Բիթքոյն. Մասնակիցը մասնակցին (peer-to-peer) էլեկտրոնային դրամական միջոցների համակարգ

Սատոշի Նակամոտո
satoshin@gmx.com
www.bitcoin.org

Անգլերենից (bitcoin.org/bitcoin.pdf) թարգմանեց Դիանա Սիսակյանը
Հովանավոր՝ cleartalks.com

Ամփոփագիր. Էլեկտրոնային դրամական միջոցների ամբողջովին մասնակիցը մասնակցին (peer-to-peer) ապակենտրոնացված տարբերակը թույլ կտա անմիջապես երկու մասնակիցների միջև առցանց գործառնություններ կատարել՝ առանց որևէ ֆինանսական հաստատության միջնորդության: Թվային ստորագրությունների օգտագործումը մասամբ լուծում է այս խնդիրը, սակայն այս մոտեցումը զրկվում է իր հիմնական առավելություններից, եթե կրկնակի ծախսերը կանխելու համար դեռ կա վստահված երրորդ կողմի միջնորդության անհրաժեշտություն: Մենք առաջարկում ենք կրկնակի ծախսի խնդրի լուծում՝ օգտագործելով մասնակիցը մասնակցին (peer-to-peer) ցանցը: Ցանցը ժամանակագրում է (timestamp) գործառնությունները՝ միավորելով դրանք hash[1] գործառույթի վրա հիմնված աշխատանքի ապացույցի (proof-of-work) հաջորդական շղթայի մեջ, այսպիսով ցանցը ձևավորում է գրառում, որը հնարավոր չէ փոխել առանց աշխատանքի ապացույցը վերագործարկելու: Շղթայի ամենաերկար տարբերակը ոչ միայն հաստատում է գործառնությունների հերթականությունը, այլ նաև ապացուցում է, որ այն բխում է ցանցի ԿՄՀ (CPU) ամենամեծ հզորության կետից: Քանի դեռ ԿՄՀ-ի հզորության մեծ մասը վերահսկվում է ցանցի հարձակմանը չմասնակցող հանգույցների կողմից, նրանք կստեղծեն ամենաերկար շղթան և կգերազանցեն հարձակվողներին: Ցանցն ինքնին ունի պարզագույն կառուցվածք:

[1] Գործառույթն իրականացնում է որոշակի ալգորիթմի կողմից կատարված կամայական երկարության տողի տվյալների վերափոխում:

Հաղորդագրությունները ուղարկվում են «best effort[2]» համաձայնության հիման վրա, իսկ հանգույցները կարող են դուրս գալ և նորից միանալ ցանցին ցանկացած պահի՝ հաստատելով աշխատանքի ապացույցի շղթայի ամենաերկար տարբերակը՝ որպես բաց թողնված գործառնությունների պատմության ապացույց։

1. Ներածություն

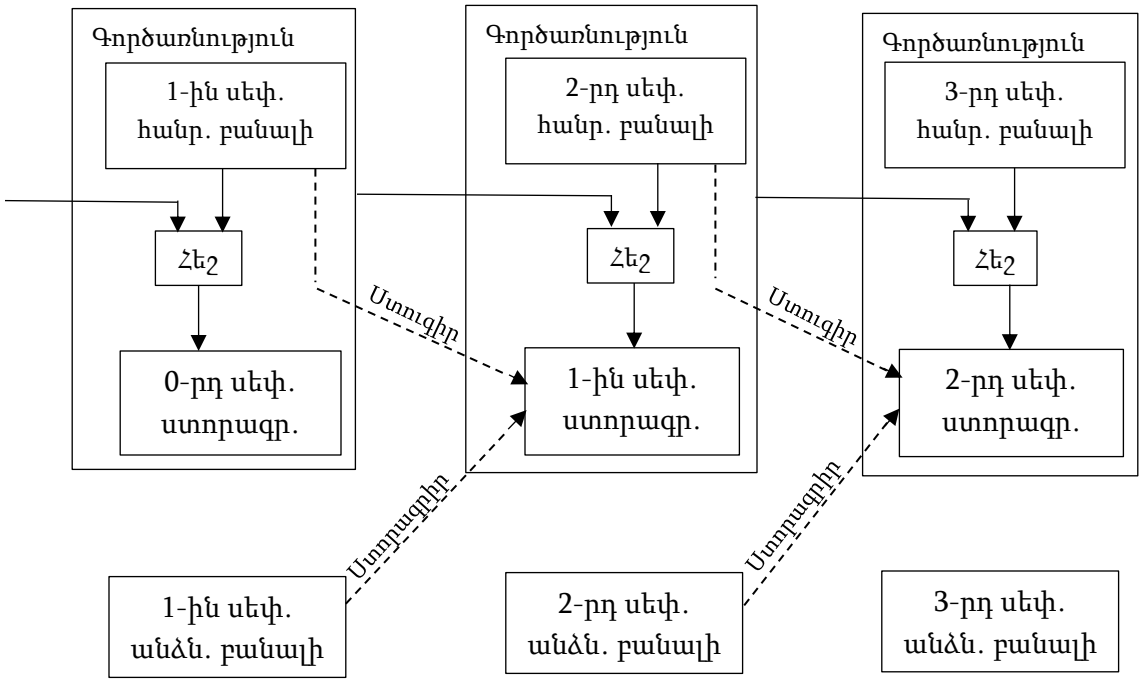
Առցանց առևտուրը սկսել է գրեթե բացառապես ապավինել ֆինանսական հաստատություններին, որոնք էլեկտրոնային վճարումներ իրականացնելիս կատարում են վստահված երրորդ կողմի դեր։ Թեև այս համակարգը գործառնությունների զգալի մասի համար բավականին լավ է աշխատում, այն հիմնված է վստահության մոդելի վրա՝ վերջինիս բոլոր խոցելի կողմերի հետ մեկտեղ։ Լիովին անշրջելի գործառնություններ իրականում հնարավոր չեն, քանի որ ֆինանսական հաստատությունները չեն կարող խուսափել միջնորդության վերաբերյալ վեճերից։ Միջնորդության ծառայության արժեքը մեծացնում է գործառնության գինը՝ սահմանափակելով գործառնությունների նվազագույն գործնական չափը, և վերացնում հերթական պարզ գործառնություններ կատարելու հնարավորությունը, իսկ գինն ավելի բարձր է անշրջելի ծառայությունների դիմաց անշրջելի վճարումներ կատարելու հնարավորության բացակայության դեպքում։ Գործառնության շրջելիության հնարավորությունը մեծացնում է վստահության անհրաժեշտությունը։ Վաճառողը պետք է զգուշանա իր հաճախորդից՝ պահանջելով նրանից ավելի շատ տեղեկություն, քան սովորաբար։ Խարդախության որոշակի տոկոսը համարվում է անխուսափելի։ Այս ծախսերից և վճարման անորոշություններից կարելի է խուսափել օգտագործելով ֆիզիկական արժույթ, սակայն գոյություն չունի չմիջնորդավորված էլեկտրոնային վճարումներ կատարելու որևէ համակարգ։ Անհրաժեշտ է էլեկտրոնային վճարային համակարգ, որը, վստահության փոխարեն՝ հիմնված է գաղտնագրային (կրիպտոգրաֆիկ) ապացույցների վրա, ինչը թույլ կտա ցանկացած երկու մասնակից կողմերին անմիջականորեն գործառնություններ իրականացնել միմյանց հետ՝ առանց վստահված երրորդ կողմի անհրաժեշտության։ Բացարձակապես անշրջելի հաշվողական գործառնությունները կպաշտպանեն վաճառողներին խարդախությունից, իսկ

