ԵՐԵՎԱՆԻ ՊԵՏԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԿԻՐԱՌԱԿԱՆ ՄԱԹԵՄԱՏԻԿԱՅԻ ՖԱԿՈՒԼՏԵՏ

ԾՐԱԳՐԱՎՈՐՄԱՆ ԵՎ ԻՆՖՈՐՄԱՑԻՈՆ ՏԵԽՆՈԼՈԳԻԱՆԵՐԻ ԱՄԲԻՈՆ

ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԿԻՐԱՌԱԿԱՆ ՄԱԹԵՄԱՏԻԿԱՅԻ ԿՐԹԱԿԱՆ ԾՐԱԳԻՐ

ՏԻԳՐԱՆ ԳԱԼՍՏՅԱՆ ԵՐՎԱՆԴԻ

ԱՎԱՐՏԱԿԱՆ ԱՇԽԱՏԱՆՔ

ԲԼՈԿՉԵՅՆԻ ԿԻՐԱՌՈՒԹՅՈՒՆԸ ԽԱՂԵՐԻ ՈԼՈՐՏՈՒՄ

«Մաթեմատիկա: Համակարգչային գիտություններ» մասնագիտությամբ և Ինֆորմատիկայի և կիրառական մաթեմատիկայի բակալավրի որակավորման աստիճանի հայցման համար

ԵՐԵՎԱՆ 2018

Ուսանող՝ _____

ստորագրություն

Գալստյան Տիգրան

ազգանուն, անուն

Ղեկավար՝ _____

ստորագրություն

Ֆ.Մ.գ.թ., ասիստենտ Ս. Ա. Խաչատրյան

գիտ. աստիճան, կոչում, ազգանուն, անուն

«Թույլատրել պաշտպանության»

Ամբիոնի վարիչ՝ _____

ստորագրություն

Ֆ.Մ.գ.դ., պրոֆեսոր Ս.Ա. Նիգիյան

գիտ. աստիճան, կոչում, ազգանուն, անուն

« ____» _____20____թ

ՀԱՄԱՌՈՏԱԳԻՐ

«ԲԼՈԿՉԵՅՆԻ ԿԻՐԱՌՈՒԹՅՈՒՆԸ ԽԱՂԵՐԻ ՈԼՈՐՏՈՒՄ»

“ПРИМЕНЕНИЕ БЛОКЧЕЙНА В ИГРАХ”

“THE APPLICATION OF BLOCKCHAIN IN GAMES”

Ուսումնասիրվել են՝

- մի շարք էլեկտրոնային արժույթներ, որոնցից խորությամբ ուսումնասիրվել է Էթերեումը(Ethereum), որը ոչ միայն էլեկտրոնային արժույթ է այլ նաեւ բաշխված համակարգիչների վրա տեղադրված օպերացիոն համակարգ
- կենտրոնացված եւ ապակենտրոնացված համակարգեր
- դետերմինիստիկ պատահական մեծության ստացման ալգորիթմներ
- Angular 5 Framework-ը
- Ionic SDK-ը
- web3.js գրադարանը
- truffle-hdwallet-provider գրադարանը
- Սոլիդիթի (Solidity) ծրագրավորման լեզուն
- Էթերեում վիրտուալ մեքենան (EVM)

Ստեղծվել են՝

- «Հաջողակը» անվամբ ապակենտրոնացված կիրառություն (DAPP)
- iOS եւ Android ՕՀ-երի համար կիրառություններ, որոնք թույլ կտան օգտագործողին գրաֆիկական ինտերֆեյսի միջոցով փոխազդել (interact) Էթերեումի բլոկչեյնի վրա գտնվող «Հաջողակը» ապակենտրոնացված կիրառության վրա

ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

ՆԵՐԱԾՈՒԹՅՈՒՆ

2009թ.-ին ստեղծվել է առաջին ապակենտրոնացված թվային արտարժույթը՝ Բիթքոինը [1], որն աշխատում է բլոկչեյն տեխնոլոգիայի հիման վրա: 2015 թ.-ին թողարկված Էթերեումը (Ethereum) [3], որը ոչ միայն թվային արտարժույթ է, այլ նաեւ մի հարթակ, որում կարող են ստեղծվել զանազան ապակենտրոնացված կիրառություններ (DAPP) եւ ապակենտրոնացված ինքնավար կազմակերպություններ (DAO), ապահովում է հարյուր տոկոսանոց թափանցիկություն:

Բլոկչեյնի թափանցիկությունը դրա հիման վրա հիմնված ապակենտրոնացված կիրառություններին դարձնում է վստահելի: Այդ իսկ պատճառով, այն դառնում է կարեւոր մի շարք ոլորտներում, մասնավորապես խաղերում, որոնցում խաղացողները գործ են ունենում իրական գումարների հետ:

Խնդրի դրվածքը. Ուսումնասիրել բլոկչեյն տեխնոլոգիան եւ օգտագործելով այն ստեղծել լոտո խաղը ու պարզել բլոկչեյնի արդյունավետությունը խաղերի ոլորտում:

Արդիականությունը. Ներկայումս էլէկտրոնային արժույթներով հետաքրքրվածությունը շատ բարձր մակարդակի վրա է գտնվում: Դրան զուգահեռ առկա է մեծ աուդիտորիա սմարթֆոնների ու համակարգիչների քանակական եւ որակական աճի շնորհիվ, որը խաղերի մեծ պահանջարկ ունի: Այսպիսով, խաղը որը հիմնված կլինի բլոկչեյն տեխնոլոգիայի վրա, անմիջապես կգրավի այդ աուդիտորիայի ուշադրությունը:

Լուծումը. Օգտագործելով Էթերեում հարթակը իրականացվել է (implemented) «Հաջողակը» խաղը: Խաղը բաղկացած է երկու կոմպոնենտներից. Էթերեում վիրտուալ մեքենայի (EVM) վրա գործող մասից՝ «բեք-էնդ» (back-end), եւ iOS ու Android ՕՀ-երի համար նախատեսված հիբրիդ կիրառություններից՝ ֆրոնտ-էնդ (front-end): «Բեք-էնդում» օգտագործվել է Սոլիդիթի (Solidity) լեզուն, ստեղծված 2014թ.-ին Գեյվին Վուդի կողմից: Այն բարձր մակարդակի լեզու է՝ նախատեսված խելացի-պայմանագրեր (smart-contract) ստեղծելու համար: Ֆրոնտ-էնդում օգտագործվել են Իոնիկ (Ionic) SDK-ը, Angular շրջանակը (framework) եւ TypeScript ծրագրավորման՝ լեզուն հիբրիդ կիրառություն ստանալու համար, ինչպես նաեւ

