Bitcoin: A Peer-to-Peer Electronic Cash System
KHMER LANGUAGE

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

Abstract. សាច់ប្រាក់អេឡិចត្រូនិកដែលប្រើប្រាស់ Peer-To-Peer ពិតប្រាកដណាស់អាចឱ្យការទូទាត់តាមប្រព័ន្ធអ៊ីនធឺណិតពីមនុស្សម្នាក់ទៅមនុស្សម្នាក់ទៀត ដោយមិនចាំបាច់មានស្ថាប័នហិរញ្ញវត្ថុដូចជាធនាគារដើម្បីដើរតួជាអន្តរការីនេះទេ។ ហត្ថលេខាឌីជីថលគឺជាផ្នែកមួយដែលធ្វើឱ្យទម្រង់នៃការទូទាត់តាមអ៊ីនធឺណិតនេះអាចដំណើរការ និងប្រតិបត្តិការធ្វើទៅបាន ប៉ុន្តែវានឹងមិនអាចទៅរួចទេ ប្រសិនបើប្រព័ន្ធនៅតែត្រូវការការជឿទុកចិត្តពីអន្តរការីផ្សេងៗ ដើម្បីការពារកំហុសដូចជាការចំណាយទ្វេដង យើងផ្តល់ជូននូវដំណោះស្រាយសម្រាប់ការចំណាយទ្វេដងដោយប្រើនូវ Peer-To-Peer Network បណ្តាញនេះកត់ត្រានូវពេលវេលារាល់ការធ្វើប្រតិបត្តិការដោយ Hashing ចូលទៅក្នុងខ្សែសង្វាក់ដោយផ្អែកលើ Proof-Of-Work បន្ទាប់មកបង្កើតជាកំណត់ត្រាដែលមិនអាចធ្វើការផ្លាស់ប្តូរបាន និងក៏មិនចាំបាច់ធ្វើឡើងវិញនូវ Proof-Of-Work នេះម្តងទៀតនេះទេ។ បន្ទាប់មកប្រតិបត្តិការដែលបានអ៊ិនគ្រីបត្រូវបានចូលរួមនៅក្នុងខ្សែសង្វាក់នៃប្រតិបត្តិការផ្សេងទៀតហើយក៏ដូចជាដើម្បីកត់ត្រាប្រតិបត្តិការទាំងនេះ ពួកគេត្រូវតែឆ្លងកាត់ Proof-Of-Work តែប៉ុណ្ណោះ។

ខ្សែសង្វាក់វែងបំផុតមិនត្រឹមតែជាភស្តុតាងនៃលំដាប់នៃព្រឹត្តិការណ៍ដែលបានឃើញនេះទេ ប៉ុន្តែអាចបញ្ជាក់ថាវាបានមកពី Pool ដែលធំបំផុតនៃកម្លាំងរបស់ CPU។ Proof-of-work របស់ Nodes (កុំព្យូទ័រក្នុងបណ្តាញ) Node ជាច្រើននឹងប្រើ CPUs ដើម្បីដោះស្រាយបញ្ហាដើម្បីផ្ទៀងផ្ទាត់ភាពត្រឹមត្រូវនៃប្រតិបត្តិការដែលកើតឡើង ដរាបណាថាមពល CPUs ភាគច្រើនរបស់ Node មិនត្រូវបានប្រើដើម្បីវាយប្រហារប្រព័ន្ធនោះទេ ខ្សែសង្វាក់នេះនឹងបន្តពង្រីកហើយវានឹងរារាំងអ្នកដែលគិតចង់វាយប្រហារប្រព័ន្ធពីការបង្កើតខ្សែសង្វាក់។ ប្រព័ន្ធបណ្តាញនេះមិនត្រូវការរចនាសម្ព័ន្ធស្មុគស្មាញទេ ហើយវិធីសាស្ត្រទំនាក់ទំនងក្នុងបណ្តាញគឺជាការខិតខំប្រឹងប្រែងដ៏ល្អបំផុត (ការបញ្ជូនត្រូវបានបញ្ជូនដោយមិនគិតពីអ្នកទទួល) Node នីមួយៗអាចចាកចេញពីបណ្តាញបានហើយត្រលប់មកវិញនៅពេលណាក៏បាន ដោយសារតែខ្សែសង្វាក់វែងបំផុតតែងតែប្រាប់ពីអ្វីដែលបានកើតឡើងនៅពេលដែល Node ទាំងនោះបាត់។

I. សេចក្តីផ្តើម

ការជួញដូរតាមអ៊ីនធឺណិតជាទូទៅពឹងផ្អែកលើស្ថាប័នហិរញ្ញវត្ថុជាភាគីទីបីដែលសមនឹងទទួលបានឥណទាន ឬអាចទុកចិត្តបាន។ ដើម្បីបញ្ជាក់ភាពត្រឹមត្រូវនៃប្រតិបត្តិការអនឡាញនោះ ទោះបីជាប្រព័ន្ធអន្តរការីនេះដំណើរការល្អសម្រាប់ប្រតិបត្តិការភាគច្រើនក៏ដោយ ប៉ុន្តែនៅតែមានចំណុចខ្សោយដែលប្រព័ន្ធនេះនៅតែត្រូវការការជឿទុកចិត្តដដែល (ឥណទាន ឬទំនុកចិត្តក្នុងន័យហិរញ្ញវត្ថុ) ដើម្បីចូលរួម។

ការចំណាយតាមរយៈអន្តរការីទាំងនេះ មានឱកាសដែលប្រតិបត្តិការអាចនឹងខុស ឬអាចត្រូវបានលុបចោល នេះដោយសារតែស្ថាប័នហិរញ្ញវត្ថុអន្តរការីអាចជួបប្រទះនឹងជម្លោះរវាងអន្តរការី (ឧ. ការបង់ប្រាក់ដែលគ្មានការអនុញ្ញាត មូលប្ប

ទានប័ត្រ ឬប័ណ្ណឥណទានដែលផុតកំណត់) ការមានអន្តរការីបង្កើតការចំណាយបន្ថែម និងថ្លៃប្រតិបត្តិការ។ ហើយការចំណាយនេះរារាំងអ្នកប្រើប្រាស់មិនឱ្យធ្វើប្រតិបត្តិការតូចៗ។ (ដោយសារតែតម្លៃនៃប្រតិបត្តិការមានតម្លៃថ្លៃជាងចំនួនប្រតិបត្តិការ) ហើយមានការចំណាយផងដែរដែលយើងប្រហែលជាត្រូវចំណាយ។

លទ្ធភាពដែលការទូទាត់របស់យើងនឹងមិនដោគជ័យ។ វាធ្វើឱ្យយើងពិបាកក្នុងការចំណាយប្រាក់លើសេវាកម្មទូទាត់តាមអ៊ីនធឺណិត។ ជាថ្នូរនឹងសេវាទូទាត់ វាមានលទ្ធភាពដែលប្រតិបត្តិការអាចនឹងខុស រួមទាំង មិនអាចត្រឡប់ទៅវិញបានទេ។ ប៉ុន្តែយើងបានទទួលសេវាកម្មរួចហើយ អាចធ្វើឱ្យយើងបាត់បង់ប្រាក់ ពេលវេលា និងឥណទាន) នៅពេលដែលប្រតិបត្តិការរបស់យើងមានសក្តានុពលក្នុងការមិនដោគជ័យ មិនត្រូវបានផ្ទៀងផ្ទាត់ ឬមិនត្រូវបានអនុម័តដោយអន្តរការីដែលដំណើរការប្រតិបត្តិការ ដូច្នេះ ភាពជឿជាក់នៃអន្តរការីគឺសំខាន់ជាង។

