Adatbiztonság és kriptográfia jegyzőkönyv
3. gyakorlat

Csutak Balázs

2018. 05. 16

Tartalomjegyzék

1. Bevezető

Ezen a gyakorlaton folytattuk a blockchain technológián alapuló kriptovaluták működésének szimulációját és vizsgálatát egy PázMoney fantázianévre hallgató saját kriptovalután. A gyakorlat alatt sikerült befejezni a félév során épített kis rendszert, melynek segítségével a PPKE-ITK hallgatói így már utalhatnak egymásnak ebből az új virtuális fizetőeszközből.

A jegyzőkönyv első részében bemutatom a rendszer új elemeit, illetve válaszolok a gyakorlatvezető által feltett kérdésekre. A második rész az elvégzett feladatok dokumentációját tartalmazza.

Mivel az első jegyzőkönyvemben [2] a kriptovaluta elméleti működését nagyvonalakban ismertettem, itt most csak a feladatok elvégzése szempontjából megértendő új részletekre térek ki, időnként hivatkozva az említett munkára.

1.1. A rendszer új elemei

A gyakorlat során használhatóra egészítettük ki a PázMoney rendszerét, azaz felkészítettük a többi felhasználó által is épített közös blockchain feldolgozására, validálására, illetve ki is próbáltuk ennek közös használatát. A rendszer némi hozzáértés mellett működőképes, azaz a közös blockchain netes tárhelyről való letöltése után az tovább építható és később visszatölthető a tárhelyre.

1.2. Double spending

A double spending a kriptovaluták esetén felbukkanó visszaélési lehetőség, melynek lényege, hogy egy felhasználó ugyanazt a pénzt kétszer is elkölti. Ennek lehetséges módja, hogy ugyanarra a tranzakcióra, mint bemenetre két kimenő tranzakcióban is hivatkozik a felhasználó.

A mi kriptovalutánkban ennek megakadályozása a `validate_transaction` függvényen belül meghívott `get_transactions` függvény feladata. Ez azokkal a tranzakciókkal tér vissza (az érvényesnek elfogadott, paraméterként kapott blockchainben visszalépegetve és összegyűjtve ezeket), melyek kedvezményezettje, avagy visszajárójának jogosultja a vizsgált felhasználó (akinek a kimenő tranzakcióját éppen ellenőrizzük), és amelyek egyik blockchain-ben rögzített tranzakció bemeneteként sem szerepelnek. Ezután a validálás már csupán annak ellenőrzése, hogy a hivatkozott tranzakciók benne vannak-e ebben a listában.

1.3. Az első blokk

Mert a rendszer indulásakor még senkinek nincs pénze, így utalni sem tud. Az egyetlen tranzakció, ami a blokkban szerepelhet, az első blokkot létrehozó bányászt illető jutalom, melynek nincs bemenete, előre meghatározott fix összeget utal, és a bányász a kedvezményezett.

1.4. Láncolás hash értékekkel

Ennek lényegét már az első jegyzőkönyv [**?**] 1.4 fejezetében részletesen leírtam. Ismétlésképpen álljon itt is: az előző block hash-ének belefoglalása biztosítja a blokkok egymásra épülését, azaz hogy a lánc egyetlen blokkja sem cserélhető ki észrevétlenül. A hash függvény alapvető biztonsági elvárásai közé tartozik, hogy nem generálható olyan bemenet, amely egy előre ismert kimenetet adna. Ennek megfelelően, nem tudunk úgy módosítani egy blokkot, hogy annak hash értéke meg ne változna. Mivel ez a hash érték a ráépülő blokkban szerepel, a blokk megváltoztatása (rekurzív módon) az összes ráépülő blokkot érvénytelenítené.

1.5. Tranzakció érvénytelenítés újrabányászással

Jelen részben arra keressük a választ, hogy hogyan lehetséges egy 10 hosszú blokklánc 5 blokkjában lévő utalást törölni a rendszerből, azaz meg nem történtté tenni az utalást.

