Bitcoin: Mfumo wa Fedha wa Kielektroniki wa Rika-kwa-Rika

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Muhtasari. Toleo halisi la rika-kwa-rika la fedha za kielektroniki lingeruhusu malipo ya mtandaoni kutumwa moja kwa moja kutoka upande mmoja kwenda mwingine bila kupitia taasisi ya kifedha. Sahihi za kidijitali zinatoa sehemu ya suluhisho, lakini faida kuu zinapotea ikiwa upande wa tatu unaoaminika bado unahitajika ili kuzuia matumizi-maradufu. Tunapendekeza suluhisho kwa tatizo la matumizi-maradufu kwa kutumia mtandao wa rika-kwa-rika. Mtandao huweka alama ya muda kwenye miamala kwa kuichanganya (hashing) katika mnyororo unaoendelea wa uthibitisho-wa-kazi (proof-of-work) unaotokana na mchanganyiko (hash), ukitengeneza rekodi ambayo haiwezi kubadilishwa bila kufanya upya uthibitisho-wa-kazi.

Mnyororo mrefu zaidi hautumiki tu kama uthibitisho wa mlolongo wa matukio yaliyoshuhudiwa, bali kama uthibitisho kwamba ulitoka katika kundi kubwa zaidi la nguvu ya CPU. Mradi idadi kubwa ya nguvu ya CPU inadhibitiwa na nodi (nodes) ambazo hazishirikiani kuushambulia mtandao, zitazalisha mnyororo mrefu zaidi na kuwashinda washambuliaji. Mtandao wenyewe unahitaji muundo mdogo. Ujumbe hutangazwa kwa msingi wa juhudi bora, na nodi zinaweza kuondoka na kujiunga tena na mtandao upendavyo, zikikubali mnyororo mrefu zaidi wa uthibitisho-wa-kazi kama uthibitisho wa kile kilichotokea wakati hazikuwepo.

1. Utangulizi

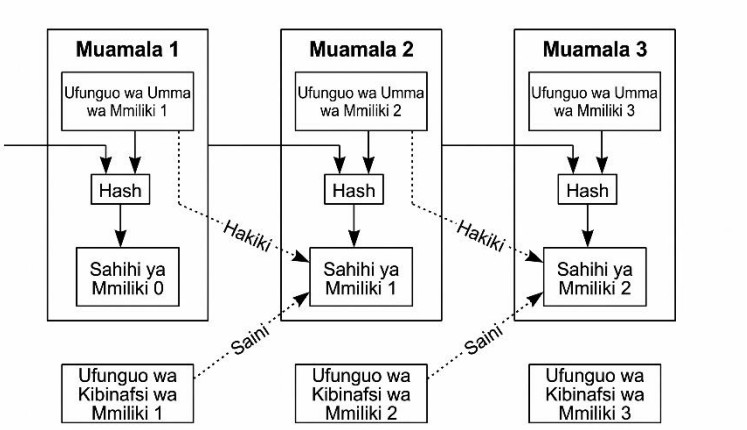
Biashara kwenye Mtandao imekuja kutegemea karibu kabisa taasisi za kifedha zinazotumika kama pande za tatu zinazoaminika ili kuchakata malipo ya kielektroniki. Wakati mfumo huu unafanya kazi vizuri vya kutosha kwa miamala mingi, bado unateseka kutokana na udhaifu wa asili wa mtindo unaotegemea uaminifu. Miamala isiyoweza kubadilishwa kabisa haiwezekani, kwani taasisi za kifedha haziwezi kuepuka kusuluhisha mizozo.

Gharama ya usuluhishi huongeza gharama za miamala, ikipunguza kiwango cha chini cha muamala kinachowezekana na kukata uwezekano wa miamala midogo ya hapa na pale, na kuna gharama kubwa zaidi katika upotezaji wa uwezo wa kufanya malipo yasiyoweza kubadilishwa kwa huduma zisizoweza kubadilishwa. Pamoja na uwezekano wa mabadiliko (reversal), hitaji la uaminifu huenea. Wafanyabiashara lazima wawe waangalifu na wateja wao, wakiwasumbua kwa taarifa zaidi kuliko ambazo wangehitaji vinginevyo. Asilimia fulani ya udanganyifu hukubalika kama jambo lisiloepukika. Gharama hizi na kutokuwa na uhakika wa malipo vinaweza kuepukika ana kwa ana kwa kutumia sarafu ya kifizikia, lakini hakuna mfumo uliopo wa kufanya malipo kupitia njia ya mawasiliano bila upande unaoaminika.

