \in L$), akkor $Pr(V(x, B) = elfogad) = 1$,

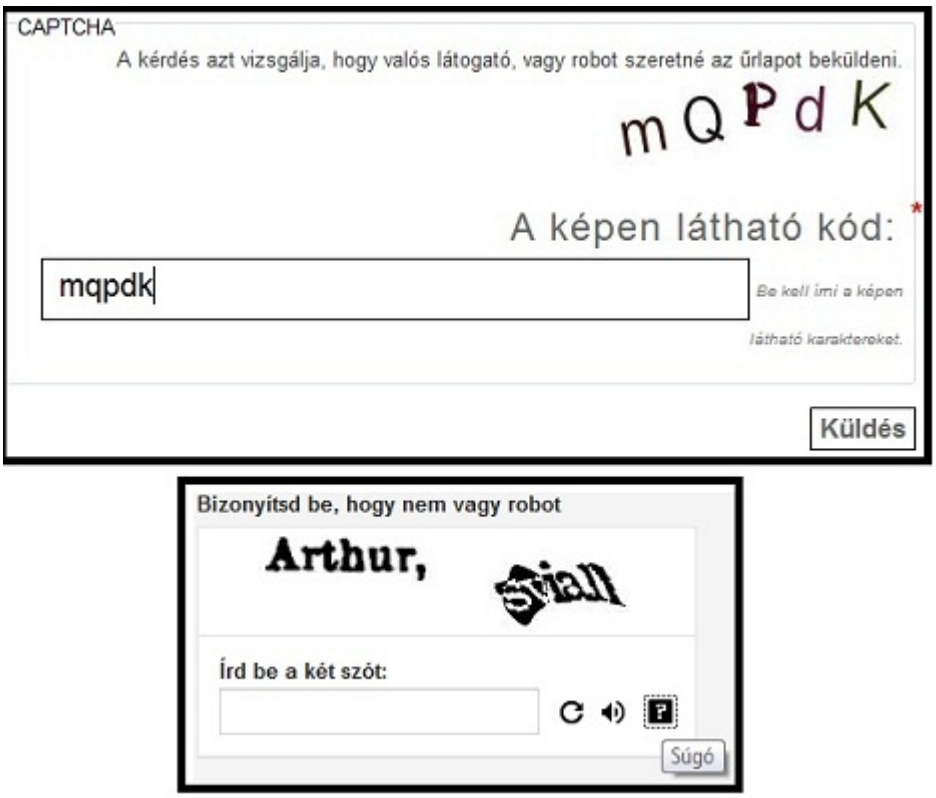
- ha nem a válasz ($x \notin L$), akkor $\forall P'$ esetleg rosszindulatú bizonyítóra is:$Pr(V(x, B') = elfogad) \leq \frac{1}{2}$. Ekkor létezik *Interaktív bizonyítási rendszer* egy eldöntendő L kérdésre. (ahol Pr(A): Az A esemény valószínűsége.)

0.1.1. Nullaismeretű bizonyítások

"Hozzon ajándékot is, mégpedig úgy, hogy mégse hozzon,
amikor pedig belép, köszönjön is, ne is!"
Mátyás király és a székely ember lánya
Illyés Gyula: Hetvenhét magyar népmese

A napjainkban alkalmazott azonosítási rendszerek (pl. jelszó vagy kérdés-válasz alapú) általában kiszivárogtatnak valamely részinformációt a titkos információról, melyet az identitását bizonyító fél felhasznál. Például, a digitális aláíráson alapuló azonosítási rendszerek kiszivárogtatják a kihívásként küldött üzenet digitális aláírását. Így az aláírt kihívást a támadó felhasználhatja. Sikeres támadást lehet megvalósítani abban az esetben, ha az identitását bizonyító fél azt a titkos aláíró kulcsot használja azonosításra, mint amit az üzenetek digitális aláírására, akkor az ellenőrző fél kihívásként valamely dokumentum hash értékét is küldheti. Így megszerzi egy általa létrehozott üzenet aláírását. Ha azt szeretnénk, hogy az azonosítási rendszer semmilyen részinformációt ne adjon ki, akkor nulla-ismeretű azonosítási rendszert kell használnunk. Leegyszerűsítve az ilyen rendszer azon kívül, hogy bizonyítja valamely állítás helyességét, más információt nem ad ki. Az ellenőrző fél azon kívül, hogy bizonyítékot kap arról, hogy az adott állítás helyes, más információhoz nem jut. Ez azt is jelenti, hogy minden, ami polinomiális időn belül kiszámítható a nulla-ismeretű bizonyítás üzeneteiből, az polinomiális időn belül kiszámítható az érvényes állításból magából is. Egy protokoll nulla-ismeretű tulajdonsága tehát azt jelenti, hogy bármit, amit az ellenőrző fél "lát" a bizonyító féllel való interakciója során, mindazt polinomiális időn belül tudja szimulálni maga is a bizonyító fél részvétele nélkül. Azt fontos megjegyezni, hogy a nulla-ismeretű tulajdonságnak akkor is teljesülnie kell, ha az ellenőrző fél nem követi a protokoll előírt lépéseit, hanem eltér tőle.

0.1.2. Alkalmazásai



Alan Turing angol matematikus, a számítógép tudomány egyik megalkotója, szerette volna megoldani azt a problémát, hogy hogyan különböztessünk meg egy fejlett számítógépet egy embertől. A feladat úgy néz ki, hogy az egyik szobában leültetünk egy embert, aki nem tudja ki ül a másik szobában. A másik szobába pedig vagy egy számítógépet rakunk, vagy egy embert. Az első ember, kérdéseket tehet fel, vagy bármi más módon társaloghat a másik szobában lévő lénnyel, és el kell döntenie, kivel ül szemben. Ezt nevezzük *Turing-tesztnek*. Ez az interaktív bizonyítás egyik formája, hiszen két fél között zajlik a kommunikáció, és az egyik be akarja bizonyítani, hogy Ő ember.

Ennek egy hétköznapi alkalmazása a Captcha, amit sokan névről nem ismernek, és ha használják is, nem értik miért. A Captcha-val szinte minden nap találkozunk, amikor például új e-mail címet szeretnénk csinálni, vagy éppen egy űrlapot töltünk ki, mint azt a képeken is láthatjuk. Ekkor egy, a képeken látható kéréssel találkozhatunk: " Kérjük, gépelje be a képen látható betűket. " A képen zavaró hátérrel eltorzított betűk és számok láthatók, ami emberi szemmel olvasható, de egy számítógép, a betűk torzulása miatt (ma még) nem képes kiolvasni.

Ha valaki nem érti, mire való ez, rákattinthat a "?" ikonra , és megtudja, hogy azért van erre szükség, mert sok program van, ami automatizálja a címek létrehozását, hogy aztán hirdetéseket küldjön szét róluk, vagy ami rosszabb, levelek

tömeges küldésével megbénítson valakit. A képen az emberi szem könnyedén felismeri a betűket, de egy számítógépet (ma még legalábbis) a betűk torzulásai és a kusza vonalak megtévesztenek, így a programozott címgenerálást ki lehet szűrni.
A mai számítógépek még megbuknak a Turing-teszten.
A hagyományos bizonyításoknál egy tételhez hozzá lehet csatolni a bizonyítást, az interaktív bizonyításnál azonban van egy hitelesítő, aki egy vagy több kérdést tesz fel, és a " bizonyítás " a kérdésekre feltett válaszoktól függ. Ez a séma sokkal erősebb a hagyományos bizonyítási formáknál. Nehéz lenne kérdés nélkül bebizonyítani, hogy mi emberek vagyunk. Azt is láthatjuk, hogy ennek a bizonyításnak a kulcsa az, hogy mindig új random karaktersorozatot kapunk, hiszen ha mindig ugyanazt kérdeznék, könnyű lenne beprogramozni a gépet, hogy jó választ adjon. Ezért szükség van a véletlen-generátorra ebben a bizonyításban.
Az interaktív bizonyítás egy másik példája a következő. Tegyük fel, hogy Péter és Kati interneten sakkoznak. Beesteledik, és abba kell hagyniuk a játékot. Ha valamelyik fél lép utoljára a másik fél egész este gondolkodhat a következő lépésén, ami nem igazságos a másik féllel szemben. Hagyományos sakkversenyen ilyenkor az történik, hogy mondjuk, Péter eldönti a következő lépését, de nem lépi meg, hanem borítékolja, és odaadja a bírónak. A borítékot Kati másnap nyithatja ki, így egyiküknek sincs előnye. De most nincs bíró, és nincs boríték sem. Péter üzenhet valamit este Katinak, amiből Kati csak reggel tudja meg Péter következő lépését. Ha már egyből elmondaná a következő lépését, akkor Kati reggelig gondolkodhatna, de mindenképpen el kell döntenie mit lép különben Ő lenne előnyben. Előre megállapodnak abban, hogy a lépéseket egy-egy négyjegyű számmal írják le: például Kf3 helyett azt mondják, hogy 9083.Ezek után Péter választ magának két 100 jegyű prímszámot, amelyek közül a kisebbik 9083... számjegyekkel kezdődik. (Számítógéppel lehet ellenőrizni, hogy egy szám prímszám-e, és be lehet bizonyítani, hogy néhány 100 százjegyű szám között, lesz prím). Legyen ez a két prímszám p, és q. Ezáltal Péter eldönti a következő lépését, és elküldi Katinak a pq szorzatot. Kati megkapja a pq szorzatot, ami körülbelül egy 200 számjegyű összetett szám, aminek a faktorizálása nem kis időbe telne, így biztos nem tudja meg másnap reggelig, hogy mi ez a lépés. Így egyikük sem kerül előnybe. Reggel pedig Péter megmondja a két prímszámot, amelyek közül a kisebbik határozza meg Péter következő lépését. Ezzel megoldottuk a borítékolás problémáját.

