Bitcoin Whitepaper - NEPALI

This is the translation of Bitcoin WP in Nepali Language by Krishna Dahal & Bibek Koirala

बिटकोइन: एक पियर-टु-पियर इलेक्ट्रोनिक मुद्रा प्रणाली

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

सारांस

एक विशुद्ध व्यक्ति-व्यक्ति बीचको इलेक्ट्रोनिक मुद्रा प्रणालीले वित्तीय संस्था विना नै एक पक्षबाट अर्को पक्षलाई पैसाको अनलाइन भुक्तानी गर्ने अनुमति दिन्छ। डिजिटल हस्ताक्षरबाट केहि समाधान मिलेतापनि जबसम्म दोहोरो-खर्च (डबल-इस्पेंडिंग) हटाउन विश्वसनीय तेस्रो व्यक्तिको आवश्यकता पर्छ तबसम्म यस्तो प्रणालीको मुख्य फाइदाहरु प्राप्त गर्न सकिदैन। व्यक्ति-व्यक्ति बीचको संझाल/नेटवर्क प्रयोग गरेर हामी "डबल-इस्पेंडिंग"को समस्या समाधान गर्ने प्रस्ताव राख्दछौं। संझाल/नेटवर्कले लेनदेनको कारोवारलाई श्रृंखलावद्ध रुपमा चलिरहेको ह्याशमा आधारित भंएको "कार्य-प्रमाणीकरण" मा टाइमस्ट्याम्प गर्छ र यसरी बनेको रेकर्ड पुन:"कार्य-प्रमाणीकरण" नगरीकन परिवर्तन गर्न सकिँदैन। सबैभन्दा लामो चेन "घटनाहरूको अनुक्रमको प्रमाण" साक्षी मात्र नभई यो कम्पुटर CPU पावरको सबैभन्दा ठूलो जमघटबाट आएको प्रमाण पनि हुन्छ । जबसम्म नेटवर्कको अधिकांश CPU पावर त्यस्ता नोडहरू द्वारा नियन्त्रित हुन्छ जसले नेटवर्कलाई मिलेमतोमा आक्रमण गर्दैनन्, तिनीहरूले सबैभन्दा लामो चेन उत्पन्न गर्छन् र आक्रमणकारी नोडहरूलाई परास्त पार्छन। यस्तो किसिमको नेटवर्कलाई न्यूनतम संरचना मात्र चाहिन्छ। सन्देशहरू सर्वश्रेष्ठ प्रयासको आधारमा प्रसारित हुन्छन्, र नोडहरू आफु अनुपस्थित हुदा के भयो भन्ने प्रमाणको रूपमा सबैभन्दा लामो "कार्य-प्रमाणीकरण"को श्रृंखलालाई स्वीकार गर्दै नेटवर्कमा आफ्नै मर्जीले छाड्न र पुन: सामेल हुन सक्दछन्।

१. परिचय

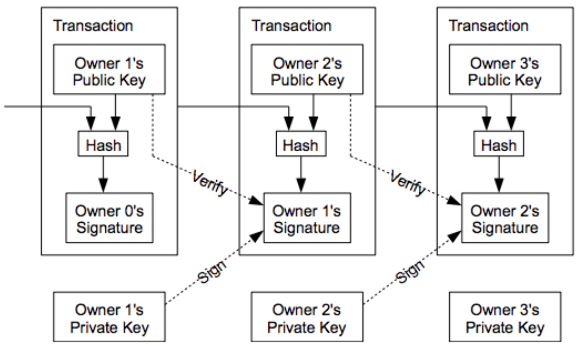
इन्टरनेटमा हुने व्यापार पूर्णरुपले वित्तीय संस्थाहरूमा भर परेको छ। विश्वसनीय तेस्रो पक्षहरूको रूपमा सेवा दिएर वित्तीय संस्थाहरू इलेक्ट्रोनिक भुक्तानीहरू प्रशोधन गर्छन। जबकि यस प्रणालीले पर्याप्त रुपमा राम्रो तरिकाले नै लेनदेनको काम गर्दछ, यो अझै पनि विश्वासमा आधारित ढांचा(मोडेल)को अंतर्निहित कमजोरीहरु बाट ग्रस्त छ। यस प्रणालीमा पूर्ण रूपले उल्टाउन नमिल्ने लेनदेनहरू वास्तवमै सम्भव छैनन्, किनक वित्तीय संस्थाहरू विवादको समयमा मध्यस्थता नगर्नबाट बच्न सक्दैनन। मध्यस्थता गर्न खर्च लाग्ने भएकाले लेनदेनको लागत बढ्छ, जसले न्यूनतम व्यावहारिक लेनदेनको आकार साँघुरो बनाउछ र साना आकस्मिक लेनदेनको सम्भावना नै हटाईदिन्छ, र यसरी अपरिवर्तनीय सेवाहरूको लागि उल्टाउन नसकिने भुक्तानीहरू गर्ने क्षमता गुमाउनुमा निकै फराकिलो खर्छ/लागत छ। लेनदेन उल्टाउने सम्भावना हुने बित्तिकै विश्वासको आवश्यकता पनि बढ्दछ । अनि व्यापारीहरू आफ्ना ग्राहकहरूबाट झन् धेरै सचेत हुनुपर्छ जसकारण उनीहरु अनावषेक धेरै भन्दा धेरै ग्राहकहरूको जानकारी लिएर ग्राहकहरूलाई नै झन्झट दिन्छन् । यसमा केही प्रतिशत ठगि पनि स्वीकार गर्नै पर्ने हुन्छ । भौतिक मुद्रा प्रयोग गरेर यी लागत तथा भुक्तानीका अनिश्चितताहरू हटाउन सकिन्छ, तर कुनै विश्वसनीय पक्ष बिना संचार माध्यममा भुक्तानी गर्न हामी संग कुनै संयन्त्र अवस्थित छैन।

