Eötvös Loránd Tudományegyetem
Természettudományi Kar

Palotay Dorka

Csalni vagy nem csalni

Szakdolgozat

Témavezető:
Szabó Csaba

Budapest, 2014.

Tartalomjegyzék

Chuck Norrisnak van a világon a legjobb pókerarca!
Ez segített neki megnyerni egy jollyval, egy makk
alsóval, egy kőr hetessel és egy bankkártyával
a '84-es póker világbajnokságot.

'83-ban.

1. Bevezetés

1933-ban Niels Bohr fiával Christiannal, Felix Blochal, Carl Friedrichel és Werner Heisenberggel egy síelés alkalmával pókerezni szeretett volna. Kártyájuk nem volt, ezért megpróbáltak fejben játszani. Ez az első ismert alkalom, amikor valakik a fejben pókerezés problémájával foglalkoztak. Bohrnak és barátainak azonban nem sikerült érdemben fejben kártyázniuk.

Ezután igen sokakat foglalkoztatott a kérdés, hogy hogyan lehetne kártyák és egy külső fél segítsége nélkül például telefonon keresztül pókerezni. Az internet megjelenésével pedig a kérdés egyre aktuálisabbá vált.

Az évek során egyre több és több póker protokoll született. Ezek a protokollok leírják, hogy a felek milyen módon kommunikálhatnak egymással, milyen típusú üzeneteket küldhetnek, illetve a játék menetét. Minden esetben felteszzük, hogy a játékosokban nem lehet bízni, ha lehetőségük adódik a csalásra, akkor azt ki is használják.

Szakdolgozatomban négy póker protokollt mutatok be. A kezdeti próbálkozásoktól az első elméletben jól működő megoldásig. A bemutatott póker protokollokat Claude Crépeau 1985-ös cikkében [6] megfogalmazott hét szempont szerint fogom vizsgálni. Ezek a következők:

1. Egyik lapot se lehessen többször kiosztani. Minden kártya legyen különböző a pakliban és a játékosok kezében is. Ha mégis előfordul, hogy valamelyik játékos olyan lapot húz, ami már más kezében is szerepel, akkor a csalást a többi játékos észre tudja venni.

2. Az osztás véletlenszerű legyen, azaz minden osztás egyenlő valószínűséggel forduljon elő. Fontos, hogy az összes játékos ugyanakkor hatással legyen a lapok osztására, így senki nem tudja befolyásolni, hogy ki mit húz.

3. Ne legyen szükség egy megbízható külső személyre. Hiszen sem egy élő személyben, sem egy gépben nem lehet teljesen megbízni.

4. A csalást nagy valószínűséggel ki lehessen szűrni. A cél, hogy a kódolás bonyolultságának függvényében exponenciálisan gyorsan csökkenjen annak a valószínűsége, hogy valaki úgy tud csalni, hogy nem bukik le. Mindeközben a protokoll műveletigénye csak polinomiálisan növekedjen.

5. A játékosok egymás kártyáiról semmilyen információt ne tudjanak szerezni. Itt nem csak arra kell gondolni, hogy ne ismerjék egymás lapjait, hanem arra is, hogy semmilyen apró részletet se tudjanak meg egymás kártyáiról, azaz ne tudják leszűkíteni a lehetséges lapok halmazát. Arra is oda kell figyelni, hogy a pakliban maradt kártyákról se tudjanak információt szerezni.

6. Amikor kettőnél több játékos van, akkor néhány játékos összefoghat, és megoszthatja egymással az információit. Azt sajnos nem lehet kiküszöbölni, hogy elmondják egymásnak a saját lapjaikat, de cél, hogy ennél több információhoz ne tudjanak hozzájutni. Ez azt jelenti, hogy egy játékos lapjait a többiek ne tudják megismerni mindaddig, amíg ő nem vesz részt a csalásban.

7. Az igazi póker játék során nagyon fontos szerepet játszik a blöffölés is, ezért a kieső játékosok nem kötelesek megmutatni a bedobott lapjaikat, így a többiek előtt rejtett marad a stratégiájuk. Ehhez hozzátartozik az is, hogy ne lehessen visszakövetni, hogy ki mikor milyen lapott húzott, illetve dobott el. Ennek megfelelően egy jó póker protokollnak úgy kell működnie, hogy a játékosok stratégiái rejtve maradjanak.

A dolgozatban elsőként bemutatott póker protokoll [1] az első publikált póker protokoll, mely két játékos esetén írja le a játék szabályait. Legnagyobb hibája, hogy némi információ ki tud szivárogni a játékosok lapjairól, és ezt felhasználva az egyik játékos nagyobb eséllyel tud nyerni.

Ezután Bárány és Füredi póker protokollját [2] ismerhetjük meg, mely segítségével már több játékos is játszhat egyszerre, de a szövetkezéssel szemben nem biztonságos.

Ezt a problémát Fortune és Merritt egy megbízható harmadik fél segítségével oldotta meg [3], de mint fent ezt már említettük szeretnénk elkerülni az osztó közreműködését.

Mindhárom protokoll hibája, hogy a játék végén a játékosoknak fel kell fedniük titkos kódolásukat, így a játék teljes menete visszakövethető, azaz mindenki megismeri a többiek stratégiáját. Igazi póker játékosok ilyen feltételek mellett nem ülnének le játszani.

Crépeau 1985-ben publikált cikkében egy olyan póker protokollt mutat be, mely ugyan már eleget tesz az első hat követelménynek, de a stratégiákat ebben is fel kell fedni a játék végén. Két évvel később azonban megjelent az első tökéletes póker protokoll [7], mely már mind a hét feltételt kielégíti. A dolgozat utolsó fejezetében ezt ismerhetjük meg.

2. Kártyázzunk kettesben

Fejben pókerezésről először Adi Shamir, Ronald L. Rivest és Leonard M. Adleman 1979-es cikkében olvashatunk.

Az említett cikkben a szerzők két látszólag ellentmondásos eredményt közöltek. Egyrészt bizonyítást adtak arra, hogy elméletileg lehetetlen fejben tisztességesen pókerezni. Majd bemutattak egy protokollt, mely segítségével megvalósítható a játék. Látni fogjuk, hogy ez a két állítás tökéletesen megfér egymás mellett.

2.1. A protokoll elméleti leírása

Először nézzük, hogy hogyan működik a szerzők által leírt protokoll, melyre a későbbiekben SRA-protkollként fogunk hivatkozni.

A protokoll két játékos esetén írja le a játék menetét. A példa kedvéért tegyük fel, hogy Réka és Lilla a két játékosunk, akik telefonon keresztül szeretnének játszani egymással.

Az SRA-protokoll alapötlete, hogy a játék elején a játékosok megegyeznek egy kódoló (E), és egy dekódoló (D) függvényben. Majd mindketten egymástól függetlenül választanak egy titkos kulcsot, melyet a játék végéig nem árulnak el. Jelölje a kulcsok halmazát $\mathcal{K}$. A fenti függvények felfoghatók úgy, hogy $\mathcal{K} \oplus \{1, 2, \ldots, 52\}$-ből képeznek az $\{1, 2, \ldots, 52\}$ halmazba.

A függvényektől a következő tulajdonságokat várjuk el:

1. $E_K(X)$ az X üzenet kódolt alakja, ahol K a kulcs.
2. $D_K(E_K(X)) = X$ minden X-re és K-ra.
3. $E_K(E_J(X)) = E_J(E_K(X))$ teljesül minden X üzenetre és K, J kulcsokra.
4. Adott X és $E_K(X)$ segítségével kiszámíthatatlan a K kulcs, ami azt jelenti, hogy polinomiális időben nem tudjuk megtalálni K-t.
5. Adott X és Y esetén polinomiális időben lehetelen két olyan K és J kulcs kiszámítása, melyre $E_K(X) = E_J(Y)$.

A játék elején feleltessük meg a kártyalapokat 52 számnak. Ezek tetszőleges számok lehetnek, melyekben a játékosok közösen megegyeznek. Az egyszerűség kedvéért tekintsük az $\{1, \ldots, 52\}$ számokat. Majd a játékosok meghatározzák a kódoló és dekódoló függvényeket, és mindketten választanak egy titkos kulcsot. Példánkban legyenek Réka függvényei E_R és D_R, Lilláé pedig E_L és D_L, ahol R Réka kulcsa, L pedig Lilláé.

Keverés.
Ahogy egy hagyományos póker játék esetében, itt is keveréssel kell kezdenünk. Ezt most Réka fogja elvégezni, aki az E_R függvény segítségével kódolja a kártyalapokat, majd az így kapott értékeket egy megkevert sorrendben elküldi Lillának. Azaz Lilla az $< E_R(1), \ldots, E_R(52) >$ sorozat egy permutációját kapja Rékától.

Osztás.
Az osztást már Lilla fogja elvégezni a következő módon. Először Réka lapjait osztja ki. Azaz kiválaszt tetszőleges öt kódolt lapot, amit visszaküld Rékának. Neki semmilyen információja nincs ezekről a lapokról, hiszen Réka kódolta őket, ugyanakkor Réka már könnyedén kiszámítja a saját lapjait a dekódoló függvénye segítségével. Így például legyenek a visszaküldött lapok $\{E_R(5), E_R(12), E_R(33), E_R(38), E_R(49)\}$. Ekkor Réka tudni fogja, hogy az ő lapjai $\{5, 12, 33, 38, 49\}$.

