SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Filip Maček

PAMETNI UGOVORI POMOĆU BLOCKCHAIN TEHNOLOGIJE

Diplomski rad

Voditelj rada:

prof.dr.sc. Luka Grubišić

Zagreb, kolovoz 2020.

Sadržaj

Uvod

Pametni ugovori naziv su za bilo kakav kompjutorski program ili transakcijski protokol koji automatski izvršava i kontrolira izvršavanje nekog ugovora i svih njegovih dijelova. Ideja je da su uvjeti ugovora kao i njegova etape izvršavanja potpuno prepušteni kompjutorskom kodu.

Glavna korisnost pametnih ugovora je u tome što se oni ne ovise o nikakvim sigurnim posrednicima ili arbitražama i svim troškovima koje takvo posredstvo nosi. Svi ostali problemi i prevare koji mogu nastati u izvršavanju ugovora također su izostavljeni jer se podrazumijeva da je kod tog ugovora napravljen u cilju obuhvaćanja svih mogućih događaja u stvarnom svijetu.

Još jedan važna odluka kod pametnih ugovora jest gdje će se taj program tj. ugovor izvršavati . Većina današnjih pametnih ugovora se izvršavaju na centralnim računalima onog jednog od sudionika tog ugovora. Ali idealni scenarij bi bio da se ugovor izvršava na serverima na kojima nitko od sudionika nema kontrola,a svi sudionica mogu vjerovati da se ugovor izvršiti po unaprijed određenim pravilima.

To dolazimo do blockchaina i jedan od mogućih rješenja gdje se taj programski kod pametnog ugovora može izvršavati. Trenutno je najveći blockchain i platforma za decentralizirano izvršavanje pametnih ugovora Ethereum.

U ovom diplomskom radu pokušat ćemo pokazati jednu moguću implementaciju i korisnost pametnih ugovora u stvarnom svijetu te ćemo pokušati pokazati kako se takva implementacija može natjecati sa sadašnjim modelom izvršavanja tih ugovora gdje velike korporacije ... TODO

Model i interakcija sa pametnim ugovorima na blockchainu je drugačiji nego današnje aplikacije koje u većini slučajeva komuniciraju sa svojim centralnim serverom pomoću API-ja (engl. Application programming interface).

Naša aplikacija je decentralizirana jer ne komunicira sa centralnim našim serverom nego se izvršava na svim kompjuterima na Ethereum blockchainu. Time naša aplikacija postaje „Dapp”, naziv koji se koristi za aplikacije tj. pametne ugovore koji se izvršavaju na Ethereum.

--- KRATAK OPIS APLIKACIJE I SVIH NJENIH DIJELOVA

Poglavlje 1

Uvod u blockchaina

1.1 Uvod

Kroz ovo poglavlje ukratko ćemo opisati glavne karakteristike i dizajn blockchain kao i njegovu povijest. Također značajna će nam biti i njegova motivacija i kakve pogodnosti blockchain donosi.

1.2 Povijest

Nakon izuma računala 1980-tih ljudi su se počeli pitati kakve su sve još inovacije moguće pomoću tog stroja. Ljudi još nisu shvaćali sve mogućnosti koje će im taj stroj omogućiti u budućnosti. Tada su još to bila izolirana računala koja su slabo bila povezana. Ali i već tada su neki istaknuti stručnjaci počeli uviđati kako da povežu ta računala u jedan koherentan sustav ili mrežu. *David Chaum* je bio pionir u tom području, i on je već 1982. u svojoj doktorskoj disertaciji [1] počeo razmišljati kako organizacije koje međusobno ne vjeruju jedna drugoj, mogu zajedničkim snagama izgraditi i održavati siguran računalni sistem kojem svi mogu vjerovati. Uvidio je veliku potrebu u privatnom i javnom sektoru za takve sisteme. Kriptografskim tehnikama bilo bi moguće takav sistem ostvariti praktičnim, gdje bi se spremljeni i podaci u opticaju mogli biti zaštićeni mehanizmom sefa (*eng. vault*)
Kasnije, 1991. *Stuart Haber i W. Scot. Sornetta* iznose prvi dizajn kriptografski osiguranog lanca blokova u kojem se ne može manipulirati vremenskim oznakama. Godinu dana nakon u svoj dizajn unose novu inovaciju binarno hash stablo (*eng. Merkle Tree).* Ono je omogućilo jednostavnu i sigurnu verifikaciju sadržaja velikih struktura podataka. Također je omogućilo da više certifikata dokumenata budu uključeni u blokove.