0.2. Érmefeldobás telefonon

0.2.1. Használt számelméleti háttér

1. Definíció. *Az R gyűrű invertálható elemei csoportot alkotnak az R-beli szorzásra, melynek egységeleme a gyűrű egységelemével egyenlő. Ez az R multiplikatív csoportja, jele* R^x.

2. Definíció. *Legyen n pozitív egész. Egy n-hez relatív prím szám akkor kvadratikus maradék mod n, ha egy alkalmas másik szám négyzetével kongruens mod n.*

3. Definíció (additív inverz). *Legyen egy szám x. Ennek additív inverze y, ha* $x + y = 0$.

1. Tétel. *Ha* (m_1,m_2)=1, *akkor az*

$$\begin{aligned} x &\equiv c_1 \pmod{m_1} \\ x &\equiv c_2 \pmod{m_2} \end{aligned}$$

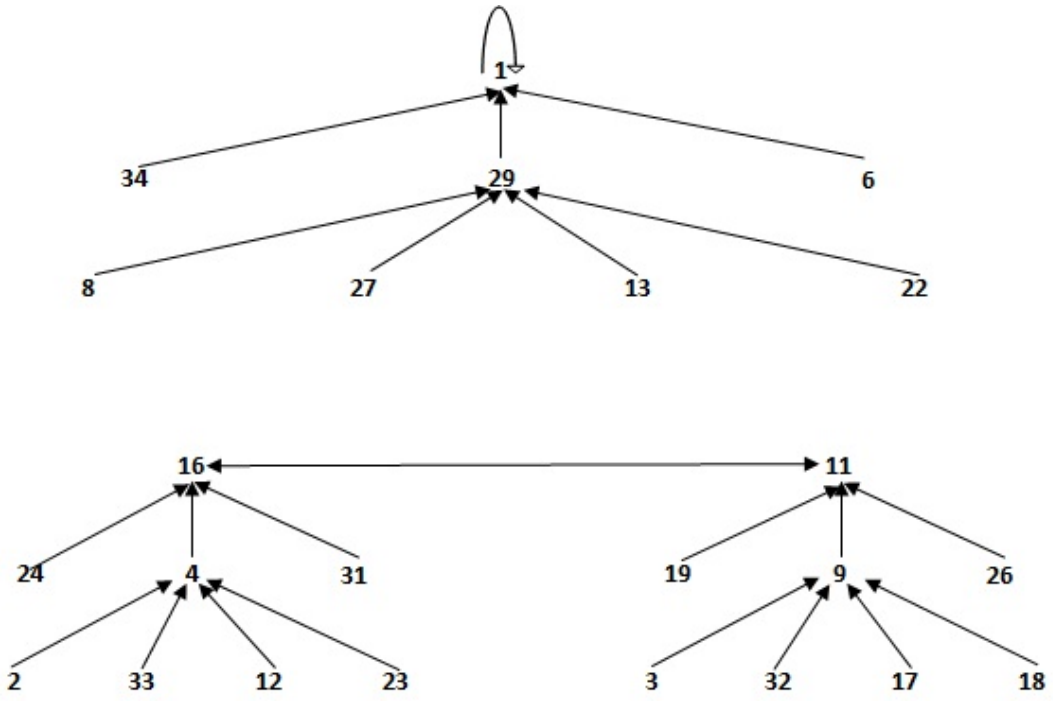
szimultán kongruenciarendszer bármilyen c_1 *és* c_2 *egész szám esetén megoldható és a megoldások egyetlen maradékosztályt alkotnak modulo* $m_1 m_2$.

2. Tétel (Kínai maradéktétel). *Legyenek az* $m_1, m_2, \ldots, m_k$ *modulusok páronként relatív prímek. Ekkor az*

$$\begin{aligned} x &\equiv c_1 \pmod{m_1} \\ x &\equiv c_2 \pmod{m_2} \\ &\vdots \\ x &\equiv c_k \pmod{m_k} \end{aligned}$$

szimultán kongruencia rendszer bármilyen $c_1, c_2, \ldots, c_k$ *egészek esetén megoldható, és a megoldások egyetlen maradékosztályt alkotnak modulo* $m_1 m_2 \cdots m_k$.

Bizonyítás: k=2 esetén az előző tétel.
Tegyük fel hogy $k-1$ kongruenciából álló rendszerekre az állítás igaz, és tekintsük a k kongruenciából álló rendszert. Itt az első $k-1$ kongruenciát kielégítő egész számok az indukciós feltevés szerint egyetlen maradékosztályt alkotnak modulo $m_1 m_2 \cdots m_{k-1}$ vagyis ahol c alkalmas egész szám. Így a k kongruenciából álló



1. ábra.

kongruencia rendszer ekvivalens az

$$\begin{aligned} x &\equiv c \pmod{m_1 m_2 \cdots m_{k-1}} \\ x &\equiv c_k \pmod{m_k} \end{aligned}$$

szimultán kongruencia rendszerrel. Az előző tételt alkalmazva erre a kongruencia rendszerre éppen k-ra vonatkozó állítást kapjuk. ■

Az *1. ábra* tanulmányozása után, azt láthatjuk, hogy a kvadratikus maradékok modulo 35 az $1, 4, 9, 11, 16$, és a 29. Azt is észrevehetjük, hogy minden kvadratikus maradéknak 4 gyöke van modulo 35. A 4 gyök közül pedig kettő-kettő additív inverzek. Ez abból adódik, hogy 35-nek két páratlan prímtényezője van.

Következőkben bizonyítjuk, hogy ha egy n páratlan egész számnak k különböző prímtényezője van, akkor minden $c \in Z_n^*$ kvadratikus maradéknak, 2^k gyöke van modulo n. Legyen $n = p_1 p_1 \cdots p_k$, ahol $p_1, p_2, \ldots, p_k$ k különböző prímszám, és tegyük fel hogy az $x^2 \equiv c \pmod{n}$ $(0 < c < n)$ kongruenciának x_i egy megoldása. Ennek a kongruenciának modulo n 2^k megoldása van, azaz c kvadratikus maradéknak 2^k különböző négyzetgyöke van, ha c és n relatív prímek.

Mivel $x^2 \equiv c \pmod{n}$ megoldható, ezért minden p_i-re $x^2 \equiv c \pmod{p_i}$ is. A két megoldás legyen a b_i és a $-b_i$.

Tekintsük a következő kongruenciarendszert:

$$\begin{aligned} x &\equiv b_1 \pmod{p_1} \\ x &\equiv b_2 \pmod{p_2} \\ &\vdots \\ x &\equiv b_k \pmod{p_k} \end{aligned}$$

$$\begin{aligned} x &\equiv -b_1 \pmod{p_1} \\ x &\equiv b_2 \pmod{p_2} \\ &\vdots \\ x &\equiv b_k \pmod{p_k} \end{aligned}$$

$$\vdots$$

$$\begin{aligned} x &\equiv -b_1 \pmod{p_1} \\ x &\equiv -b_2 \pmod{p_2} \\ &\vdots \\ x &\equiv -b_k \pmod{p_k} \end{aligned}$$

. Minden gyöknél két előjel közül választhatunk, vagy "+"-t vagy "-"-t. Ezt 2^k féle képen tehetjük meg. A kínai maradéktételből következik, hogy mindegyik kongruencia rendszernek van megoldása. Így 2^k különböző gyöke lesz minden kvadratikus maradéknak. Ezért a $x^2 \equiv c \pmod{n}$,$((x,n) = 1)$ kongruenciának pontosan 2^k megoldása van.
Ha ismerjük n prímfaktorizációját, akkor ezzel a módszerrel polinom időben megvizsgálhatjuk egy számról, hogy az gyök-e, és polinom időben tudunk keresni is egy gyököt.
A következő három algoritmust használjuk fel a feladatunk megoldásához.:
Következőkben csak az $n = p_1 p_2 \cdots p_k$ számokat tekintjük, ahol minden p_i különböző.

Kvadratikus maradék: Bemenetként kap két számot n-et, és $x \in Z_n^*$-ot. Kimenetként eldönti, hogy x kvadratikus maradék Z_n^*-ban, vagy sem.