Kile kinachohitajika ni mfumo wa malipo ya kielektroniki unaozingatia uthibitisho wa kriptografia badala ya uaminifu, ukiruhusu pande mbili zozote zilizo tayari kufanya muamala moja kwa moja bila hitaji la upande wa tatu unaoaminika. Miamala ambayo kimahesabu haiwezekani kubadilishwa ingewalinda wauzaji dhidi ya udanganyifu, na mifumo ya kawaida ya amana (escrow) inaweza kutekelezwa kwa urahisi ili kuwalinda wanunuzi. Katika andiko hili, tunapendekeza suluhisho kwa tatizo la matumizi-maradufu kwa kutumia seva ya alama ya muda iliyosambazwa ya rika-kwa-rika ili kuzalisha uthibitisho wa kimahesabu wa mpangilio wa matukio wa miamala. Mfumo huu ni salama mradi nodi waaminifu kwa pamoja wanadhibiti nguvu nyingi zaidi za CPU kuliko kikundi chochote cha nodi washambuliaji wanaoshirikiana.

2. Miamala

Tunafafanua sarafu ya kielektroniki kama mnyororo wa sahihi za kidijitali. Kila mmiliki huhamisha sarafu kwa mwingine kwa kutia sahihi kidijitali mchanganyiko (hash) wa muamala uliopita na ufunguo wa umma (public key) wa mmiliki anayefuata na kuongeza hivi mwishoni mwa sarafu. Mlipwa anaweza kuhakiki sahihi ili kuhakikisha mnyororo wa umiliki.

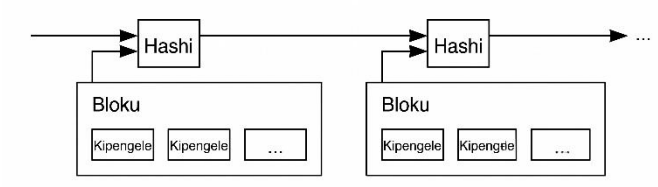


Shida bila shaka ni kwamba mlipwa hawezi kuhakiki kuwa mmoja wa wamiliki hakutumia sarafu hiyo mara mbili (double-spend). Suluhisho la kawaida ni kuanzisha mamlaka kuu inayoinamiwa, au "mint" (kiwanda cha sarafu), ambacho hukagua kila muamala kwa ajili ya matumizi-maradufu. Baada ya kila muamala, sarafu lazima irudishwe kwenye "mint" ili kutoa sarafu mpya, na ni sarafu zilizotolewa moja kwa moja kutoka kwenye "mint" pekee ndazo zinazoaminika kutotumika mara mbili. Shida ya suluhisho hili ni kwamba hatima ya mfumo mzima wa pesa inategemea kampuni inayozalisha sarafu, huku kila muamala ukilazimika kupita kwao, kama ilivyo benki.

Tunahitaji njia kwa mlipwa kujua kwamba wamiliki wa awali hawakutia sahihi miamala yoyote ya awali. Kwa malengo yetu, muamala wa kwanza kabisa ndio unaohesabiwa, kwa hivyo hatujali kuhusu majaribio ya baadaye ya kutumia mara mbili. Njia pekee ya kuthibitisha kutokuwepo kwa muamala ni kuwa na ufahamu wa miamala yote. Katika mtindo unaozingatia "mint", mint ilikuwa na ufahamu wa miamala yote na kuamua ni upi uliofika kwanza. Ili kufanikisha hili bila upande unaoaminika, miamala lazima itangazwe hadharani [1], na tunahitaji mfumo wa washiriki kukubaliana juu ya historia moja ya mpangilio ambao miamala hiyo ilipokelewa. Mlipwa anahitaji uthibitisho kwamba wakati wa kila muamala, idadi kubwa ya nodi (nodes) ilikubali kuwa ndio uliopokelewa kwanza.