Iako je bilo mnoštvo akademski radove i napretka u kriptografiji do 2008. nije postojala nikakva veća i značajnija implementacija blockchain u svijetu. Tada je u jeku najveće financijske krize skupina ljudi ili jedan čovjek (što do danas nije poznato) pod pseudonimom Satoshi Nakamoto objavljuje članak *eng. Bitcoin: A Peer-to-Peer Electronic Cash System. [2]* U njemu iznosi osnovne teze i potrebe za neovisnim i decentraliziranim novčanim sustavom:

- Potpuno neovisan elektronički novac bi omogućio direktno slanje novca od jednog sudionika do drugog bez prisustva neke financijske institucija.

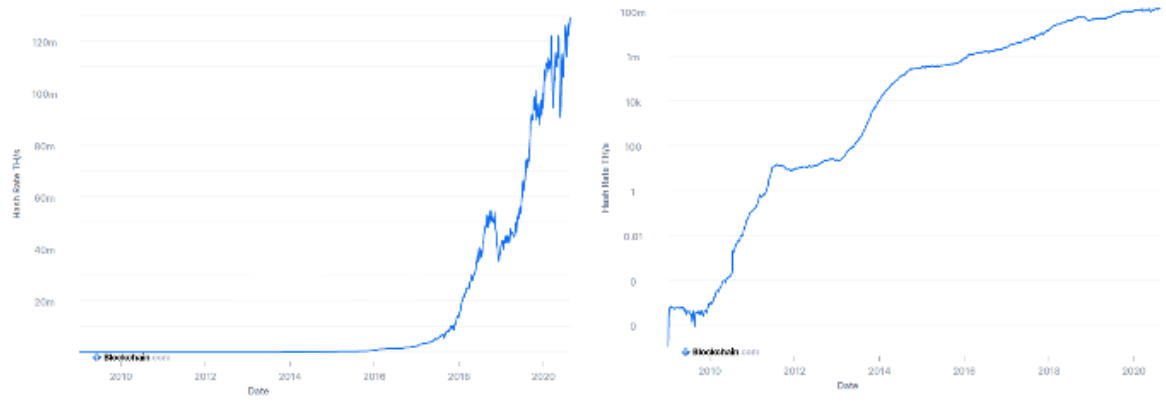
- Takav sustav koji se temelji na povjerenju u jedan centralni entitet ima svoje nedostatke. Neke od njih su mogućnost poništavanja transakcija i blokiranja računa sudionika te time izgon iz financijskog sustava i nemogućnosti sudjelovanja u financijskim uslugama.

- Transakcijski troškovi su povećani prisustvom posredovatelja

U tom članku je preporučeno elektronički naplatni sustav koji se temelji na kriptografiji, a ne na povjerenju. Dopuštajući da bilo koja dva sudionika sudjeluju u financijskoj transakciji bez potrebe da vjeruju jedan drugome.

Problem dvostruke potrošnje (*eng. double-spending problem*) već potrošenog novca riješen je na način da se umrežena računala na Bitcoin mreži slože oko toga da postoji jedan jedan opći vremenski lanac gdje bi se generirao kriptografski dokaz o kronološkom poretku transakcija.