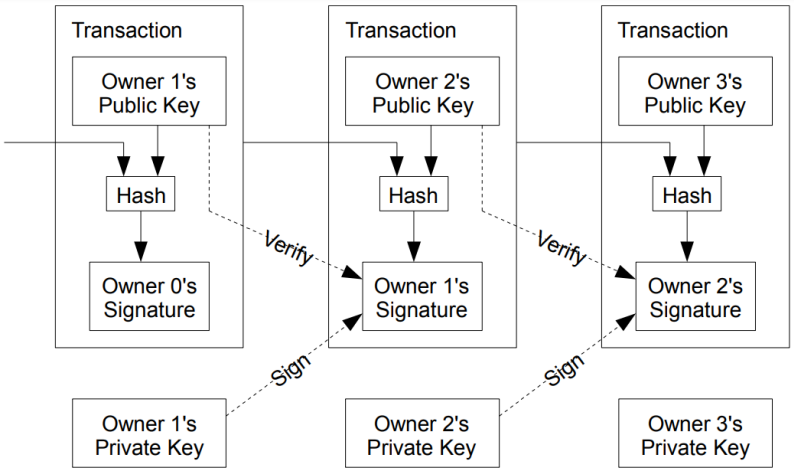
សូម្បីតែអាជីវកែខ្លួនឯងក៏ជួបបញ្ហាច្រើនជាងនេះដែរ។ នៅពេលអតិថិជនស្នើសុំព័ត៌មានដែលលើសពីការចាំបាច់ដើម្បីបញ្ជាក់ពីភាពគួរឱ្យទុកចិត្ត (ដូចជាការស្នើសុំព័ត៌មានផ្ទាល់ខ្លួនពីពាណិជ្ជករ។ នេះគឺជាអ្វីដែលអាចត្រូវបានអភិវឌ្ឍបន្ថែមទៀតដើម្បីប្រព្រឹត្តបទឧក្រិដ្ឋ) ដោយសារតែអ្នកទិញគិតថាការបោកប្រាស់អាចកើតឡើង។

ការចំណាយបន្ថែមដ៏មានសក្តានុពលនៃប្រព័ន្ធទូទាត់តាមអ៊ីនធឺណិតដែលមានកំហុសអាចត្រូវបានយកឈ្នះដោយប្រើរូបិយប័ណ្ណរូបវន្តជំនួសវិញ។ ប៉ុន្តែប្រើលុយពិត ឬតាមវិធីណាក៏ដោយ មិនអាចលុបបំបាត់ឈ្មួញកណ្តាលក្នុងការបង់ប្រាក់បានទេ។ (អន្តរការីសម្រាប់ប្រាក់ពិតប្រាកដគឺជារដ្ឋាភិបាលដែលកំណត់តម្លៃនៃប្រាក់) ហើយនៅពេលដែលអន្តរការីមិនអាចត្រូវបានគេលុបបំបាត់បាន យើងនឹងត្រូវខ្ជះខ្ជាយលុយដែលមិនចាំបាច់លើប្រតិបត្តិការ។ សេរីភាពក្នុងការធ្វើប្រតិបត្តិការ ដូច្នេះហើយការទិញទំនិញតាមអ៊ីនធឺណិតមានកំណត់។

អ្វីដែលយើងត្រូវការគឺប្រព័ន្ធទូទាត់អេឡិចត្រូនិចដែលផ្អែកលើ Cryptographic Proof ជំនួសឱ្យការប្រើអន្តរការីហិរញ្ញវត្ថុដែលអាចទុកចិត្តបាន។ នេះនឹងអនុញ្ញាតឱ្យមនុស្សពីរនាក់ធ្វើប្រតិបត្តិការជាមួយគ្នាដោយមិនចាំបាច់មានឈ្មួញកណ្តាលដើម្បីគ្រប់គ្រងពួកគេនេះទេ។ ប្រតិបត្តិការដែលបានបញ្ជាក់មិនអាចត្រូវបានប្តូរឬលុបចោលបានទេ នេះជួយការពារអ្នកលក់ពីការក្លែងបន្លំទម្រង់ផ្សេងៗ។ ហើយប្រព័ន្ធ escrow ស្វ័យប្រវត្តិអាចត្រូវបានបង្កើតយ៉ាងងាយស្រួលក្នុងការការពារអ្នកទិញ។ ក្នុងឯកសារនេះ យើងបង្ហាញពីវិធីសាស្ត្រការពារការចំណាយទ្វេដងដោយសកម្មក្នុងប្រព័ន្ធ peer-to-peer ដែលកត់ត្រាពេលវេលានៃប្រតិបត្តិការ។ ហើយនឹងបង្កើតបញ្ហាដែលទាមទារថាមពលស៊ីភីយូដើម្បីដោះស្រាយ ដើម្បីបញ្ជាក់ថាពួកគេពិតជាបានកើតឡើង និងកើតឡើងតាមលំដាប់ពេលវេលាត្រឹមត្រូវ។ មិនបានកែសម្រួល ឬត្រូវបានរំខានដោយអ្នកដែលចង់វាយប្រហារប្រព័ន្ធ ប្រព័ន្ធមានសុវត្ថិភាព ប្រសិនបើ Node ភាគច្រើននៅតែប្រើថាមពលស៊ីភីយូ ដើម្បីបញ្ជាក់ប្រតិបត្តិការ វាមិនត្រូវបានប្រើដើម្បីវាយប្រហារប្រព័ន្ធខ្លួនឯងនេះទេ។

II. ប្រតិបត្តិការ (Transactions)

ពួកយើងបានកំណត់កាក់ឌីជីថលគឺជាខ្សែសង្វាក់នៃហត្ថលេខាឌីជីថល ដែលម្ចាស់កាក់នីមួយៗផ្ទេកាក់ទៅមនុស្សបន្ទាប់ដោយការចុះហត្ថលេខាជាឌីជីថលនៅក្នុងលេខហត្ថលេខានៃប្រតិបត្តិការមុន និង Public Key នៃម្ចាស់បន្ទាប់ដោយបន្ថែមវាទៅចុងបញ្ចប់នៃកាក់ អ្នកទទួលប្រាក់នឹងអាចផ្ទៀងផ្ទាត់ហត្ថលេខាដើម្បីផ្ទៀងផ្ទាត់ខ្សែសង្វាក់នៃភាពជាម្ចាស់។



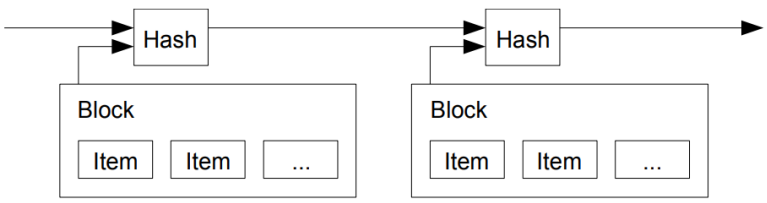
ពិតណាស់បញ្ហាគឺអ្នកទទួលប្រាក់មិនអាចផ្ទៀងផ្ទាត់ថាម្ចាស់ម្នាក់ណា មិនបានចំណាយទ្វេដងនោះទេ (double-spend) វិធីសាមញ្ញមួយដើម្បីដោះស្រាយបញ្ហានេះគឺត្រូវប្រើ អាជ្ញាធរកណ្តាលដែលគួរឱ្យទុកចិត្ត ឬ Mint ត្រួតពិនិត្យរាល់ប្រតិបត្តិការសម្រាប់ការប្រើប្រាស់ឡើងវិញថាតើ (Double-Spend) ឬអត់។ ក្នុងរាល់ប្រតិបត្តិការ កាក់ត្រូវតែត្រលប់ទៅអន្តរការី ដើម្បីបង្កើតកាក់ថ្មី ហើយមានតែកាក់ដែលត្រូវបាន Mint យកពីអន្តរការីប៉ុណ្ណោះ ដែលយើងអាចជឿជាក់បានថា វាមិនត្រូវបានគេរៀបចំឡើងវិញ (Double-Spend)។