Négyzetgyökvonás: Bemenetként megkapja n egész számot, és x kvadratikus maradékot modulo n. Kiszámol egy y-t melynek négyzete x, az algoritmus kimenete pedig ez az y gyök lesz.

Faktorizáló: Bemenetként megkap egy n egész számot. Kiszámolja n prímfelbontását. Ez a prímfelbontás lesz a kimenet.

Ezekre az algoritmusokra úgy tekintünk, hogy létezik egy olyan gép, amely polinom időben kiszámolja. Következőkben bebizonyítjuk, hogy a négyzetgyökvonás és a faktorizáló algoritmus számítási igénye egyenértékű.
Legyen n egy összetett szám, melynek a prímfelbontását $p_1p_2...p_k$-t ki tudjuk számolni a faktorizációs algoritmussal.
A prímfelbontás segítségével megkereshetjük bármely x kvadratikus maradék egy gyökét. Prím modulus esetében tudunk gyököt számolni polinom időben, nem a gyökvonás algoritmussal, melyből már megtudjuk a kvadratikus maradék gyökét modulo n. Minden $1 < i < k$-ra kiszámoljuk a kvadratikus maradék gyökét modulo p_i. Tehát, ha ismerjük a prímfelbontást, akkor a kínai-maradéktétel segítségével tudunk gyököt is számolni.
Számunkra fontosabb lesz az, hogy ha bármely kvadratikus maradék gyökét ki tudjuk számolni a négyzetgyökvonás algoritmussal, akkor n prímfelbontását is fel tudjuk írni.
Szemléltető célból $k = 2$ esetet vizsgáljuk. Az általánosítás, hogy tetszőleges n-re is igaz, hasonlóan bizonyítható.
Azt állítjuk, hogy ha egy kvadratikus maradék gyökei közül kiválasztunk kettőt, úgy hogy azok nem additív inverzei egymásnak, akkor ezek segítségével kiszámolható n egy prímtényezője.
Tekintsük az x kvadratikus maradékot. Válasszuk ki két gyökét s-et, és t-t úgy, hogy s ne legyen additív invereze t-nek. Ekkor $g = lnko(s+t, n)$ osztója lesz n-nek. Ehhez azt kell belátnunk, hogy $g \neq 1$, és $g \neq n$.
Az $x \equiv s^2 - t^2 \equiv (s+t)(s-t) \equiv 0 \pmod{n}$, ezért az n osztja az $(s+t)(s-t)$-t. A g csak úgy lehet egyenlő 1-gyel, ha n osztja $s-t$-t. Ha n osztja $s-t$-t, akkor $s-t \equiv 0 \pmod{n}$, ami ellentmond annak a feltevésünknek, hogy s nem additív inverze t-nek. Így beláttuk hogy $g \neq 1$. A $g = n$, pedig akkor lenne igaz, ha $s + t \equiv 0 \pmod{n}$. Ez azonban ismét ellentmond a feltevésünknek, és ezzel beláttuk, hogy $g = n$ sem igaz.
Illusztrálásként az *1. ábráról* a 4 kvadratikus maradéknak gyökei a $2, 33, 12$, és a 23. A 4 gyök közül válasszunk ki kettőt, melyek nem additív inverzei egymásnak.

Legyenek ezek a 33, és a 23. Az előzőek miatt $lnko(33 + 23, 35) = 7$ a 35-nek egy megfelelő tényezője. Hasonlóképpen, ha a 29 kvadratikus maradék 4 gyöke a $8, 27, 13, 22$ közül választunk kettőt, akkor a $lnko(27 + 13, 35) = 5$ másik prímtényezőt kapjuk.

Így ha tetszőleges kvadratikus maradéknak ki tudjuk számolni az összes gyökét, akkor az előzőek alapján a prímtényezőket is kiszámolhatjuk. Kvadratikus maradékot úgy tudunk keresni, hogy választunk egy $t \in Z_n^*$ számot, amit négyzetre emelünk. Ennek a kvadratikus maradéknak pedig, ki tudjuk számolni egy gyökét, s-et, a gyökvonás algoritmussal. Ha s additív inverze t-nek, akkor választunk egy másik t-t, de ha nem, akkor $lnko(s + t, n)$ kiszámolásával megkapjuk n egy prímtényezőjét. Addig választunk újabb t-ket, amíg meg nem kapjuk n összes prímtényezőjét. Ezzel n felbontható prímtényezők szorzatára a gyökvonás segítségével.

Ha ismert egy kvadratikus maradék egy gyöke, akkor annak segítségével még nem lehet faktorizálni.

0.2.2. Érmefeldobás telefonon

A telefonbeszélgetés közben történő érmefeldobás nem igazságos, hiszen aki az érmefeldobást végzi, az bármit mondhat annak eredményéről. Olyan játékra lenne szükségünk, az érmefeldobás helyett, melynek bármelyik kimenetele $\frac{1}{2}$ valószínűséggel következik be. Ha valamelyik fél csal, akkor annak ki kell derülnie a játék során.

Rabin és Blum erre fejlesztett ki egy protokollt[4]. A protokollt a *2. ábrán* láthatjuk, és ezt mindkét félnek be kell tartania. Ha valamelyik nem tartja be és ez kiderül, akkor a másik automatikusan nyer. Ha mindkét fél követi a protokollt, akkor pontosan $\frac{1}{2}$ a valószínűsége annak, hogy valamelyik nyer.

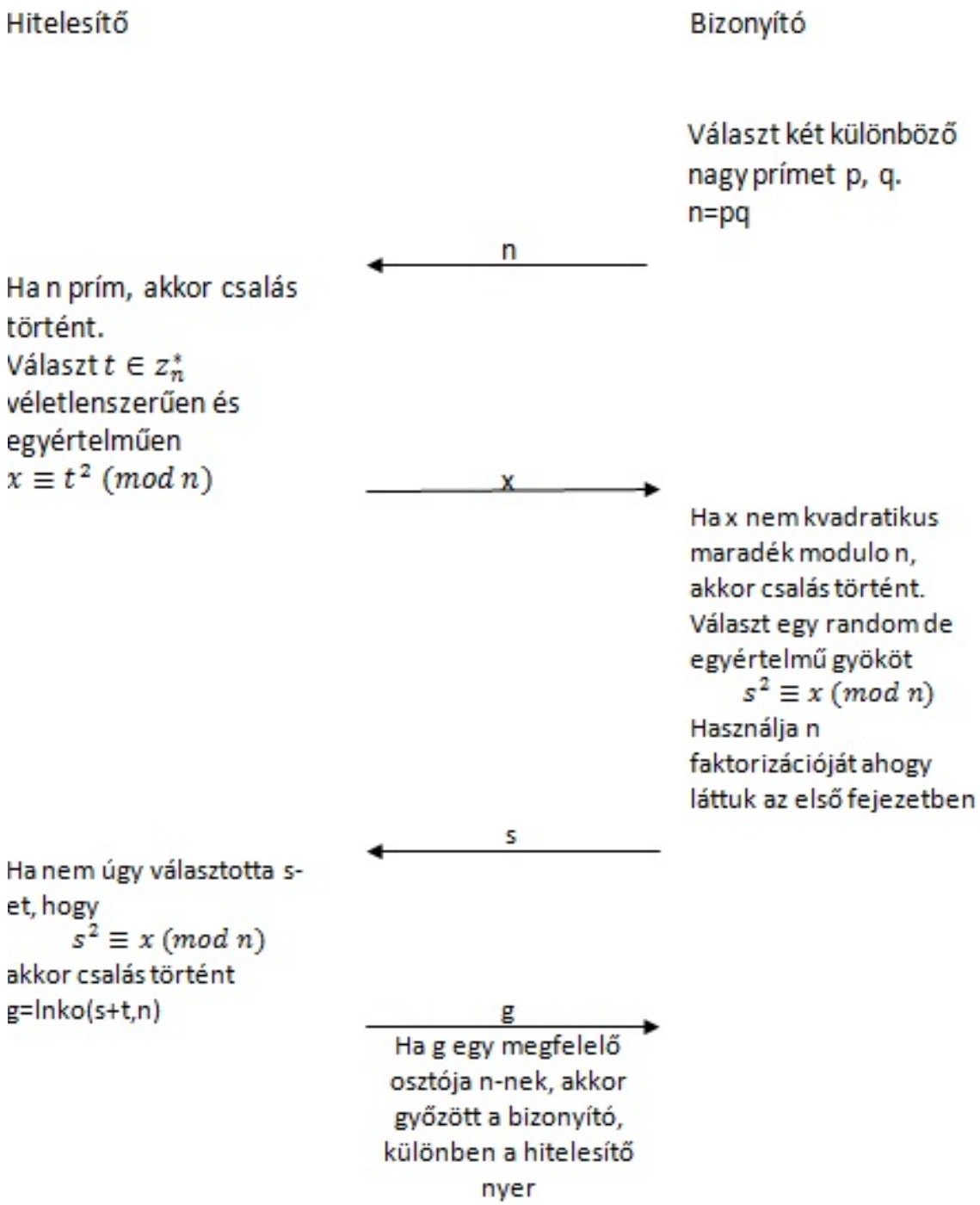
A bizonyító választ két elég nagy prímszámot p-t és q-t, melyek szorzatát n-nel jelöljük. Ezt az n számot elmondja a hitelesítőnek. A hitelesítő választ egy Z_n^*-beli elemet, t-t. A t négyzetre emelésével kiszámol egy kvadratikus maradékot x-et. A kvadratikus maradékot elmondja az bizonyítónak. Az n szám $k = 2$ prímszám szorzata, így minden kvadratikus maradéknak $2^k = 4$ különböző gyöke van. Mivel a bizonyító ismeri n prímfelbontását, így a már látott módon, ki tudja számolni az x kvadratikus maradék többi gyökét is. A bizonyító kiválasztja x egy gyökét s-et, és ezt elmondja a hitelesítőnek. Az s vagy egyenlő t vagy additív inverze t-nek, vagy egyik sem. Ha s nem additív inverze t-nek, és nem t, akkor a $g = (s + t, n)$ egy megfelelő tényezője lesz n-nek, ezt beláttuk az előző fejezetben. Ezzel a hitelesítő ki tudja számolni n egy prímtényezőjét, és megnyeri a játékot. Ha azonban a bizonyító a t gyököt választja, vagy annak additív inverzét, akkor ebből a hitelesítő nem tud meg semmilyen új információt. Ekkor a hitelesítő nem tudja kiszámolni a prímtényezőket, és a bizonyító nyeri a játékot. Ha a bizonyító $s \equiv \pm t \pmod{n}$ közül választ akkor nyer, ha pedig a másik kettő közül, akkor elveszti a játékot.