Ezután saját magának választ öt az eddigiektől különböző lapot a Réka által kódolt halmazból. Így biztosítva van, hogy ugyanaz a lap nem osztható ki kétszer. Legyenek ezek $\{E_R(2), E_R(15), E_R(23), E_R(35), E_R(40)\}$. Ezután a kiválasztott lapokat kódolja a saját kulcsa segítségével és az így kapott $\{E_L E_R(2), E_L E_R(15), E_L E_R(23), E_L E_R(35), E_L E_R(40)\}$ halmazt elküldi Rékának.

Ekkor Réka dekódolja a kapott kódokat a saját kulcsa szerint, és visszaküldi Lillának az eredményt, mely a példánkban
$\{E_L(2), E_L(15), E_L(23), E_L(35), E_L(40)\}$. Itt látható, hogy miért fontos a függvényekre vonatkozó harmadik tulajdonság. Ugyanis, ha a kiválasztott függvények kompozíciójára nem teljesülne a kommutativitás, akkor Réka nem tudná dekódolni a Lillától kapott lapokat.

Ezek után Lilla már könnyen kiszámítja a lapjait, melyek $\{2, 15, 23, 35, 40\}$.

Húzás.
Újabb lapok húzása ugyanúgy történik, mint az osztás. Mindig Lilla húz mindkettőjüknek a Réka által kódolt lapokból, így nem fordulhat elő, hogy kétszer ugyanazt a lapot kihúzzák.

Lapdobás.
Amikor a játékosok el szeretnék dobni valamelyik lapjukat, akkor egyszerűen bemondják a saját kulcsuk szerint kódolva, hogy melyiket szeretnék áthelyezni a dobott lapok halmazába. Persze ekkor némi információt megtudunk a másik stratégiájáról, mégpedig, azt hogy az éppen eldobott lapot melyik körben húzta.

A játék vége.
A játék végén a játékosok megosztják egymással titkos kulcsaikat, ezzel felfedve a kezükben lévő lapokat. Ennek segítségével ellenőrizhetik, hogy a másik tisztességesen játszott-e a játék során.

2.2. A lehetetlenség bizonyítása

Ebben a fejezetben azt mutatjuk meg, hogy annak ellenére, hogy a szerzők megadtak egy látszólag működőképes protokollt, hogy fordulhat elő, hogy a probléma megoldása elméletben mégis lehetetlen.

A fejben pókerezés során alapvető fontosságú, hogy ugyanúgy, ahogy a valódi játék esetében, itt se tudjanak meg semmilyen információt a játékosok egymás lapjairól. Ugyanakkor az is lényeges, hogy véletlenül sem fordulhasson elő, hogy két játékos ugyanazt a lapot kapja. Shamir, Rivest és Adleman cikkükben bebizonyítják, hogy két játékos esetén elméletileg lehetetlen, hogy ennek a két követelménynek egyidejűleg eleget tegyünk.

Ennek bizonyításához az egyszerűség kedvéért tételezzük fel, hogy két játékosunk Réka és Lilla egy három lapból álló paklival játszanak. Legyenek a kártyák X, Y és Z. A játék során a felek üzeneteket küldenek egymásnak. Egy ilyen üzenetsorozat eredményeként, a játékosok kapnak egy-egy lapot. Tételezzük fel, hogy az $M_1, M_2, \ldots, M_n$ üzenetsorozat után Rékánál az X, Lillánál az Y kártya van. Jelöljük S_R-rel azon lapok halmazát, melyet Réka kaphat az $M_1, M_2, \ldots, M_n$ üzenetek eredményeként. Hasonlóan Lilla esetében legyen ez a halmaz S_L. Nyilvánvaló, hogy $X \in S_R$, valamint $Y \in S_L$.

Amennyiben S_R kizárólag ebből az egyetlen elemből állna, akkor sérülne az a feltétel, hogy a játékosok semmilyen információval nem rendelkezhetnek egymás lapjairól. Hiszen Lillának nem kell mást tennie, mint véges sok próbálkozással kiszámítania, hogy az adott üzenetsorozat a különböző esetekben milyen eredményre vezethet. Ha minden esetben azt a választ kapja, hogy Rékához az X kártya került, akkor biztos lehet benne, hogy Rékánál ez a lap van.

Ha az S_R halmazban mindhárom lap szerepelne, akkor előfordulhatna, hogy Réka ugyanazt a lapot kapja, mint Lilla, hiszen az azt jelenti, hogy létezik olyan eset, amikor Réka és Lilla független döntéseinek eredményeként Rékához is Y kerül. Most már csak azt kell megvizsgálnunk, hogy mi történik, ha S_R két lapot, azaz X-et és Z-t tartalmazza. Hasonlóan az eddigi gondolatmenetehez S_L is csak két lapot tartalmazhat. Azonban ekkor Z az S_L halmazban is benne van, így megintcsak lehetséges, hogy Réka és Lilla ugyanazt a lapot kapják, nevezetesen Z-t.

2.3. Konkrét példa

Most, hogy áttekintettük a protokoll elméleti hátterét, vizsgálujk meg a Shamir, Rivest és Adleman által javasolt kódoló és dekódoló függvényeket.

Legyen

$$E_K(X) = X^K \ (mod\ n)$$

ahol n egy nagy prím, melyre $(K, n-1) = 1$.

Az ehhez tartozó dekódoló függvény:

$$D_K(Y) = Y^L \ (mod\ n)$$

ahol

$$L \cdot K \equiv 1 \ (mod\ n-1)$$

Nézzük meg, hogy a függvények valóban teljesítik-e a velük szemben támasztott követelményeket.

1. $E_K(X) = X^K \ (mod\ n)$ az X üzenet kódolt alakja, ahol K a kulcs.
2. $D_K(E_K(X)) = D_K(X^K) = X^{KL} \equiv X \ (mod\ n)$ az Euler-Fermat tétel szerint, hiszen $L \cdot K \equiv 1 \ (mod\ n-1)$.
3. $E_K(E_J(X)) = E_J(E_K(X)) = X^{KJ} \ (mod\ n)$.
4. Adott X és X^K esetén a kulcs meghatározásához az ún. diszkrét logaritmus problémát kell megoldanunk. Ugyan a probléma valódi nehézsége nem ismert, de jelenleg úgy hisszük, hogy ennek megoldása nehéz, azaz polinom időben nem végezhető el.
5. Ugyancsak a diszkrét logaritmus probléma nehézsége miatt adott X és Y esetén két olyan K és J kulcs kiszámítása, melyre $E_K(X) = E_J(Y)$ polinom időben nem végezhető el. Ez a tulajdonság biztosítja, hogy a játék végén a játékosok ne adhassanak meg más kódot, mint amit az elején választottak.

A játék kezdetén a játékosok választanak maguknak egy-egy K kulcsot, melyre teljesül, hogy $(K, n-1) = 1$. Nyilvánvaló, hogy ebben az esetben a kártyákat nem feleltethetjük meg az $\{1, \ldots, 52\}$ halmaznak, hiszen ekkor az 1-nek megfelelő lap kódja végig 1 maradna. Ezt figyelembe véve a játékosok kiválasztanak 52 tetszőleges számot. A játék pedig úgy zajlik, ahogy azt az előző fejezetben láthattuk.

2.4. Hol a hiba?

Ilyen feltételek mellett egy a matematikában jártas póker játékos biztos nem ülne le játszani. Eddig végig arról beszéltünk, hogy alapvető fontosságú, hogy a játékosok lapjai rejtve maradjanak. De az, hogy a konkrét lapokat nem ismerjük, még közel sem elegendő feltétel. Bármilyen apró részinformáció nagy segítség lehet a játék során. Már az is sokat segíthet, ha tudjuk, hogy milyen színű kártya van a játékostársunk kezében, vagy le tudjuk szűkíteni a lehetséges lapok halmazát. És bizony ez ennél a protokollnál könnyedén megtehető.

A probléma a következő egyszerű állításból adódik: *Ha k páratlan, akkor a^k akkor és csak akkor kvadratikus maradék modulo p, ha a kvadratikus maradék modulo p.*

Ez hol okoz problémát a játék során? A protokoll úgy indul, hogy minden kártyalapnak megfeleltetünk egy számot. Ezeket a számokat mindkét fél ismeri, és ezek hatványaival kódolják a lapokat a játékosok. Az $(R, n-1) = (L, n-1) = 1$ feltétel miatt tudjuk, hogy R és L páratlan kulcsok. Így a fenti észrevétel alapján látható, hogy ha egy laphoz rendelt eredeti szám kvadratikus maradék volt modulo n, akkor a kódolás után kapott érték is az lesz, hasonlóan kvadratikus nemmaradék esetén. Ezt az információt felhasználva, ha például tudjuk, hogy a nagy értékű lapok többsége kvadratikus maradék mod n, akkor Lilla magának választva a kvadratikus maradékokat, és Rékának osztva a kvadratikus nemmaradékokat, nagyobb eséllyel nyerhet a játékban.

2.5. Összefoglalva

A bevezetésben leírt hét követelmény között több is van, melyeknek nem tud megfelelni az SRA protokoll. Mint azt az előzőekben láthattuk a kvadratikus maradékok tulajdonságai miatt a lapokról bizonyos információ kiszivároghat. Az így kapott információk segítségével az egyik játékosnak lehetősége nyílik csalni, és ez természetesen befolyásolja azt is, hogy az egyes osztások milyen valószínűséggel fordulhatnak elő. Mindemellett a játékosok startégiája sem marad rejtve, hiszen a játék végén a kulcsok felfedésével a játék teljes egészében visszakövethető.