បញ្ហានៃវិធីសាស្ត្រនេះគឺថាអ្វីគ្រប់យ៉ាងនៅក្នុងប្រព័ន្ធហិរញ្ញវត្ថុអាស្រ័យលើអង្គការអន្តរការី រាល់ប្រតិបត្តិការត្រូវតែឆ្លងកាត់អន្តរការីនេះដូចទៅនឹងប្រព័ន្ធធនាគារដែរ។ យើងត្រូវការប្រព័ន្ធមួយដែលអ្នកទទួលប្រាក់អាចដឹងថាលុយមិនត្រូវបានប្រើពីមុនទេ។ សម្រាប់ហេតុផលនេះ យើងរាប់ប្រតិបត្តិការដំបូងបំផុតជាប្រតិបត្តិការដំបូង យើងមិនខ្វល់ថាប្រតិបត្តិការនេះត្រូវប្រើឡើងវិញឬអត់នោះទេ។

មធ្យោបាយតែមួយគត់ដែលយើងអាចបញ្ជាក់ថាមិនមានប្រតិបត្តិការដែលបាត់គឺដោយការពិនិត្យមើលប្រតិបត្តិការទាំងអស់។ នៅក្នុងប្រព័ន្ធដែលប្រើគំរូអន្តរការី អន្តរការីមើលប្រតិបត្តិការទាំងអស់ ហើយសម្រេចចិត្តថាមួយណាមកមុនគេ ម្យ៉ាងវិញទៀត វាគឺអាចធ្វើទៅបានដើម្បីសម្រេចបាននូវបញ្ហានេះដោយមិនប្រើអន្តរការីដែលអាចទុកចិត្តបាន។ ប្រតិបត្តិការទាំងអស់ត្រូវតែប្រកាសជាសាធារណៈ ដូច្នេះហើយយើងត្រូវការប្រព័ន្ធដែលអនុញ្ញាតឱ្យអ្នកចូលរួមផ្ទៀងផ្ទាត់ប្រតិបត្តិការពីប្រវត្តិប្រតិបត្តិការភ្លាមៗនៅពេលដែលពួកគេទទួលបាន។ អ្នកទទួលត្រូវតែពិនិត្យមើលនៅពេលធ្វើប្រតិបត្តិការនីមួយៗ។ ថ្នាំងភាគច្រើនត្រូវតែយល់ព្រមថានេះជាលើកដំបូងរបស់ពួកគេដែលបានទទួលកាក់។

III. ម៉ាស៊ីនបម្រើត្រាពេលវេលា (Timestamp Server)

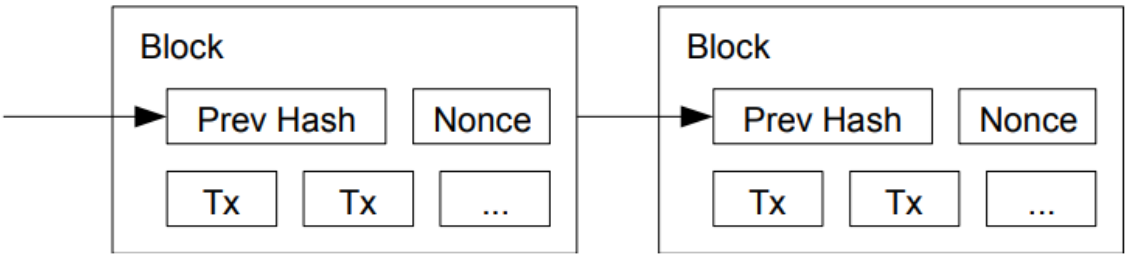
ដំណោះស្រាយដែលយើងមានគឺចាប់ផ្តើមជាមួយ Timestamp Server (ម៉ាស៊ីនមេពេលវេលា) ធ្វើការដោយយក hash នៃប្លុក នៃធាតុដែលត្រូវបោះត្រាពេលវេលា និងចែកចាយ hash ស្រដៀងនឹងប្រព័ន្ធកាសែត ឬ Usenet post ។ ត្រាពេលវេលាបង្ហាញថាទិន្នន័យត្រូវតែមាននៅពេលវេលា ដើម្បីទទួលបាន hash។ ត្រាពេលវេលានីមួយៗរួមបញ្ចូលត្រាពេលវេលាពីមុនទៅកាន់ hash ដោយបង្កើតខ្សែសង្វាក់ជាមួយនឹងត្រាពេលវេលាដែលនៅពីមុខវា។



IV. កស្តុតាងនៃការងារ(Proof Of Work)

ដើម្បីអនុវត្តការចែកចាយម៉ាស៊ីនមេត្រាពេលវេលាតាមមូលដ្ឋានពីម្នាក់ទៅម្នាក់ យើងនឹងត្រូវការប្រើប្រព័ន្ធកស្តុតាងនៃការងារ (Proof Of Work) ស្រដៀងទៅនឹង Adam Back's Hashcash ជាជាងការប្រើប្រាស់កាសែត ឬ Usenet Post។ កស្តុតាងនៃការងារពាក់ព័ន្ធនឹងការស្កេនរកតម្លៃដែលនៅពេលដែល hashed ដោយប្រើប្រាស់ SHA-256, Hash ចាប់ផ្តើមជាមួយនឹងចំនួននៃសូន្យប៊ីត។ ការងារជាមធ្យមដែលត្រូវការគឺ អិចស្ប៉ូណង់ស្យែលក្នុងចំនួននៃសូន្យប៊ីតដែលត្រូវការ ហើយអាចត្រូវបានផ្ទៀងផ្ទាត់ដោយការប្រតិបត្តិនូវ Hash តែមួយ។

សម្រាប់បណ្តាញត្រាពេលវេលា យើងនឹងបង្កើត Proof of work ដោយបង្កើនចំនួន Nonces ក្នុងប្លុករហូតដល់យើងរកឃើញតម្លៃ។ នៅពេលដែលការខិតខំប្រឹងប្រែង ស៊ីភីយូត្រូវបានចំណាយដើម្បីធ្វើឱ្យវាបំពេញនូវកស្តុតាងនៃការងារ ធ្វើឱ្យ Block hash ក្លាយជាលេខ 0 ដែលមានន័យថាថាមពលដំណើរការដែលប្រើដោយ CPU នឹងកើនឡើងរហូតដល់វាគ្រប់គ្រាន់ដើម្បីបំពេញតាម Proof of Work ដោយ Block មិនអាចផ្លាស់ប្តូរបានទេ ហើយនឹងដំណើរការឡើងវិញនៅពេលដែលប្លុកត្រូវបានភ្ជាប់ទៅខ្សែសង្វាក់។ នៅពេលដែលប្លុកនៅពេលក្រោយត្រូវបានភ្ជាប់ ការងារដែលត្រូវផ្លាស់ប្តូរត្រូវតែរួមបញ្ចូលក្នុងការធ្វើឡើងវិញនូវប្លុកទាំងអស់បន្ទាប់ពីវា។