3. Seva ya Alama ya Muda (Timestamp Server)

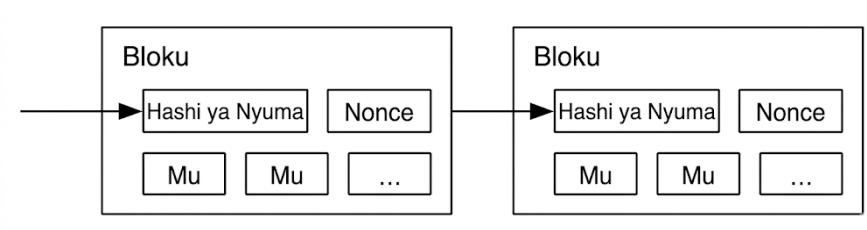
Suluhisho tunalopendekeza linaanza na seva ya alama ya muda. Seva ya alama ya muda hufanya kazi kwa kuchukua mchanganyiko (hash) wa kundi la vitu (block of items) vitakavyowekwa alama ya muda na kuchapisha mchanganyiko huo kwa mapana, kama vile kwenye gazeti au chapisho la Usenet [2-5]. Alama ya muda inathibitisha kwamba data lazima ilikuwepo wakati huo, ni wazi, ili kuingia kwenye mchanganyiko huo. Kila alama ya muda inajumuisha alama ya muda iliyotangulia katika mchanganyiko wake, ikitengeneza mnyororo, huku kila alama ya muda ya ziada ikiimarisha zile zilizotangulia.



4. Uthibitisho-wa-Kazi (Proof-of-Work)

Ili kutekeleza seva ya alama ya muda iliyosambazwa kwa msingi wa rika-kwa-rika, tutahitaji kutumia mfumo wa uthibitisho-wa-kazi unaofanana na Hashcash ya Adam Back [6], badala ya magazeti au machapisho ya Usenet. Uthibitisho-wa-kazi unahusisha kutafuta thamani ambayo inapochanganywa (hashed), kama vile kwa SHA-256, mchanganyiko (hash) huanza na idadi fulani ya biti za sifuri. Wastani wa kazi inayohitajika huongezeka kielelezo (exponential) kulingana na idadi ya biti za sifuri zinazohitajika na inaweza kuhakikiwa kwa kutekeleza hashi moja tu.

Kwa mtandao wetu wa alama ya muda, tunatekeleza uthibitisho-wa-kazi kwa kuongeza namba ya nasibu (*nonce*) katika bloku hadi thamani ipatikane ambayo inaipatia hashi ya bloku biti za sifuri zinazohitajika. Mara tu juhudi za CPU zimeshatumika kuifanya itosheleze uthibitisho-wa-kazi, bloku haiwezi kubadilishwa bila kufanya upya kazi hiyo. Kadiri bloku za baadaye zinavyounganishwa baada yake, kazi ya kubadilisha bloku hiyo itajumuisha kufanya upya bloku zote zinazofuata.



Uthibitisho-wa-kazi pia unatatua tatizo la kuamua uwakilishi katika ufanyaji maamuzi wa wengi. Ikiwa wengi wangezingatia anwani-moja-ya-IP-kura-moja, mfumo ungeweza kuvurugwa na mtu yeyote anayeweza kutenga IP nyingi. Uthibitisho-wa-kazi kimsingi ni CPU-moja-kura-moja. Uamuzi wa wengi unawakilishwa na mnyororo mrefu zaidi, ambao una juhudi kubwa zaidi ya uthibitisho-wa-kazi iliyowekezwa ndani yake. Ikiwa idadi kubwa ya nguvu ya CPU inadhibitiwa na nodi waaminifu, mnyororo wa waaminifu utakua kwa kasi zaidi na kuzipita mnyororo wowote shindani. Ili kurekebisha bloku ya zamani, mshambuliaji angelazimika kufanya upya uthibitisho-wa-kazi wa bloku hiyo na bloku zote baada yake na kisha kuifikia na kuipita kazi ya nodi waaminifu. Tutaonyesha baadaye kwamba uwezekano wa mshambuliaji aliye polepole kuwafikia unapungua kielelezo kadiri bloku zinazofuata zinavyoongezwa.

Ili kufidia ongezeko la kasi ya maunzi (hardware) na mabadiliko ya maslahi katika kuendesha nodi kwa muda, ugumu wa uthibitisho-wa-kazi unamuliwa na wastani unaobadilika unaolenga idadi fulani ya wastani ya bloku kwa saa. Ikiwa zinazalishwa kwa kasi sana, ugumu huongezeka.