Ugyanakkor azt is láttuk, hogy minden lapot csak egyszer lehet kihúzni, a játékosok egymás kódjait nem tudják gyorsan megfejteni, illetve nincs szükség harmadik fél részvételére a játékban.

3. Csapatban könnyebb

Az eddigekben azzal foglalkoztunk, hogy hogyan pókerezhet két játékos fejben. Elsőként 1983-ban Bárány és Füredi cikkében olvashatunk egy olyan protokollt, mely segítségével már többen is játszhatnak.

3.1. A protokoll leírása három játékos esetén

Először megnézzük, hogy hogyan működik Bárány és Füredi protokollja három játékos esetén. A fejezet végén pedig a protokoll általános leírását is olvashatjuk.

Az első lényeges különbség, amire oda kell figyelnünk több játékos esetén, a kommunikációs csatornák kiépítésénél jelentkezik. Két játékosnál ez igen egyszerű volt, csak annyit kellett biztosítanunk, hogy a játékosok tudjanak egymással kommunikálni. Több játékos esetén viszont már előfordulhat, hogy bizonyos információt csak a játékosok egy kisebb csoportjával szeretnénk megosztani. Bárány és Füredi protokolljához kétféle kommunikációs csatornára lesz szükségünk. Egyrészt fontos, hogy a játékosok meg tudjanak osztani információt az összes többi játékossal egyidejűleg, ugyanakkor arra is szükség lesz, hogy páronként egy titkos csatornán keresztül tudjanak kommunikálni egymással.

Legyenek a játékosaink Réka, Lilla és Ákos.

Keverés.
Ahogy ezt korábban is tettük, feleltessük meg a kártyapaklit az $\{1, \ldots, 54\}$ halmaznak. Először mindhárom játékos kiválasztja a pakli egy-egy tetszőleges permutációját. Legyenek ezek rendre R, L és A. Valamint legyen a játékosok kezében lévő lapok halmaza rendre H_R, H_L és H_A. (Ezek a kezdetben mind megegyeznek az üres halmazzal.)

A játék elején Lilla és Ákos elküldi az L és A permutációkat Rékának, aki visszaküldi Lillának az AR^{-1}, Ákosnak pedig az LR^{-1} permutációt. Nagyon fontos, hogy ezeket a titkos csatornákon keresztül küldjék egymásnak.

A következő táblázatban azt láthatjuk, hogy az egyes játékosok milyen információval rendelkeznek a játék során:

Réka	R, L, A, H_R
Lilla	L, AR^{-1}, H_L, $R(H_R)$, $R(H_L)$, $R(H_A)$
Ákos	A, LR^{-1}, H_A, $R(H_R)$, $R(H_L)$, $R(H_A)$

1. táblázat. A játékosok információi

Húzás.
Tegyük fel, hogy Lilla szeretne egy új lapot. Ezt Ákostól kapja a következő módon. Ákos választ egy tetszőleges x számot, ami nem szerepel az $R(H_R)$, $R(H_L)$, $R(H_A)$ halmazokban, ezzel biztosítva, hogy ne lehessen kétszer ugyanazt a lapot kiosztani. Majd elküldi $LR^{-1}(x)$-et Lillának. Ebből Lilla kiszámítja a lapját L^{-1} segítségével. A kapott lapja $y = R^{-1}(x)$ lesz. Lilla hozzáadja H_L-hez y-t, és Lilla és Ákos is hozzáadja x-et $R(H_L)$-hez.

Ákos teljesen hasonló módon húz Lilla segítségével.

Látható, hogy Réka szerepe különböző, így ha ő szeretne egy új lapot, azt máshogy megkapni. Akár Lillától, akár Ákostól kaphatja. Tegyük fel, hogy Ákos segítségével húz egy új lapot. Ekkor Ákos újfent választ egy x-et, ami nem szerepel az $R(H_R)$, $R(H_L)$, $R(H_A)$ halmazokban, és ezt az x-et mind Rékának, mind pedig Lillának elküldi. Ezután Réka beveszi a H_R halmazba $y = R^{-1}(x)$-et, és Ákos és Lilla hozzáadják $R(H_R)$-hez x-et.

Rékának különleges szerepe van a játékban. Az ő permutációjára gondolhatunk úgy, mint a megkevert paklira. A többi játékos mind tudja, hogy ki melyik lapot tartja a kezében a megkevert pakliból. Így, amikor új lapot osztanak, akkor olyat választanak, ami még senkinél sem szerepel, így a játék során végig biztosak lehetünk abban, hogy a játékosoknál különböző lapok vannak. Ugyanakkor csak Réka tudja, hogy az ő permutációjával megkevert pakliban melyik lap melyiknek felel meg, így Lilla és Ákos választása teljesen véletlenszerű lesz.

A játék vége.
A játék végén, a játékosok a választott permutációkat megosztják egymással. Így felfedik kártyáikat, és megbizonyosodhatnak arról, hogy mindenki tisztességesen játszott.

3.2. Hol a hiba?

Vajon okos dolog lenne-e ilyen feltételek mellett igazi pénzben játszani? Látszólag a protokoll minden szükséges kérdést kezel. Meg tudjuk keverni a paklit, tudunk lapokat osztani, tudjuk, hogy mindenkinél különböző lapok vannak, és a végén még ellenőrizni is tudjuk, hogy mindenki valóban olyan lapokat birtokolt-e, mint amiket állított. Ennek ellenére mégsem jó ötlet egy ilyen protokollal működtetni például egy online póker játékot. Gondoljunk csak bele, mi történne, ha két játékos összebeszélne. Ha például Réka és Lilla osztja meg egymással az információit, akkor ismerik R-et is és azt is, hogy R szerint kinél milyen lapok vannak. Így azonnal tudni fogják Ákos lapjait is. Ha Lilla és Ákos fognak össze, akkor mivel ismerik AR^{-1}-et és A-t is, így könnyedén kiszámíthatják R-et, és ezáltal már tudni fogják Réka lapjait is.

Nyilvánvaló, hogy egy online póker játék esetén reménytelen azt elvárni, hogy a játékosok ne szövetkezzenek. Sokszor ez még akkor is elkerülhetetlen, amikor egy asztal körül ülve játszuk a játékot, és mindenki lát mindenkit. Azonban, ha otthon ülünk a számítógép előtt, akkor semmi nem garantálja, hogy a többi játékos nem játszik össze. A cél az, hogy az összefogásból a lehető legkevesebb előnyük származzon. Az nyilvánvaló, hogy saját lapjaikat meg tudják osztani egymással, de azt szeretnénk elérni, hogy ennél többet ne is tudjanak. A többiek lapjáról semmilyen információt ne tudjanak szerezni. Erre láthatunk egy megoldást a következő fejezetben.

3.3. A protokoll általános leírása

Előtte azonban nézzük Bárány és Füredi protokolljának általános leírását tetszőleges számú játékos esetén.

Legyenek a játékosok P_1, P_2, ..., P_n. Értelmezzük P_{i+1}-et a szokásos módon, kivéve, ha $i = n-1$. Ekkor legyen $P_{i+1} = P_1$. Jelöljük a játékosok kezében lévő lapok halmazát L_1, L_2, ..., L_n-nel, amik a játék kezdetén egyenlőek az üres halmazzal.

Keverés.

1. Minden P_i játékos ($i \in \{1, 2, \ldots, n\}$) kiválasztja egy tetszőleges π_i permutációját az $1, \ldots, 52$ számoknak.

2. Minden P_i játékos ($i \in \{1, 2, \ldots, n-1\}$) elküldi az általa választott permutációt P_n-nek egy titkos csatornán, úgy hogy a többiek ne tudják meg.

3. P_n minden P_{i+1} játékosnak ($i \in \{1, 2, \ldots, n-1\}$) elküldi $\pi_i \pi_n^{-1}$-et.

Húzás.
P_n húz egy új lapot:

1. P_1 választ egy tetszőleges $1 \leq x \leq 52$ számot, melyre $x \notin \bigcup_{i=1}^{n} \pi_n(L_i)$, és elküldi az összes többi játékosnak.

2. P_n új lapja $\pi_n^{-1}(x)$, amit hozzávesz L_n-hez.

3. A többi játékos x-et berakja az $\pi_n(L_n)$ halmazba.

P_i ($i \neq n$) húz egy új lapot:

1. P_{i+1} választ egy tetszőleges $1 \leq x \leq 52$ számot, melyre $x \notin \bigcup_{i=1}^{n} \pi_n(L_i)$, és elküldi az összes többi játékosnak, kivéve P_n-nek.

2. P_{i+1} elküldi $\pi_i\pi_n^{-1}(x)$-et P_i-nek.

3. P_i kiszámítja $\pi_n^{-1}(x)$-et ($\pi_n^{-1}(x) = \pi_i^{-1}\pi_i\pi_n^{-1}(x)$), majd berakja az L_i halmzaba.

4. Minden játékos, kivéve P_n, hozzáveszi x-et $\pi_n(L_i)$-hez.

3.4. Összefoglalva

A játék végén a játékosok permutációik felfedésével igazolják, hogy nem csaltak. Azonban ahogy azt a 3.2 fejezetben láthattuk, egy csalási lehetőség még így is marad. Ha ugyanis néhány játékos összefog, akkor lehetőségük nyílik megismerni mások lapjait. Ha biztosak lehetnénk benne, hogy a játékosok nem osztják meg egymással az információikat, akkor ez a protokoll biztonságos lenne, és minden egyéb követelménynek eleget tenne, de sajnos ebben sosem lehetünk biztosak.