A protokollt mindkét félnek követnie kell. Ha valamelyik nem követi, akkor az egyből kiderül és elveszti a játékot. A bizonyító nem tud észrevétlenül csalni, mert t annyira titkos, hogy ez megakadályozza a bizonyítót abban, hogy s-t úgy válassza, meg hogy $s \equiv \pm t$ legyen.

Ha a bizonyító olyan n-et küld a hitelesítőnek, aminek 2-nél több prímtényezője van, akkor csökken annak a valószínűsége, hogy s-et számára kedvezően választja meg. Ha a prímtényezők száma k, akkor ez a valószínűség $\frac{1}{2}$ helyett 2^{-k+1}. Ha n-et prímszámnak választja meg a bizonyító, akkor az már a protokoll során kiderül, amikor a hitelesítő leellenőrzi, hogy n valóban összetett szám-e.

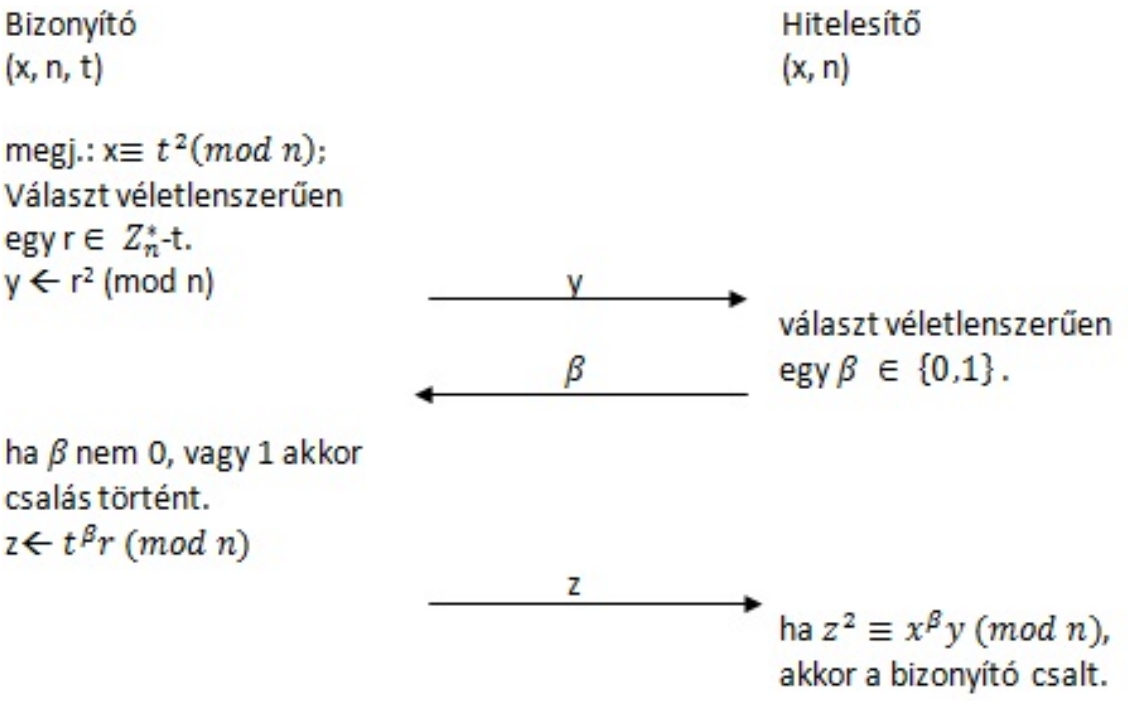
Hasonlóan beláthatjuk, hogy a hitelesítő sem tud csalni. Kevés az esélye annak, hogy a másik segítsége nélkül ismerné n prímfelbontását.

A hitelesítőnek csak egy gyök ismerete lehetne segítség a prímfaktorizációban, más


Hitelesítő
Bizonyító
Választ két különböző nagy prímet p, q.
n=pq
n
Ha n prím, akkor csalás történt.
Választ $t \in z_n^*$ véletlenszerűen és egyértelműen
$x \equiv t^2 \ (mod\ n)$
x
Ha x nem kvadratikus maradék modulo n, akkor csalás történt.
Választ egy random de egyértelmű gyököt
$s^2 \equiv x \ (mod\ n)$
Használja n faktorizációját ahogy láttuk az első fejezetben
s
Ha nem úgy választotta s-et, hogy
$s^2 \equiv x \ (mod\ n)$
akkor csalás történt
g=lnko(s+t,n)
g
Ha g egy megfelelő osztója n-nek, akkor győzött a bizonyító, különben a hitelesítő nyer

2. ábra.

információt nem kap a bizonyítótól. Ha a bizonyító, csak olyan gyököt árul el, amire az $s \equiv \pm t \pmod{n}$ kongruencia igaz, az nem segítség a hitelesítőnek, hiszen t additív inverzét ki tudja számolni magától is.
Fischer rámutatott egy hiányosságra ebben az érvelésben[5]. Azt mutatta meg, hogy ha a hitelesítő nem ismeri n prímfelbontását, és nem ismeri t értéket sem, akkor egy s gyök segítség lehet számára a prímfaktorizációban.
Feltesszük, hogy létezik olyan x kvadratikus maradék, melynek gyöke, s, segítségével könnyen fakotrizálható n. Így a hitelesítő kereshet egy ilyen x kvadratikus maradékot, amit elmond a bizonyítónak. Ennek gyökét a bizonyító kiszámolja, és elárulja a hitelesítőnek, aki ennek segítségével felbonthatja n-et prímtényezők szorzatára. Ezáltal a hitelesítő tud csalni.
Olyan gyököt, amely segítségével tényleg felbontható az n, még nem találtak. Ez azonban nem azt jelenti, hogy nincs ilyen. Senki nem bizonyította még, hogy nem létezik ilyen gyök.
Ennek hiánya miatt nem bizonyítható a *2. ábrán* szereplő protokoll igazságossága, azonban ez nem azt jelenti, hogy nem lehet tisztességesen érmefeldobást játszani telefonon. A megoldást Goldwasser, Micali és Rackoff dolgozta ki, a *"zero-knowledge"* interaktív bizonyítást[3]. Ennek az ötletnek a felhasználásával kerüljük ki Fischer kifogását az alábbiak szerint.
A protokollban az a hiba, hogy a bizonyító anélkül, hogy tudná, hogy a hitelesítő tényleg ismeri t értékét, elárulja x egy gyökét s-t. Emiatt a bizonyítónak nem kellene ilyen könnyen elárulni x egy gyökét, mert ez segítség lehet a hitelesítőnek.
A hitelesítő azonban nem szeretné felfedni t értékét, mert a bizonyító ekkor úgy választaná meg s-t, hogy az $s \equiv \pm t \pmod{n}$ legyen. Így a bizonyítónak van lehetősége csalni.
A hibák kikerülésére dolgozott ki egy protokollt Goldwasser, Micali és Rackoff, úgy hogy kiegészítették a *2. ábrán* látottakat [3].