[2] «Best effort basis». նախաձեռնություն իրականացնելու համաձայնություն է, առանց որևէ երաշխիքի, որ այն կպսակվի հաջողությամբ։

պարզ «escrow»[3] պահուստային մեխանիզմները հեշտությամբ կարող են կիրառվել գնորդներին պաշտպանելու համար: Այս աշխատությունում մենք առաջարկում ենք կրկնակի ծախսի խնդրի լուծում՝ օգտագործելով գործառնությունների ժամանակագրական հաջորդականության հաշվողական ապացույց ստեղծող մասնակիցը մասնակցին (peer-to-peer) բաշխված ժամանակագրական սերվերը: Համակարգն անվտանգ է այնքան ժամանակ, քանի դեռ ազնիվ մասնակից-հանգույցները միասին վերահսկում են ԿՄՀ-ի ավելի շատ հզորություն, քան հարձակվող հանգույցների ցանկացած համագործակցող խումբ:

2. Գործառնություններ

Սահմանենք էլեկտրոնային մետաղադրամը որպես թվային ստորագրությունների շղթա: Յուրաքանչյուր սեփականատեր մետաղադրամը փոխանցում է հաջորդին՝ թվային կերպով ստորագրելով նախորդ գործառնության հեշը (hash), և հաջորդ սեփականատիրոջ հանրային բանալին՝ ավելացնելով այդ տեղեկատվությունը մետաղադրամին: Ստացողը կարող է ստուգել ստորագրությունները՝ սեփականության շղթան վավերացնելու համար:



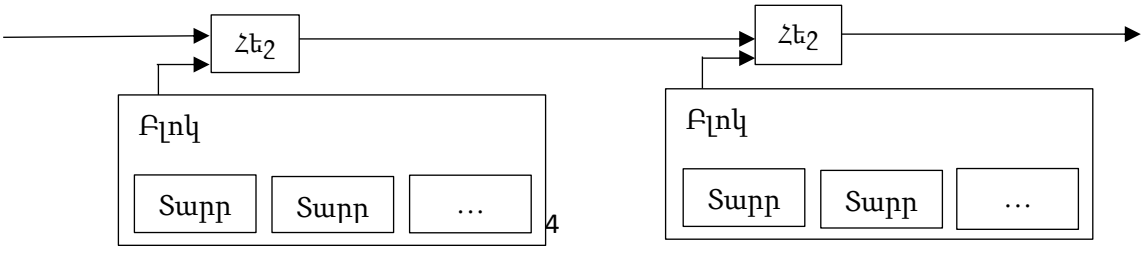
[3] «Escrow-հաշիվ». հատուկ պայմանական հաշիվ, որի վրա գրանցում են գույք, փաստաթղթեր կամ դրամական միջոցներ մինչև որոշակի հանգամանքների առաջացումը կամ որոշակի պարտավորությունների կատարումը:

Խնդիրը, անտարակույս, այն է, որ ստացողը չի կարող պարզել, թե քանի անգամ է տվյալ մետաղադրամի նախորդ սեփականատերը այն ծախսել: Խնդրի ընդհանուր լուծումը վստահված կենտրոնական մարմնի կամ դրամահատարանի ներգրավումն է, որը ստուգում է յուրաքանչյուր գործառնություն: Մետաղադրամը պետք է վերադարձվի դրամահատարան յուրաքանչյուր գործառնությունից հետո, որպեսզի թողարկվի նոր մետաղադրամ, և վստահելի են միայն դրամահատարանից թողարկված մետաղադրամները: Այս լուծման թերությունն այն է, որ ամբողջ դրամական համակարգի ճակատագիրը կախված է դրամահատարանը ղեկավարող ընկերությունից, և յուրաքանչյուր գործառնություն պետք է կատարվի նրա միջնորդությամբ՝ ինչպես բանկի միջոցով:

Մեզ պետք է միջոց, որի օգնությամբ ստացողն իմանա, որ մետաղադրամի նախորդ սեփականատերերը չեն ստորագրել ավելի վաղ գործառնություններ: Մեր նպատակին հասնելու համար մենք հաշվի ենք առնում միայն ամենավաղ կատարված գործառնությունը, ուստի մենք հաշվի չենք առնում կրկնակի ծախսերի հետագա փորձերը: Գործառնության բացակայությունը հաստատելու միակ միջոցը բոլոր գործառնությունների մասին տեղյակ լինելն է: Դրամահատարանի վրա հիմնված մոդելում դրամահատարանը տեղյակ է եղել բոլոր գործառնությունների մասին և որոշել, թե որն է տեղի ունեցել առաջինը: Այս ամենը առանց վստահված երրորդ կողմի իրականացնելու համար, գործառնությունները պետք է հանրայնորեն հայտարարվեն [1], և մեզ անհրաժեշտ է համակարգ, որում մասնակիցները համաձայնության կգան գործառնությունների շղթայի հաջորդականության պատմության շուրջ: Ստացողին անհրաժեշտ է ապացույց, որ յուրաքանչյուր գործառնության ժամանակ, հանգույցների մեծամասնությունը համաձայնել է, որ այդ գործառնությունն առաջինն է ստացել:

3. Ժամանակագրական սերվեր

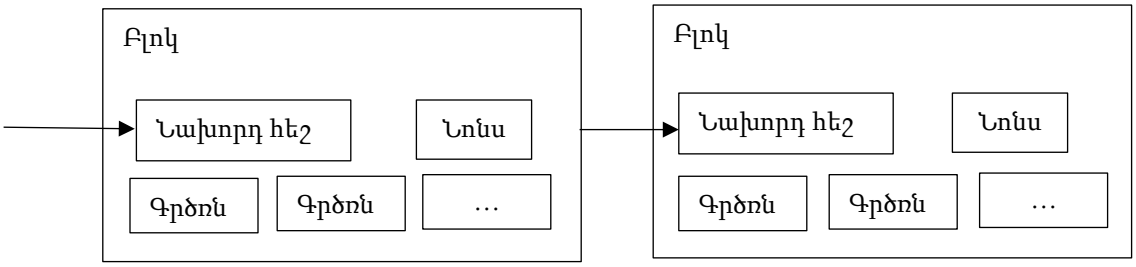
Մեր առաջարկած լուծումը սկսվում է ժամանակագրական սերվերից: Վերջինս գործում է վերցնելով տարրերի (item) բլոկի հեշը և այն հանրայնորեն հրապարակելով, ինչպես օրինակ թերթում կամ Usenet գրառման մեջ [2-5]: Յուրաքանչյուր ժամանակագրություն իր հեշում ներառում է նախորդ ժամանակագրությունը, այսպիսով կազմելով շղթա, որում յուրաքանչյուր ժամանակագրություն ամրացնում է նախորդները:



4. Աշխատանքի ապացույց (Proof-of-work)

Որպեսզի մասնակիցը մասնակցին համակարգի հիման վրա իրագործենք բաշխված ժամանակագրական սերվերը, մեզ հարկավոր է օգտագործել Ադամ Բեքի «Hashcash» տեխնոլոգիայի [6] նմանությամբ աշխատանքի ապացույց համակարգը՝ թերթի կամ Usenet-ի գրառումների փոխարեն: Աշխատանքի ապացույցը ներառում է մի արժեքի սկանավորում (օրինակ՝ SHA-256), որը, երբ հեշավորվում է, վերջինիս հեշը սկսվում է զրոյական թվի բիթով: Պահանջվող աշխատանքի ծավալը ցուցչայնորեն (էքսպոնենցիալորեն) կախված է զրոյական բիթերի քանակից, իսկ սկանավորված արժեքը կարելի է ստուգել ընդամենը մեկ հեշի միջոցով:

Մեր ժամանակագրական ցանցի համար մենք իրականացնում ենք աշխատանքի ապացույցը՝ բլոկում ավելացնելով նոնս (nonce)[4] պարամետր, մինչև գտնվի մի արժեք, որը բլոկի հեշին տալիս է պահանջվող զրո բիթերը: Ճիշտ այն պահին, երբ բլոկը բավարարում է աշխատանքի ապացույցի պայմանը, վերջինիս արդյունքը համարվում է անշրջելի, և բլոկը չի կարող փոփոխվել առանց վերագործարկման: Քանի որ հետագա բլոկները շղթայված են դրանից հետո, բլոկը փոխելու աշխատանքը կներառի դրան հաջորդող բոլոր բլոկների վերագործարկումը:



Աշխատանքի ապացույցը հեշավորման միջոցով նաև լուծում է մեծամասնության կողմից աջակցվող տարբերակի որոշման հարցը: Եթե մեծամասնությունը հիմնված է «մեկ IP հասցե՝ մեկ ձայն» համակարգի վրա, ապա այն հնարավոր կլինի «ջարդել» IP հասցեների մեծամասնությունը վերահսկելու դեպքում: Ըստ էության, աշխատանքի ապացույցը հիմնված է «մեկ ԿՄՀ՝ մեկ ձայն» սկզբունքի վրա: Ամենաերկար շղթան ներկայացնում է մեծամասնության որոշումը, որում ներդրված է ամենամեծ թվով աշխատանքի ապացույց: Եթե ԿՄՀ-ի հզորության մեծ մասը վերահսկվում է ազնիվ հանգույցների կողմից, ապա ազնիվ շղթան ամենաարագ աճը կունենա և կգերազանցի բոլոր մրցակից շղթաները: Նախորդ բլոկներից որևէ մեկը փոփոխելու համար հարձակվողը պետք է կրկնի բլոկի աշխատանքի ապացույցը և դրանից հետո

[4] «Nonce» («number only used once») հապավում է, որը բլոկչեյնում հեշավորված կամ գաղտնագրված բլոկին ավելացված թիվ է:

վերագործի բոլոր մյուս բլոկները, այնուհետև հասնի ազնիվ հանգույցների աշխատանքին և գերազանցի դրանք: Հաջորդիվ մենք կցուցադրենք, որ հաջորդ բլոկների ավելացման պատճառով ավելի դանդաղ գործող հարձակվողի՝ ազնիվ հանգույցների աշխատանքին հասնելու հավանականությունը ցուցչայնորեն նվազում է:

Սարքախմբի (hardware) արագության բարձրացման և ժամանակի ընթացքում գործարկվող հանգույցների քանակի տատանումները փոխհատուցելու համար հարկավոր է, որ փոփոխվի աշխատանքի ապացույցի հեշավորման բարդությունը (difficulty)՝ ապահովելու բլոկների գեներացման համաչափ արագություն: Եթե դրանք շատ արագ են գեներացվում, ապա բարդությունը մեծանում է:

5. Ցանց

Ցանցի կառավարման քայլերը հետևյալն են.