4. Csak a kezemet figyeljék!

Ebben a fejezetben Fortune és Merritt protokollját tekintjük át, mely már védett a szövetkezéssel szemben. Ha két vagy akár több játékos összefog, akkor ennél a protokollnál csak egymás lapjait ismerik meg, a többiekéről semmilyen információt nem tudnak szerezni. Ehhez azonban szükség lesz egy megbízható külső személyre. Nevezzük őt osztónak. Természetesen, a fejben pókerezés problémája egy olyan osztó segítségével, aki végigköveti a játékot, triviális lenne. Ebben az esetben azonban az osztó csak a játék legelején vesz részt.

A protokollhoz kétféle kommunikációs csatornára van szükség. Egyrészt minden játékosnak egy titkos csatornán keresztül kell kommunikálnia az osztóval, úgyhogy a többi játékos ne szerezhessen semmilyen információt a küldött üzenetekről. Valamint a játékosoknak egymással is meg kell osztaniuk információkat, de ez már történhet úgy, hogy mindenki hallja.

Ez a protokoll is, csakúgy mint az előző, permutációkon alapszik. A játék elején a játékosok titkos permutációikat egyirányú függvények segítségével kódolják. Az egyirányú függvények olyan függvények, mely kiszámítása könnyű, de invertálása már nehéz feladat. Így a kódok ismeretében a többi játékos nem tudja kiszámítani az eredeti permutációt, viszont a játék végén mindenki könnyedén felfedheti saját permutációját.

4.1. A protokoll leírása három játékos esetén

Keverés.
Most megint képzeljük el, hogy Réka, Lilla és Ákos szeretnének játszani. Ehhez segítségükre van az osztó. A lapokat a szokásos módon megfeleltetjük az $\{1, \ldots, 52\}$ halmaznak. Az osztó a játék elején kiválasztja egy titkos π permutációját a lapoknak, amit nem oszt meg a játékosokkal. Réka, Lilla és Ákos szintén választanak egy-egy permutációt, legyenek ezek rendre α, β, γ. Mindhárman elküldik a választott permutációt a titkos csatornán keresztül az osztónak, aki kiszámítja a $\Delta = \alpha^{-1}\beta^{-1}\gamma^{-1}\pi^{-1}$ permutációt, és elküldi a játékosoknak.

A játék során végig ismert lesz, hogy a π permutáció szerint mely lapokat osztották már ki, így elkerülhető, hogy kétszer ugyanazt a lapot kihúzzák.

Húzás.
Nézzük, hogy is történik egy új lap kihúzása. Tegyük fel, hogy Ákos szeretne egy új lapot húzni. Ekkor kiválaszt egy olyan $y = \pi(x)$ értéket, amit még senki nem húzott ki, és ezt, valamint $\Delta(y)$-t megosztja a többiekkel. Ákos x értékét szeretné megtudni. Ehhez először Réka kiszámítja az $\alpha(\Delta(y)) =$

$\beta^{-1}\gamma^{-1}\pi^{-1}(y)$ értéket, és ezt elmondja a többieknek. Ezután Lilla alkalmazza a saját permutációját, melynek eredménye: $\beta(\alpha(\Delta(y))) = \gamma^{-1}\pi^{-1}(y)$. Ebből pedig Ákos már könnyedén megkapja x-et: $\gamma(\beta(\alpha(\Delta(y)))) = \pi^{-1}(y) = x$.

Igen ám, de mi történik, amikor Réka vagy Lilla szeretne lapot húzni? Ez a módszer olyankor nem használható. Persze a probléma könnyen orvosolható. Ahelyett, hogy mindenki egy permutációt választ a legelején, mindannyian egyből hármat küldenek el az osztónak. Ekkor Réka permutációi α_1, α_2, α_3. Lilla permutációi β_1, β_2 és β_3. Ákosé pedig γ_1, γ_2 és γ_3. Ezek után az osztó három Δ-t küld vissza. $\Delta_1 = \beta_1^{-1}\gamma_1^{-1}\alpha_1^{-1}\pi^{-1}$ segítségével Réka tud új lapot húzni. A $\Delta_2 = \gamma_2^{-1}\alpha_2^{-1}\beta_2^{-1}\pi^{-1}$ permutációt Lillának való osztáskor használjuk. És végül Ákos lapjait a $\Delta_3 = \alpha_3^{-1}\beta_3^{-1}\gamma_3^{-1}\pi^{-1}$ permutáció segítségével kapja.

A játék vége.
A játék végén minden játékos megosztja a titkos permutációit, így le tudják ellenőrizni, hogy mindenki tisztességesen játszott.

Csalási lehetőségek.
A protokoll leírásából látható, hogy olyan lapot nem húzhatnak a játékosok, ami már valamely játékostársuk kezében van. De valóban teljesül-e az, hogy ha ketten összefognak, akkor annál nagyobb előnyre nem tudnak szert tenni, minthogy megismerik egymás lapjait? Tegyük fel, hogy az osztó tisztességesen játszik, és a permutációkat titokban tartja. Az a kérdés, hogy ha két játékos szövetkezik akkor a játék során szerzett információkból tudnak-e a harmadik társuk lapjaira következtetni. Ennek vizsgálatához a következő táblázatban összefoglaljuk, hogy az egyes játékosok milyen információval rendelkeznek a játék során.

Réka	α_1, α_2, α_3
Lilla	β_1, β_2, β_3
Ákos	γ_1, γ_2, γ_3
Mindenki	Δ_1, Δ_2, Δ_3
Mindenki a Réka által választott y-okról	$\beta_1\Delta_1(y)$, $\gamma_1\beta_1\Delta_1(y)$
Mindenki a Lilla által választott y-okról	$\gamma_2\Delta_2(y)$, $\alpha_2\gamma_2\Delta_2(y)$
Mindenki a Ákos által választott y-okról	$\alpha_3\Delta_3(y)$, $\beta_3\alpha_3\Delta_3(y)$

2. táblázat. A játékosok információi

Tegyük fel, hogy Lilla és Ákos összefognak Réka ellen. Legyen a kettejük kezében összesen k darab lap. Mekkora eséllyel tudják kitalálni Réka egy lapját, vagy esetleg egy olyat, ami még a pakliban van?

Ahhoz, hogy Réka lapjait megtudják a $\Delta_(y) = \beta_1^{-1}\gamma_1^{-1}\alpha_1^{-1}\pi^{-1}(y)$ kifejezést kell megfejteniük, vagy egyszerűen a $\pi(x) = y$ egyenletet kell megoldaniuk. Az első esetben $\gamma_1\beta_1\Delta_1(y) = \alpha_1^{-1}\pi^{-1}(y)$ mindenki számára ismert. Azonban α_1-et Réka választotta véletlenszerűen, ezért Lilla és Ákos a saját lapjaik kiszámításához megkapott információkon túlmenően semmit nem tudnak α_1-ről. A második esetben ugyanez a gondolat elmondható, hiszen π is tetszőleges permutáció,melyet az osztó választott, és melyről feltettük, hogy az osztó semmilyen többlet információt nem szivárogtat ki.

Így mindkét esetben annak az esélye, hogy eltalálják, hogy mely lapot fedi y $1/(52-k)$.

Egy még a pakliban lévő lap esetén ugyanez a helyzet, hiszen ekkor is az a feladat, hogy egy olyan x-et találjanak, melyre $\pi(x) = y$.

4.2. A protokoll általános leírása

Most pedig következzen a protkoll leírása teszőleges számú játékos esetén. Legyenek a játékosok P_1, P_2, ..., P_n.

Keverés.

1. Az osztó választ egy véletlen π permutációt.
2. Minden P_i játékos ($i \in \{1, 2, \ldots, n\}$):
   (a) Kiválasztja n tetszőleges permutációját az $1, \ldots, 52$ számoknak: $\{\pi_{i,1}, \ldots, \pi_{i,n}\}$.
   (b) Egy titkos csatornán keresztül elküldi a $\{\pi_{i,1}, \ldots, \pi_{i,n}\}$ halmazt az osztónak.
   (c) Egy egyirányú függvény segítségével kódolja a permutációit, majd ezt megosztja a többiekkel.
3. Az osztó a következő lépéseket teszi:
   (a) Minden i-re kiszámítja a $\pi_i = \pi_{i+1,i}^{-1}\pi_{i+2,i}^{-1}\cdots\pi_{n,i}^{-1}\pi_{1,i}^{-1}\cdots\pi_{i,i}^{-1}\pi^{-1}$ permutációt, ahol $i \in \{1, 2, \ldots, n\}$.
   (b) Az összes játékosnak elküldi a $\{\pi_1, \pi_2, \ldots, \pi_n\}$ halmazt.

Húzás.
P_i húz egy új lapot:

1. P_i kiválaszt egy olyan $y = \pi(x)$ lapot, amit még senki nem húzott, és ezt valamint $\pi_i(y)$-t megosztja a többiekkel.

2. Sorban minden P_j játékos a $P_{i+1}, P_{i+2} \ldots, P_n, P_1, \ldots, P_{i-1}$ sorrendben a következőket teszi:

   (a) megkapja az x_{j-1} értéket az előző játékostól,

   (b) kiszámítja az $x_j = \pi_{j,i}(x_{j-1})$ értéket,

   (c) elküldi x_j-t a következő játékosnak.