हामीलाई विश्वासको सट्टा क्रिप्टोग्राफिक प्रमाणमा आधारित इलेक्ट्रोनिक भुक्तानी प्रणाली चाहिएको हो, जसले कुनै पनि दुई इच्छुक पक्षहरूलाई तेस्रो पक्षको आवश्यकता विना नै एकअर्कासँग प्रत्यक्ष लेनदेन गर्न अनुमति दिन्छ । गणनात्मक रुपमा उल्टाउन नमिल्ने लेनदेनहरू भएपछि बिक्रेताहरू ठगिबाट

जोगिन्छन, र नियमित एस्क्रो प्रणाली लागू गरेर सजिलैसँग खरिदकर्ताहरूको पनि सुरक्षा गर्न सकिन्छ। यस पेपरमा हामी "डबल-इस्पेंदिंग"को समस्या समाधान गर्न व्यक्ति-व्यक्ति बीचमा वितरण भएको टाइमस्ट्याम्प सर्भर प्रयोग गर्नेछौं जसले लेनदेनको कालानुक्रमिक गणनात्मक प्रमाण उत्पन्न गर्छ। जबसम्म इमानदार नोडहरूले सामूहिक रूपमा आक्रमणकारी नोडहरूको तुलनाभन्दा बढी CPU पावर नियन्त्रण गर्छन तबसम्म यस प्रणाली सुरक्षित रहन्छ।

२. लेनदेन

हामीले इलेक्ट्रोनिक मुद्रालाई डिजिटल हस्ताक्षरको श्रृंखलाको (चेनको) रुपमा परिभाषित गर्छौं। प्रत्येक धनीले अघिल्लो लेनदेनको ह्याश र क्रमश अर्को धनीको सार्वजनिक कुञ्जिलाई डिजिटल रुपमा हस्ताक्षर गरी मुद्राको (डिजिटल हस्ताक्षरको श्रृंखला) अन्त्यमा तिनीहरुलाई समावेश गरेर मुद्राको स्थान्तरण गर्दछ। एक भुक्तानीकर्ताले स्वामित्वको श्रृंखला जाँच गरेर हस्ताक्षरहरू प्रमाणित गर्न सक्छ।