Ehhez arra van szükség, hogy a felhasználó kicserélje a lánc 5. blokkját olyanra, melyben nem szerepel a meg nem történtté tenni kívánt utalás. Az előző pontban leírtak szerint ezzel a lánc 6-10 blokkjai a hash értékek nem egyezése miatt érvényüket vesztenék, az új 5. blokk már csak egy 5 hosszú lánc utolsó eleme lenne. Mivel általában a rendszer felhasználói a leghosszabb láncot fogadják el érvényesnek, ahhoz, hogy ezt az új láncot a felhasználó általánosan elfogadottá tegye, még legalább 5 blokkot ki kéne bányásznia (feltéve, hogy közben az eredeti, módosítani kívánt láncot senki nem építi tovább). Mivel valós rendszerekben egyik felhasználó sem rendelkezik akkora számítási kapacitással, hogy 5 blokk lemaradást behozzon, ez a gyakorlatban kivitelezhetetlen.

1.6. A `Get_transactions` függvény

Ezen fügvény szerepét már az 1.2 részben részletesen leírtam: a `get_transactions` lényege, hogy paraméterként kapott felhasználó bejövő (ő a kedvezményezett, vagy a visszajáróra jogosult) és még el nem használt (azaz kimenő tranzakcióban nem hivatkozott) tranzakcióit az ugyancsak paraméterként kapott blokkláncból kigyűjti, és listába rendezve visszaadja.

2. Elvégzett feladatok

2.1. Blokklánc validálása

Első lépésben felkészítettük a rendszert egy külső forrásból kapott blokklánc validálására, hogy az esetleges csalásoknak áldozatul nem esve használhassuk a közös láncot. Ehhez lényegében kiegészítettük a 1.2 részben már említett `validate_chain` függvényt a beleírt kommentek alapján. Ennek működésének megértése a mellékletként csatolt kódban levő kommentek és a fent leírt elmélet mellett már triviális, így erre most itt nem térek ki.

2.2. Blokklánc betöltése

Következő lépésben letöltöttük a blokkláncot a közös tárhelyről **??**, és importáltuk ezt a már megírt `load_chain` függvénnyel. Ez az eljárás meghívja az előző lépésekben már összerakott ellenőrző funkciókat, így biztosak lehetünk benne, hogy csak érvényes lánc kerül betöltésre.

2.3. Pénzkeresés

A rendszerben két módon van lehetőségünk pénzhez jutni: egyrészt tovább építhetjük (akár üres, csak bányászjutalmat tartalmazó) blokkokkal a láncot, másrészt kaphatunk pénzt a többi felhasználótól utalással.

A gyakorlat során kaptam 8 egység PázMoney-t Bartha Barnabástól [4]. Ehhez ő a közös tárhelyre feltöltött publikus kulcsomat használta. Emellett két, csak bányászjutalmat tartalmazó blokk kibányászásából is nyertem 20 egységet.

2.4. Utalás

A feladat megoldásához Bartha Barnabásnak [4] utaltam vissza 15 egység PázMoney-t. Ehhez letöltöttem a publikus kulcsát a közös tárhelyről, és ennek segítségével a [**?**] jegyzőkönyvben leírtak szerint összeraktam a tranzakciót.

2.5. Tranzakció beillesztése a láncba

A tranzakció beillesztéséhez kibányásztam egy blokkot, melyben csak ez az egy tranzakció szerepelt. A blokkot beillesztettem a láncba.

2.6. Közzététel

Végül, az ily módon 3 blokkal hosszabb láncot visszatöltöttem a közös tárhelyre. A folyamatok ellenőrzéséhez az egyenlegemet minden lépés után lekérdeztem, és megfelelt az elvárásoknak.

3. Következtetések

A gyakorlat során a feladatkiírásban szereplő teendőket sikerült maradéktalanul végrehajtani, a kód működéséhez szükséges részeket megírni és kiegészíteni. A feladat végzése során kipróbáltam a kriptovalutai rendszerek működésének alaplépéseit, így megismerve és megértve a fontos részeket.

A jegyzőkönyv mellékletét képezi a gyakorlaton készített python kód. Ez, és jelen jegyzőkönyv az [1] weboldalon is elérhetők.

Hivatkozások

[1] Csutak Balázs weboldala, adatbiztonság és kriptográfia

`http://users.itk.ppke.hu/~csuba/Kripto`

[2] Csutak Balázs: Adatbiztonság és kriptográfia, 1. gyakorlat jegyzőkönyv (megtalálható a weboldalon)

[3] A kriptovaluta megosztott részeinek működtetéséhez használt közös tárhely: `https://drive.google.com/drive/folders/19eqqO2obmCFY-ykhZhMDruQS0KTYl3ry`

[4] Bartha Barnabás publikus kulcsa: `https://drive.google.com/drive/folders/16eQx4gfR37Ey8ivAjDHtVuTpHs7X1dZE`