Takav sustav je siguran dok većina pošteni čvorova na Bitcoin mreži kontrolira više procesorske moći nego napadačka skupina čvorova.Takav sustav je robustan u svojoj nestrukturiranoj jednostavnost



Slika 1.1: Hash rate Bitcoin mreže kroz vrijeme (linearna, logaritamska skala)

Na slici 1.1

Iako u početku odbačen i ne hvaljen od strane akademika i financijskih institucija koji su u njemu vidjeli pokušaj promijene starih vrijednosti. Mreža je aktivna i rastuća, te bez prestanka niti kakvih smetnji pouzdano izvršava svoju funkciju.

Kasnije 2014. godine izlazi članak i dizajn novog tipa blockchaina po imenu Ethereum. On je naslijedio opću funkcionalnost slanja novčane valute od Bitcoin protokola, ali je u svojoj implementaciji dodao još jednu jako važno novinu. A to je da čvorovi na Ethereum mreži još mogu i izvršavati posebno napisanu programsku logiku u programskom jeziku nazvanom Solidity. Time je otvoren put pametnim ugovorima te je i svaki takav programski isječak na Ethereumu i prozvan *eng. Smart Contract* . U budućim poglavljima ćemo pobliže opisati unutrašnji dizajn Ethreumu, osnovne značajke programskog jezika Solidity te kako je moguće da se programski kod paralelno izvršava na svim računalima.

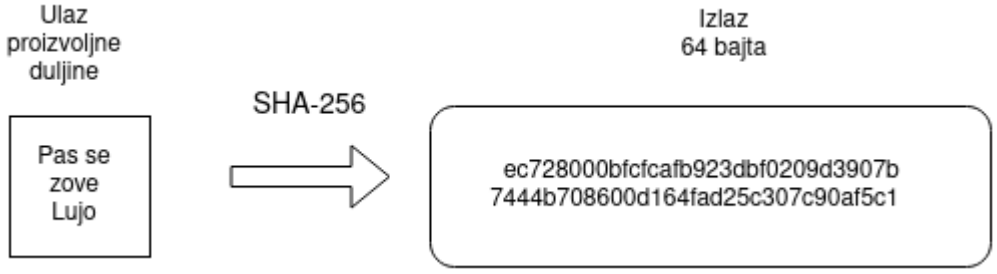
1.3 Osnovne karakteristike

U ovom poglavlju ćemo ukratko opisati glavne dijelove i karakteristike blockchaina fokusirajući se na inicijalni dizajn Bitcoin protokola.

Kriptografska hash funkcija

U pozadini blockchaina je posebna klasa kriptografske funkcije koji ima svojstvo da je ulazni nezavisna varijabla proizvoljne veličine, a rezultat te funkcije je fiksne veličine. Takva funkcija je jednosmjerna te joj se ne može izračunati inverz.

Time je jedini mogući algoritam za pronalaska ulaznog parametra za zadanu vrijednost funkciji , onaj u kojem se eng. brute-force algoritmom provjeravaju sve vrijednosti. U ovom slučaju koristi je posebna hash funkcija imenom SHA256 koja vraća 256 bitni broj koji je prikazan kao heksadecimalni broj sa 64 znamenke rade lakše čitljivosti.



Slika 1.2 SHA-256 hash funkcija

Transakcije

Na blockchainu elektronska valuta se definira kao lanac digitalnih kriptografski potpisa. Svaki sudionika transferira željenu količinu digitalnog novca koji on posjeduje tako da pomoću svog privatnog ključa potpisuje skup koji sadrži sljedeće informacije:

- Hash posljednje vlastite transakcije
- Javnih ključ primatelja te digitalne transakcije