5. Mtandao

Hatua za kuendesha mtandao ni kama ifuatavyo:

1. Miamala mipya inatangazwa kwa nodi zote.
2. Kila nodi hukusanya miamala mipya kwenye bloku.
3. Kila nodi inafanya kazi kutafuta uthibitisho-wa-kazi mgumu kwa ajili ya bloku yake.
4. Nodi inapopata uthibitisho-wa-kazi, inatangaza bloku hiyo kwa nodi zote.
5. Nodi zinakubali bloku hiyo ikiwa tu miamala yote ndani yake ni sahihi na haijatumiwa tayari.
6. Nodi zinaonyesha kukubali kwao bloku hiyo kwa kufanya kazi ya kuunda bloku inayofuata katika mnyororo, zikitumia hashi ya bloku iliyokubaliwa kama hashi iliyotangulia.

Nodi daima huchukulia mnyororo mrefu zaidi kuwa ndio sahihi na zitaendelea kufanya kazi ya kuurefusha. Ikiwa nodi mbili zinatangaza matoleo tofauti ya bloku inayofuata kwa wakati mmoja, baadhi ya nodi zinaweza kupokea moja au nyingine kwanza. Katika hali hiyo, zinafanya kazi kwenye ile ya kwanza waliyoipokea, lakini zinahifadhi tawi lingine ikiwa litakuwa refu zaidi. Sare hiyo itavunjwa wakati uthibitisho-wa-kazi

unaofuata utakapopatikana na tawi moja kuwa refu zaidi; nodi ambazo zilikuwa zikifanya kazi kwenye tawi lingine basi zitahamia kwenye lile refu zaidi.

Matangazo ya miamala mipya si lazima yafikie nodi zote. Mradi yafikie nodi nyingi, yataingia kwenye bloku baada ya muda mfupi. Matangazo ya bloku pia yanastahimili ujumbe uliopotea. Ikiwa nodi haipokei bloku, itaiomba itakapopokea bloku inayofuata na kutambua kuwa imekosa moja.

6. Motisha (Incentive)

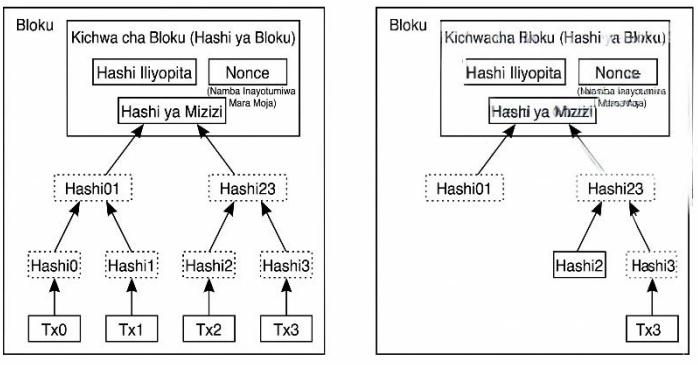
Kwa kawaida, muamala wa kwanza katika bloku ni muamala maalum unaoanzisha sarafu mpya inayomilikiwa na muundaji wa bloku hiyo. Hii inaongeza motisha kwa nodi kusaidia mtandao, na inatoa njia ya kusambaza sarafu mwanzoni kwenye mzunguko, kwani hakuna mamlaka kuu ya kuzitoa. Ongezeko la mara kwa mara la kiasi fulani cha sarafu mpya linafanana na wachimbaji dhahabu wanaotumia rasilimali kuongeza dhahabu kwenye mzunguko. Katika hali yetu, ni muda wa CPU na umeme ndio unaotumika.

Motisha pia inaweza kufadhiliwa kwa ada za miamala. Ikiwa thamani ya pato (output) ya muamala ni ndogo kuliko thamani ya pembejeo (input) yake, tofauti hiyo ni ada ya muamala ambayo huongezwa kwenye thamani ya motisha ya bloku iliyo na muamala huo. Mara baada ya idadi iliyopangwa mapema ya sarafu kuingia kwenye mzunguko, motisha inaweza kuhamia kikamilifu kwenye ada za miamala na kuwa haina mfumuko wa bei kabisa.