«web3.js» գրադարանը՝ կիրառությունից Էթերեումի բլոկչեյնին համասեռ ցանցով (P2P) միանալու եւ փոխազդելու համար:

ԲԼՈԿՉԵՅՆ

Անժխտելի է այն փաստը, որ բլոկչեյնը հնարամիտ գյուտ է, որը հայտնի է «Սատոշի Նակամոտո» կեղծանվամբ եւ հանդիսանում է մի խումբ մարդկանց կամ անհատի մտքի արգասիքը: Սակայն ստեղծվելուց հետո, այն աստիճանաբար վերաճել է՝ դառնալով մի շատ ավելի մեծ բան եւ յուրաքանչյուրի մոտ ծագող մի շատ կարեւոր, այն՝ ի՞նչ է բլոկչեյնը:

Ի՞ՆՉ Է ԲԼՈԿՉԵՅՆԸ

Բլոկչեյնը *փաստերի* «գրքույկ» է, կրկնօրինակված տարբեր համակարգիչներում հավաքված լինելով համասեռ ցանցում (P2P network) [6]: Փաստերը կարող են ամեն ինչ լինել, սկսած դրամական փոխանցումների արձանագրություններից մինչեւ կոնտենտի ստորագրությունը: Ցանցի անդամները անհայտ անհատներ են, որոնք կոչվում են հանգույցներ: Ցանցի ներսում ուղարկողի եւ ստացողի անվտանգ ճանաչումն ու հաղորդակցությունը տեղի է ունենում օգտվելով կրիպտոգրաֆիկ ֆունկցիաներից: Երբ հանգույցը փաստ է ուզում ավելացնել «գրքույկին», համաձայնեցում (consensus) է ձեւավորվում ցանցում, որպեզի որոշվի թե «գրքույկի» որ մասում պետք է ավելացվի փաստը. այդ համաձայնեցումը կոչվում է *բլոկ* [4]:

Բլոկչեյնը թույլ է տալիս ցանցի ոչ վստահված անդամների միջեւ ապահով կերպով կիսել եւ/կամ մշակել տվյալներ: Տվյալը կարող է լինել ցանկացած ինֆորմացիա:

Տեխնիկական տեսանկյունից, բլոկչեյնը նորարարություն է՝ հիմնված երեք հասկացությունների վրա. համասեռ ցանցերի (P2P networks), հանրային-բանալու (public-key) կրիպտոգրաֆիայի եւ պատահական մաթեմատիկական մարտահրավերների որոշումից բաղկացած բաշխված համաձայնեցումից

(consensus): Այս երեք հասկացություններից ոչ մեկն էլ նորույթ չէ: Այդ երեքի միաձուլումը հաշվողական աշխարհում բեկում է մտցնում:

Բլոկչեյնին կարելի է նայել որպես տվյալների շտեմարան, պատճենված եւ համաժամեցված այնքան անգամ, ինչքան հանգույց կա, կամ, որպես սուպերհամակարգիչ՝ կազմված ցանցի հանգույցներում առկա կենտրոնական կամ գրաֆիկական մշակիչ սարքերի (CPUs/GPUs) համադրությունից: Այս սուպերհամակարգիչը կարող ենք օգտագործել տվյալներ պահելու եւ մշակելու համար, ինչպես կանեինք օրինակ՝ REST API-ի դեպքում, բացի դրանից, անհրաժեշտություն չկա ունենալու սեփական բեք-էնդ հատվածը:

Բլոկչեյնի ամենահայտնի կիրառությունը դեռ Բիթքոինն է:

Բլոկչեյն տեխնոլոգիան եւ ինտրիգային է, եւ հուզիչ: Կարո՞ղ է բլոկչեյնը լինել այն հեղափոխությունը, որը կանխատեսել են գուրուները: Դա տեսանելի կլինի հաջորդ տասնամյակում:

ՀԱՄԱՁԱՅՆՈՒԹՅԱՆ (CONSENSUS) ԳԱԼՈՒ ՓՈՒԼԸ. ՄԱՅՆԻՆԳ

Բլոկչեյն տեխնոլոգիայում համաձայնության (consensus) գալու գործընթացի ամենա-տարածված երկու տեսակներն են՝ աշխատանքի ապացույցը (proof of work) եւ բաժնետեր լինելու ապացույցը (proof of stake): Կխոսենք միայն աշխատանքի ապացույցի մասին:

Աշխատանքի ապացույցը մի արձանագրություն (protocol) է, որի հիմնական նպատակը այնպիսի կիբեր հարձակումների կանխումն է, ինպես, օրինակ՝ ծառայության բաշխված մերժումն է (DDoS), որը նպատակաուղղված է կեղծ հարցումների միջոցով համակարգչային ռեսուրսներ սպառելուն [9]:

Աշխատանքի ապացույց կոնցեպտը հավանաբար ամենահանճարեղ մտահաղացումն էր Նակամոտոյի Բիթքոինի զեկույցում, հրատարակված 2008թ.-ին, քանի որ այն թույլ է տալիս առանց վստահության անհրաժեշտության բաշխված համաձայնագիր ունենալու: Առանց վստահության անհրաժեշտության բաշխված համաձայնագիր նշանակում է, որ, եթե ուզում ես ինչ որ մեկին փող ուղարկել եւ/կամ

ստանալ, դու ստիպված չես վստահել երրորդ կողմին: Ավանդական վճարման մեթոդներ (օրինակ՝ Visa, Mastercard, PayPal, բանկեր) օգտագործելիս ստիպված ես վստահել երրորդ կողմին գումարը փոխանցելեւ հարցը: Հանգույցները ունեն իրենց անձնական «գրքույկը» որտեղ պահում են փոխանցումների եւ հաշվեկշռի պատմությունը ամեն հաշվի համար: Օրինակ՝ եթե Արամը $100 ուղարկի Ռոբին, վստահված երրորդ կողմը Արամի հաշվից կգանձի նշված գումարը եւ կավելացնի Ռոբի հաշվին: Այսպիսով, նրանք ստիպված են վստահել երրորդ կողմին, որ նա ճիշտ կկատարի այդ պրոցեսը:

Էլեկտրոնային արժույթների դեպքում բոլոր հանգույցներն ունեն «գրքույկը», այսպիսով, ոչ ոք ստիպված չի վստահել երրորդ կողմին, քանի որ ցանկացածը կարող է ուղղակիորեն հաստատել գրված ինֆորմացիան:

Այսպիսով, մայնինգի արդյունքում բլոկ է ավելացվում բլոկչեյնին:

Մայնինգը ունի երկու նպատակ.