3. P_i megkapja x_{i-1}-et P_{i-1}-től.

4. P_i kiszámítja $\pi_{i,i}(x_{i-1}) = x$-et.

5. Minden játékos megjegyzi, hogy P_i $y = \pi(x)$-et húzta.

4.3. Összefoglalva

Fortune és Merritt protokollja csakúgy, mint az előző kettő, úgy biztosítja, hogy a játék tisztességes legyen, hogy a játékosok végül elárulják permutációikat. Tehát a hetedik feltételnek ez a protokoll sem tesz eleget. Az osztó közreműködésével ugyan a feltételek közül öt teljesül, de célunk az, hogy a játékot egy külső fél segítsége nélkül is lehessen tisztességesen játszani.

5. S egy álom által elvégezni

Ebben a fejezetben Claude Crépeau póker protokollját ismerhetjük meg. Ez a protokoll eleget tesz a bevezetőben leírt mind a hét követelménynek. Legnagyobb eredménye, hogy a játékosok startégiája rejtett marad, a titkos kódokat nem kell felfedni a játék végén.

5.1. A protokoll alapötlete

Keverés.
Legyenek a játékosok $P_1, P_2, \ldots, P_n$. A kártyapaklinak megint feleltessük meg az $\{1, 2, \ldots, 52\}$ számokat. A játék elején a játékosok mindannyian kiválasztják a pakli egy-egy permutációját, melyet végig titokban tartanak. Legyen a P_i játékos által választott permutáció π_i. Ekkor a megkevert pakli $\pi_n \ldots \pi_2\pi_1$ lesz.

Húzás.
Nézzük meg, hogy mi történik, ha P_i szeretne húzni egy lapot. Annak érdekében, hogy a játékosok különböző lapokat húzzanak, húzáskor az $\{1, \ldots, 52\}$ számok közül választanak egy olyat, amelyet korábban még senki. Tegyük fel, hogy P_i k-t választja. Ekkor P_i húzott lapja $\pi_n \ldots \pi_2\pi_1(k)$ lesz. Ezt az értéket pedig úgy kapja meg, hogy először P_1 kiszámítja $\pi_1(k)$-t, majd ezt közzéteszi. Ebből P_2 meg tudja határozni $\pi_2\pi_1(k)$-t, majd szép sorban egészen P_{i-1}-ig a játékosok kiszámolják $\pi_{i-1} \ldots \pi_2\pi_1(k)$ értékét, melyet megosztanak a többiekkel. Ezután P_i titokban meghatározza $\pi_i\pi_{i-1} \ldots \pi_2\pi_1(k)$-t. Végül P_i sorban megszerzi a $\pi_{i+1} \ldots \pi_1(k)$, $\ldots$, $\pi_n \ldots \pi_1(k)$ értékeket. Ezeket azonban már nem nyilvánosan. A cél az, hogy P_i ezeket úgy kapja meg, hogy közben a játékosok, akiktől megszerzi az információkat, ne tudják, hogy mit árultak el. Az erre vonatkozó stratégia leírása a 5.2 alfejezetben olvasható.

Crépeau első cikkében a játék úgy zajlott, hogy a végén a játékosok megosztották egymással titkos permutációikat, és így ellenőrizték le, hogy mindenki tisztességesen játszott-e. Azonban a most vizsgált protokoll legérdekesebb kérdése, hogy hogyan bizonyíthatjuk, hogy nem csaltunk, és valóban olyan kártyát húztunk, ami nem volt más kezében, anélkül, hogy felfednénk a lapjainkat, ezáltal pedig a stratégiánkat.

5.2. A titkos titok

Crépeau protokolljának egyik alappillére, hogy úgy szerezzünk információt játékostársainktól, hogy közben ők ne tudják, hogy mit árulnak el nekünk.[5]

Képzeljük el, hogy Réka sok titkos információval rendelkezik. A titkok pontos tartalma ugyan nem ismert, de tudjuk, hogy mikre vonatkoznak. Lilla szeretné megtudni ezek közül az egyiket, melyet ő választ ki. Réka hajlandó ugyan átadni ezt az információt, de közben a többi titokról semmit nem szeretne elárulni. Ugyanakkor Lilla úgy szeretné megszerezni az információt, hogy Réka ne tudja meg, hogy melyikre volt kíváncsi a sok titok közül.

Milyen csalási lehetőségekre kell odafigyelnünk? Fontos, hogy Réka ne sejthesse, hogy Lilla melyik információra kíváncsi, valamint az is, hogy ne küldhessen Lillának hamis információkat. Lilla esetében pedig meg kell akadályoznunk, hogy egyszerre több titokról is információt szerezhessen.

5.2.1. A titkok kódolása

Legyenek $x_1, x_2, \ldots, x_n$ Réka titkai, melyek legfeljebb t bit hosszúak (ha valamelyik rövidebb, akkor 0-ákkal kiegészítjük). Jelölje az x_i titok j-edik bitjét b_{ij}, ahol $1 \leq i \leq n$ és $1 \leq j \leq t$. Először Réka választ két nagy véletlenszerű prímet, p-t és q-t. Kiszámítja $m = pq$ értékét, és meghatároz egy kvadratikus nemmaradékot modulo m, melynek Jacobi szimbóluma $+1$. Legyen ez y.

Ilyen y-t polinom időben tudunk találni. Arra van szükségünk ugyanis, hogy y kvadratikus nemmmaradék legyen mind modulo p, mind pedig modulo q. Először keresünk egy kvadratikus nemmaradékot modulo p. Véletlenszerűen kiválasztunk egy $a \neq 0$ értéket, majd teszteljük, hogy kvadratikus maradék-e. Ehhez kiszámítjuk $a^{\frac{p-1}{2}}$ értékét. Ha a nem kvadratikus nemmaradék, akkor egy új a-val próbálkozunk. Annak a valószínűsége, hogy a kiválasztott a kvadratikus nemmmaradék $1/2$, így várhatóan 2 próbálkozás elég lesz. Végül az így kapott a értéket modulo p is teszteljük.

Ezután Réka minden titok minden egyes bitjét kódolja a következő módon. Minden b_{ij}-hez választ egy véletlenszerű $c_{ij} \in \mathbf{Z}_m^*$ számot, és kiszámítja $z_{ij} = c_{ij}^2 y^{b_{ij}}$ modulo m értékét.

Amennyiben $b_{ij} = 0$, akkor z_{ij} kvadratikus maradék lesz modulo m, hiszen éppen c_{ij} négyzetével egyenlő. Amennyiben $b_{ij} = 1$, akkor z_{ij} kvadratikus nemmaradék, mivel y is az, c_{ij}^2 kvadratikus maradék, és egy kvadratikus maradék, valamint egy kvadratikus nemmaradék szorzata kvadratikus nemmaradék.

Ezek után Réka elküldi Lillának m-et, y-t, és az összes z_{ij}-t, míg p-t és q-t titokban tartja.

A kvadratikus maradékokra vonatkozó sejtés szerint polinom időben nem eldönthető m és y ismeretében, hogy egy adott z_{ij} kvadratikus maradék-e. Ezért Lilla a kapott z_{ij}-kből nem tudja gyorsan kitalálni, hogy milyen információt rejtenek.

Csalási lehetőségek.
Rékának esélye nyílna a csalásra, ha hazudna Lillának, és az átadott y kvadratikus maradék modulo m, nem pedig kvadratikus nemmaradék. Hiszen ekkor, mint azt a következő fejezetben látni fogjuk, a kérdésekből Réka tud következtetni arra, hogy Lilla melyik titokra kíváncsi. Ezért Rékának meg kell győznie Lillát arról, hogy m és y eleget tesznek a követelményeknek.

Ahhoz, hogy Lilla meggyőződjön arról, hogy y valóban kvadratikus nemmaradék, különböző maradékosztályokat küld Rékának, akinek az a feladata, hogy eldöntse ezekről, hogy kvadratikus maradék vagy nemmaradék modulo m. A küldött számokat a következő képpen választja meg: véletlenszerűen választ egy r értéket és elküldi $r^2 \ mod \ m$-et vagy $yr^2 \ mod \ m$-et. Azt, hogy épp melyiket küldi, azt is véletlenszerűen dönti el. Mivel Réka ismeri m prímtényezős felbontását, ezért könnyedén ki tudja számítani egy tetszőleges x értékről, hogy kavdratikus maradék-e. Ha y valóban kvadratikus nemmaradék, akkor $yr^2 \ mod \ m$ is kvadratikus nemmaradék lesz modulo m. Ellenkező esetben azonban $yr^2 \ mod \ m$ kvadratikus maradék. Tudjuk, hogy $r^2 \ mod \ m$ y-tól függetlenül mindig kvadratikus maradék. Így ha Réka hazudott, és y kvadratikus maradék, akkor minden esetben kvadratikus maradékot kap Lillától, így csak úgy csalhatna, ha mindig sikerül eltalálnia, hogy a Lilla által küldött érték, megfelelő y mellett mikor lenne kvadratikus nemmaradék.

Itt természetesen még oda kell figyelni arra is, hogy Lilla a feltett kérdésekkel ne tudjon csalni, azaz amellett, hogy ellenőrzi, hogy y valóban kvadratikus nemmaradék, más információt ne tudjon szerezni. Például egy olyan x értékről kérdezi meg, hogy kvadratikus maradék-e, amelyet nem a fent említett két módszer valamelyikével kapott meg. Ennek részleteit [11] cikkben olvashatjuk.