ប្រព័ន្ធកស្តុតាងនៃការងារក៏ជួយដោះស្រាយបញ្ហានៃការស្វែងរកហានិភ័យភាគច្រើនផងដែរ នេះក៏ជាការសម្រេចចិត្តនៅក្នុងប្រព័ន្ធផងដែរ ប្រសិនបើការបោះឆ្នោតភាគច្រើនជាប្រព័ន្ធបោះឆ្នោត 1 ip 1 វាអាចមានឱកាសដែលនរណាម្នាក់នឹងបង្កើត ips ច្រើន។

ប្រព័ន្ធត្រួតពិនិត្យ៖ ប្រព័ន្ធកស្តុតាងនៃការងារគឺជាប្រព័ន្ធ 1 ស៊ីភីយូ 1 ការបោះឆ្នោត ហានិភ័យភាគច្រើននៅក្នុងប្រព័ន្ធនោះគឺផ្អែកលើខ្សែសង្វាក់ កស្តុតាងនៃការងារដែលវែងបំផុតត្រូវបានបង្កើតឡើងដោយដំណើរការដែលវែងបំផុតផងដែរ ប្រសិនបើថាមពលនៃការដឹកយករ៉ែរបស់ CPU ក៏ត្រូវបានគ្រប់គ្រងដោយថ្នាំងស្មោះត្រង់តាមខ្សែសង្វាក់ផងដែរ។

ភាពត្រឹមត្រូវគឺជាតំណភ្ជាប់ត្រឹមត្រូវអាចបង្កើតខ្សែសង្វាក់វែងបំផុត បើប្រៀបធៀបនឹងអ្នកដទៃព្យាយាមខ្សែសង្វាក់ដើម្បីដោះស្រាយប្លុកពីមុន អ្នកវាយប្រហារត្រូវតែដំណើរការឡើងវិញនូវ Proof of work នៅក្នុងប្លុកនោះ។ ហើយត្រូវតែបង្កើតប្លុកឱ្យបានរហ័ស និងរហូតដល់មានខ្សែសង្វាក់ពេញលេញវែងជាងខ្សែសង្វាក់ថ្នាំងស្មោះត្រង់ ដែលយើងនឹងពន្យល់នៅពេលក្រោយ។

បន្ទាប់ពីឈានដល់លទ្ធភាពចំនួនប្លុកដែល អ្នកវាយប្រហារអាចបង្កើតនឹងថយចុះជាលំដាប់ជាមួយនឹងរាល់ប្លុកបន្ថែមចូលមកក្នុងខ្សែសង្វាក់ ហើយដោយសារតែល្បឿនដំណើរការ កស្តុតាងនៃការងារនៅក្នុង Hardware នីមួយៗនឹងខុសគ្នា។ និងក្នុងរយៈពេលយូរ ថាមពលដំណើរការនឹងកើនឡើងជាបណ្តើរៗ។ កស្តុតាងនៃតម្លៃភាពលំបាកនៃការងារនឹងត្រូវបានកំណត់ដោយផ្អែកលើចំនួនមធ្យមនៃប្លុកដែលបានបង្កើតរៀងរាល់ម៉ោង ប្រសិនបើពួកគេត្រូវបានបង្កើតលឿនពេក ការលំបាកនឹងកើនឡើង។

V. បណ្តាញ (Network)

ដំហានក្នុងការដំណើរការបណ្តាញនេះគឺត្រូវធ្វើទៅតាមលក្ខណៈខាងក្រោម៖

1. ប្រតិបត្តិការថ្មីទាំងអស់គឺត្រូវចែកចាយទៅកាន់ថ្នាំងទាំងអស់
2. ថ្នាំងនីមួយៗប្រមូលប្រតិបត្តិការថ្មីៗទៅក្នុងប្លុក
3. ថ្នាំងនីមួយៗដំណើរការក្នុងការស្វែងរក Proof Of Work នៃការងារលំបាកៗសម្រាប់ប្លុករបស់វា
4. នៅពេលដែលថ្នាំងរកឃើញ Proof Of Work វានឹងធ្វើការចែកចាយប្លុកទៅកាន់ថ្នាំងទាំងនេះ
5. ថ្នាំងទទួលយកប្លុកបាន លុះត្រាតែប្រតិបត្តិការទាំងអស់នៅក្នុងវាមានសុពលភាព និងមិនទាន់បានចំណាយ
6. ថ្នាំងបង្ហាញពីការទទួលយកប្លុករបស់ពួកគេដោយធ្វើការលើការបង្កើតប្លុកបន្ទាប់ នៅក្នុងខ្សែសង្វាក់ដោយប្រើ hash នៃប្លុកដែលទទួលយកជា hash មុន។

Nodes តែងតែចាត់ទុកខ្សែសង្វាក់វែងបំផុតថាត្រឹមត្រូវ ហើយនឹងបន្តដំណើរការពង្រីកវា។ ប្រសិនបើថ្នាំងពីរចែកចាយកំណែផ្សេងគ្នានៃប្លុកបន្ទាប់ក្នុងពេលដំណាលគ្នា, ថ្នាំងខ្លះអាចទទួលបានមួយ ឬផ្សេងទៀតជាមុនសិន។ ក្នុងករណីនេះ, ពួកគេធ្វើការលើសាខាទីមួយដែលពួកគេបានទទួល ប៉ុន្តែរក្សាទុកសាខាផ្សេងទៀត ក្នុងករណីដែលវាវែង។ ចំណងនឹងខូចនៅពេលដែលរកឃើញកស្តុតាងនៃការងារបន្ទាប់ ហើយសាខាមួយនឹងកាន់តែវែង។ ថ្នាំងដែលធ្វើការលើសាខាផ្សេងទៀតនឹងប្តូរទៅផ្នែកដែលវែងជាងនេះ។

ការផ្សាយប្រតិបត្តិការថ្មីមិនចាំបាច់ទៅដល់ថ្នាំងទាំងអស់នោះទេ។ ដរាបណាពួកគេឈានដល់ថ្នាំងជាច្រើន មិនយូរប៉ុន្មានពួកគេនឹងចូលទៅក្នុងប្លុកមួយនេះ។ ការចែកចាយប្លុកទាំងអស់នេះគឺជាធន់នៃការទម្លាក់សារផងដែរ។ បើសិនជាថ្នាំងមិនបានទទួលប្លុក វានឹងធ្វើការស្នើសុំ នៅពេលដែលទទួលប្លុកបន្ទាប់ និងដឹងថាវាអាចខកខានដោយប្រការណាមួយ។