1) Նոր գործառնությունները հեռարձակվում են դեպի բոլոր հանգույցներ:
2) Յուրաքանչյուր հանգույց նոր գործառնությունները հավաքում է բլոկի մեջ:
3) Յուրաքանչյուր հանգույց աշխատում է գտնել ընթացիկ բարդության աշխատանքի ապացույց (proof-of-work) իր բլոկի համար:
4) Երբ հանգույցը գտնում է այն, վերջինս հեռարձակում է այդ բլոկը բոլոր մյուս հանգույցներին:
5) Հանգույցներն ընդունում են բլոկը միայն այն դեպքում, եթե դրանում կատարված բոլոր գործառնությունները վավեր են, և որոնց դրամական միջոցները դեռ չեն ծախսվել:
6) Հանգույցները բլոկի վերաբերյալ հայտնում են համաձայնություն և միևնույն ժամանակ աշխատում շղթայում նոր բլոկ ստեղծել օգտագործելով ընդունված բլոկի հեշը՝ որպես նախորդ հեշ:

Հանգույցները միշտ ճշմարիտ են համարում միայն ամենաերկար շղթան և այն շարունակաբար ընդլայնում են: Եթե երկու հանգույցները համաժամանակորեն հեռարձակում են հաջորդ բլոկի այլատեսակ տարբերակներ, որոշ հանգույցներ կարող են սկզբում ստանալ մեկը, իսկ հետո՝ մյուսը: Այդ դեպքում հանգույցներից յուրաքանչյուրը աշխատում է շղթայի իր ստացած առաջին տարբերակի հետ՝ փրկելով մյուս ճյուղը այն դեպքում, եթե վերջինս ավելի վաղ էր սկսել երկարել: Այս կապը կխզվի, եթե հայտնաբերվի հաջորդ աշխատանքի ապացույցը, և մյուս ճյուղը կերկարի: Մյուս ճյուղի հետ աշխատող հանգույցները այնուհետև անցում կկատարեն ավելի երկար ճյուղին:

Նոր գործառնությունների հեռարձակումը դեպի բոլոր հանգույցներ պարտադիր չէ: Քանի դեռ նրանք հասնում են բազմաթիվ հանգույցների, նրանք շուտով կհայտնվեն բլոկում: Բլոկերի հեռարձակումները նաև հանդուրժող են բաց թողնված հաղորդագրությունների նկատմամբ: Եթե հանգույցը բլոկ չի ստանում, ապա այն կպահանջի այն, իսկ երբ ստանա հաջորդ բլոկը, կհասկանա, որ մեկը բաց է թողել:

6. Խրախուսանք (Incentive)

Ըստ պայմանական համաձայնության՝ բլոկի առաջին գործառնությունը հատուկ գործառնություն է, որը ստեղծում է նոր մետաղադրամ, իսկ վերջինս պատկանում է բլոկի ստեղծողին: Սա խրախուսում է հանգույցներին աջակցել ցանցին և հնարավորություն է տալիս սկզբնական շրջանում մետաղադրամները մտցնել շրջանառության մեջ, քանի որ չկա դրանք թողարկող կենտրոնական մարմին: Շրջանառության մեջ գտնվող մետաղադրամների քանակի կայուն աճը կարելի է համեմատել ոսկու արդյունահանման հետ, որտեղ ոսկու կրիպտոհանքափորները ներդնում են իրենց միջոցները այդ շրջանառության մեջ: Մեր դեպքում, այդ գործառույթը իրականացնում են ԿՄՀ-ի ծախսվող ժամանակն ու էլեկտրաէներգիան:

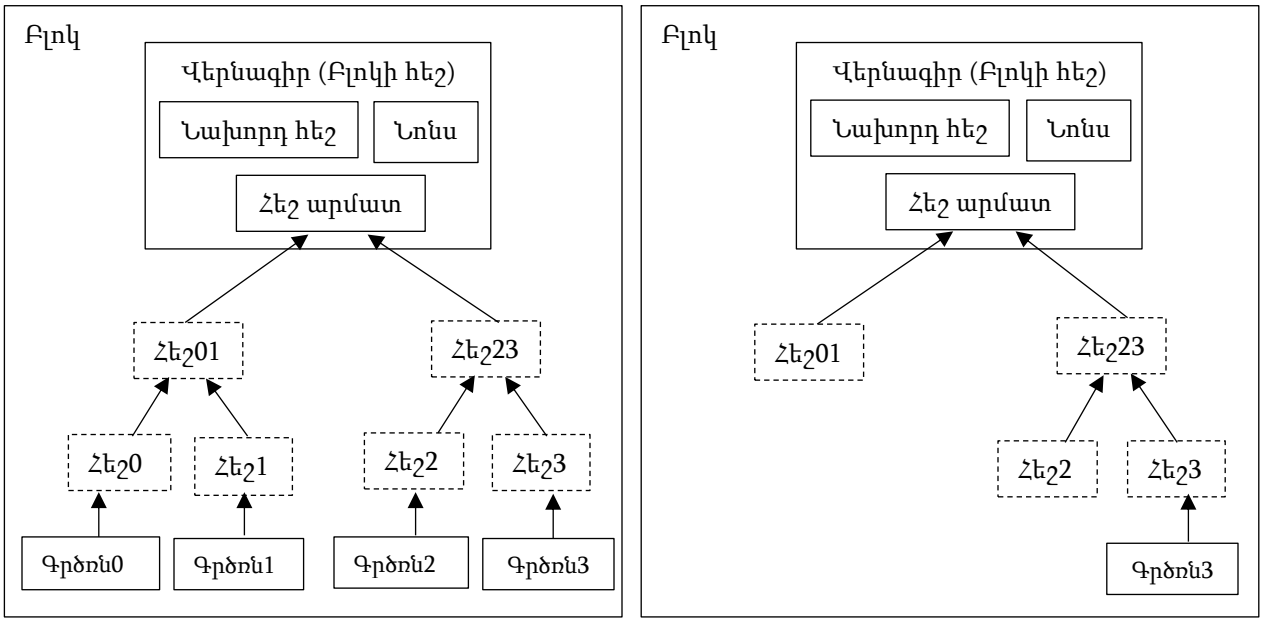
Խրախուսանքը կարող է ֆինանսավորվել նաև գործառնությունների վճարներով: Եթե գործառնության ելքային արժեքը փոքր է դրա մուտքային արժեքից, ապա տարբերությունը գործառնության վճարն է, որը ավելացվում է այդ գործառնությունը պարունակող բլոկի խրախուսական արժեքին: Երբ կանխորոշված թվով մետաղադրամներ մտնեն շրջանառության մեջ, խրախուսանքը կարող է ամբողջովին տրամադրվել գործառնության վճարներից և լիովին զերծ մնալ գնաճից:

Խրախուսանքը կարող է օգնել «քաջալերել» հանգույցներին ազնիվ մնալ: Եթե ագահ հարձակվողը կարողանա հավաքել ավելի շատ ԿՄՀ հզորություն, քան բոլոր ազնիվ հանգույցները, նա պետք է ընտրի հետևյալ երկու տարբերակների միջև. օգտագործել այդ ռեսուրսը մարդկանց խաբելու համար՝ իր իսկ վճարումները ետ գողանալով, կամ օգտագործել այն նոր մետաղադրամներ թողարկելու համար: Նրա համար ավելի շահավետ կլինի հետևել խաղի կանոններին, և առաջնորդվել այնպիսի կանոններով, որոնք իրեն կշնորհեն ավելի շատ նոր մետաղադրամներ, քան բոլորինը միասին վերցրած՝ համակարգին և սեփական հարստության վավերականությանը ծանրագույն վնաս պատճառելու փոխարեն:

7. Կոշտ սկավառակի ծավալի տնտեսում

Երբ մետաղադրամի վերջին գործառնությունը ընկնում է բավականաչափ թվով բլոկների տակ, դրանից առաջ ծախսված գործառնությունները կարելի է ջնջել՝ կոշտ

սկավառակի ծավալը տնտեսելու համար: Դա հեշտացնելու համար՝ առանց բլոկի հեշը կոտրելու, գործառնությունները հեշավորվում են Մերքլի հեշ-ծառում (Merkle tree[5]) [7][2][5]՝ բլոկի հեշի մեջ ներառելով միայն արմատը: Այնուհետև, հին բլոկների ծավալը կարելի է սեղմել՝ կտրելով ծառի ճյուղերը: Ներքին հեշերը պահելու անհրաժեշտություն չկա:



Մերքլի ծառի հեշավորված գրծոն. Գրծոն0-2ը բլոկից կտրելուց հետո

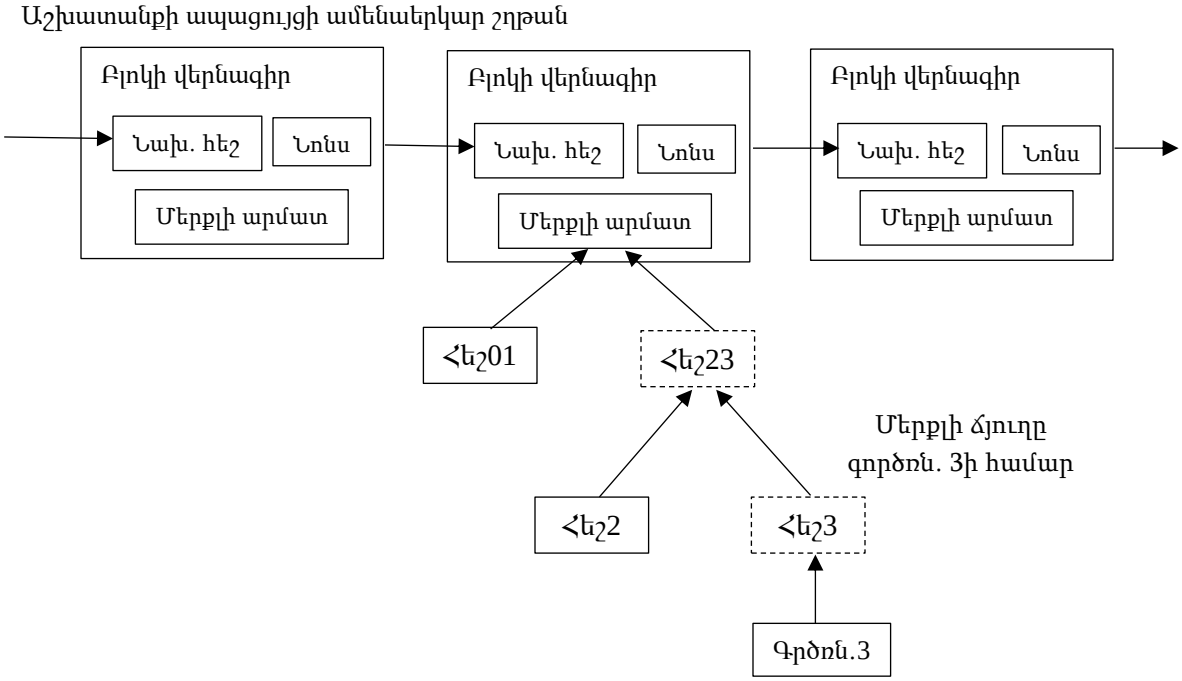
Առանց գործառնությունների՝ բլոկի վերնագիրը կկազմի մոտ 80 բայթ: Եթե ենթադրենք, որ բլոկները ստեղծվում են 10 րոպեն մեկ, ապա տարեկան ստացվում է 80 բայթ * 6 * 24 * 365 = 4,2 ՄԲ : Հաշվի առնելով այն, որ 2008 թ.-ի դրությամբ համակարգիչները միջինում վաճառվում են 2 ԳԲ օպերատիվ հիշողությամբ, և Մուրի օրենքը, որը կանխատեսում է տարեկան 1,2 ԳԲ ընթացիկ աճ, տվյալների պահպանումը չպետք է խնդիր լինի, նույնիսկ եթե բլոկի վերնագրերը պահվեն հիշողության մեջ:

8. Վճարման պարզեցված ստուգում

Հնարավոր է ստուգել վճարումները՝ առանց ողջ ցանցային հանգույցը գործարկելու: Օգտատերը միայն պետք է պահպանի աշխատանքի ապացույցի ամենաերկար շղթայի բլոկների վերնագրերի պատճենները, որոնք նա կարող է ստանալ ցանցի

[5] Հեշ ծառ, որում յուրաքանչյուր «տերև» (հանգույց) պիտակավորված է տվյալների բլոկի ծածկագրային հեշով:

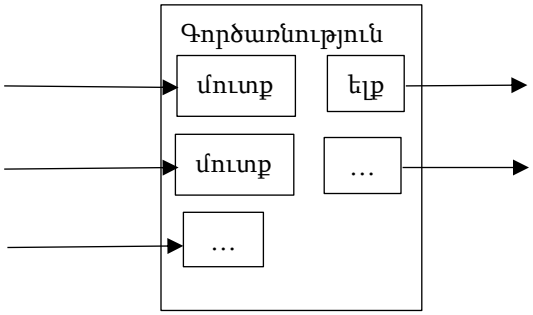
հանգույցների հարցումով, մինչև համոզվի, որ իր մոտ է ամենաերկար շղթան, և ստանալ Մերքլի ճյուղը, որը կապում է գործառնությունը այն բլոկի հետ, որում այն ժամանակագրվել է: Օգտատերը ինքնուրույն չի կարող ստուգել գործառնությունը, սակայն ստանալով բլոկի հղումը՝ նա կարող է հավաստել, որ ցանցի հանգույցն այն ընդունել է, և դրանից հետո ավելացված բլոկները հետագայում հաստատում են, որ ցանցն ընդունել է տվյալ գործառնությունը:



Ստուգման այս մեթոդը հուսալի է, քանի դեռ ազնիվ հանգույցները վերահսկում են ցանցը, սակայն այն ավելի խոցելի է, եթե ցանցում հարձակվողն ունի ռեսուրսների գերակշռություն: Թեև ցանցային հանգույցները կարող են ինքնուրույնաբար ստուգել գործառնությունները, ստուգման պարզեցված մեթոդը կարող է խաբվել հարձակվողի կեղծված գործառնություններով այնքան ժամանակ, քանի դեռ հարձակվողը շարունակում է վերահսկել ցանցը: Սրանից պաշտպանվելու ռազմավարություններից մեկը ցանցային հանգույցներից զգուշացումներ ընդունելն է, երբ նրանք հայտնաբերում են անվավեր բլոկ: Այս ամենն օգտատիրոջ համակարգչի ծրագրակազմին (software) թույլ կտա ներբեռնել ամբողջական բլոկը, իսկ զգուշացված գործառնություններին՝ հաստատել կեղծ տվյալների առկայությունը: Հաճախակի վճարումներ ստացող ձեռնարկությունները, հավանաբար, դեռ կցանկանան գործարկել իրենց սեփական հանգույցները՝ ավելի կայուն անվտանգության և ստուգման ավելի մեծ արագության համար:

9. Արժեքների համակցում և բաժանում

Թեև հնարավոր կլիներ մետաղադրամներով կառավարել առանձին-առանձին, յուրաքանչյուր փոխանցվող ցենտի համար առանձին գործառնություն կատարելը բարդ կլիներ: Արժեքների բաժանումն ու համակցումը իրականացնելու նպատակով՝ գործարքները պարունակում են բազմաթիվ մուտքեր և ելքեր: Սովորաբար նախորդ, ավելի մեծ գործառնությունից տեղի է ունենում մեկ մուտքագրում, կամ մի քանի մուտքեր, որոնք համակցում են ավելի փոքր գումարներ, և առավելագույնը երկու ելք. մեկը վճարման համար, իսկ մյուսը՝ մանրը (առկայության դեպքում) վճարողին վերադարձնելու:

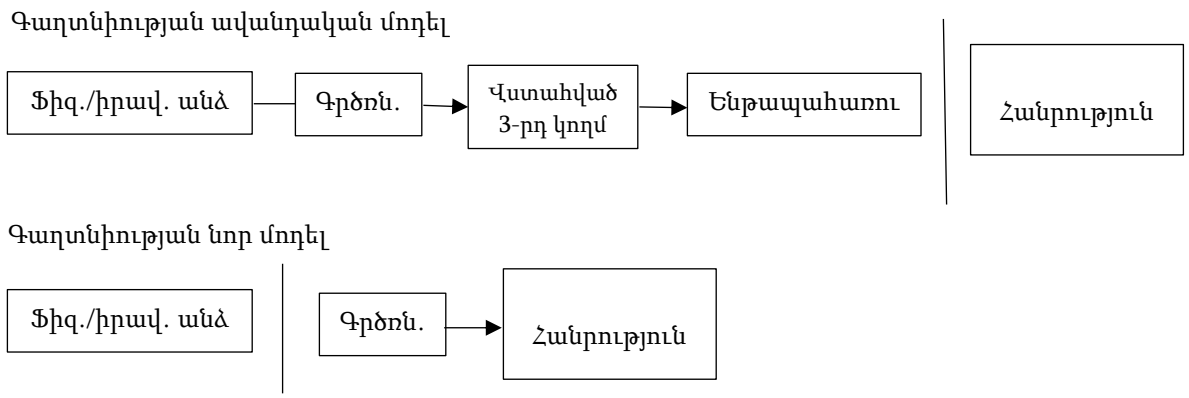


Հարկ է նշել, որ «fan-out»[6]-ը, որի ժամանակ մեկ գործառնությունը կախված է մի քանի գործառնությունից, իսկ դրանք էլ՝ ավելի մեծ թվով գործառնություններից, այս դեպքում խնդիր չէ: Գործառնության պատմության ամբողջական օրինակի դուրս բերման անհրաժեշտություն երբեք չկա:

10. Գաղտնիություն

Ավանդական բանկային մոդելը ապահովում է գաղտնիություն՝ սահմանափակելով տեղեկատվության հասանելիությունը ներգրավված կողմերի և վստահված երրորդ կողմի համար: Բոլոր գործառնությունները հրապարակայնորեն հայտարարելու անհրաժեշտությունը բացառում է վերոնշյալ մեթոդի կիրառումը, սակայն գաղտնիությունը դեռևս կարող է պահպանվել՝ խաթարելով տեղեկատվական հոսքը դեպի մեկ այլ կետ, եթե անանուն են պահվել հանրային բանալիները: Հանրությանը հասանելի կլինի միայն այն, որ ինչ-որ մեկը գումար է ուղարկում մեկ ուրիշին՝ առանց գործառնության մասնակիցների մասին տեղեկատվության: Սա նման է ֆոնդային սակարանների (բորսա) կողմից հրապարակված տեղեկատվությանը, որտեղ հանրայնացվում են առանձին գործառնությունների ժամանակն ու ծավալը՝ առանց նշելու, թե ում միջև են տեղի ունեցել գործառնությունները:

[6] Մուտքերի թիվը, որոնք կարող են միացվել տվյալ ելքին



Որպես լրացուցիչ firewall[7] (հրարգելիչ) պաշտպանություն, յուրաքանչյուր գործառնության համար պետք է օգտագործվի նոր բանալիների զույգ, որը կկանխարգելի տարբեր վճարումների կապն իրենց ընդհանուր սեփականատիրոջ հետ: Բազմամուտքային գործառնությունների դեպքում որոշ կապեր դեռևս անխուսափելի են. դրանք անպայման ցույց են տալիս, որ իրենց մուտքերը պատկանում էին նույն սեփականատիրոջը: Հնարավոր վտանգը բանալիի սեփականատիրոջ բացահայտման մեջ է. այդ դեպքում կապը կարող է բացահայտել այլ գործառնություններ, որոնք պատկանել են նույն սեփականատիրոջը:

11. Հաշվարկներ

Մենք դիտարկում ենք այն սցենարը, երբ հարձակվողը ազնիվ շղթայից ավելի արագ է փորձում այլընտրանքային շղթա ստեղծել: Նույնիսկ եթե նա հասնի իր նպատակին, դա համակարգը բաց ու հասանելի չի դարձնում կամայական փոփոխությունների համար. օրինակ՝ օդից արժեքներ ստեղծելու կամ այլ մարդկանց մետաղադրամները յուրացնելու համար: Հանգույցները չեն ընդունի անվավեր գործառնությունը կամ այն բլոկը, որը պարունակում է այդ գործառնությունը: Հարձակվողը կարող է միայն փորձել փոխել իր սեփական գործառնություններից մեկը՝ վերջերս ծախսած գումարը ետ վերցնելու նպատակով:

Ազնիվ շղթայի և հարձակվողի միջև մրցավազքը կարելի է բնութագրել որպես Երկանդամ պատահական պրոցես[8] (Binominal Random Walk): Հաջող իրադարձությունն (event) այն է, երբ ազնիվ շղթան երկարացվում է մեկ բլոկով՝ մեծացնելով իր առաջատարությունը +1-ով, իսկ ձախողված իրադարձությունն այն է, երբ հարձակվողի շղթան է երկարացվում մեկ բլոկով՝ նվազեցնելով բացը (gap) -1-ով:

[7] այն պաշտպանում է (ցանցը կամ համակարգը) անօրինական մուտքից

[8] Մաթ. «Պատահական պրոցեսը» մաթեմատիկական օբյեկտ է, որը նկարագրում է մի ուղի, որը բաղկացած է պատահական քայլերից որոշ մաթեմատիկական տարածության վրա, ինչպիսին օրինակ ամբողջ թվերն են:

Հավանականությունը, որ հարձակվողը մի քանի բլոկի տարբերությունը կկրճատի նման է Խաղացողի սնանկացման խնդրին[9] (Gambler's Ruin Problem): Ենթադրենք՝ անսահմանափակ վարկով խաղամոլը սկսում է պակասորդով և անսահման թվով փորձեր է ձեռնարկում, որպեսզի խաղի ընթացքում պարտված միջոցները վերականգնի: Հարձակվողի՝ հաջողության և ազնիվ մասնակիցներին հասնելու հավանականությունը հաշվարկվում է հետևյալ կերպ [8].

p = հավանականություն, որ ազնիվ հանգույցը կգտնի հաջորդ բլոկը
q = հավանականություն, որ հարձակվողը կգտնի հաջորդ բլոկը
q_z = հավանականություն, որ հարձակվողը երբևէ կհասնի z թվով բլոկներին

$$q_z = \begin{cases} 1 & if\ p \le q \\ (q/p)^z & if\ p > q \end{cases}$$

Հաշվի առնելով մեր ենթադրությունը, որ $p > q$, հավանականությունը ցուցչայնորեն նվազում է, քանի որ այն բլոկների թիվը, որոնց հարձակվողը պետք է հասնի, աճում է: Քանի որ համակարգը ամբողջովին գործում է հարձակվողի դեմ, առանց մեծ ուժով առաջ նետվելու՝ նրա հաղթանակի հնարավորությունները հօդս են ցնդում, մինչ նա շարունակաբար ետ է մնում:

Այժմ մենք կուսումնասիրենք, թե նոր գործառնության ստացող կողմը որքան պետք է սպասի, նախքան լիովին վստահ լինելը, որ ուղարկողը չի կարող փոխել գործառնությունը: Ենթադրենք՝ ուղարկողը հարձակվողն է, որը ցանկանում է, որ ստացողը հավատա, որ վճարումը կատարվել է, այնուհետև փոխել գործառնությունը՝ վերադարձնելով գումարն ինքն իրեն: Երբ այս ամենը տեղի է ունենում, ստացողին ուղարկվում է զգուշացում, մինչդեռ հարձակվող-ուղարկողը լիահույս է, որ արդեն շատ ուշ է:

Ստացողը ստեղծում է նոր բանալիների զույգ և ստորագրումից անմիջապես առաջ հանրային բանալին տալիս է ուղարկողին: Սա թույլ չի տալիս ուղարկողին նախապես պատրաստել բլոկների շղթա՝ շարունակաբար աշխատելով դրա վրա, մինչև նրան հաջողվի բավականաչափ առաջ ընկնել, և թույլ չի տա կատարել գործառնությունն այդ պահին: Երբ գործառնությունն ուղարկվել է, խաբեբան սկսում է գաղտնի կերպով աշխատել զուգահեռ շղթայի վրա, որը պարունակում է իր գործառնության այլընտրանքային տարբերակը:

Ստացողը սպասում է, մինչև գործառնությունը ավելացվի բլոկին, իսկ z բլոկները կցվեն դրա ետևից: Նա չգիտի հարձակվողի առաջընթացի ճշգրիտ չափը, բայց

[9] Հավանականության տեսության խնդիր

ենթադրելով, որ եթե ազնիվ բլոկների առաջացման միջին արագությունը հայտնի արժեք է, ապա հարձակվողի հավանական առաջընթացը մաթեմատիկական ակնկալիքով ենթարկվում է Պուասոնի բաշխմանը.

$$\lambda = z\frac{q}{p}$$

Որպեսզի ստանանք հարձակվողի՝ ազնիվ հանգույցներին հասնելու հավանականությունը, մենք բազմապատկում ենք նրա ստեղծած բլոկների քանակը այն հավանականությամբ, որ նա կկարողանա լրացնել մնացած տարբերությունը.

$$\sum_{k=0}^{\infty}\frac{\lambda^k e^{-\lambda}}{k!}\begin{cases}(q/p)^{(z-k)} & if\ k \le z\\ 1 & if\ k > z\end{cases}$$

Վերախմբավորում ենք՝ բաշխման անսահման «պոչը» համակցելուց խուսափելու համար...

$$1-\sum_{k=0}^{z}\frac{\lambda^k e^{-\lambda}}{k!}\left(1-(q/p)^{(z-k)}\right)$$

Փոխադրելով C լեզու…

```
#include <math.h>

double AttackerSuccessProbability(double q, int z)

{

    double p = 1.0 - q;

    double lambda = z * (q / p);double
    sum = 1.0;

    int i, k;

    for (k = 0; k <= z; k++)

    {

        double poisson = exp(-lambda);for
        (i = 1; i <= k; i++)

            poisson *= lambda / i;

        sum -= poisson * (1 - pow(q / p, z - k));

    }

    return sum;
}
```

Գործարկելով որոշ արդյունքներ՝ մենք կարող ենք տեսնել, որ հավանականությունը z-ով ցուցչայնորեն իջնում է։

```
q=0.1
z=0     P=1.0000000
z=1     P=0.2045873
z=2     P=0.0509779
z=3     P=0.0131722
z=4     P=0.0034552
z=5     P=0.0009137
z=6     P=0.0002428
z=7     P=0.0000647
z=8     P=0.0000173
z=9     P=0.0000046
z=10    P=0.0000012

q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
z=30    P=0.0001522
z=35    P=0.0000379
z=40    P=0.0000095
z=45    P=0.0000024
z=50    P=0.0000006
```

Լուծելով $P < 0{,}1\%$՝ մենք ստանում ենք.

```
P < 0.001

q=0.10   z=5
q=0.15   z=8
q=0.20   z=11
q=0.25   z=15
q=0.30   z=24
q=0.35   z=41
q=0.40   z=89
q=0.45   z=340
```

12. Եզրակացություն

Մենք առաջարկեցինք էլեկտրոնային գործառնությունների ապակենտրոնացված և չմիջնորդավորված համակարգ։ Ելակետը թվային ստորագրություններից պատրաստված մետաղադրամների սովորական կառուցվածքն է, որը տրամադրում է սեփականության նկատմամբ ուժեղ վերահսկողություն, սակայն առանց կրկնակի ծախսերը կանխելու միջոցի՝ այն թերի է։ Այս խնդրին լուծում տալու համար մենք առաջարկեցինք մասնակիցը մասնակցին (peer-to-peer) ցանց՝ օգտագործելով աշխատանքի ապացույցը (proof-of-work), որպեսզի գրանցենք գործառնությունների հանրային պատմությունը, որը փոխելը հարձակվողի համար դառնում է հաշվողականորեն անիրագործելի, եթե ազնիվ հանգույցները վերահսկում են

ԿՄՀ-ի հզորության մեծ մասը: Ցանցն ամուր է իր կառուցվածքի պարզությամբ: Հանգույցներն աշխատում են միանգամից՝ քիչ համակարգվածությամբ: Նրանց նույնականացման անհրաժեշտություն չկա, քանի որ հաղորդագրությունները չեն ուղղորդվում դեպի որևէ հստակ կետ և պետք է առաքվեն միայն «best effort» համաձայնության հիման վրա: Հանգույցները կարող են դուրս գալ և նորից միանալ ցանցին ցանկացած պահի՝ հաստատելով աշխատանքի ապացույցի շղթան՝ որպես բաց թողնված գործառնությունների պատմության ապացույց: Նրանք քվեարկում են իրենց ԿՄՀ-ի հզորությամբ, վավեր բլոկների նկատմամբ հայտնում համաձայնություն՝ միևնույն ժամանակ աշխատելով ընդլայնել բլոկը, և մերժում անվավեր բլոկները՝ հրաժարվելով դրանք շարունակել: Որևէ անհրաժեշտ կանոն և խրախուսանք կարող է գործարկվել այս համաձայնության (կոնսենսուսի) մեխանիզմով:

Օգտագործված գրականության ցանկ

[1] W. Dai, "b-money," http://www.weidai.com/bmoney.txt, 1998.

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conferenceon Computer and Communications Security*, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security andPrivacy*, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957.