Rékának még azt kell bebizonyítania, hogy m a megfelelő formájú, azaz prímtényezős felbontásában csak két prím szerepel. Ennek bizonyításához azt használjuk ki, hogy ha egy adott m szám prímtényezős felbontásában i darab prím szerepel, akkor $\mathbf{Z}_m^*$ azon elemeinek, melyek Jacobi-szimbóluma $+1$ pontosan $1/2^{i-1}$ része kvadratikus maradék. Ezt felhasználva Réka és Lilla generálnak k véletlenszerű elemet $\mathbf{Z}_m^*$-ből. Azoktól, amelyek Jacobi-szimbóluma -1 eltekintenek, majd a többiről Réka sorban bebizonyítja, hogy kvadratikus maradék vagy kvadratikus nemmaradék. Így, ha Lilla az esetek $\frac{3}{8}$-ában kvadratikus nemmaradékot kap, úgy elfogadja, hogy m két prímtényező szorzata.

5.2.2. A kérdések kódolása

Most nézzük meg, hogy Lilla hogyan tudhatná meg egy adott b_{ij} értékét anélkül, hogy Réka elárulná p-t és q-t. Ehhez Lilla választ egy tetszőleges

$r \in \mathbf{Z}_m^*$ számot és egy a-t, melynek értéke véletlenszerűen 0 vagy 1. Ezek után kiszámítja a a következő értéket: $\tilde{q} = z_{ij} r^2 y^a$ modulo m. Az így kapott $\tilde{q}$-t elküldi Rékának, aki eldönti $\tilde{q}$-ról, hogy kvadratikus maradék vagy kvadratikus nemmaradék, majd ezt megosztja Lillával. (Ezt, mivel ismeri p-t és q-t, könyen megteheti).

Ennek ismeretében Lilla már tudni fogja b_{ij}-t. Ekkor

$$\tilde{q} = z_{ij} r^2 y^a = c_{ij}^2 r^2 y^{b_{ij}+a}$$

Így $\tilde{q}$ akkor lesz kvadratikus maradék, ha y kitevője 0 vagy 2, azaz $b_{ij} = a$. Ellenkező esetben kvadratikus nemmaradék. Mivel Lilla ismeri a-t, ezért $\tilde{q}$ segítségével b_{ij}-t is meg tudja határozni.

A kérdés az, hogy $\tilde{q}$ segítségével Réka ki tudja-e találni, hogy Lilla melyik b_{ij}-re volt kíváncsi. Mivel r és a véletlenszerűen választott elemek voltak, ezért $\tilde{q}$ is egy véletlenszerű eleme $\mathbf{Z}_m^*$-nak, valamint $\tilde{q}$ Jacobi szimbóluma $+1$. Így Réka nem tudhatja, hogy melyik titokra kérdezett rá Lilla.

Ezek szerint, ha Lilla megszeretné tudni az egyik x_i titkot, akkor t kérdést kell feltennie Rékának, egyenként minden bitre, és ezek eredményeként meg is kapja x_i-t.

Ehhez Lilla legyárt $n \cdot t$ kérdést, minden titok minden bitjére egyet-egyet, tehát minden z_{ij}-hez fog tartozni egy kérdés. A titkok sorrendjét Lilla megkeveri. Ehhez legyen σ egy tetszőleges permutációja az $\{1, 2, \ldots, n\}$ elemeknek, és legyen $\tilde{q}_{kj} = z_{ij} r_{kj}^2 y^{a_{kj}} \bmod m$, ahol $i = \sigma^{-1}(k)$, r_{kj} egy véletlenszerű eleme $\mathbf{Z}_m^*$-nak, a_{kj} pedig egy véletlenszerű $0-1$ érték. Legyen a kérdések egy sorozata $Q =< \tilde{q}_{kj} | 1 \leq k \leq n, 1 \leq j \leq t >$. Q-t úgy képzelhetjük el, mint egy $n \cdot t$ nagyságú táblázat, melynek minden sora egy-egy titok t bitjéhez tartozó kérdésekből áll.

Mivel $\tilde{q}_{kj}$-k mind véletlenszerű elemei $\mathbf{Z}_m^*$-nak, ezért $\mathbf{Z}_m^*$ tetszőleges $n \cdot t$ $+1$ Jacobi szimbólumú elemeiből álló táblázat megfelelhet a kérdések táblázatának, ezért Réka a kapott halmazból nem tud arra következtetni, hogy melyik kérdés melyik titokra vonatkozik.

Csalási lehetőségek.
Az eddig vázolt protokoll esetén azonban egyelőre még több csalási lehetőség is van:

- Lilla feltehet t kérdést úgy, hogy a kérdések különböző titkokra vonatkoznak.
- Lilla egy kérdésen belül több bitről is szerezhet információt, ha a kérdés például $\tilde{q} = z_{ij} z_{kj} r^2 y^a$ modulo m alakú.

Tehát az egyetlen lépés, ami még hátra van, hogy Lilla bebizonyítsa Rékának, hogy az elküldött kérdések megfelelnek a követelményeknek. Mindezt úgy kell megtennie, hogy közben Réka ne ismerje meg a kiválasztott σ permutációt, hiszen akkor már tudna következtetni a kérdésekre.

Ehhez Réka és Lilla megegyeznek egy s biztonsági paraméterben. Lilla a már legyártott Q táblázat mellé készít még s darab hasonló táblázatot. Ehhez választ s tetszőleges permutációt $(\sigma_1, \ldots, \sigma_n)$, $n \cdot t \cdot s$ véletlenszerű elemet $\mathbf{Z}_m^*$-ből, és ugyanennyi $0-1$ értéket. Ezek segítségével a fent leírt módon elkészíti a P_1, P_2, $\ldots$, P_s táblázatokat, melyek szintén megfelelő kérdéseket tartalmaznak a titkokra vonatkozóan. Ezek után Lilla elküldi ezeket Q-val együtt Rékának. Ezután Réka kiválasztja az $\{1, 2, \ldots, s\}$ halmaznak egy tetszőleges részhalmazát. Legyen ez X. Ahhoz, hogy Lilla meggyőzze Rékát a kérdések helyességéről a következőket kell tennie:

Egyrészt minden $k \in X$ érték esetén bebizonyítja, hogy P_k tisztességes kérdésekből áll. Ehhez elárulja σ_k-t, valamint a P_k elkészítéséhez használt $\mathbf{Z}_m^*$-beli értékeket, és $0-1$ értékeket.

Másrészt minden $k \notin X$ esetén Lilla bebizonyítja, hogy Q pontosan akkor tartalmaz megfelelő kérdéseket, ha P_k is. Ehhez elárulja $\sigma_k^{-1}\sigma$-t, és megmutatja, hogy Q kérdéseit meg tudja feleltetni P_k kérdéseinek.

Lilla csak úgy tudna csalni, ha minden $k \in X$ esetén a legyártott P_k táblázat tisztességes kérdéseket tartalmaz, míg a többi táblázatban az eredetinek megfelelően csalás van. Azonban mivel a táblázatokat Lilla azelőtt küldi el Rékának, hogy ő meghatározná X-et, ezért ennek az esélye mindössze 2^{-s}.

5.2.3. A titok megszerzése

- Réka kiválasztja p-t és q-t. Kiszámolja $m = pq$-t és elküldi Lillának, y-al együtt.
- Réka minden titok minden bitjéhez legyártja z_{ij}-t, és az így kapott táblázatot elküldi Lillának.
- Lilla létrehozza Q-t, a kérdések táblázatát. Ezt átadja Rékának és bebizonyítja, hogy a kérdések tisztességesek.
- Lilla elküldi $k = \sigma(i)$-t Rékának, mely megmutaja, hogy mely kérdésekre vár választ. Ezek épp az x_i titokra vonatkoznak.
- Réka minden $\tilde{q}_{kj}$-ről $(1 \leq j \leq t)$ Q-ból elárulja, hogy kvadratikus maradék, vagy kvadartikus nemmaradék. És hogy bizonyítsa, hogy nem csal, ha $\tilde{q}_{kj}$ kvadratikus maradék, akkor elküldi egy négyzetgyökét, ha pedig kvadratikus nemmaradék, akkor $\tilde{q}_{kj}y$ egy négyzetgyökét küldi el.

(Létezik olyan randomizált algoritmus, mellyel polinom időben található négyzetgyök p és q ismeretében.)

- Lilla kiszámítja a b_{ij} bitek értékét ($1 \leq j \leq t$), ezzel megkapja x_i-t.

5.3. A permutáció az permutáció

Crépeau póker protokollja permutációkra épül, melyeket a játékosok a játék elején egymástól függetlenül, véletlenszerűen választanak ki, majd elkódolva megosztják társaikkal. Annak érdekében, hogy a játékosok stratégiája rejtve maradjon, azt szeretnénk, ha a játék végén nem kellene felfedniük titkos permutációikat. Így azonban akár az is előfordulhat, hogy valamely játékos kódjai nem is egy permutációt takarnak, hanem például pár lapot kivett a pakliból, míg néhányat többször is berakott. Ahhoz, hogy a játék tisztességes legyen, fontos, hogy biztosak legyünk benne, hogy semelyik játékos nem próbál ilyen módon csalni.