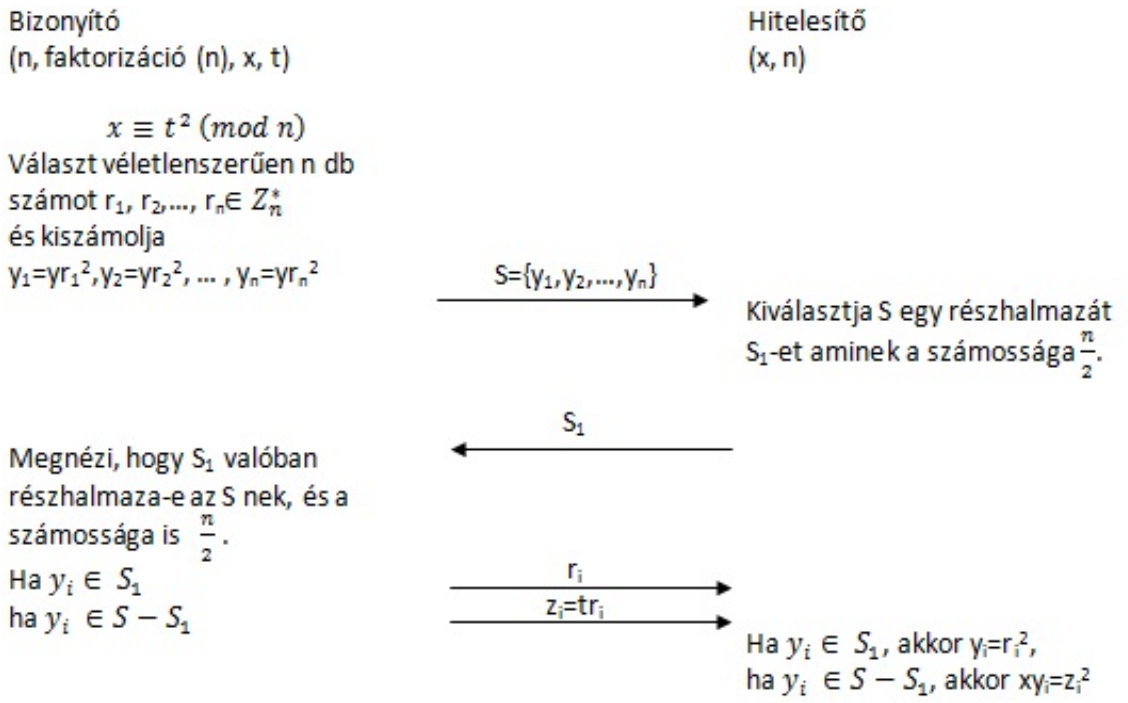


3. ábra.

0.2.3. Négyzetgyök ismeretének bizonyítása

Az előző fejezetben beláttuk, hogy a bizonyítónak bármely kvadratikus maradék gyökét ismerni kell mod n, mert ismeri a prímfelbontását n-nek. A *3. ábrán* szereplő protokoll segítségével a hitelesítő meggyőződhet arról, hogy a bizonyító valóban ismeri-e x kvadratikus maradék gyökét,t-t. Ezt a protokollt Tompa és Woll dolgozta ki [6].

Ha a bizonyító esetleg nem ismeri x egy gyökét, vagy elhajlik a protokolltól, akkor ezt a hitelesítő $\frac{1}{2}$ valószínűséggel veszi észre. Mivel elég nagy a valószínűsége annak, hogy a bizonyító észrevétlenül csal, ezért érdemes megismételni a protokollt, hogy csökkenjen ez a valószínűség. Ha a protokollt τ-szor megismételjük, akkor $\frac{1}{2}$-ről $2^{-\tau}$-ra csökken a valószínűsége annak, hogy a bizonyító észrevétlenül tud csalni. A *3. ábrán* lévő protokollal a bizonyító igazolhatja, hogy ismeri t-t anélkül, hogy elárulna az értékét. A bizonyító random választ egy $r \in Z_n^*$ számot. Az r négyzetre emelésével kiszámol egy kvadratikus maradékot, y-t. Ezt az y kvadratikus maradékot átküldi a hitelesítőnek. Ezt követően a hitelesítő $\beta = 0$, és $\beta = 1$ közül választhat. Ha $\beta = 0$-t választja, akkor a bizonyító $z = r$ számot küldi vissza. Ezután a hitelesítő a $z^2 \equiv x^0 y \pmod{n}$ kongruenciával ellenőrizheti, hogy a bizonyító tényleg nem csalt. A $z = r$ szám tényleg kielégíti a kongruenciát hiszen.:



4. ábra.

$$\begin{aligned} r^2 &\equiv x^0 y \pmod{n} \\ r^2 &\equiv y \pmod{n}. \end{aligned}$$

Ha a hitelesítő azonban a $\beta = 1$-et választja, akkor a bizonyító a $z = tr$ számot fogja vissza küldeni. A hitelesítő ezt is a $z^2 \equiv x^1 y \pmod{n}$ kongruenciával ellenőrizheti. A $z = tr$ szám is kielégíti a kongruenciát, hiszen:

$$\begin{aligned} (tr)^2 &\equiv xy \pmod{n} \\ t^2 r^2 &\equiv xy \pmod{n} \\ xy &\equiv xy \pmod{n} \end{aligned}$$

A bizonyító ismeri r és tr értékét, azaz ismeri az $r^{-1}tr = t$ értékét is. Ezzel igazoltuk, hogy a bizonyító ismeri t értékét.

A bizonyító mindig vagy r, vagy tr érékét, árulja el a hitelesítőnek. Ha a hitelesítő is ismerné mindkettő értékét, akkor ő is simerné t értékét. Ennek a kiküszöbölésére létezik egy másik *subprotokoll* is. Ezt a *4.ábrán* láthatjuk. A *4. ábrán* szereplő protokollban kevesebb az esélye a bizonyítónak, hogy észrevétlenül csaljon.

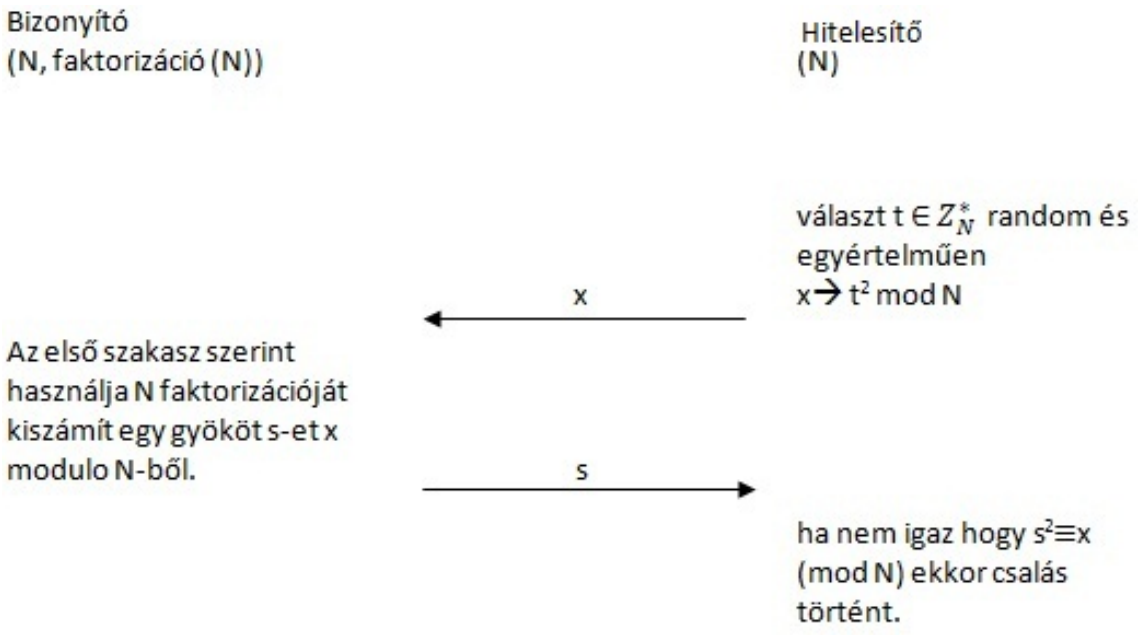
A bizonyító random választ n db $r_i \in Z_n^*$ számot. Az n db szám legyen az $r_1, r_2, \ldots, r_n$. A bizonyító az r_i számokat négyzetre emelve n db kvadratikus maradékot számol ki. Jelöljük őket $y_1, y_2, \ldots, y_n$-nel. A bizonyító átküldi a hitelesítőnek a kiszámolt kvadratikus maradékokat, az $S_1 = \{y_1, y_2, \ldots, r_n\}$ halmazt. A

hitelesítő ezután kiválaszt a kvadratikus maradékok közül $\frac{n}{2}$ darabot, és ezt átküldi a bizonyítónak. A bizonyító a megkapott kvadratikus maradék gyökeit átküldi a hitelesítőnek, a többi kvadratikus maradékot pedig megszorozza x-szel, és a szorzatnak a gyökét küldi át a hitelesítőnek. A hitelesítő leellenőrzi, hogy a megfelelő gyököket kapta-e meg.