1. Հաստատել գործարքի լեգիտիմությունը կամ խուսափել, այսպես կոչված կրկնակի-ծախսումից
2. Ստեղծել նոր էլեկտրոնային արժույթ՝ մայներներին կատարած աշխատանքի դիմաց վճարելու համար:

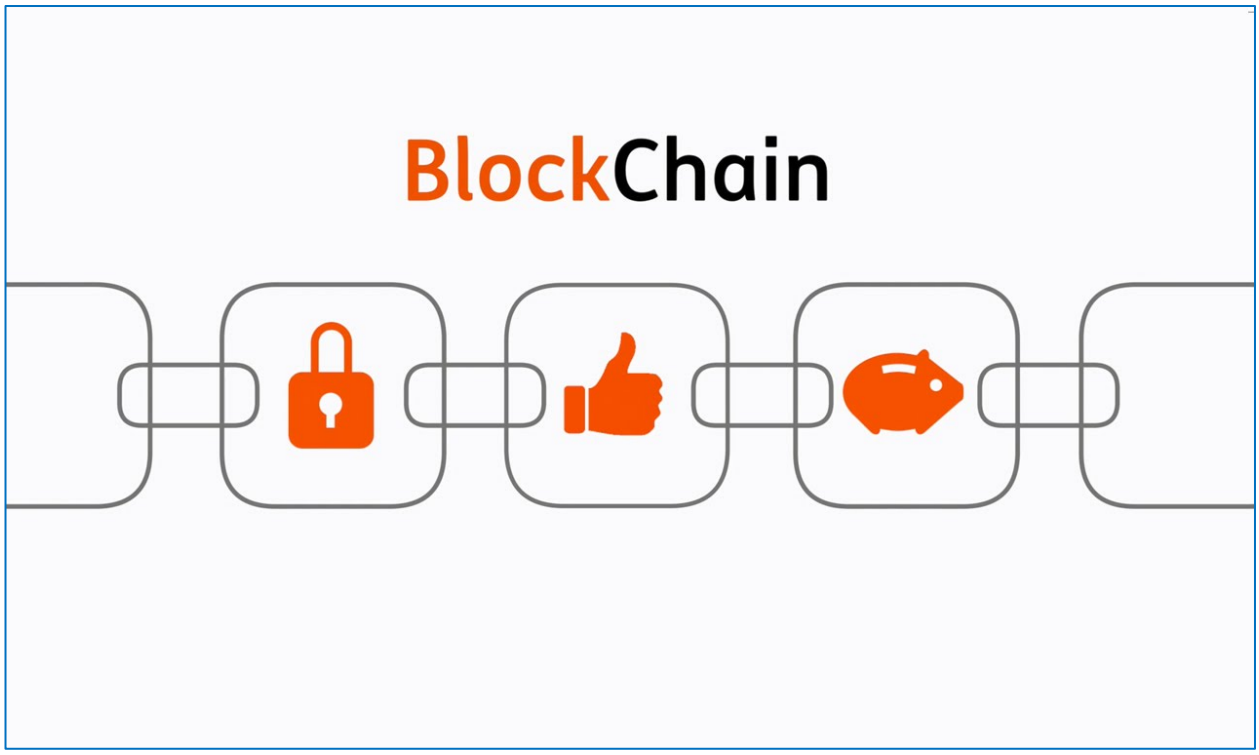
Գործարք կատարելիս տեղի են ունենում հետեւյալ պրոցեսները.

- Գործարքները միավորվում են բլոկի մեջ
- Մայներները հաստատում են ամեն բլոկի մեջ եղած գործարքների լեգիտիմությունը (դրա կատարման համար, մայներներ պետք է լուծեն «մաթեմատիկական հանելուկներ», հայտնի որպես աշխատանքի ապացույց)
- Ամեն բլոկի համար պարգեւ է տրվում այն մայներին, ով առաջինն է լուծում խնդիրները
- Հաստատված գործարքները պահվում են բլոկչեյնում

Այս «մաթեմատիկականական հանելուկները» մի հատկություն ունեն. *ասիմետրիկություն*: Ցանցի բոլոր մայներները մրցում են այս մաթեմատիկական խնդրի լուծումը առաջինը գտնելու համար:

Երբ մայները գտնում է ճիշտ լուծումը, այն հայտարարում է ամբողջ ցանցին՝ ավելացնում է բլոկը ցանցին, եւ ստանում է պարգեւ էլեկտրոնային արժույթով:

Տեխնիկական տեսանկյունից, մայնինգը հակադարձ հեշավորման օպերացիա է, որը որոշում է մի թիվ:



Նկ. 1.1. Բլոկչեյնի գրաֆիկական տեսքի օրինակ

«ՀԱՋՈՂԱԿԸ» ԽԱՂԸ

Էթերեում հարթակի թողարկումից հետո, սկսվում է դրա կիրառումը նաեւ խաղերում: Էթերեումը թույլ է տալիս ստեղծել հարյուր տոկոսանոց թափանցիկ խաղեր, որոնք կարող են գրավել խաղերի շատ սիրահարների: Ամենաարագը օնլայն խաղատները ինտեգրեցին, արդեն իսկ առկա խաղերը, էթերեում հարթակի հետ:

ԽԱՂԻ ՆԿԱՐԱԳԻՐԸ

Խաղին կարող է մասնակցել x քանակի խաղացող: Մասնակցելու համար խաղացողը տոմս է գնում, y ETH գնով, եւ գցում արկղի մեջ: Երբ վաճառվում է x քանակությամբ տոմս, պատահականորեն արկղից հանվում է մեկ տոմս եւ շահողին է փոխանցվում x * y ETH՝ էլեկտրոնային արժույթ:

ԽԱՂԻ ԳՐԱՖԻԿԱԿԱՆ ԻՆՏԵՐՖԵՅՍԸ

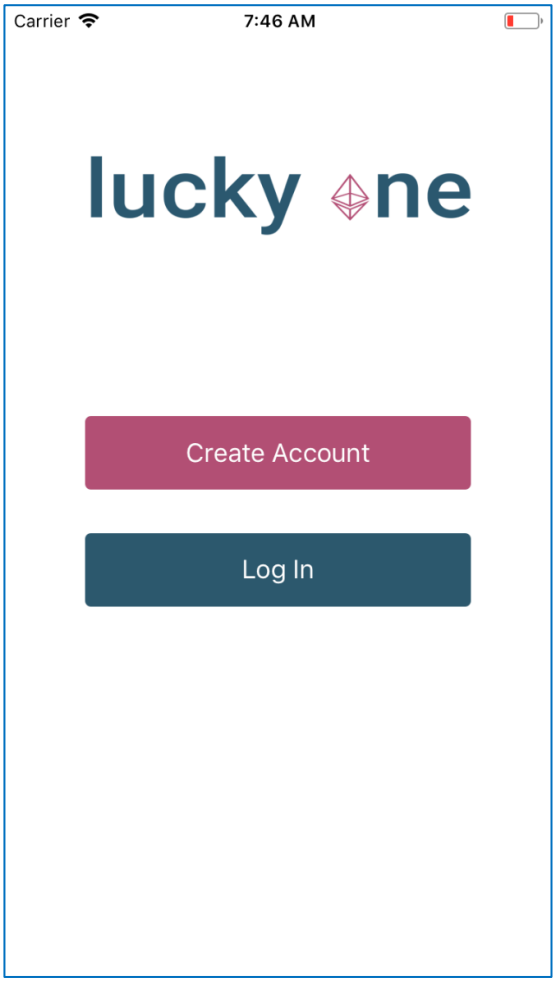
Մինչեւ գրաֆիկական ինտերֆեյսին անցնելը պետք է նշել, որ խաղը հնարավոր է խաղալ նաեւ առանց գրաֆիկական ինտերֆեյսի: Ունենալով ՄետաՄասկ [1] ընդլայնումը (extension) բրաուզերի վրա, remix.ethereum.org IDE-ի օգնությամբ կարելի է գնել տոմս գործարքի միջոցով:

Վերոհիշյալ կամ գրաֆիկական տարբերակով տոմս գնելուց հետո, խաղացողը կարող է նայել իր կատարած գործարքը Բլոկչեյնի վրա: Այդ հարցում կօգնի արդեն իսկ հայտնի ԷթերՍկան[2] (EtherScan) բլոկ-ուսումնասիրողը: Ընդամենը անհրաժեշտ է խաղացողի հանրային բանալին (public-key) փնտրել ԷթերՍկանում:

[1] ՄետաՄասկը թույլ է տալիս փոխազդել ապակենտրոնացված կիրառությունների հետ: Ավելին՝ metamask.io

[2] Էթերեումի բլոկչեյնի հանրային լինելը յուրաքանչյուրին թույլ է տալիս ուսումնասիրել ցանկացած բլոկ: Ավելին՝ etherscan.io

Նկ. 2.1. ԷթերՍկանի կայքի մի հատված



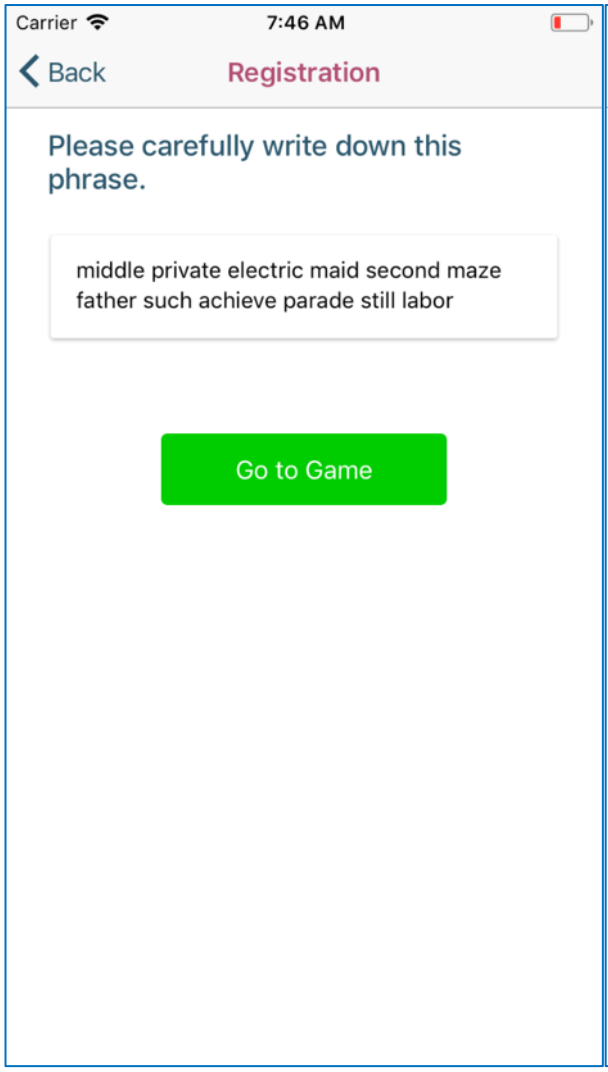
Նկ. 2.2. Կիրառության մուտքի էջ

«Հաջողակը» խաղը խաղալու համար անհրաժեշտ է ունենալ մնեմոնիկով (mnemonic) մուտք գործվող Էթերեումի հաշիվ, կամ այն ստեղծել խաղի տրամադրած գրաֆիկական ինտերֆեյսի միջոցով:

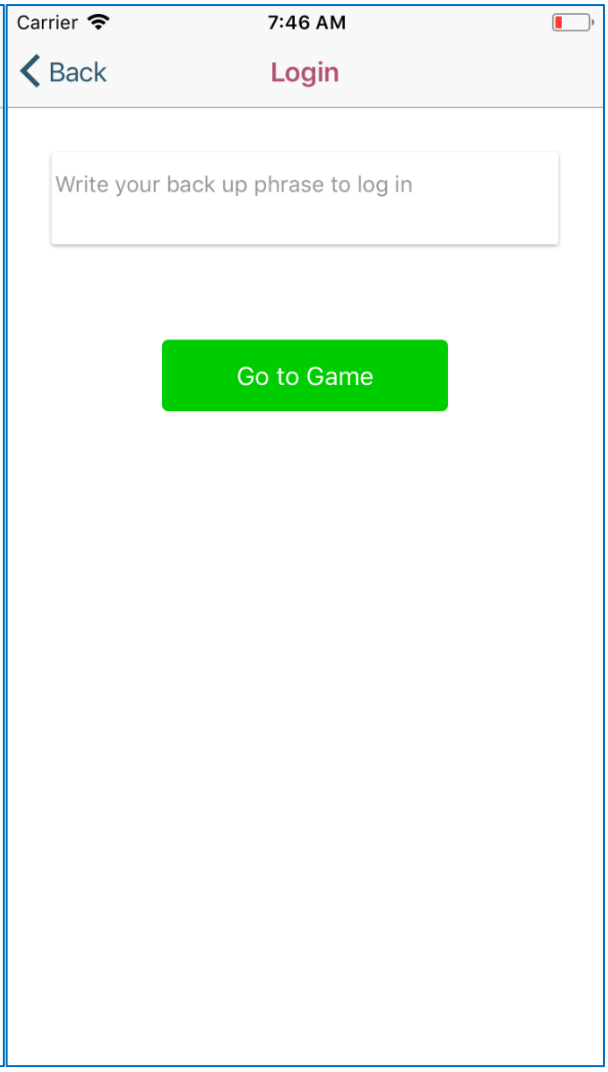
Խաղի առաջին՝ **մուտք գործելու էջում** *(տես Նկ. 2.2.-ը)* կա երկու կոճակ. «ՍՏԵՂԾԵԼ ՀԱՇԻՎ» (Create Account) եւ «ՄՈՒՏՔ ԳՈՐԾԵԼ» (Log In): «ՍՏԵՂԾԵԼ ՀԱՇԻՎ»-ը խաղացողի համար գեներացնում է 12 բառից կազմված մնեմոնիկ, որը պետք է հիշել, ինչպես օրինակ՝ գաղտնաբառը, եւ ապահովել նրա գաղտնիությունը, քանի որ ցանկացածը կարող կառավարել հաշվի վրա առկա միջոցները մնեմոնիկ արտահայտությունը ունենալու դեպքում: «ՄՈՒՏՔ ԳՈՐԾԵԼ» կոճակը տանում է մուտք գործելու համար նախատեսված էջ, որտեղ խաղացողից կպահանջվի արդեն ունեցած մնեմոնիկ արտահայտությունը: Մնեմոնիկ արտահայտությամբ մուտք գործվող հաշիվները շատ ավելի ապահով են: Այն օգտագործվում է անձնական բանալի (private-key) գեներացնելու համար, որը անհրաժեշտ է բլոկչեյնում գործարքներ ստորագրելիս: Տոմս գնելն արդեն իսկ գործարք է:

Հաշվի ստեղծման էջում *(տես Նկ. 2.3.-ը)* խաղացողին տրվում է նոր գեներացված մնեմոնիկ արտահայտություն: «Գնալ դեպի խաղը» (Go to Game) կոճակը սեղմելուց բացվում է խաղի հիմնական մասը:

Մուտքի էջում *(տես Նկ. 2.4.-ը)* կա մնեմոնիկ արտահայտության համար նախատեսված մուտքագրման դաշտ եւ «Գնալ դեպի խաղը» կոճակը, որը կատարում է վերոհիշյալ ֆունկցիոնալը:



Նկ. 2.3. Հաշիվ ստեղծելու էջ



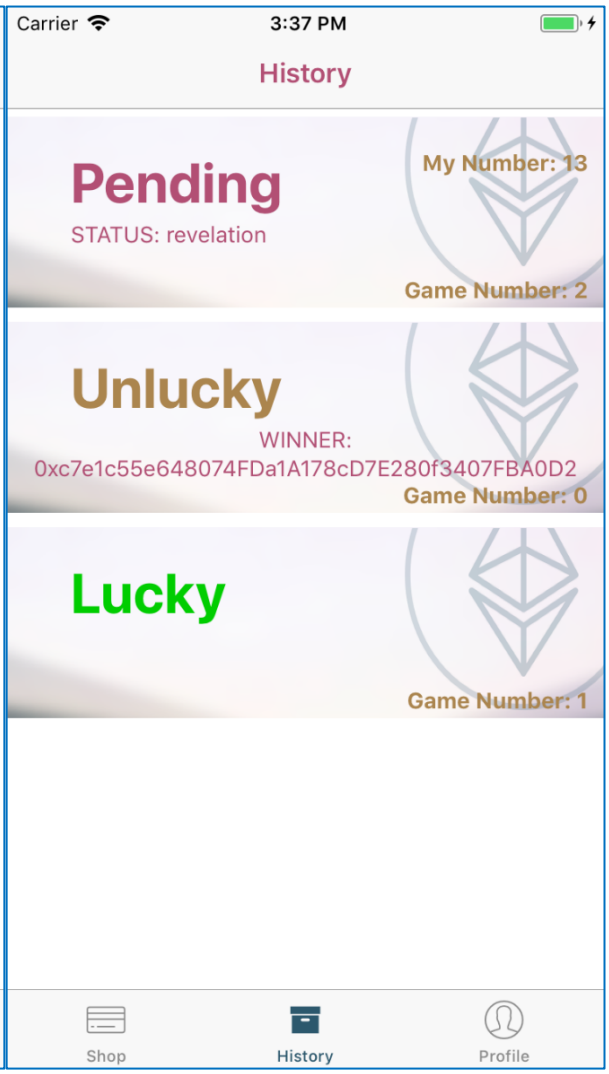
Նկ. 2.4. Հաշիվ մուտք գործելու էջ

Հիմնական մասը բաղկացած է երեք էջերից՝ տոմս գնելու էջ, պատմության էջ, պրոֆիլի էջ:

Պրոֆիլի էջում *(տես Նկ. 2.5.-ը)* խաղացողը կարող է գտնել իր հաշվեկշիռը՝ վերեւի մասում, իր հանրային հասցեն (public-key) եւ հաշվից դուրս գալու կոճակը: Խաղացողը կարող է իր հաշիվը լիցքավորել օգտվելով օնլայն փոխանակման

կետերից (exchange): Օրինակ՝ eToro-ից, HitBTC-ից, YObit-ից եւ այլն: Հաշվեկշիռը ցույց է տրվում Էթերեում էլեկտրոնային արժույթով (ETH):