Motisha inaweza kusaidia kuhimiza nodi kubaki waaminifu. Ikiwa mshambuliaji mlafi anaweza kukusanya nguvu nyingi zaidi za CPU kuliko nodi zote waaminifu, atalazimika kuchagua kati ya kuitumia kuwahadaa watu kwa kuiba malipo yake ya nyuma, au kuitumia kuzalisha sarafu mpya. Anapaswa kuona ni faida zaidi kucheza kwa kufuata sheria, sheria ambazo zinampendelea kwa sarafu mpya nyingi zaidi kuliko kila mtu mwingine akijumuishwa, kuliko kudhoofisha mfumo na uhalali wa utajiri wake mwenyewe.

7. Kudai Tena Nafasi ya Diski (Reclaiming Disk Space)

Mara tu muamala wa hivi punde katika sarafu unapozikwa chini ya bloku za kutosha, miamala iliyotumika kabla yake inaweza kutupwa ili kuokoa nafasi ya diski. Ili kuwezesha hili bila kuharibu hashi (hash) ya bloku, miamala huchanganywa (hashed) katika Mti wa Merkle (Merkle Tree) [7][2][5], huku mzizi pekee ukiwa umejumuishwa kwenye hashi ya bloku. Bloku za zamani zinaweza kisha kupunguzwa kwa kukata matawi ya mti huo. Hashi za ndani hazihitaji kuhifadhiwa.

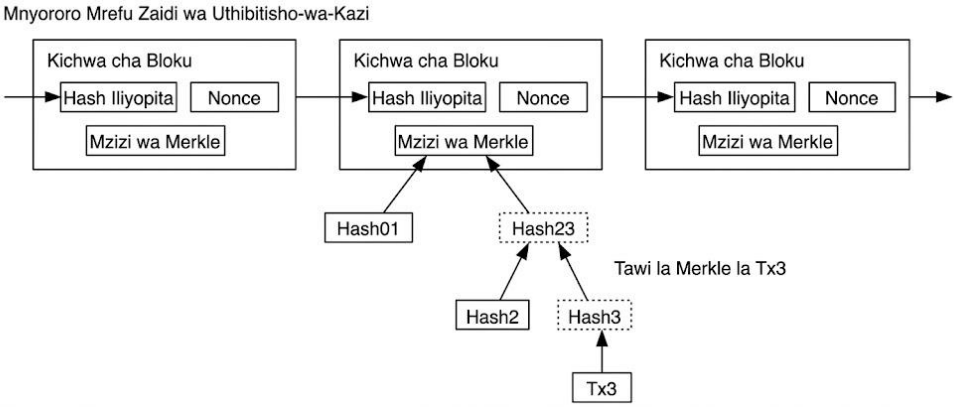


Miamala Iliyopunguzwa katika Mti wa Merkle | Baada ya Kupunguza Tx0-2 kutoka Bloku

Kichwa cha bloku (block header) kisicho na miamala kingekuwa na takriban baiti (bytes) 80. Ikiwa tunakadiria bloku zinazalishwa kila dakika 10, baiti 80 * 6 * 24 * 365 = 4.2MB kwa mwaka. Kwa mifumo ya kompyuta ambayo kwa kawaida inauzwa ikiwa na RAM ya 2GB kufikia mwaka 2008, na Sheria ya Moore (Moore's Law) ikitabiri ukuaji wa sasa wa 1.2GB kwa mwaka, uhifadhi haupaswi kuwa tatizo hata kama vichwa vya bloku lazima vihifadhiwe kwenye kumbukumbu (memory).

8. Uhakiki wa Malipo Rahisi (Simplified Payment Verification)

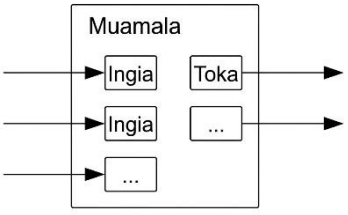
Inawezekana kuhakiki malipo bila kuendesha nodi kamili ya mtandao. Mtumiaji anahitaji tu kuwa na nakala ya vichwa vya bloku vya mnyororo mrefu zaidi wa uthibitisho-wa-kazi, ambao anaweza kuupata kwa kuuliza nodi za mtandao hadi atakapojiridhisha kuwa anao mnyororo mrefu zaidi, na kupata tawi la Merkle linalounganisha muamala na bloku uliowekewa alama ya muda. Hawezi kuukagua muamala yeye mwenyewe, lakini kwa kuuunganisha na sehemu fulani katika mnyororo, anaweza kuona kwamba nodi ya mtandao imekubali, na bloku zilizoongezwa baada yake zinathibitisha zaidi kuwa mtandao umeukubali.