A protokoll során a hitelesítő nem tudja meg, hogy a bizonyító x melyik gyökét ismeri, hiszen az r_i random választás volt így tr_i is random gyöke lesz xy_i-nek. Belátjuk, hogy a bizonyító nem tudja meggyőzni a hitelesítőt, hogy ismeri a gyököt, még akkor sem, ha csal.

Tegyük fel, hogy a bizonyító nem ismeri a gyököt és random választ y_i-ket. A bizonyító el tudja küldeni y_i gyökeit, és néhány i-re pedig el tudja küldeni xy_i gyökét is. Ha a bizonyító ki tudja számolni egy y_i-re mind a kettőt, akkor megtalálta x-nek egy gyökét, a $\frac{z_i}{r_i} = t$-t.

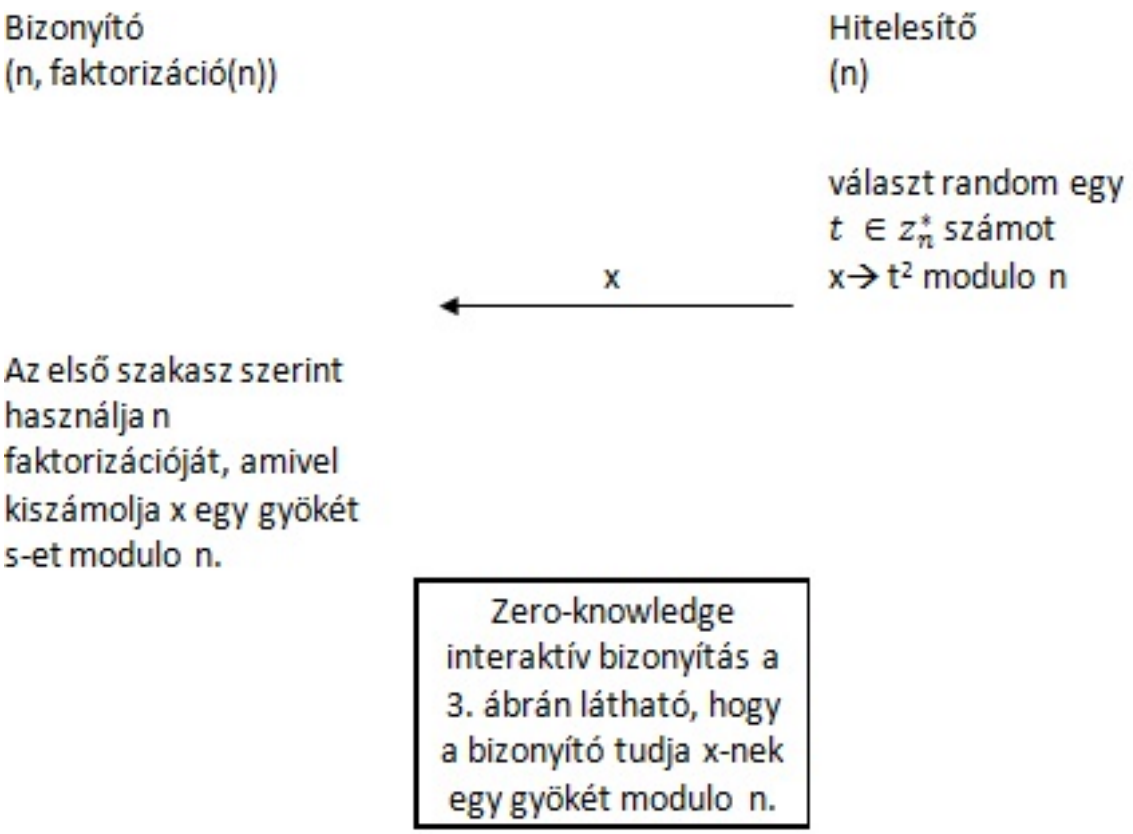
Ha a bizonyító egyik i-re sem tudta kiszámolni mind a kettőt, akkor egyetlen választása van S_1-ből ami lehetővé teszi, hogy a hitelesítőt meggyőzze. Ebben az esetben, annak a valószínűsége, hogy a bizonyító meggyőzi a hitelesítőt kisebb, mint $\frac{1}{\binom{n}{\frac{n}{2}}} \leq \frac{1}{n^k}$ minden k-ra.



5. ábra.

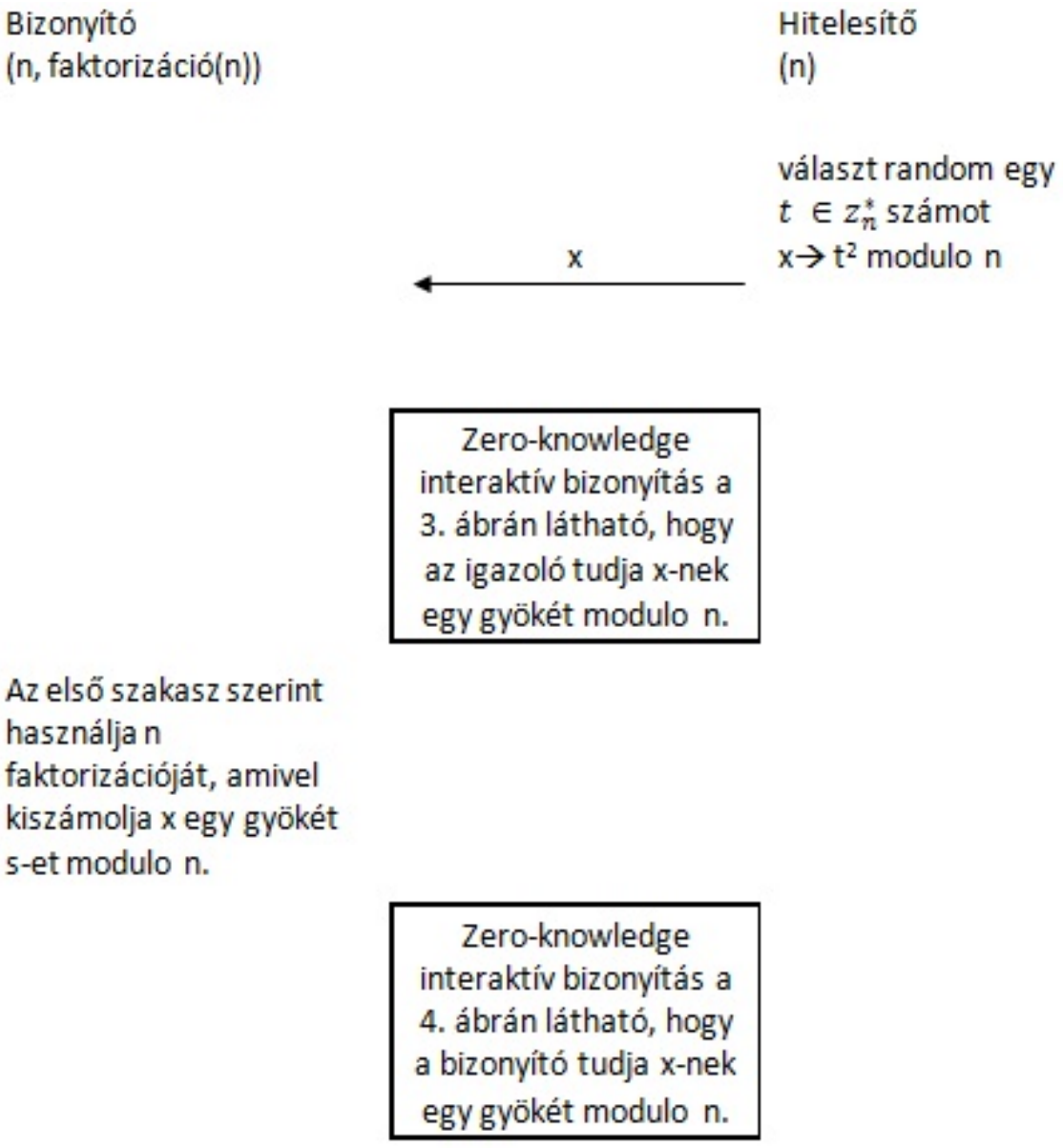
0.2.4. Faktorizáció ismeretének bizonyítása

Ebben a fejezetben olyan protokollt keresünk, amely segítségével bizonyítható a faktorizáció ismerete, nulla-ismeretű interaktív bizonyítással. A faktorizáció ismeretének bizonyítására Tompa és Woll dolgoztak ki egy protokollt [6]
Az első részben elegendő volt, hogy a bizonyító megmutatta, hogy bármely kvadratikus maradék gyökét mod n ki tudja számolni. Az első protokollt a *6. ábrán* láthatjuk.
A hitelesítő választ random egy t számot, melyet négyzetre emel, és a kvadratikus maradékot, x-et, elküldi a bizonyítónak. A bizonyító kiszámolja egy gyökét s-et az x kvadratikus maradéknak. Ezt a gyököt azonban nem árulja el a hitelesítőnek, csak a *3. ábrán* látott módon meggyőzi a hitelesítőt, hogy ismeri s értékét. Mint a négyzetgyök ismereténél itt is növelhetjük a hitelesítő bizalmát, a protokoll megismétlésével. Előfordulhat, hogy a hitelesítő csal, és random választ egy x számot, amit elküld a bizonyítónak. Ekkor, ha a bizonyító nem mondja, hogy csalás történt, akkor a hitelesítő tudni fogja, hogy x kvadratikus maradék.
Így olyan információt árul el a bizonyító a hitelesítőnek, amit magától nem tudott volna kiszámolni, ezért ez nem nulla-ismeretű bizonyítás.
Ezt a hibát a *7. ábrán* lévő protokoll már kijavítja. A *7. ábrán* látott protokoll során a hitelesítő random választ egy t értéket, és ezt négyzetre emeli. Az így kapott x kvadratikus maradékot elmondja a bizonyítónak. Ezután a hitelesítőnek bizonyítania kell, hogy ismeri t értékét is a *3. ábrán* szereplő protokoll alapján. Miután a bizonyító megkapta x-t, szintén a *3.ábrán* látottak alapján bebizonyítja, hogy ki



6. ábra.

tudja számolni x bármely gyökét. Ezzel pedig a bizonyító meggyőzi a hitelesítőt, hogy ismeri a prímfelbontását n-nek.