VI. ការលើទឹកចិត្ត (Incentive)

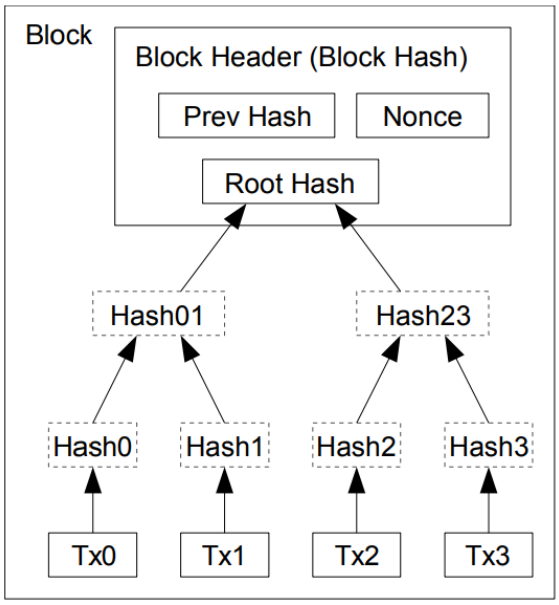
ជាធម្មតាប្រតិបត្តិការដំបូងនៅក្នុងប្លុកគឺ ជាប្រតិបត្តិការពិសេសដែលចាប់ផ្តើមកាក់ថ្មីដែលជាកម្មសិទ្ធិរបស់អ្នកបង្កើតប្លុក។ វាបន្ថែមការលើកទឹកចិត្តសម្រាប់ថ្នាំងដើម្បីគាំទ្របណ្តាញ, និងផ្តល់នូវវិធីមួយដើម្បីចែកចាយកាក់ និងចរាចរដំបូង ដោយសារតែមិនមានអាជ្ញាធរកណ្តាលចេញឱ្យពួកគេ។ ការបន្ថែមចំនួនកាក់ថ្មីគឺស្រដៀងគ្នាទៅនឹងអ្នករុករករ៉ែមាសដែលចំណាយធនធានដើម្បីចរាចរមាស។ ក្នុងករណីរបស់យើង វាជាពេលវេលាស៊ីភីយូ និងអគ្គិសនីដែលត្រូវចំណាយ។

ការលើកទឹកចិត្តក៏អាចត្រូវបានផ្តល់មូលនិធិជាមួយនឹងថ្លៃប្រតិបត្តិការផងដែរ។ ប្រសិនបើលទ្ធផលតម្លៃនៃប្រតិបត្តិការគឺតិចជាងតម្លៃបញ្ចូលរបស់វា ភាពខុសគ្នាគឺជាថ្លៃប្រតិបត្តិការដែលត្រូវបានបន្ថែមទៅតម្លៃលើកទឹកចិត្តនៃប្លុកដែលមានប្រតិបត្តិការ។ នៅពេលដែលចំនួនកាក់ដែលបានកំណត់ទុកជាមុនបានចរាចរ ការលើកទឹកចិត្តអាចផ្លាស់ប្តូរទាំងស្រុងទៅថ្លៃប្រតិបត្តិការ និងជាគ្មានអតិផរណាទាំងស្រុង។

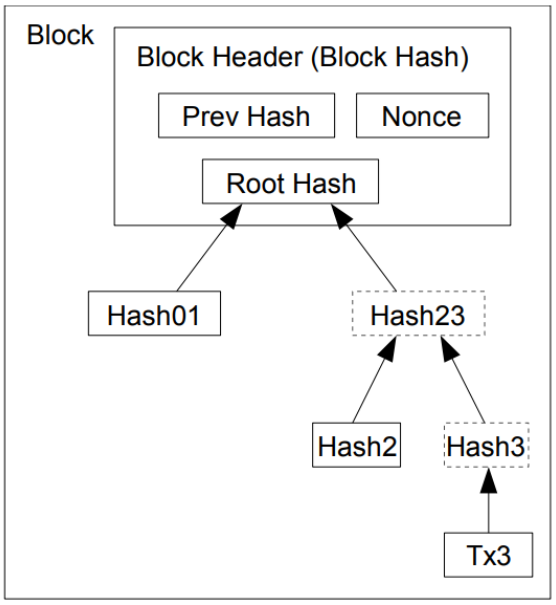
ការលើកទឹកចិត្តអាចជួយលើកទឹកចិត្តឱ្យថ្នាំងរក្សាភាពស្មោះត្រង់ ប្រសិនបើអ្នកវាយប្រហារលោភលន់អាចប្រមូលផ្តុំថាមពលស៊ីកីយូច្រើនជាងថ្នាំងទាំងអស់។ គាត់នឹងត្រូវជ្រើសរើសរវាងការប្រើវាដើម្បីបោកប្រាស់មនុស្សដោយការលួចយកការទូទាត់របស់គាត់មកវិញ ឬប្រើវាដើម្បីបង្កើតកាក់ថ្មី។ គាត់គួរតែរកឱ្យឃើញថាវាចំណេញច្រើនជាងក្នុងការលេងដោយច្បាប់ច្បាប់បែបនេះដែលអនុគ្រោះឱ្យគាត់ជាមួយនឹងកាក់ថ្មីច្រើនជាងអ្នកផ្សេងទៀតរួមបញ្ចូលគ្នា ជាជាងធ្វើឱ្យខូចប្រព័ន្ធ និងសុពលភាពនៃទ្រព្យសម្បត្តិផ្ទាល់ខ្លួនរបស់គាត់។

VII. ការទាមទារទំហំ Disk ឡើងវិញ (Reclaiming Disk Space)

នៅពេលដែលប្រតិបត្តិការចុងក្រោយបំផុតនៅក្នុងកាក់មួយត្រូវបានកប់នៅក្រោមប្លុកគ្រប់គ្រាន់។ ប្រតិបត្តិការដែលបានចំនាយ មុនពេលវាអាចត្រូវបានបោះចោល ដើម្បីសន្សំ Disk Space ដើម្បីជួយសម្រួលដល់បញ្ហានេះដោយមិនចាំបាច់បំបែក hash របស់ប្លុក ប្រតិបត្តិការត្រូវបាន hashed នៅក្នុង Merkle Tree [7][2][5] ដោយមានតែ root ប៉ុណ្ណោះដែលរួមបញ្ចូលនៅក្នុង hash របស់ block។ បន្ទាប់មកប្លុកចាស់អាចបង្រួមបានដោយការគៀបចេញពីមែក។ Hash ខាងក្នុងមិនចាំបាច់រក្សាទុកទេ។



Transactions Hashed in a Merkle Tree



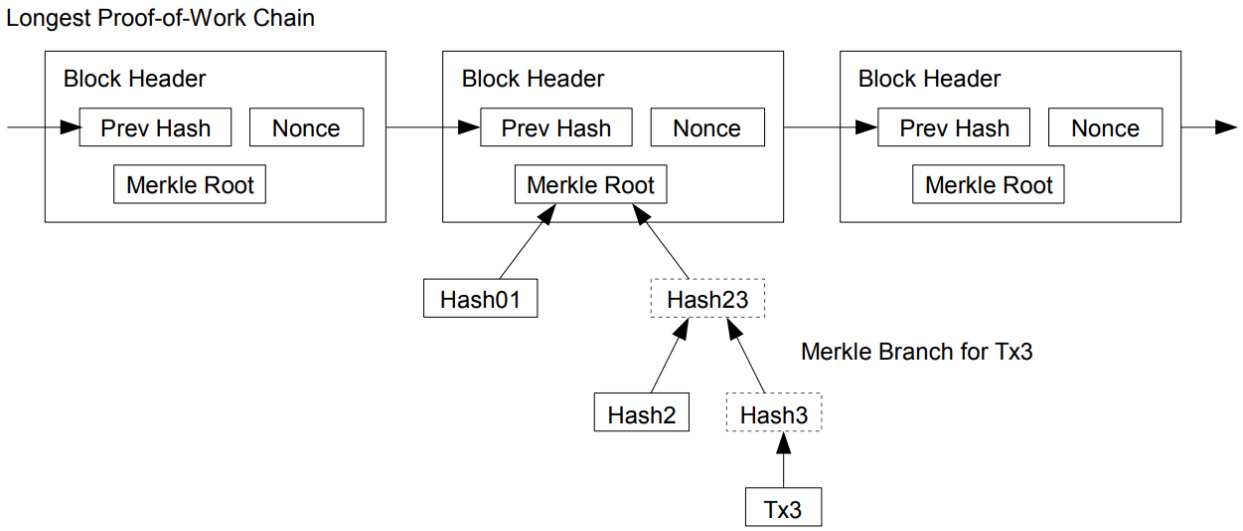
After Pruning Tx0-2 from the Block

ប្លុកដែលមិនមានប្រតិបត្តិការមានទំហំចំនួន ៨០ bytes។ បើសិនជាយើងឧបមាថាប្លុកទាំងអស់ត្រូវបានបង្កើតក្នុងរយៈពេល ១០ នាទីម្តង, ៨០ bytes * ៦ * ២៤ * ៣៦៥ = 4.2MB per year។ ជាមួយនឹងប្រព័ន្ធកំព្យូទ័រជាធម្មតាលក់ជាមួយនឹងទំហំ