Kama ilivyo, uhakiki ni wa kuaminika mradi nodi waaminifu wanadhibiti mtandao, lakini una uwezekano mkubwa wa kuvamiwa ikiwa mtandao utazidiwa nguvu na mshambuliaji. Wakati nodi za mtandao zinaweza kuhakiki miamala zenyewe, njia iliyorahisishwa inaweza kudanganywa na miamala ya kughushi ya mshambuliaji kwa muda mrefu mradi mshambuliaji anaweza kuendelea kuuzidi nguvu mtandao. Mkakati mmoja wa kujilinda dhidi ya hili ungekuwa kukubali arifa kutoka kwa nodi za mtandao zinapogundua bloku batili, ikichochea programu ya mtumiaji kupakua bloku nzima na miamala iliyoarifiwa ili kuthibitisha kutolingana huku. Biashara zinazopokea malipo ya mara kwa mara pengine bado zitapenda kuendesha nodi zao wenyewe kwa ajili ya usalama huru zaidi na uhakiki wa haraka zaidi.

9. Kuchanganya na Kugawanya Thamani (Combining and Splitting Value)

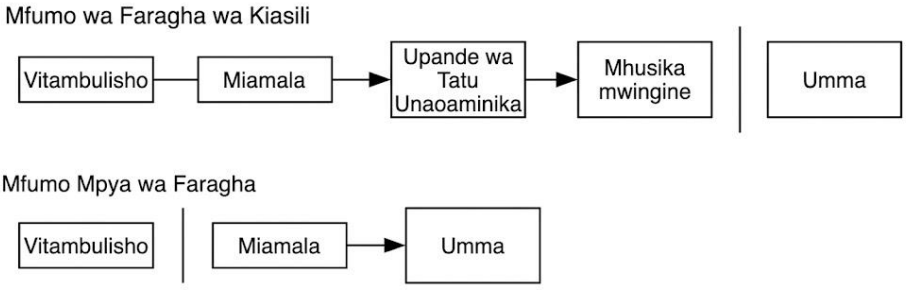
Ingawa ingewezekana kushughulikia sarafu moja moja, ingekuwa vigumu kufanya muamala tofauti kwa kila senti katika uhamisho. Ili kuruhusu thamani kugawanywa na kuchanganywa, miamala huwa na pembejeo (inputs) na matokeo (outputs) mengi. Kwa kawaida kutakuwa na pembejeo moja kutoka kwa muamala mkubwa uliopita au pembejeo nyingi zinazochanganya kiasi kidogo, na zaidi ya matokeo mawili: moja kwa ajili ya malipo, na moja ya kurudisha chenji, ikiwa ipo, kurudi kwa mtumaji.



Inapaswa kuzingatiwa kuwa kusambaa kwa miamala (*fan-out*), ambapo muamala mmoja unategemea miamala mingine kadhaa, na miamala hiyo inategemea mingine mingi zaidi, si tatizo hapa. Hakuna kamwe hitaji la kutoa nakala kamili inayojitegemea ya historia ya muamala.

10. Faragha (Privacy)

Mtindo wa jadi wa kibenki unafanikisha kiwango fulani cha faragha kwa kuzuia upatikanaji wa taarifa kwa wahusika wanaohusika na upande wa tatu unaoaminika. Ulazima wa kutangaza miamala yote hadharani unazuia njia hii, lakini faragha bado inaweza kudumishwa kwa kukata mtiririko wa taarifa katika sehemu nyingine: kwa kuweka funguo za umma (*public keys*) bila majina. Umma unaweza kuona kwamba kuna mtu anamtumia kiasi fulani mtu mwingine, lakini bila taarifa inayounganisha muamala huo na mtu yeyote. Hii inafanana na kiwango cha taarifa zinazotolewa na soko la hisa, ambapo muda na ukubwa wa biashara binafsi, ile "tape", huwekwa hadharani, lakini bila kusema wahusika walikuwa nani.