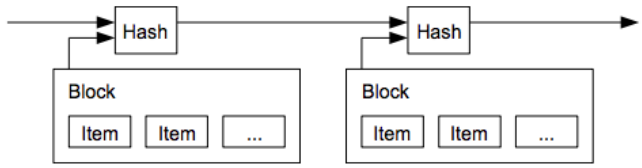


स्वभाभिक समस्या यो हो कि भुक्तानीकर्ताले पूर्व मालिक मध्ये कुनै एकले मुद्राको दोहोरो खर्च गरेको बारे प्रमाणित गर्न सक्दैन। यसको एक सामान्य समाधान भनेको एउटा विश्वसनीय केन्द्रीय प्राधिकरण, वा टकसालको स्थापना हो जसले दोहोरो खर्चको लागि प्रत्येक लेनदेन जाँच गर्छ। प्रत्येक लेनदेन पछि, नयाँ मुद्रा जारी गर्न पुरानो मुद्रालाई टकसालमा फिर्ता ल्याउनुपर्छ, र टकसालबाट सिधै जारी गरिएका मुद्राहरू मात्र दोहोरो खर्च नभएको विश्वास गर्न सकिन्छ। यो समाधानको समस्या यो हो कि सम्पूर्ण मुद्रा प्रणालीको भाग्य टकसाल चलाउने कम्पनीमा निर्भर हुन्छ र एउटा बैंक जस्तै प्रत्येक लेनदेन तिनीहरूमार्फत नै जानु पर्ने हुन्छ।

अघिल्लो धनिहरूले कुनै पनि पहिलेको कारोबारमा हस्ताक्षर गरेका छैनन् भनेर भुक्तानकर्ताले जान्नको लागि हामीलाई एउटा तरिका चाहिन्छ। यसका लागि, सबैभन्दा प्रारम्भिक लेनदेनलाई मात्र मान्य ठानिन्छ जसकारण समयको अन्तरालमा दोहोरो खर्च गर्ने प्रयासहरूको हामी वास्ता गर्दैनौं। लेनदेनको अनुपस्थिति पुष्टि गर्ने एक मात्र तरिका भनेको सबै लेनदेनहरुको बारेमा सचेत हुनु हो। टकसालमा आधारित मोडेलमा टकसाल सबै लेनदेनको बारेमा सचेत थियो र कुन लेनदेन पहिले आइपुग्यो भनेर आफै निर्णय गर्थ्यो। यस्तो मोडेल एक विश्वसनीय पक्ष बिना पूरा गर्न लेनदेन सार्वजनिक रूपमा घोषणा गरिनु पर्छ [१], र लेनदेनहरु प्राप्त भएको कालक्रमको एकल इतिहासमा सहभागीहरू सहमत हुन हामीलाई एउटा प्रणाली चाहिन्छ। प्रत्येक लेनदेनको समयमा उक्त लेनदेन पहिलो पटक प्राप्त भएको कुरामा अधिकांश नोडहरूको सहमति भुक्तानीकर्तालाई प्रमाण स्वरुप चाहिन्छ।

३. टाइमस्ट्याम्प सर्भर

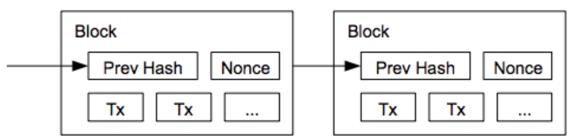
हामीले प्रस्ताव गरेको समाधान टाइमस्ट्याम्प सर्भरबाट सुरु हुन्छ। टाइमस्ट्याम्प सर्भरले टाइमस्ट्याम्प गर्नका लागि ब्लकको ह्यास लिएर काम गर्दछ र ह्यासलाई सर्बत्र रूपमा "अखबार वा युस्नेट पोस्ट [२-५]मा" जसरी प्रकाशित गर्दछ। टाइमस्ट्याम्पले के प्रमाणित गर्दछ भने ह्यासमा समावेश हुनका लागि उक्त डाटा टाइमस्ट्याम्पको बेला अवस्य नै अवस्थित भएको हुनुपर्दछ। प्रत्येक टाइमस्ट्याम्पले यसको ह्यासमा अघिल्लो टाइमस्ट्याम्प समावेश गरेर एउटा श्रृंखला (चेन) बनाउँछ; यसरी प्रत्येक अतिरिक्त टाइमस्ट्याम्पले अघिल्लो टाइमस्ट्याम्पलाई बलियो बनाउँछ।



४. कार्य-प्रमाणीकरण

पियर-टु-पियर आधारमा वितरित टाइमस्ट्याम्प सर्भर लागू गर्न हामीले अखबार वा युजनेट पोस्टहरू भन्दा एडम ब्याकको ह्यासक्यास [6] जस्तै कार्य-प्रमाणीकरण प्रणाली प्रयोग गर्न आवश्यक छ। कार्य-प्रमाणीकरणमा एउटा मानको खोजी गरिन्छ जुन SHA-256 जस्तो ह्यास गर्दा, ह्यास शून्य बिट्सको संख्याबाट सुरु हुन्छ। यसमा आवश्यक औसत कार्य शून्य बिट्सको संख्यामा घातीय छ र यो एकल ह्यास कार्यान्वयन गरेर प्रमाणित गर्न सकिन्छ।