Ezért a játék elején a játékosoknak meg kell győzniük egymást, hogy a kódjaik valóban a $\{1, \ldots, 52\}$ számok egy permutációját takarják. Ezt úgy tehetik meg, hogy két kódolt permutációról megmutatják, hogy ugyanazokat az elemeket tartalmazzák, majd az egyiket kikódolva bebizonyítják, hogy valóban permutációk.

Tegyük fel, hogy Réka beszeretné bizonyítani Lillának, hogy kódjai valóban egy permutációt takarnak.

Legyen $X = \{x_1, x_2, \ldots, x_n\}$ elemek egy sorozata, és σ valamint σ' két tetszőleges permutációja X elemeinek. Tegyük fel, hogy minden elem legfeljebb t bit hosszú, és legyen $b_{i,j}$ x_i j-edik bitje ($1 \leq i \leq n, 1 \leq j \leq t$). Ezt már a korábbiakban is látott módon kódolja Réka. Választ két nagy prímet, veszi ezek szorzatát, és az így kapott $m = pq$ értéket, valamint egy y kvadratikus nemmaradékot megosztja Lillával. Ezek után veszi $\mathbf{Z}_m^*$ tetszőleges $r_{i,j}$ illetve $r'_{i,j}$ elemét, és legyen $b_{\sigma(i),j}$ titkos kódolása $\hat{b}_{i,j} = r_{i,j}^2 y^{b_{\sigma(i),j}} \bmod m$, míg $b_{\sigma'(i),j}$ kódja pedig $\hat{b}'_{i,j} = r'^2_{i,j} y^{b_{\sigma'(i),j}} \bmod m$.

Ezeket úgy képzelhetjük el, mint két $n \cdot t$ nagyságú táblázatot, melynek minden sora X egy-egy elemének bitjeit kódolja. A sorok pedig a σ illetve σ' permutációk szerint vannak sorban. Réka azt szeretné bebizonyítani Lillának, hogy a két táblázat sorait meg tudjuk feleltetni egymásnak úgy, hogy ugyanazt az elemet kódolják.

Ehhez megegyeznek egy s biztonsági paraméterben. Ez azt adja meg, hogy a következő lépéseket hány alkalommal fogják elvégezni. Minél nagyobb s, Réka annál kisebb eséllyel tud csalni. Minden lépésben Réka kiválasztja az $\{x_1, x_2, \ldots, x_n\}$ elemek egy újabb permutációját, majd megint gyárt egy

ennek megfelelő táblázatot tetszőleges $c_{i,j} \in \mathbf{Z}_m^*$ elemek segítségével. Legyenek az így kapott új kódok $\bar{b}_{i,j} = c_{i,j}^2 y^{b_{\rho(i),j}} \bmod m$ alakúak, ahol $1 \leq i \leq n$, $1 \leq j \leq t$.

Ezután Lilla véletlenszerűen választ egy $0-1$ értéket (például pénzfeldobással). Ha 0-t választ, akkor Réka bebizonyítja, hogy a σ permutáció szerinti táblázat megfeleltethető a ρ szerinti táblázatnak. Ezt úgy teszi meg, hogy megmutatja, hogy melyik sor melyiknek felel meg, és itt ha képezzük a megfelelő $\hat{b}_{\sigma^{-1}(i),j} \cdot \bar{b}_{\rho^{-1}(i),j}$ szorzatokat, akkor Réka ezeknek megadja egy-egy négyzetgyökét. A fentiek szerint $\hat{b}_{\sigma^{-1}(i),j}$ és $\bar{b}_{\rho^{-1}(i),j}$ a $b_{i,j}$ bitet kódolják.

Ha $b_{i,j} = 0$, akkor $\hat{b}_{\sigma^{-1}(i),j} \cdot \bar{b}_{\rho^{-1}(i),j} = r_{\sigma^{-1}(i),j}^2 \cdot c_{\rho^{-1}(i),j}^2$, azaz Réka elküldi Lillának a $r_{\sigma^{-1}(i),j} \cdot c_{\rho^{-1}(i),j}$ értéket.

Ha $b_{i,j} = 1$, akkor $\hat{b}_{\sigma^{-1}(i),j} \cdot \bar{b}_{\rho^{-1}(i),j} = r_{\sigma^{-1}(i),j}^2 \cdot c_{\rho^{-1}(i),j}^2 \cdot y^2$, azaz Réka elküldi Lillának a $r_{\sigma^{-1}(i),j} \cdot c_{\rho^{-1}(i),j} \cdot y$ értéket.

Látható, hogy az így képzett szorzatok csak akkor lesznek kvadratikus maradékok, ha valóban ugyanazt a b_{ij} értéket kódolják.

Hasonlóan, ha Lilla 1-et választ, akkor Réka azt mutatja meg, hogy a σ' permutáció szerinti táblázat megfeleltethető a ρ szerinti táblázatnak.

Majd a végén Réka kikódolja a σ' szerinti kódokat, és megmutatja Lillának, hogy ez valóban egy permutációt takart.

Látható, hogy minden lépésben Réka választ egy permutációt, majd Lilla véletlenszerű döntésének függvényében vagy azt bizonyítja, hogy ez valóban egy permutáció, vagy azt, hogy megfeleltethető az eredeti σ permutáció kódolásának. Azaz Réka csak úgy csalhatna, ha minden lépésben, amikor Lilla 0-át választ, akkor úgyanúgy rossz táblázatot ad meg, mint az elején, ha pedig Lilla 1-et mond, akkor valóban egy permutációnak megfelelő táblázatot készít. Mivel azonban ezeket a táblázatokat azelőtt készíti el, hogy Lilla döntene, ezért mindössze $1/2^s$ esélye van a csalásra.

Crépeau póker protokollja esetén minden P_i játékosnak be kell bizonyítania, hogy valóban a $\{1, \ldots, 52\}$ számok egy permutációját választotta. Ehhez a fenti módszert alkalmazzák $\sigma = \pi_i$ és $\sigma' = I$ választással, ahol I az identikus permutáció, és sorban minden játékostársukkal elvégzik a lépéseket.

5.4. A protokoll

Keverés.
Minden P_i $(1 \leq i \leq n)$ játékos elvégzi a következő lépéseket:

- Kiválasztja egy tetszőleges π_i permutációját az $\{1, 2, \ldots, 52\}$ számoknak.

- Választ két véletlen nagy prímet, p_i-t és q_i-t. Ezekből kiszámítja $m_i =$

p_iq_i-t, és ezt y_i-vel együtt elküldi a többieknek. y_i egy modulo m_i kvadratikus nemmaradék, melynek Jacobi-szimbóluma $+1$.

- Bebizonyítja, hogy m_i és y_i megfelel a követelményeknek.
- Elvégzi π_i kódolását a 5.2.1 alfejezetben leírtak szerint, majd megosztja a többi játékossal. A titkos információk, amiket P_i elkódol a $\pi_i(k)$ értékek, ahol $1 \leq k \leq 52$. Így, ha az $\{1, 2, \ldots, 52\}$ számokat használjuk a kártyák jelölésére, akkor egy $52 \cdot 6$-os táblázatot jelent a permutációk kódolása.
- Elkészíti a kérdések táblázatát minden P_j esetén a $\pi_j(1), \pi_j(2), \ldots, \pi_j(52)$ értékekhez $(1 \leq j \leq n)$. (ld. 5.2.2 fejezet)
- Megmutatja, hogy π_i valóban egy permutáció a 5.3 alfejezetben leírtak szerint.

Húzás.
A játék kezdetén a pakli minden lapja szabad lapként van megjelölve. Amikor egy P_i játékos húzni szeretne egy új lapot, akkor kiválaszt egy tetszőleges számot $\{1, 2, \ldots, 52\}$ közül, majd ezt kihúzottnak jelöli. Legyen ez k. Ezután P_1 kiszámítja $\pi_1(k)$-t, amit megoszt a többiekkel. Utána P_2 alkalmazza az így kapott értéken a permutációját, és közli a többiekkel $\pi_2\pi_1(k)$-t. Ezt folytatják egyészen az $i-1$-dik játékosig, aki nyilvánosságra hozza $\pi_{i-1} \ldots \pi_1(k)$ értékét. Ezután P_i titokban kiszámítja $\pi_i\pi_{i-1} \ldots \pi_1(k)$-t. Végül sorban egészen az utolsó játékosig P_i titkosan a 5.2.3 fejezetben leírtaknak megfelelően megszerzi a szükséges permutációkat, azaz $\pi_{i+1} \ldots \pi_1(k)$-től $\pi_n \ldots \pi_1(k)$-ig mindet. Így a végén megkapott $\pi_n \ldots \pi_1(k)$ érték fog megfelelni P_i kihúzott lapjának.

A kérdés az, hogy hogyan lehetünk biztosak abban, hogy mindenki olyan lapot húz, amit a többiek nem húztak korábban anélkül, hogy köteleznénk a játékosokat, hogy a játék végén megosszák titkos permutációikat. Ennek egyik fontos lépése, hogy miután P_i kiszámítja $\pi_i\pi_{i-1} \ldots \pi_1(k)$-t minden előtte lévő P_j $(1 \leq j \leq i-1)$ játékosnak be kell bizonyítania, hogy nem használta még a titkos kérdéseit $\pi_i\pi_{i-1} \ldots \pi_1(k)$ kiszámítására. Ehhez nem kell mást tenniük, mint elárulni $\sigma_j\pi_{i-1} \ldots \pi_1(k)$-t, azaz megadni, hogy a kérdéseik táblázatának mely sora vonatkozik a $\pi_i\pi_{i-1} \ldots \pi_1(k)$ értékre. Ez megtehető úgy, hogy a kérdések permutációjának egy kódolását a játékosok a játék elején közzéteszik, majd ebben a lépésben kikódolják a megfelelő értéket.