២GB នៃ RAM នៅក្នុងឆ្នាំ២០០៨ និង ច្បាប់របស់ Moore ព្យាករណ៍ពីកំណើនបច្ចុប្បន្ន ១.២GB ក្នុងមួយឆ្នាំ ការផ្ទុកមិនគួរជាបញ្ហាទេ ទោះបីជាបបមកថាប្លុកត្រូវតែរក្សាទុកក្នុងអង្គចងចាំក៏ដោយ។

VIII. ការផ្ទៀងផ្ទាត់ការទូទាត់សាមញ្ញ

ដើម្បីផ្ទៀងផ្ទាត់ការទូទាត់ដោយមិនដំណើរការថ្នាំងបណ្តាញពេញលេញ អ្នកប្រើប្រាស់គ្រាន់តែត្រូវការរក្សាច្បាប់ចម្លងនៃក្បាលប្លុកនៃខ្សែសង្វាក់កស្តុតាងនៃការងារដែលវែងបំផុត ដែលគាត់អាចទទួលបានដោយការសាកសួរថ្នាំងបណ្តាញរហូតដល់គាត់ជឿជាក់ថាគាត់មានខ្សែសង្វាក់វែងបំផុតហើយទទួលបាន Merkle Tree ការភ្ជាប់ប្រតិបត្តិការទៅនឹងប្លុកដែលវាត្រូវបានបោះត្រាពេលវេលា។ គាត់មិនអាចពិនិត្យមើលប្រតិបត្តិការបានដោយខ្លួនគាត់ផ្ទាល់ទេ ប៉ុន្តែដោយការភ្ជាប់វាទៅកន្លែងមួយនៅក្នុងខ្សែសង្វាក់ គាត់អាចមើលឃើញថាថ្នាំងបណ្តាញបានទទួលយកវា និងប្លុកបន្ថែមបន្ទាប់ពីវាបញ្ជាក់បន្ថែមថាបណ្តាញបានទទួលយកវា។

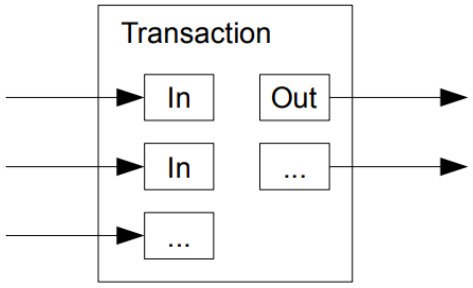


ដូចនេះការផ្ទៀងផ្ទាត់គឺអាចទុកចិត្តបាន ដរាបណាថ្នាំងគឺជាអ្នកគ្រប់គ្រងបណ្តាញ ប៉ុន្តែមានភាពងាយរងគ្រោះជាងប្រសិនបើ បណ្តាញត្រូវបានគ្រប់គ្រងដោយអ្នកវាយប្រហារ។ ខណៈពេលដែលថ្នាំងបណ្តាញអាចផ្ទៀងផ្ទាត់ប្រតិបត្តិការសម្រាប់ខ្លួនគេ វិធីសាស្ត្រសាមញ្ញអាចត្រូវបានបោកបញ្ឆោតដោយប្រតិបត្តិការប្រឌិតរបស់អ្នកវាយប្រហារ ដរាបណាអ្នកវាយប្រហារអាចបន្តគ្រប់គ្រងបណ្តាញ។

យុទ្ធសាស្ត្រមួយដើម្បីការពារប្រឆាំងនឹងការងារនេះ គឺដើម្បីទទួលយកការជូនដំណឹងពីថ្នាំងបណ្តាញ នៅពេលដែលពួកគេរកឃើញប្លុកមិនត្រឹមត្រូវ ដោយជំរុញឱ្យកម្មវិធីរបស់អ្នកប្រើទាញយកប្លុកពេញលេញ និងប្រតិបត្តិការជូនដំណឹងដើម្បីបញ្ជាក់ពីភាពមិនស៊ីសង្វាក់គ្នា។ អាជីវកម្មដែលទទួលបានការទូទាត់ញឹកញាប់នឹងនៅតែចង់ដំណើរការថ្នាំងផ្ទាល់ខ្លួនរបស់ពួកគេសម្រាប់សុវត្ថិភាពឯករាជ្យបន្ថែមទៀត និងការផ្ទៀងផ្ទាត់លឿនជាងមុន។

IX. ការរួមបញ្ចូលគ្នានិងការបំបែកតម្លៃ (Combining and Splitting Value)

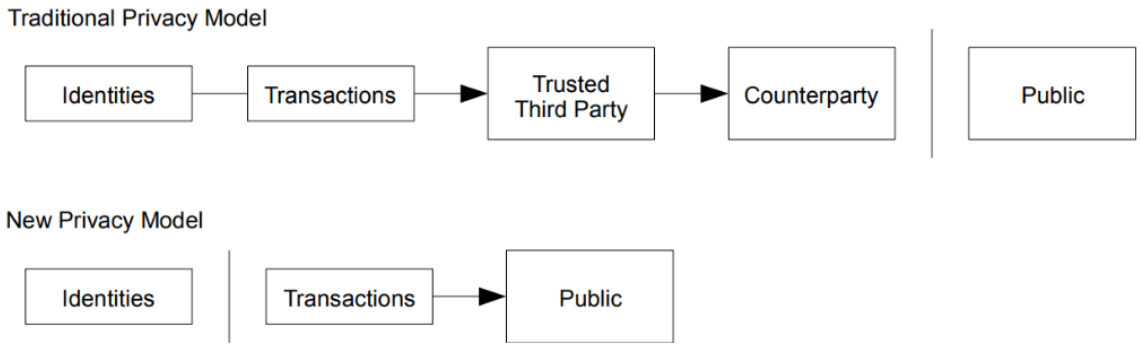
ទោះបីជាវាអាចគ្រប់គ្រងកាក់ដោយឡែកពីគ្នាក៏ដោយ វាពិតជាមិនអាចទទួលយកបានក្នុងការធ្វើប្រតិបត្តិការដាច់ដោយឡែកសម្រាប់រាល់សេនក្នុងការផ្ទេរ។ ដើម្បីអនុញ្ញាតឱ្យតម្លៃត្រូវបានបំបែក និងបញ្ចូលគ្នា ប្រតិបត្តិការមានធាតុបញ្ចូល និងលទ្ធផលច្រើន។ ជាធម្មតាវានឹងមានការបញ្ចូលតែមួយពីប្រតិបត្តិការពីមុនធំជាង ឬការបញ្ចូលច្រើនដែលរួមបញ្ចូលគ្នានូវចំនួនតូចជាងហើយនៅលទ្ធផលភាគច្រើនពីរ៖ មួយសម្រាប់ការទូទាត់ និងមួយត្រឡប់ការផ្លាស់ប្តូរប្រសិនបើមាន ត្រឡប់ទៅអ្នកផ្ញើវិញ។