Bizonyító
(n, faktorizáció(n))
Hitelesítő
(n)
választ random egy
$t \in z_n^*$ számot
x→ t² modulo n
x
Zero-knowledge interaktív bizonyítás a 3. ábrán látható, hogy az igazoló tudja x-nek egy gyökét modulo n.
Az első szakasz szerint használja n faktorizációját, amivel kiszámolja x egy gyökét s-et modulo n.
Zero-knowledge interaktív bizonyítás a 4. ábrán látható, hogy a bizonyító tudja x-nek egy gyökét modulo n.

7. ábra.

0.3. Hogyan snapszerozzunk telefonon

A snapszer, más néven snapszli vagy hatvanhat egy kártyajáték. Magyar kártyával játszhatja 2-4 játékos. A játék célja: legalább 66 pont (innen a hatvanhat név) elérése bemondásokból és ütésekből.

Ha négyen játszanak, a 7-es, és 8-as lapokat kiveszik a pakliból. A 9-es 0 pontot ér. Két részletben 6 (3-3) lapot osztanak, talon nincs. Az, aki először kap lapot, első három lapja után megnevez egy lapot; ennek színe az adu, és az a partnere, akinek a hívott lap a kezében van. (Meghívhatja önmagát is, azt remélve, hogy megkapja a lapot a második osztásfordulóban, vagy az addig kiosztott lapok egyikével.) Az, hogy kik a partnerek, nem közölhető.

A snapszer kézből vagy az első kijátszás után is bemondható. A hívó indul. A 66 elérésétől függetlenül addig kell játszani, amíg a meghívott lap ki nem jön. Ha a hívó saját ütéseivel és bemondásaival elér 66-ot anélkül, hogy a meghívott lap kijött volna, egyedül számolhat, partnere nem kap pontot. Ha a lap már kijött, az ellenpár is befejezheti 66-tal. Mivel talon nincs, színre szín vagy adu és felülütés kötelező. A partnerek "együtt sírnak, együtt nevetnek", azaz együtt számolják a pontjaikat, de együtt is veszíthetnek.

Ha valaki elérte a 12 pontot, a többi játékos a hármas játék szabályai szerint játszhat tovább; ezután a kettes szerint.

Szabályok 3 játékos esetén a játékmenet eltérései a négyes játékhoz képest: két részletben 8 (4-4) lapot osztanak, a hívó első négy lapjából csak színt hív, ez az adu. A másik kettő ellen a hívó egyedül játszik.

Ha ketten játszanak, ki kell venni a 7, 8, 9-es lapokat. Az osztó 3-3 lapot ad, egyet felüt, ennek színe az adu, majd még 2-2 lapot oszt. A maradék lapokat lefelé fordítva ("csukva") a felütött adura helyezi, úgy hogy az adu kilátszik Húsz és negyven. Az egy kézben lévő azonos színű király és felső 20-at, ha az adu színéből van, 40 pontot ér - de a végelszámolásnál csak akkor számít be a pont, ha ütések is vannak mellette. Ha kettő is van kézben, csak egy mondható be. A bemondott húsz vagy negyven után ki kell hívni a királyt vagy a felsőt. Összesen elvileg 100 pontot lehet bemondani.

Ahhoz hogy ezt a kártyajátékot telefonon tudjuk játszani, először is azzal a problémával kell megküzdenünk, hogy hogyan lehet kiosztani a kártyákat a játékosok között úgy, hogy senki ne tudja mit kapott a másik.

0.3.1. Kártyapakli kiosztása két személy között

András és Peti egy pakli kártyából szeretnének 5-5 lapot osztani maguknak. Az osztást úgy kell megvalósítani, hogy mindkét fél tudja a saját lapjait, de csak a saját lapjait a másikét nem.
A szemléletesség kedvéért képzeljük el azt, hogy a lapok kódolása a következőképpen valósul meg: Mind a 32 lapot külön-külön dobozokba helyezzük el. Ezekre a dobozokra olyan lakatokat teszünk, melyeket csak az tud kinyitni aki rárakta. Tehát kódok esetében a kódot csak a tulajdonos fejtheti meg. Egy dobozon több lakat is lehet, ez azt jelenti, hogy egy kódolt üzenetet tovább kódolhatunk.
Ezek alapján az osztás így fog kinézni. András a 32 lapot beteszi véletlen sorrendben 32 dobozba és lelakatolja őket. Majd átadja a dobozokat Petinek, vagyis elküldi a kódolt lapokat. Peti kiválaszt tetszőlegesen 5 dobozt, amelyeket lelakatol saját lakatjaival, és visszaküldi Andrásnak. András leveszi a dobozokról a saját lakatjait, és újra átadja a dobozokat Petinek. Így Peti megkapja a saját 5 dobozát, amin már csak a saját lakatjai vannak, vagyis a saját kódolása, amit már meg tud fejteni és tudja, milyen lapok kerülnek hozzá.
András hasonlóan kapja meg lapjait. A megmaradt lelakatolt dobozok közül még választ 5-t Peti és elküldi azokat Andrásnak. A dobozokon csak András lakatjai vannak, leveszi a lakatokat a dobozról és megtudja a lapjait.
A lap kiosztásában szereplő lakatok, illetve kódok alkalmazása azon múlik, hogy használatuk sorrendje felcserélhető. Ez azt jelenti, hogy a lakatokat különböző sorrendben vehetjük le, mint ahogy felkerültek.
Az előző játékban szereplő dobozok helyett most megszámozzuk a kártyákat, a lakatokat pedig úgy valósítjuk meg, hogy kódoljuk a számokat. A lakatokat tetszőleges sorrendben vehetjük le a dobozokról így a kódolásnak is olyannak kell lennie, amit tetszőleges sorrendben dekódolhatunk. Ehhez egy kommutatív kódolásra van szükségünk.
A kódolásokat egyértelműen kell meghatározni, és az üzeneteknek melyeket a játék során küldtek egymásnak, összhangban kell lenni a kódolással. Különben egy két lapot kicserélhetnének a játékosok olyanra, ami számukra esetleg kedvezőbb. Fontos még, hogy olyan kódolásra van szükségünk, amely semmilyen információt nem árul el a lapokról, (akár a színét sem) mert kis információ is nagy előnyt jelent a játékban.
Shamir, Rivest és Adleman javasoltak egy kommutatív kódolást, az RSA-t [9]. András és Peti választ két nagyon nagy prímszámot p-t és q-t, ezeknek a szorzata legyen n. A r és q segítségével kiszámolható a $\varphi(n) = (r-1)(q-1)$. Péter és András választanak maguknak egy titkos kulcsot $A = a$ és $P = p$, úgy hogy

$lnko(a, \varphi(n)) = lnko(p, \varphi(n)) = 1$. Az a és p számoknak kiszámítják az inverzeiket modulo $\varphi(n)$, úgy hogy $aa_i \equiv 1 \pmod{\varphi(n)}$ és $pp_i \equiv 1 \pmod{\varphi(n)}$. Ekkor a kódolás úgy fog kinézni, hogy $E_{a_i}(x) \equiv x^a \pmod{n}$ és $D_{a_i}(x) \equiv x^{a_i} \pmod{n}$. Peti kódolása ugyanígy.
A dekódolás pedig a következők miatt működik.

$$x \equiv E_{a_i}(x)^{a_i} \pmod{n}$$

akkor

$$E_{a_i}(x)^{a_i} \equiv (x^a)^{a_i} \equiv x^{aa_i} \pmod{n}$$

Mivel

$$\begin{aligned} aa_i &\equiv 1 \pmod{r-1}, \\ aa_i &\equiv 1 \pmod{q-1} \end{aligned}$$

a Kis-Fermat tétel kimondja, hogy

$$\begin{aligned} x^{aa_i} &\equiv x \pmod{r}, \\ x^{aa_i} &\equiv x \pmod{q}. \end{aligned}$$