Kama kizuizi cha ziada (firewall), jozi mpya ya ufunguo inapaswa kutumiwa kwa kila muamala ili kuzuia miamala hiyo kuunganishwa na mmiliki mmoja. Baadhi ya uunganishaji bado hauepukiki na miamala ya pembejeo-nyingi (multi-input), ambayo lazima ifunue kuwa pembejeo zao zilikuwa zinamilikiwa na mmiliki huyo huyo. Hatari ni kwamba ikiwa mmiliki wa ufunguo atafunuliwa, uunganishaji unaweza kufunua miamala mingine iliyomilikiwa na mmiliki huyo huyo.

11. Mahesabu (Calculations)

Tunafikiria mazingira ya mshambuliaji anayejaribu kuzalisha mnyororo mbadala kwa kasi zaidi kuliko mnyororo wa waaminifu. Hata kama hili likifanikiwa, halifanyi mfumo kuwa wazi kwa mabadiliko ya kiholela, kama vile kuunda thamani kutoka kusikojulikana au kuchukua pesa ambazo hazijawahi kumilikiwa na mshambuliaji. Nodi hazitakubali muamala batili kama malipo, na nodi waaminifu hazitakubali kamwe bloku iliyo na miamala hiyo. Mshambuliaji anaweza tu kujaribu kubadilisha mmoja wa miamala yake mwenyewe ili kurudisha pesa alizotumia hivi karibuni.

Mbio kati ya mnyororo wa waaminifu na mnyororo wa mshambuliaji zinaweza kufafanuliwa kama Matembezi ya Nasibu ya Binomial (Binomial Random Walk). Tukio la mafanikio ni mnyororo wa waaminifu kurefushwa kwa bloku moja, ikiongeza uongozi wake kwa +1, na tukio la kushindwa ni mnyororo wa mshambuliaji kurefushwa kwa bloku moja, ikipunguza mwanya kwa -1.

Uwezekano wa mshambuliaji kuwafikia (catching up) kutoka kwa upungufu (deficit) fulani unafanana na tatizo la Gambler's Ruin (Maangamizi ya Mcheza Kamari). Tuseme mcheza kamari mwenye mkopo usio na kikomo anaanza na upungufu na anacheza majaribio mengi yasiyo na mwisho ili kujaribu kufikia usawa (breakeven). Tunaweza kuhesabu uwezekano wa yeye kufikia usawa, au kwamba mshambuliaji atawahi kuufikia mnyororo wa waaminifu, kama ifuatavyo [8]:

p = uwezekano wa nodi mwaminifu kupata bloku inayofuata
q = uwezekano wa mshambuliaji kupata bloku inayofuata
$\mathbf{q_z}$ = uwezekano wa mshambuliaji kuwahi kufikia (catch up) akiwa nyuma kwa bloku **z**

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

Kwa kuzingatia dhana yetu kwamba **p > q**, uwezekano hupungua kielelezo (exponentially) kadiri idadi ya bloku ambazo mshambuliaji anapaswa kuzifikia inavyoongezeka. Kukiwa na uwezekano mdogo dhidi yake,

ikiwa hafanyi jaribio la bahati la kusonga mbele mapema, nafasi zake zinakuwa ndogo sana kadiri anavyozidi kuachwa nyuma.

Sasa tunafikiria ni muda gani mpokeaji wa muamala mpya anahitaji kusubiri kabla ya kuwa na uhakika wa kutosha kwamba mtumaji hawezi kubadilisha muamala huo. Tunadhani mtumaji ni mshambuliaji ambaye anataka kumfanya mpokeaji aamini kuwa amemlipa kwa muda, kisha aubadilishe ili ajilipe mwenyewe baada ya muda fulani kupita. Mpokeaji ataalamishwa jambo hilo likitokea, lakini mtumaji anatumaini kuwa itakuwa imekwishachelewa.

Mpokeaji anazalisha jozi mpya ya ufunguo na kumpa mtumaji ufunguo wa umma muda mfupi kabla ya kutia sahihi. Hii inamzuia mtumaji kuandaa mnyororo wa bloku mapema kwa kuufanyia kazi mfululizo hadi awe na bahati ya kutosha kufika mbali zaidi mbele, kisha kutekeleza muamala wakati huo. Mara tu muamala unapotumwa, mtumaji asiye mwaminifu anaanza kufanya kazi kwa siri kwenye mnyororo sambamba ulio na toleo mbadala la muamala wake.