Ezek után biztosak lehetünk abban, hogy $\pi_i\pi_{i-1} \ldots \pi_1(k)$-t senki nem tudja mégegyszer kiszámítani anélkül, hogy a P_i játékos tudna róla. Hiszen minden előtte lévő játékos elárulta, hogy mely kérdések vonatkoznak a kérdéses permutációra, míg az utána következők nyilvánosan teszik fel neki

kérdéseiket, így biztosan lebuknak. Az még előfordulhat, hogy P_i közreműködésével számítja ki valaki ezt az értéket, de arról már volt szó, hogy a játékosok összefogását teljes mértékben nem lehet kizárni, ugyanakkor ebben az esetben a csaló játékos csak P_i lapját fogja megismerni, más előnye nem származhat a csalásból.

Lapok felfedése és dobása.
Végül még azt kell megvizsgálnunk, hogy hogyan lehet felfedni lapjainkat, illetve eldobni azokat, amelyekre már nincs szükségünk. Jó megoldásnak tűnik, ha a játékosok lapdobáskor egyszerűen bemondják az eldobni kívánt lap eredeti kódját, azaz ha az eldobni kívánt lap $\pi_n \dots \pi_1(k)$, akkor k-t. Hasonlóan egy lap felfedésekor elárulják a $\pi_i\pi_{i-1} \dots \pi_1(k)$, $\dots$, $\pi_n \dots \pi_1(k)$ permutációkat. Ugyanakkor ez a módszer sértené a protokollal szemben támasztott utolsó követelményt, mégpedig azt, hogy a játékosok stratégiája rejtve maradjon. Hiszen így a játékostársaink megtudnák, hogy az eldobott illetve kijátszott lapokat mikor húztuk.

Legyen K_i azon lapok π_i permutáció szerinti sorozata, melyeket P_i játékos húzott. Ezen belül jelölje D_i az eldobott lapok egy sorozatát, H_i pedig a kézben tartott lapokét. P_i a H_i-beli lapok egy permutációjával fogja biztosítani, hogy a többiek ne tudják, hogy melyik lapot mikor húzta. A kihúzott $\pi_i\pi_{i-1} \dots \pi_1(k)$ lapokat behelyezi a H_i halmazba. Mielőtt azonban kijátszana vagy eldobna egy lapot létrehozza a H_i-beli lapok egy új titkos permutációját, melyről bebizonyítja a 5.3 fejezetben bemutatott eljárás segítségével, hogy valóban az elemek egy permutációja.

Ha P_i elszeretne dobni egy lapot, akkor a kapott kódok segítségével egyszerűen elmondja, hogy melyiket helyezi át H_i-ből D_i-be.

Ha pedig P_i felfedné egy kártyáját, akkor egyszerűen elárulja $\pi_i\pi_{i-1} \dots \pi_1(k)$-t, majd dekódolja az ennek megfelelő értéket H_i-ben. Ezután sorban minden rákövetkező P_j $(i+1 \leq j \leq n)$ játékos esetén P_i felfedi $\pi_j\pi_{j-1} \dots \pi_1(k)$ értékét, valamint P_j is dekódolja ezt a permutációját. Így a végén mindenki számára ismertté válik $\pi_n \dots \pi_1(k)$, azaz P_i lapja.

5.5. A biztonság növelése

Crépeau a csalás valószínűségét még tovább kívánta csökkenteni. Az ötlete az, hogy π_1, π_2, $\dots$, π_n minden értékéhez egy véletlen információt társítunk. Esetünkben ez egy véletlenszerűen választott $0-1$ sorozat, mely megfelelően hosszú ahhoz, hogy nagyon kis eséllyel lehessen véletlenül eltalálni. Így, amikor egy játékos megszeretné tudni egy szám adott permutációhoz tartozó értékét, akkor az érték mellett ezt a titkos információt is meg kell szereznie. A játék végén azok a játékosok, akik felfedik a lapjaikat, ezeket a plusz in-

formációkat is megosztják a többiekkel, akik ezt össze tudják hasonlítani az eredeti $0-1$ sorozattal, melyek, ha senki sem csalt, meg fognak egyezni.

A játék elején a játékosok megegyeznek egy s számban, mely a $0-1$ sorozatok hossza lesz. Ezután minden játékos véletlenszerűen hozzárendel a pakli minden lapjához egy-egy ilyen sorozatot. Jelölje a P_i játékos által a k laphoz rendelt számot $\tau_i(k)$. Tehát P_i játékos a következő titkos információkkal rendelkezik:

$$< \pi_i(1), \tau_i(1) >, < \pi_i(2), \tau_i(2) >, \ldots, < \pi_i(52), \tau_i(52) >$$

Így, ha P_j meg szeretné tudni $\pi_i(k)$ értékét, akkor $\tau_i(k)$-t is meg kell szereznie. Tehát, ha csalni próbálna, és $\pi_i(k)$ helyett $\pi_i(k')$-t szerzi meg, akkor a játék végén nem tudná bizonyítani, hogy ő $\pi_i(k)$-t olvasta, hiszen nem ismeri $\tau_i(k)$-t, csak $\tau_i(k')$-t.

5.6. Hol a hiba?

A fejezet elején azt mondtuk, hogy Crépeau póker protokollja az első tökéletes megoldás. Akkor hogy fordulhat elő mégis, hogy a protokoll hibájáról is beszélnünk kell? A válasz igen egyszerű. A dolgozat elején bemutatott feltételeknek ugyan eleget tesz a fenti protokoll, ugyanakkor ezeken a szempontokon kívül mást is szemelőtt kell tartanunk. Az internet elterjedéséig a fejben pókerezés problémája egy igen érdekes elméleti kérdése volt a kriptográfiának, ugyanakkor a gyakorlat szempontjából kevésbé volt fontos. Azonban az online póker megjelenésével, illetve egyéb kapcsolódó kriptográfiai kérdések miatt egyre inkább előtérbe került a protokollok gyakorlati alkalmazhatósága. A gyakorlatban azonban nem elhanyagolható szempont a protokoll gyorsasága, illetve műveletigénye. Crépeau protokollját használva 1994-ben 8 órába telt, hogy megkeverjenek egy paklit [12]. Ebből látható, hogy a protokoll annak ellenére, hogy látszólag egy teljes megoldást ad a fejben pókerezésre, sajnos mégsem tökéletes. Mindazonáltal a póker protokollok fejlődése szempontjából igen nagy jelentőséggel bír, és Crépeau után számos újabb megoldás született, melyek közül sok már gyakorlatban is alkalmazható, alacsonyabb műveletigényű.

Hivatkozások

[1] Shamir, A., Rivest R. and Adleman L., "Mental Poker", Mathematical Gardner, 1981, pp. 37-43.

[2] Bárány, I. and Füredi, Z., "Mental Poker with Three or More Player", Information and Control 59, 1983, pp. 84-93.

[3] Fortune, S. and Merrit, M., "Poker Protocol", Advances in Cyptology: Proc. of CRYPTO 84, G. R. Blakley and D. Chaum, eds., Lecture Notes in Computer Science 196, Springer-Verlag, Berlin, 1985, pp. 454-464.

[4] Brassard, G. and Crépeau C.,"Zero-Knowledge Simulation of Boolean Circuit", CRYPTO 86, 1987, pp. 223-233.

[5] Brassard G., Crépeau C. and Robert J.-M.,"All-or-Nothing Disclosure of Secret", CRYPTO 86, 1987, pp. 234-238.

[6] C. Crépeau, "A Secure Poker Protocol That Minimizes the Effect of Player Coalitions", Advances in Ctyptolgy: Proceedings of CRYPTO 85, H. C. Williams ed., Lecture Notes in Computer Science 218, Springer-Verlag, Berlin, 1986, pp 73-86.

[7] C. Crépeau, "A zero-knowledge poker protocol that achieves confidentiality of the players' strategy or how to achieve an electronic poker face", Advances in Cyptology: CRYPTO 86, A. M. Odlyzko, eds., Lecture Notes in Computer Science 263, Springer-Verlag, Berlin, 1987, pp. 239-247.

[8] J. Castella-Rocá, "Contributions to Mental Poker", PhD thesis, Universitat Autónoma de Barcelona, 2005.

[9] Galil, Z., S. Haber and M. Yun "A Private Interactive Test of a Boolean Predicate and Minimum-Knowledge public-Key Cryptosystem" Proceedings of the 26th Annual IEEE Symposium on the Foundations of Computer Science, 1985, pp. 360-371.

[10] Goldwasser, S. and Micali S.,"Probabilistic Encryptio", Journal of Computer and System Sciences 28, 1984, pp. 270-299.

[11] Goldwasser, S., S. Micali and C. Rackoff, "The knowledge Complexity of Interactive Proof-Systems" Proceedings of the 17th Annual ACM Symposium on thr Theory of Computing, 1985, pp. 291-304.

[12] J. Edwards. "Implementing electronic poker: A practical exercise in zero knowledge interactive proofs." Masters thesis, Department of Computer Science, University of Kentucky, 1994.