Նկ. 2.5. Պրոֆիլի էջ

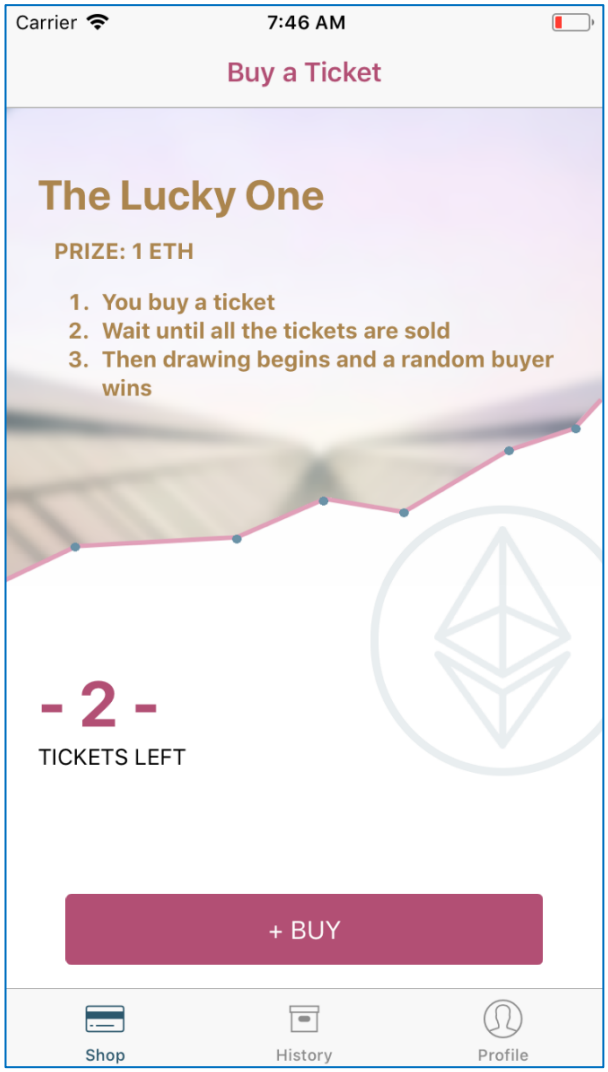


Նկ. 2.6. Պատմության էջ

Պատմության էջի *(տես Նկ. 2.6.-ը)* առաջին քարտում երեւում են ընթացիկ խաղի եւ տոմսի կարգավիճակները: Ընթացիկ քարտում տոմսը գտնվում է սպասողական (pending) վիճակում, որից հետո կա երկու ելք. հաջող (lucky) կամ անհաջող (unlucky): Անդրադառնանք տոմսի վրայի երկու տվյալներին: Իմ համարը (my number) ապակենտրոնացված կիրառությունում խաղացողի համարն է: DAPP-ում ընտրվում է դետերմինիստիկ պատահական թիվ (ավելի մանրամասն հաջորդ էջում) եւ այդ համարի խաղացողին է փոխանցվում մրցանակային ֆոնդը: «STATUS»-ը՝ խաղի կարգավիճակը, ցույց է տալիս թե ինչ փուլում է գտնվում խաղը: Կա երկու փուլ. տոմսերի վաճառք (selling tickets) եւ բացահայտում (revelation):

Տոմսերի վաճառք կարգավիճակը նշանակում է, որ դեռ չվաճառված տոմսեր են առկա: Բոլոր տոմսերի վաճառվելուց հետո կարգավիճակը փոխվում է բացահայտում (revelation) կարգավիճակի: Այս կարգավիճակում մոբայլում գեներացված պատահական թիվը ԲԱՑԱՀԱՅՏ կերպով ուղարկվում է ապակենտրոնացված կիրառությանը: Մոբայլում գեներացվող պատահական թվի մասին ավելի մանրամասն կխոսվի «Խանութի» էջը ներկայացնելիս:

Խաղի ավարտից հետո, ինչպես արդեն նշվեց, տոմսը դառնում է հաջողակ կամ անհաջողակ: Հաջողակ տոմսի վրա գրված է խաղի համարը եւ «Lucky» արտահայտությունը: Անհաջողակ տոմսի վրա գրված է խաղի համարը, «Unlucky» արտահայտությունը եւ հաղթող խաղացողի հանրային բանալուն (public-key): Բոլոր խաղացողները կարող են ստուգել թե արդյոք նշված՝ հաղթող, հանրային բանալուն կատարվել է մրցանակի փոխանցում: Դա կարելի է անել, արդեն նշված, ԷթերՍկանի միջոցով:



Նկ.2.7. Խանութի էջ

Խանութի էջում *(տես Նկ. 2.7.-ը)* կարելի տեսնել խաղի կանոնները եւ առկա տոմսերի քանակը: Ներքեւի հատվածում կա գնել (buy) կոճակը: Գնել կոճակը սեղմելիս տեղի են ունենում հետեւյալ գործողու-թյունները.

1. Մոբայլում գեներացվում է պատահա-կան թիվ եւ պահվում սարքի հիշողության մեջ
2. Թիվը եւ խաղացողի հանրային բանալին միասին հեշավորվում են
3. Ստացված տողային արտահայ-տությունը տոմսի արժեքի չափով էթերեումի հետ ուղարկվում է ապակենտրոնացած կիրառությանը
4. Ծրագիրը սպասում է բոլոր տոմսերի վաճառվելուն

5. Վաճառվելուց հետո ապակեն-տըրոնացված կիրառությանը մոբայլ կիրառությունից ուղարվում է սարքի հիշողությունում պահված պատահական մեծությունը