Mivel p és q különböző prímszámok, alkalmazva a Kínai maradéktételt ezekre a kongruenciákra

$$x^{aa_i} \equiv x \pmod{rq}.$$

Így
q

$$E_{a_i}(x)^{a_i} \equiv x \pmod{n}$$

András kódolása és dekódolása legyen $E_A = E_{a_i}$ és $D_A = D_{a_i}$, Petié pedig $E_P = E_{p_i}$ és $D_P = D_{p_i}$. Minden x számra legyen igaz az, hogy $E_A(D_P(x)) = D_P(E_A(x))$ és $E_P(D_A(x) = D_A(E_P(x)$.
A kártyákat jelöljük az $1, \ldots, 32$ számokkal. A kártyakiosztás a következő módon történik.
András kódolja a saját kódjaival a lapokat $E_A(1), E_A(2), \ldots, E_A(32)$, és a kódolt lapokat random sorrendben pedig átküldi Petinek. Peti választ tetszőleges 5 kártyát $E_A(2), E_A(15), E_A(24), E_A(10), E_A(30)$, ezeket visszaküldi Andrásnak, így András már ismeri a lapjait.
Peti ismét választ random 5 kártyalapot $E_A(7), E_A(13), E_A(27), E_A(31), E_A(9)$. Ezeket a lapokat tovább kódolja a saját kódjával, így
$E_P(E_A(7)), E_P(E_A(13)), E_P(E_A(27)), E_P(E_A(31)), E_P(E_A(9))$. A már duplán kódolt lapokat visszaküldi Andrásnak. András dekódolja a saját kódolását a lapokról. Mivel $D_a(E_P(x)) = E_P(D_a(x)$, ezért $D_a(E_P(E_a(x))) = E_P(D_a(E_a(x))) = E_P(x)$, így már csak Peti titkosítása kódolja a lapokat, amit dekódol és megtudja a lapjait. A játék után András és Peti elárulják egymásnak a titkosításukat, ezáltal ellenőrizhetik egymást, hogy a másik megfelelő kártyákkal játszott-e. (előfordulhat, hogy olyan kártyákat mondanak, ami éppen nekik megfelel, ez addig működik, amíg egy kártyát nem próbál mindkét játékos használni)

0.3.2. Néhány gyanús kártya

Miután az előző protokoll megjelent egy műszaki cikkben, Lipton észrevette, hogy ez a protokoll még mindig elárul egy kis információt a lapokról [10].
A kódulásunk lényege a hatványozás. Tudjuk, hogy minden k-ra x^k akkor és csak akkor kvadratikus maradék, ha x is az. Emiatt, ha van egy kártyánk amit egy számmal jelölünk, és ez kvadratikus maradék, akkor ha ezt elkódoljuk, továbbra is kvadratikus maradék lesz. Ezért a kódolás után kapott számokból eldönthetjük melyek kvadratikus maradékok, és így megtudhatjuk mely ezek a kártyák. Így néhány kártyáról tudunk némi információt.
Ha p nagy szám, akkor a p-nél kisebb számok fele kvadratikus maradék. Néhány p-re nehéz kiszámolni, hogy mely számok lesznek kvadratikus maradékok.Természetesen tudják úgy kódolni a kártyákat, hogy minden kártya kvadratikus maradék legyen, vagy akár úgy is, hogy egyik sem. Ezt a problémát kiküszöbölve sem biztos, hogy nincs más olyan tulajdonság, amit megtart a kódolás. 3 évvel később Lipton javaslatára Micali és Goldwasser kidolgozott egy új protokollt a játékra [11]. A fontos eredménye a munkájuknak az, hogy a másik fél kártyáinak kitalálása

egyenértékűen más nehéz probléma megoldásával.(faktorizáció, vagy egy számról eldönteni, hogy kvadratikus maradék-e összetett modulusra). A protokollt két játékosra dolgozták ki. András összekeveri a kártyákat és kódolja azokat A-val, és elküldi Petinek. Peti szintén kódolja a saját kódjával a lapokat P-vel, majd elküldi Andrásnak.

Micali és Goldwasser megmutatja, hogy Peti, hogy tudja megkérni Andrást hogy, dekódolja a lapokat úgy, hogy ne ismerje, mely lapokról van szó. A protokoll, mely tartalmazza hogyan kell 1 lapot megfelelően kiosztani, meghaladja dolgozatunk terjedelmét.

Ez a technika a protokolljuk kulcsa. Lényegében Peti minden kódolt kártyáról kérdez, de csak annyi információt kap, hogy azt az egy kártyát dekódolja.(Ez egy későbbi probléma, hogy Peti, ne kapjon olyan információt, amivel több kártyát is tud dekódolni.) Ha már tudjuk, hogyan osszunk ki egy lapot, akkor a következő módon oszthatják ki a kártyákat András és Peti.

András választ egy számot, vagyis egy kódolt lapot Peti kódolt lapjai közül például $E_P(3)$, amiről nem tudja pontosan melyik kártya. Ezt a lapot ő tovább kódolja a saját kódjával, majd $E_A(E_P(3))$-t elküldi Petinek. Peti dekódolja a titkosítását $D_P(E_A(E_P(3)))$, így megkapja az $E_A(3)$-t. Erről Peti nem tudja milyen lap, hiszen kódolva van. Ezután $E_A(3)$-t visszaküldi Andrásnak. András így már dekódolhatja a kártyát és megtudja milyen lapot húzott. Ezt a lapot kiveszi a kódolt kártyák listájáról. Ezután Peti fog egy lapot választani András kódolt lapjai közül. Ezt tovább kódolja, majd visszaküldi Andrásnak. András dekódolja, majd a dekódolt lapot visszaküldi Petinek, aki így megtudhatja milyen lapot húzott. Peti szintén kiveszi a húzott lapot a kódolt kártyáinak listájáról. Arra, hogy kivegyük a már kihúzott kártyákat, természetesen azért van szükség, hogy a másik ne húzhassa ki ugyanazt a lapot, mint amit már egyszer kihúztak. E miatt nem működik ez a protokoll több személyre.

Ha egy harmadik személy is játszana, Timi, nem tudnánk eldönteni, kitől húzza a lapokat. Ha Andrástól húzza, lehet Peti lapjai közül választana egyet, és hasonlóan fordítva is, ha Peti lapjai közül húz, lehet András lapjait választja. Így ez a protokoll csak két személyre működik.

Létezik olyan protokoll, amely három, négy, vagy több személy között is lehetővé teszi a kártyakiosztást. Készítettek olyan protokollt, amelyben a játékosok egymás között osztják ki a lapokat. Ebben az esetben, ha két játékos összefog, előfordulhat, hogy sikerül csalniuk. Van olyan protokoll is, ahol egy osztó osztja ki a kártyákat úgy, hogy nem tudja milyen kártyákat oszt ki, és a megmaradt lapok pedig az osztóé lesznek. Ebben az esetben az osztónak van lehetősége csalni.

Irodalomjegyzék

[1] Babai László, "Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity class", *Journal of Computer and System Sciences, Volume 36, Issue 2, April 1988, Pages 254-276*,

[2] Martin Tompa Zero Knowledge Interactive Proofs of Knowledge *IBM Research Division Thomas J. Watson Research CEnter P.O. Box 218*, Yorktown Heights, New York 10598

[3] M. J. Fischer, S. Micali, and Rackoff, "A Secure Protocol for the Oblivious Transfer" *Eurocrypt 84.*,

[4] M. O. Rabin, " How to Exchange Secrets", unpublished manidcript, *1981*,

[5] D. Angluin and D. Lichtenstein, " Provable Security of Cryptosystems: a Survey", *TEchnical Report TR-288, Yale University, October 1983.*,

[6] M. Tompa and H. Woll, "Random SElf-REducibility and Zero Knowledge Interactive Proofs of Possesion of Information", *28th Annual Símposium on Foundations of Computer Science*,Los Angeles, California, October 1987, 472-482.

[7] S. Goldwasser és M. Sipser. "Private coins versus public coins in interactive proof systems." *In S. Micali, szerkesztő, Randomness and Computation, volume 5 of Advances in Computing Research, pages 73-90. JAI Press, 1989. A preliminary version appeared in Proc. 18th Ann. ACM Symp. on Theory of Computing,* 1986, pp.59-68.

[8] S. Goldwasser, S. Micali, and C. Rackoff, "The Knowledge Complexity of Interactive Proof-System", *Procedings of the Seventeenth Annual ACM Símposium Theory of Computing*, Providence Rhode ISland, May 1985, 291-304.

[9] Shamir A.,Rivest R., Adleman L., "Mental Poker", in *Mathematical Gardner*, D.E. Klarner, ed., Wadsworth, 1981, pp.

[10] Lipton R.," How to Cheat at Mental Poker, "Processings of the AMS Short Course in Cryptography 1981.

[11] Goldwasser S., Micali S., "Probabilistic Encryption and Howto Play Mental Poker Keeping Secret All Partial Information, "Proceedings of the 14th STOC, pp, 365-377, 1982.