វាគួរតែត្រូវបានកត់សម្គាល់ថា fan-out ដែលប្រតិបត្តិការមួយអាស្រ័យលើប្រតិបត្តិការជាច្រើនហើយប្រតិបត្តិការទាំងនោះពឹងផ្អែកលើជាច្រើនទៀតមិនមែនជាបញ្ហានៅទីនេះទេ។ មិនដែលមានតម្រូវការក្នុងការទាញយកច្បាប់ចម្លងឯកសារជ្រៀតជ្រែកនៃប្រវត្តិប្រតិបត្តិការនោះទេ។

X. ឯកជនភាព (Privacy)

គំរូធនាគារបែបប្រពៃណីសម្រេចបាននូវកម្រិតនៃភាពឯកជនដោយកំណត់ការចូលប្រើព័ត៌មានដល់ភាគីពាក់ព័ន្ធ និងភាគីទីបីដែលគួរឱ្យទុកចិត្ត។ ភាពចាំបាច់ក្នុងការប្រកាសប្រតិបត្តិការទាំងអស់ជាសាធារណៈរារាំងវិធីសាស្ត្រនេះ ប៉ុន្តែភាពឯកជននៅតែអាចរក្សាបានដោយការបំបែកលំហូរព័ត៌មាននៅកន្លែងផ្សេងទៀត៖ ដោយរក្សាសោសាធារណៈ (PUBLIC KEY) ជាអនាមិក។ សាធារណជនអាចមើលឃើញថានរណាម្នាក់កំពុងផ្ញើចំនួនទឹកប្រាក់ទៅឱ្យអ្នកផ្សេង ប៉ុន្តែដោយគ្មានព័ត៌មានដែលភ្ជាប់ប្រតិបត្តិការទៅនរណាម្នាក់ឡើយ។ នេះគឺស្រដៀងគ្នាទៅនឹងកម្រិតនៃព័ត៌មានដែលចេញផ្សាយដោយផ្សារហ៊ុន ដែលពេលវេលា និងទំហំនៃការជួញដូរបុគ្គល "Tape" ត្រូវបានបង្ហាញជាសាធារណៈ ប៉ុន្តែដោយមិនបានប្រាប់ថាតើភាគីណាជានរណានោះទេ។



ក្នុងនាមជាជញ្ជាំងភ្លើង (Firewall) បន្ថែម សោថ្មីគួរតែត្រូវបានប្រើសម្រាប់ប្រតិបត្តិការនីមួយៗ ដើម្បីការពារពួកវាពីការភ្ជាប់ជាមួយម្ចាស់ទូទៅ។ ការភ្ជាប់មួយចំនួននៅតែមិនអាចជៀសបានជាមួយនឹងប្រតិបត្តិការពហុបញ្ចូល ដែលចាំបាច់បង្ហាញថាធាតុចូលរបស់ពួកគេត្រូវបានគ្រប់គ្រងដោយម្ចាស់តែមួយ។ ហានិភ័យគឺថាប្រសិនបើម្ចាស់សោត្រូវបានបង្ហាញ ការតភ្ជាប់អាចបង្ហាញពីប្រតិបត្តិការផ្សេងទៀតដែលជាកម្មសិទ្ធិរបស់ម្ចាស់ដូចគ្នា។

XI. ការគណនា (Calculations)

យើងពិចារណាលើសេណារីយ៉ូនៃអ្នកវាយប្រហារដែលព្យាយាមបង្កើតខ្សែសង្វាក់ជំនួសលឿនជាងខ្សែសង្វាក់។ ទោះបីជាត្រូវបានសម្រេចក៏ដោយវាមិនបោះឱ្យប្រព័ន្ធបើកចំហចំពោះការផ្លាស់ប្តូរតាមអំពើចិត្តនេះទេ ដូចជាការបង្កើតតម្លៃចេញពីខ្យល់ស្តើង ឬយកលុយដែលមិនធ្លាប់ជារបស់អ្នកវាយប្រហារនោះទេ។ ថ្នាំងនឹងមិនទទួលយកប្រតិបត្តិការមិនត្រឹមត្រូវជាការទូទាត់ទេ ហើយថ្នាំង នឹងមិនទទួលយកប្តូរដែលមានពួកវាទេ។ អ្នកវាយប្រហារអាចព្យាយាមផ្លាស់ប្តូរប្រតិបត្តិការផ្ទាល់ខ្លួនរបស់គាត់ដើម្បីយកប្រាក់ដែលគាត់បានចំណាយថ្មីៗនេះមកវិញ។

ការប្រណាំងរវាងខ្សែសង្វាក់ស្មោះត្រង់ និងខ្សែសង្វាក់អ្នកវាយប្រហារអាចត្រូវបានកំណត់ថាជា Binomial Random Walk។ ព្រឹត្តិការណ៍ជោគជ័យគឺខ្សែសង្វាក់ស្មោះត្រង់ត្រូវបានពង្រីកដោយប្តូកមួយ បង្កើនការនាំមុខដោយ +1 ហើយព្រឹត្តិការណ៍បរាជ័យគឺខ្សែសង្វាក់របស់អ្នកវាយប្រហារត្រូវបានពង្រីកដោយប្តូកមួយ កាត់បន្ថយគម្លាតដោយ -1 ។

ប្រូបាប៊ីលីតេនៃអ្នកវាយប្រហារដែលចាប់បានពីឱនភាពដែលបានផ្តល់ឱ្យគឺស្រដៀងគ្នាទៅនឹងបញ្ហាបំផ្លាញអ្នកលេងល្បែង។ ឧបមាថាអ្នកលេងល្បែងដែលមានឥណទានគ្មានដែនកំណត់ចាប់ផ្តើមនៅឱនភាព ហើយលេងសាកល្បងចំនួនគ្មានកំណត់ ដើម្បីព្យាយាមឈានដល់ចំណុចចំណេញ។ យើងអាចគណនាប្រូបាប៊ីលីតេដែលគាត់មិនធ្លាប់ឈានដល់ចំណុចចំណេញ ឬថាអ្នកវាយប្រហារធ្លាប់ចាប់បានខ្សែសង្វាក់ស្មោះត្រង់ ដូចខាងក្រោម [8]៖

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind

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

ដោយមានការសន្មត់របស់យើងថា p > q ប្រូបាប៊ីលីតេធ្លាក់ចុះជានិទស្សន្ត ដោយសារចំនួនប្តូកដែលអ្នកវាយប្រហារត្រូវតាមទាន់ការកើនឡើង។ ជាមួយនឹង Exponentially ប្រសិនបើគាត់មិនបង្កើតសំណាងទៅមុខមុននេះទេ ឱកាសរបស់គាត់នឹងរលាយបាត់បន្តិចម្តងៗ នៅពេលដែលគាត់ធ្លាក់បន្ថែមទៀត។

ឥឡូវនេះ យើងពិចារណាថាតើអ្នកទទួលប្រតិបត្តិការថ្មីត្រូវរង់ចាំរយៈពេលប៉ុន្មាន មុនពេលដែលប្រាកដថាអ្នកផ្ញើមិនអាចផ្លាស់ប្តូរប្រតិបត្តិការបានទេ។ យើងសន្មត់ថាអ្នកផ្ញើគឺជាអ្នកវាយប្រហារដែលចង់ធ្វើឱ្យអ្នកទទួលជឿថាគាត់បានបង់ប្រាក់ឱ្យ