हाम्रो टाइमस्ट्याम्प नेटवर्कको लागि हामीले ब्लकको ह्यासलाई आवश्यक शून्य बिट्स दिने मान नभेटेसम्म ब्लकमा नन्स बढाएर कार्य-प्रमाणीकरण कार्यान्वयन गर्छौं। कार्य-प्रमाणीकरणलाई आवश्यक पर्ने CPU शक्ति एक पटक खर्च गरिसकेपछि उक्त काम पुन: नगरी ब्लक परिवर्तन गर्न सकिँदैन। पछाडिका ब्लकहरू यसको पछि श्रृंखलावद्धरुपमा बाँधिएका हुनाले ब्लक परिवर्तन गर्न लाग्ने कार्यमा उक्त ब्लक पछिका सबै ब्लकहरू पुन: परिवर्तन गर्नै पर्ने हुन्छ।



कार्य-प्रमाणीकरणले बहुमतको निर्णयमा प्रतिनिधित्व निर्धारण गर्ने समस्यालाई पनि समाधान गर्छ। यदि बहुमत एक-आईपी-ठेगाना-एक-भोटमा आधारित भयो भने धेरै आईपीहरू आवंटित गर्न सक्षम जो कोहीले यसलाई विकृत गर्न सक्छ। कार्य-प्रमाणीकरण अनिवार्य रूपमा एक-सीपीयू-एक-भोट हो। बहुमतको निर्णय सबैभन्दा लामो श्रृंखलाद्वारा प्रतिनिधित्व गरिएको हुन्छ, जसमा कार्य-प्रमाणीकरणको लागि सबैभन्दा धेरै श्रम लगानी गरिएको हुन्छ । यदि CPU पावरको बहुमत इमानदार नोडहरूद्वारा नियन्त्रित छ भने इमानदार चेन सबैभन्दा छिटो बढ्नेछ र कुनै पनि प्रतिस्पर्धी चेनहरूभन्दा अगाडि बढ्छ । विगतको ब्लक परिमार्जन गर्नको लागि आक्रमणकारीले ब्लकको कार्य-प्रमाणीकरण र त्यस पछिका सबै ब्लकहरू पुन: प्रमाणीकरण गर्नुपर्दछ र त्यसपछि इमानदार नोडहरूको कामलाई भेटाएर उछिन्नुपर्छ । हामी पछि देखाउनेछौं कि पछिल्ला ब्लकहरू थपिएपछि सुस्त आक्रमणकारीले भेटाउने सम्भावना तीव्र रूपमा घट्छ।

समयसंगै हार्डवेयरको बढ्दो गति र नोडहरू चलाउन हुने भिन्न रुचिलाई प्रतिवाद गर्न कार्य-प्रमाणीकरणको कठिनाई प्रति घण्टा ब्लकहरूको औसत संख्यालाई लक्षित गर्दै चलिरहेको औसतद्वारा निर्धारण गरिन्छ। यदि ब्लकहरू धेरै छिटो उत्पन्न भएमा कार्य-प्रमाणीकरणको कठिनाई बढ्दछ।

५. नेटवर्क

नेटवर्क चलाउनको लागि निम्नानुसारका चरणहरू छन्:

१) नयाँ लेनदेन सबै नोडहरूमा प्रसारण गरिन्छ।

२) प्रत्येक नोडले नयाँ लेनदेनहरूलाई ब्लकमा सङ्कलन गर्दछ।

३) प्रत्येक नोडले आफ्नो ब्लकको लागि कठिन "कार्य-प्रमाणीकरण" फेला पार्न काम गर्दछ।

४) जब कुनै नोडले "कार्य-प्रमाणीकरण" फेला पर्दछ, यसले उक्त ब्लक सबै नोडहरूलाई प्रसारण गर्दछ।

५) यदि ब्लकमा भएका सबै लेनदेनहरू मान्य छन् र पहिले नै खर्च गरिएको छैनन् भने नोडहरूले त्यस ब्लकलाई स्वीकार गर्दछन्।

६) नोडहरूले ब्लकको स्वीकृति व्यक्त गर्न अघिल्लो ह्यासको रूपमा स्वीकृत भएको ब्लकको ह्यास प्रयोग गरेर चेनमा अर्को ब्लक सिर्जना गर्ने काम गर्दछन् ।