Transakcije nisu privatne, već javne. Time je omogućeno da se svaka transakcija vidi i provjeri je li ispravna.

```
Input:
Previous tx: f5d8ee39a430901c91a5917b9f2dc19d6d1a0e9cea205b009ca73dd04470b9a6
Index: 0
scriptSig: 304502206e21798a42fae0e854281abd38bacd1aeed3ee3738d9e1446618c4571d10
90db022100e2ac980643b0b82c0e88ffdfec6b64e3e6ba35e7ba5fdd7d5d6cc8d25c6b241501

Output:
Value: 5000000000
scriptPubKey: OP_DUP OP_HASH160 404371705fa9bd789a2fcd52d2c580b65d35549d
OP_EQUALVERIFY OP_CHECKSIG
```

Slika 1.3: Primjer Bitcoin Transakcija

Ulaz *(engl. Input)* u transakciju je hash prijašnje transakcije. Ovdje se transakcija sastoji od jednog ulaznog podatka, ali moguće je i više. Time se iznosi tih ulaznih transakcija zbrajaju, te od ukupne sume oduzimaju transakcijski troškovi. *ScriptSig* je zajednička vrijednost javnog ključa i digitalnog potpisa pošiljatelja koji služi za provjeru da je pošiljatelj stvarno potpisao transakciju.

Izlaz (*eng. Output)* sadrži instrukcije za slanje digitalne valute. Vrijednost (*eng. value*) je broj Satoshija (1 BTC = 100,000,000 Satoshija) koja će biti izlazna vrijednost ove transakcija. U ovom slučaju to je 50 BTC.

Bitcoin koristi *eng. scripting* sustav za provjeru valjanosti transakcije. Svaka output transakcije sadrži *eng. scriptPubKey* koji je „blokirajuća" skripta koja zapravo zaključava output ove transakcije te čeka vlasnika tog digitalnog novca da svojom „odblokirajućom" skriptom *scriptSig* nastavi niz i napravi novu transakciju. Detaljni ovog procesa su izvan ovog diplomskog rada. Ali ovim cijelim mehanizmom se postiže drugačiji dizajn digitalnog novca u odnosu na papirnati državni novac. Ovdje sudionici nikad ne „posjeduju" u doslovnom smislu elektronički novac, nego imaju prava (eng. rights) da prenesu digitalnih novac sa jedne adrese na drugu. Time je transakcija zapravo prijenos prava sa pošiljatelja na primatelja.

Blokovi

Povezivanje blokova

Algoritam konsenzusa

Sudionici

Motivacija sudionika na protoklu

```
class UserListAdapter (private var users: List<User>,private val navController:
NavController):ListAdapter<User,UserListAdapter.ViewHolder>(UserItemDiffCallback()),KoinComponent{
    // Context global var
    private lateinit var context:Context
    private val userRepository:UserRepository by inject()
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {
        val view= LayoutInflater.from(parent.context).inflate(R.layout.user_row,parent,false)
        context = parent.context
        return ViewHolder(view)
    }


    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        holder.username.text=users[position].username
        holder.address.text=users[position].address


        holder.connect_button.setOnClickListener {
            MaterialDialog(context).show {
                input( hint = "Type your password", inputType = InputType.TYPE_CLASS_TEXT ) { dialog,text
->
                    val password = HashUtils.doubleHash(text.toString())
                    val username=users[position].username
                    //Check if password is valid
                    val user:User = userRepository.getUserByUsername(username)
                    val bundle_username = bundleOf("username" to username)
                    if(user.password == password) {
                        navController.navigate(R.id.action_user_list_to_dashboard,bundle_username)
                    }else {
                        val toast = Toast.makeText(context,"Incorrect password",Toast.LENGTH_SHORT)
                        val toastView= toast.view
                        toastView.setBackgroundColor(Color.RED)
                        toast.show()
                    }
                }
                positiveButton(R.string.login_button_text)
            }
        }
```

Bibliografija

[1] David Lee Chaum, *Computer Systems Established, Maintained and Trusted by Mutually Suspicious Groups*, *(*1982.) University of California, Berkley

[2] Satoshi Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, (2008.) https://bitcoin.org/bitcoin.pdf