គាត់មួយ រយៈ បន្ទាប់មកប្តូរវាមកសងវិញដោយខ្លួនឯងបន្ទាប់ពីពេលវេលាបានកន្លងផុតទៅ។ អ្នកទទួលនឹងត្រូវបានជូនដំណឹង នៅពេលដែលវាកើតឡើង ប៉ុន្តែអ្នកផ្ញើសង្ឃឹមថាវានឹងយឺតពេលហើយ។

អ្នកទទួលបង្កើតគូសោថ្មី (New key pair) ហើយផ្តល់សោសាធារណៈដល់អ្នកផ្ញើភ្លាមៗមុនពេលចុះហត្ថលេខា។ នេះរារាំងអ្នកផ្ញើពីការរៀបចំខ្សែសង្វាក់នៃប្លុកជាមុនដោយធ្វើការបន្តរហូតដល់គាត់មានសំណាងគ្រប់គ្រាន់ដើម្បីឈានទៅមុខបានឆ្ងាយ បន្ទាប់មកប្រតិបត្តិប្រតិបត្តិការនៅពេលនោះ។ នៅពេលដែលប្រតិបត្តិការត្រូវបានផ្ញើ អ្នកផ្ញើដែលមិនស្មោះត្រង់ចាប់ផ្តើមធ្វើការដោយសម្ងាត់នៅលើខ្សែសង្វាក់ប៉ារ៉ាឡែលដែលមានកំណែជំនួសនៃប្រតិបត្តិការរបស់គាត់។

អ្នកទទួលរង់ចាំរហូតដល់ប្រតិបត្តិការត្រូវបានបន្ថែមទៅប្លុក ហើយប្លុក z ត្រូវបានភ្ជាប់បន្ទាប់ពីវា។ គាត់មិនដឹងថាចំនួនជាក់លាក់នៃដំណើរការដែលអ្នកវាយប្រហារបានធ្វើនោះទេ ប៉ុន្តែការសន្មត់ថាប្លុកស្មោះត្រង់បានចំណាយពេលជាមធ្យមរំពឹងទុកក្នុងមួយប្លុក វឌ្ឍនភាពសក្តានុពលរបស់អ្នកវាយប្រហារនឹងជាការចែកចាយ Poisson ជាមួយនឹងតម្លៃរំពឹងទុក៖

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

ដើម្បីទទួលបានប្រូបាប៊ីលីតេដែលអ្នកវាយប្រហារនៅតែអាចចាប់បានឥឡូវនេះ យើងគុណនឹងដង់ស៊ីតេ Poisson សម្រាប់ចំនួនវឌ្ឍនភាពនីមួយៗដែលគាត់អាចធ្វើបានដោយប្រូបាប៊ីលីតេដែលគាត់អាចចាប់បានពីចំណុចនោះ៖

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

ការរៀបចំឡើងវិញដើម្បីដៀសវាងការបូកសរុបកន្ទុយគ្មានកំណត់នៃការចែកចាយ...

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

Converting to C code...

```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;
}
```

ការដំណើរការលទ្ធផលមួយចំនួន យើងអាចមើលឃើញប្រូបាប៊ីលីតេធ្លាក់ចុះដោយនិទស្សន្តជាមួយ z ។

```
q=0.1                          q=0.3
z=0     P=1.0000000            z=0     P=1.0000000
z=1     P=0.2045873            z=5     P=0.1773523
z=2     P=0.0509779            z=10    P=0.0416605
z=3     P=0.0131722            z=15    P=0.0101008
z=4     P=0.0034552            z=20    P=0.0024804
z=5     P=0.0009137            z=25    P=0.0006132
z=6     P=0.0002428            z=30    P=0.0001522
z=7     P=0.0000647            z=35    P=0.0000379
z=8     P=0.0000173            z=40    P=0.0000095
z=9     P=0.0000046            z=45    P=0.0000024
z=10    P=0.0000012            z=50    P=0.0000006
```

ដំណោះស្រាយ P តិចជាង 0.1%...

```
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
```

XII. សេចក្តីសន្និដ្ឋាន (Conclusion)

យើងបានដាក់ស្នើប្រព័ន្ធសម្រាប់ប្រតិបត្តិការអេឡិចត្រូនិកដោយមិនពឹងផ្អែកលើការជឿទុកចិត្ត។ យើងបានចាប់ផ្តើមជាមួយនឹងក្របខណ្ឌធម្មតានៃកាក់ដែលផលិតចេញពីហត្ថលេខាឌីជីថល ដែលផ្តល់នូវការគ្រប់គ្រងដ៏រឹងមាំនៃភាពជាម្ចាស់ ប៉ុន្តែវាមិនពេញលេញដោយគ្មានមធ្យោបាយទប់ស្កាត់ការចំណាយទ្វេដងនោះទេ។ ដើម្បីដោះស្រាយបញ្ហានេះ យើងបានស្នើរបណ្តាញ peer-to-peer ដោយប្រើកស្តុតាងនៃការងារ ដើម្បីកត់ត្រាប្រវត្តិនៃប្រតិបត្តិការសាធារណៈ ដែលវាក្លាយទៅជាមិនសមហេតុផលក្នុងការគណនាយ៉ាងឆាប់រហ័សសម្រាប់អ្នកវាយប្រហារក្នុងការផ្លាស់ប្តូរ ប្រសិនបើថ្នាំងស្មោះត្រង់គ្រប់គ្រងថាមពលស៊ីភីយូភាគច្រើន។ បណ្តាញមានភាពរឹងមាំនៅក្នុងភាពសាមញ្ញដែលមិនមានរចនាសម្ព័ន្ធរបស់វា។

ថ្នាំងដំណើរការទាំងអស់ក្នុងពេលតែមួយជាមួយនឹងការសម្របសម្រួលតិចតួច។ ពួកគេមិនចាំបាច់ត្រូវបានកំណត់អត្តសញ្ញាណទេ ដោយសារមិនត្រូវបានបញ្ជូនទៅកាន់កន្លែងជាក់លាក់ណាមួយទេ ហើយគ្រាន់តែត្រូវបញ្ជូនតាមមូលដ្ឋានការខិតខំប្រឹងប្រែងដ៏ល្អបំផុតប៉ុណ្ណោះ។ ថ្នាំងអាចចាកចេញ និងចូលរួមបណ្តាញឡើងវិញតាមឆន្ទៈ ដោយទទួលយកខ្សែសង្វាក់កស្តុតាងនៃការងារជាកស្តុតាងនៃអ្វីដែលបានកើតឡើងខណៈពេលដែលពួកគេទៅ។ ពួកគេបោះឆ្នោតដោយប្រើថាមពលស៊ីភីយូរបស់ពួកគេបង្ហាញពីការទទួលយកប្តូរដែលមានសុពលភាពដោយធ្វើការលើការពង្រីកពួកវា និងបដិសេធប្តូរមិនត្រឹមត្រូវដោយបដិសេធមិនធ្វើការលើពួកវា។ ច្បាប់ និងការលើកទឹកចិត្តដែលត្រូវការអាចត្រូវបានអនុវត្តដោយប្រើយន្តការឯកភាពនេះ។

ឯកសារយោង

[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, no 2, 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 Conference on 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 and Privacy, IEEE Computer Society, pages 122-133, April 1980.

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

បកប្រែដោយ៖ ឈូក ទិត្យបញ្ញារ៉ា

Github: https://github.com/Panhara28/bitcoin-whitepaper-in-khmer

Facebook: https://www.facebook.com/chhouk.titpanhara.3

Telegram: @panhara28

X: https://twitter.com/PanharaTuarus