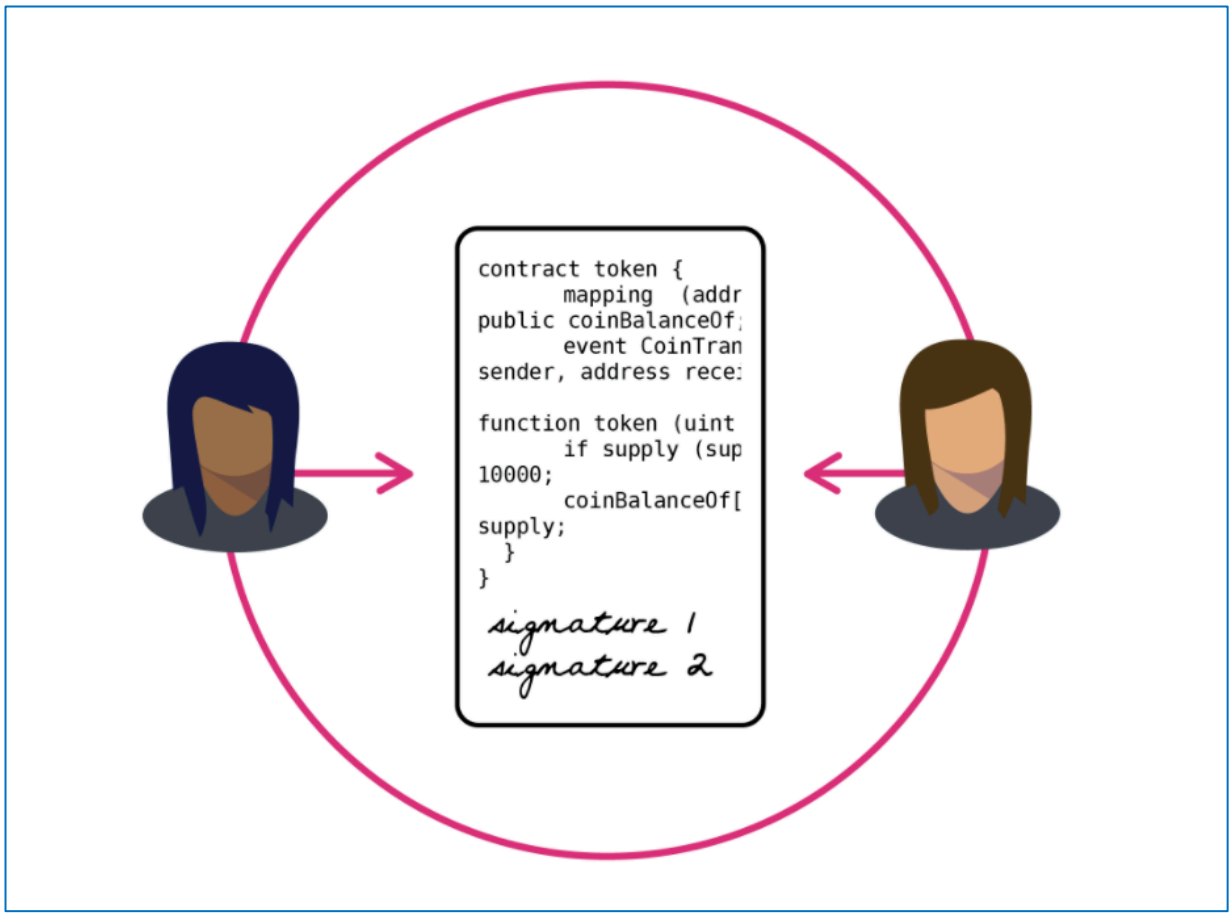
Մոբայլում գեներացված եւ պահված պատահական մեծությունը կօգտագործվի ապակենտրոնացված կիրառությունում՝ դետերմինիստիկ պատահական թիվ ստանալու համար:

Գնել ֆունկցիոնալը կաշխատի միայն հաշվեկշռի վրա տոմսի գնին մեծ կամ հավասար էլեկտրոնային արժույթ ունենալու դեպքում:

Մոբայլ կիրառության կոդը տեղադրված է Գիթհաբում (Github) հետեւյալ հղմամբ՝ https://github.com/Galti/bachelor_thesis/tree/master/mobile_app

«ՀԱՋՈՂԱԿԸ» ԱՊԱԿԵՆՏՐՈՆԱՑՎԱԾ ԿԻՐԱՌՈՒԹՅՈՒՆԸ

Խելացի պայմանագիրը համակարգչային արձանագրություն է նախատեսված թվայնորեն հեշտացնելու, հաստատելու կամ հզորացնելու պայմանագրի բանակցություններն ու կատարումը: Դրանք թույլ են տալիս, առանց երրորդ կողմի, կատարել վստահելի գործարքներ:



Նկ.2.8. Խելացի պայմանագիր երկու կողմի միջեւ

Խելացի պայմանագրեր գրելու համար, կան մի քանի լեզուներ՝

1. Մուտան (Mutan)՝ նման ԳՈ (GO) լեզվին, արգելված է (deprecated)
2. LLL, լիսպանման լեզու է (Lisp-like), հազվադեպ է օգտագործվում
3. Սերպենտ (Serpent), Պիթոնանման (Python-like) լեզու է, խորհուրդ չի տրվում օգտագործել այն

4. Սոլիդիթի (Solidity), բարձր կարգի, կոնտրակտակողմնորոշված (contract-oriented) լեզու, որով գրվում է խելացի պայմանագրերի մեծամասնությունը

Խաղի խելացի պայմանագիրը գրվել է Սոլիդիթի լեզվով, քանի որ ամենակայունը[3] դա է:

Պայմանագրի հիմնական ֆունկցիոնալը հետեւյալն է՝

1. Ստեղծել խաղ
2. Ստեղծել խաղի համար նախատեսված տոմսեր
3. Ապահովել տոմսերը գնելու ֆունկցիա
4. Սպասել մինչեւ բոլոր տոմսերը կվաճառվեն
5. Գնորդներից հավաքել *պատահական* թվերը
6. Այդ թվերի միջոցով ստանալ *դետերմինիստիկ* պատահական թիվ
7. Գոյացած մրցանակային ֆոնդը փոխանցել գնորդին եւ գնալ (goto) 1 կետին

Կանդրադառնանք միայն 3, 5, 6 եւ 7 կետերին: Մինչեւ նշված կետերին անդրադառնալը հասկանանք, թե ինչի համար է անհրաժեշտ, որ պատահական թիվը լինի դետերմինիսստիկ:

Էթերեում վիրտուալ մեքենան (ԷՎՄ) պարզ, բայց շատ հզոր Թյուրինգ-ամբողջական 256 բիթանոց վիրտուալ մեքենա է [5], որը թույլ է տալիս յուրաքանչյուրին իրականացնել կամայական ԷՎՄ Բայթկոդ: ԷՎՄ-ը Էթերեում արձանագրության մի մասն է եւ վճռական դեր է խաղում Էթերեումի էկոհամակարգում: Այն թույլ է տալիս յուրաքանչյուրին իրականացնել կամայական կոդ հուսալի միջավայրում, որտեղ իրականացման արդյունքը երաշխավորված է եւ ամբողջովին դետերմինիստիկ: Դա արված է բոլոր հանգույցների համաձայնության գալու պրոցեսը անխափան կատարելու համար: Այդ իսկ պատճառով լեզվում չկա `RANDOM`՝ պատահական թիվ գեներացնող ֆունկցիա: Խաղում հաղթողին որոշելու

[3] https://github.com/ethereum/wiki/wiki/Programming-languages-intro

համար մեզ անհրաժեշտ է *պատահական* թիվ: Այդ պատահական թիվը պետք է բոլոր հանգույցներում նույնը լինի՝ այսինքն պետք է ունենանք *դետերմինիստիկ պատահական թիվ,* որպեսզի ցանցը կարողանա համաձայնության գալ:

Տոմս գնելու ֆունկցիա. Տոմս գնելուց կատարվում է հետեւյալը՝

1. ստուգվում է, արդյոք գնորդը տրամադրել է սահմանված գնի չափով Էթերեում
2. ստուգվում է, որ ամեն մասնակից չկարողանա մեկից ավել տոմս գնել
3. զանգվածում պահվում է գնորդի հանրային բանալին, որպես իդենտիֆիկատոր, եւ տրամադրած՝ մոբայլում գեներացված, պատահական թվի ու հանրային բանալու հեշ կոդը

Թվերը հավաքելու ֆունկցիան. Տոմս գնելուց խաղացողների սարքում գեներացված եւ պահված պատահական թվի ու խաղացողի հանրային բանալու հեշ կոդը ուղարկվում է խելացի պայմանագրին՝ ապակետրոնացված կիրառությանը: Երբ բոլոր տոմսերը վաճառվում են, բոլոր տոմս գնած խաղացողներից հավաքվում են սարքերի վրա պահված պատահական թվերը:

Թվերը հավաքելու ֆունկցիան կատարում է հետեւյալը՝

1. Ստուգում է, որ խաղացողի ուղարկած թիվը չտարբերվի այն թվից, որի հետ հեշավորել էր իր հանրային բանալին
2. Թիվը ԲԱՑԱՀԱՅՏ ավելացնում է խելացի պայմանագրի «տվյալների շտեմարանում»՝ բլոկչեյնում:

1-ին քայլի ստուգումը կատարվում է `keccak256()` հեշավորման ֆունկցիայով [2]: Ուղարկված թիվը եւ գլոբալ տիրույթում եղած՝ `msg.sender`, դիմողի հանրային բանալին հեշավորվում են եւ ստացված հեշ կոդը համեմատվում է տոմս գնելուց ուղարկված հեշ կոդի հետ: Համապատասխանելու դեպքում թիվը ընդունվում է, այսինքն ավելացվում է բլոկչեյնում: Թիվը օգտագործվելու է դետերմինիստիկ պատահական թիվ ստանալու համար:

Դետերմինիստիկ պատահական թիվը գեներացվում է հետեյալ կերպ. հավաքված թվերը միմյանց հետ բիթային կամ գործողություն են արվում, ստացված

արդյունքը խաղացողների քանակի վրա է բաժանվում: Ստացված մնացորդ (նշ. x) կլինի *դետերմինիստիկ պատահական թիվը* [7]:

Տոմսերի վաճառքից հավաքված Էթերեումը **փոխանցվում է** x-րդ տոմսը գնած խաղացողին:

Դետերմինիստիկ պատահական թվի գեներացումը կատարվում է երկու փուլով: Առաջին փուլում հավաքվում է մոբայլում գեներացված պատահական թվի եւ խաղացողի հանրային բանալու հեշ կոդը: Երկրորդ փուլում հավաքվում են պատահական թվերը եւ ստուգվում դրանց ճշտությունը հեշավորման միջոցով: Գեներացումը տեղի է ունենում երկու փուլով քանի, որ բլոկչեյնը բաց է հանրության առաջ: Այսինքն, եթե կատարեինք մեկ փուլով, վերջին խաղացողը կարող էր կառավարել ստացվող պատահական թիվը: Իսկ, երկու փուլով կատարելու դեպքում, ստացվող պատահական թիվը կարող է կանխատեսվել միայն մեկ դեպքում՝ երբ բոլոր մասնակիցները մասնակցեն «կեղծարարությանը», որն անիմաստ է:

Ապակենտրոնացված կիրառության կոդը տեղադրված է Գիթհաբում (Github) հետեւյալ հղմամբ՝

https://github.com/Galti/bachelor_thesis/tree/master/smart_contract

ԵԶՐԱԿԱՑՈՒԹՅՈՒՆ

Օգտագործելով բլոկչեյն տեխնոլոգիաներից Էթերեումը, հնարավոր եղավ ստեղծել մի խաղ, որը հարյուր տոկոսանոց թափանցիկ է եւ կիրառելի: Հենվելով ստացած արդյունքների վրա կարելի է զանազան խաղեր տեղափոխել բլոկչեյն հարթակի վրա խաղացողների մոտ վստահություն ձեռք բերելու համար: Օրինակ JackPot խաղը:

Խաղը ստեղծվել է հիմնականում է բլոկչեյն տեխնոլոգիան ուսումնասիրելու համար: Ուսումնասիրության արդյունքում նկատվել են նաեւ, որ բացի կառավարման (փաստաթղթաշրջանառություն, թափանցիկ ընտրությունների անցկացում) եւ բանկային (շատ ավելի էժան փոխանցումներ) ոլորտներից, այն կարելի է օգտագործել նաեւ այլ ոլորտներում:

Բլոկչեյն տեխնոլոգիայի դրական կողմերը՝

- թափանցիկությունը
- անփոփոխելիությունը (immutability)

Բլոկչեյն տեխնոլոգիայի բացասական կողմերը՝

- արտադրողականությունը (performace)
- հանգույցներում գործող համակարգիչների մեծ քանակի հոսանք ծախսելը

ԳՐԱԿԱՆՈՒԹՅՈՒՆ

1. Bitcoin, Wikipedia

   // URL: https://en.wikipedia.org/wiki/Bitcoin, 22.04.2018

2. **Ch. Dannen,** Introducing Ethereum and Solidity, Appres 2017

3. Ethereum, Wikipedia

   // URL: https://en.wikipedia.org/wiki/Ethereum, 22.04.2018

4. **F. Zaninotto**, The Blockchain Explained to Web Developers, marmelab
   // URL: https://marmelab.com/blog/2016/04/28/blockchain-for-web-developers-the-theory.html, 19.04.2018

5. **J. Wilcke**, Optimising the Ethereum Virtual Machin

   // URL: https://medium.com/@jeff.ethereum/optimising-the-ethereum-virtual-machine-58457e61ca15, 29.04.2018

6. **I. Bashir**, Mastering Blockchain, Packt Publishing 2017

7. Miners can't be trusted, GitHub

   // URL: https://github.com/randao/randao/blob/master/README.md, 25.04.2018

8. Proof of Work vs Proof of Stake: Basic Mining Guide, Blockgeeks

   // URL: https://blockgeeks.com/guides/proof-of-work-vs-proof-of-stake/, 24.04.2018