Mlipwa anasubiri hadi muamala utakapoongezwa kwenye bloku na bloku **z** ziwe zimeunganishwa baada yake. Hajui kiasi halisi cha maendeleo ambayo mshambuliaji amefanya, lakini kwa kudhani kuwa bloku waaminifu zilichukua wastani wa muda unaotarajiwa kwa kila bloku, maendeleo yanayoweza kufanywa na mshambuliaji yatakuwa Usambazaji wa Poisson (Poisson distribution) ukiwa na thamani inayotarajiwa:

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

Ili kupata uwezekano kwamba mshambuliaji bado anaweza kufikia (catch up) sasa, tunazidisha msongamano wa Poisson (Poisson density) kwa kila kiasi cha maendeleo ambacho angeweza kuwa amefanya kwa uwezekano kwamba angeweza kufikia kutoka hatua hiyo:

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

Kupanga upya ili kuepuka kujumlisha mkia usio na mwisho wa usambazaji (distribution)...

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

Inabadilisha kwenda kwenye kodi ya C...

```
#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;
}
```

Tukichakata baadhi ya matokeo, tunaweza kuona uwezekano ukipungua kielelezo (exponentially) kulingana na **z**.

```
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
```

Tukitafuta suluhisho la **P** chini ya **10%**...

```
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. Hitimisho (Conclusion)

Tumependekeza mfumo wa miamala ya kielektroniki bila kutegemea uaminifu. Tulianza na mfumo wa kawaida wa sarafu zilizoundwa kwa sahihi za kidijitali, ambao unatoa udhibiti thabiti wa umiliki, lakini haujakamilika bila njia ya kuzuia matumizi-maradufu. Ili kutatua hili, tulipendekeza mtandao wa rika-kwa-rika ukitumia uthibitisho-wa-kazi kurekodi historia ya umma ya miamala ambayo haraka inakuwa haiwezekani kimahesabu kwa mshambuliaji kuibadilisha ikiwa nodi waaminifu wanadhibiti idadi kubwa ya nguvu ya CPU.

Mtandao ni imara katika urahisi wake usio na muundo maalum. Nodi hufanya kazi zote kwa mara moja kwa uratibu mdogo. Hazihitaji kutambuliwa, kwani ujumbe hauelekezwi mahali popote mahususi na unahitaji tu kufikishwa kwa msingi wa juhudi bora. Nodi zinaweza kuondoka na kujiunga tena na mtandao upendavyo, zikikubali mnyororo wa uthibitisho-wa-kazi kama uthibitisho wa kile kilichotokea wakati hazikuwepo. Zinapiga kura kwa nguvu zao za CPU, zikionyesha kukubali kwao bloku sahihi kwa kufanya kazi ya kuzirefusha na kukataa bloku batili kwa kugoma kuzifanyia kazi. Sheria na motisha zozote zinazohitajiwa zinaweza kusimamiwa kupitia mfumo huu wa maelewano (consensus mechanism).

Marejeo

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

[2] H. Massias, X.S. Avila, na J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," (Ubunifu wa huduma salama ya kuweka alama ya muda kukiwa na mahitaji madogo ya uaminifu), katika *Mkutano wa 20 wa Benelux wa Nadharia ya Habari*, Mei 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," (Jinsi ya kuweka alama ya muda kwenye hati ya kidijitali), katika *Journal of Cryptology*, juzuu la 3, na. 2, kurasa 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," (Kuboresha ufanisi na uaminifu wa uwekaji alama ya muda wa kidijitali), katika *Sequences II: Methods in Communication, Security and Computer Science*, kurasa 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," (Majina salama kwa mfuatano wa biti), katika *Proceedings of the 4th ACM Conference on Computer and Communications Security*, kurasa 28-35, Aprili 1997.

[6] Back, "Hashcash - a denial of service counter-measure," (Hashcash - kipimo cha kuzuia kunyimwa huduma), http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," (Itifaki za mifumo ya kriptografia ya funguo za umma), katika *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, kurasa 122-133, Aprili 1980.

[8] W. Feller, "An Introduction to Probability Theory and its Applications," (Utangulizi wa Nadharia ya Uwezekano na Maombi yake), 1957.

Translated by:
Collins Mwanga
Bitsacco Community Developer & Educator