नोडहरूले सधैं सबैभन्दा लामो श्रृंखलालाई सही श्रृंखला मान्छन् र यसलाई विस्तार गर्ने काम गरिरहन्छन । यदि दुई नोडहरूले एकैसाथ आगामी ब्लकको भिन्न-भिन्न संस्करणहरू प्रसारण गरे भने, केही नोडहरूले यौटा र अरु केही नोडहरूले अर्को संस्करण प्राप्त गर्न सक्छन्। यस्तो अवस्थामा तिनीहरूले प्राप्त गरेको पहिलो संस्करणको ब्लकमा काम गर्छन्, तर अर्को संस्करणको श्रृंखला पनि लामो हुन सक्ने मामलामा हुँदा त्यसलाई शाखाको रुपमा सुरक्षित राख्छन् । यी दुई संस्करणहरुको छिनोफानो तब हुन्छ जब अर्को कार्य-प्रमाणीकरण फेला पर्छ र त्यसबाट एउटा शाखा लामो हुन्छ; अनि अर्को शाखामा काम गरिरहेका नोडहरू त्यसपछि लामो शाखामा स्विच हुन्छन् ।

नयाँ लेनदेनको प्रसारणहरू सबै नोडहरूमा पुग्न आवश्यक हुदैन। जबसम्म लेनदेनको प्रसारणहरू धेरै भन्दा धेरै नोडहरूमा पुग्छन्, तिनीहरू कुनै आगामी ब्लकमा अवश्य समावेश हुन्छन् । । छुटेका सन्देशहरूबाट ब्लक प्रसारणहरू प्रभावित हुँदैनन्। यदि नोडले कुनै ब्लक प्राप्त गरेन भने अर्को ब्लक प्राप्त गर्दा पहिलेको ब्लकबारे अनुरोध गर्छ र यसरी कुनै ब्लक छुटेको भए थाहा हुन्छ ।

६. प्रोत्साहन

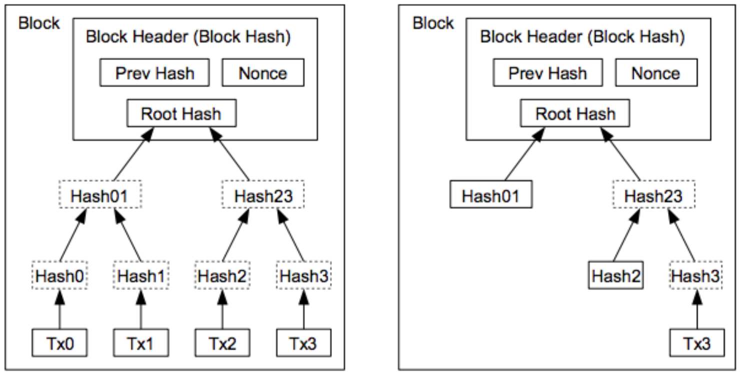
परम्परा अनुसार ब्लकमा हुने पहिलो लेनदेन एक विशेष लेनदेन हो जसले ब्लकको सिर्जनाकर्ताको स्वामित्वमा रहेको एक नयाँ मुद्रा सुरु गर्छ । यसले नोडहरूका लागि नेटवर्कलाई समर्थन गर्न प्रोत्साहन थप्दछ, र मुद्रा जारी गर्ने कुनै केन्द्रीय अधिकारी नभएकोले सुरुमा मुद्राहरू परिसंचरणमा वितरण गर्ने तरिका प्रदान गर्दछ। स्थिर मात्रामा नयाँ मुद्राहरू निरन्तर थप्नु भनेको सुन खानीमा काम गर्नेहरूले स्रोतहरू खर्चेर सुनलाई प्रचलनमा थप्न गर्नु जस्तै हो। हाम्रो केसमा यो CPU को समय र बिजुलीको खर्च हो ।

नोडहरुको प्रोत्साहनलाई लेनदेनको शुल्कले पनि वित्त पोषित गर्न सकिन्छ। लेनदेनको आउटपुट मूल्य इनपुट मूल्य भन्दा कम हुँदा यसको भिन्नता भनेको लेनदेन शुल्क हो जुन लेनदेन समावेश भएको ब्लकको प्रोत्साहन मूल्यमा थपिन्छ। एक पटक पूर्वनिर्धारित संख्याको मुद्रा प्रचलनमा प्रवेश गरेपछि प्रोत्साहन पूर्ण रूपले लेनदेन शुल्कमा परिवर्तन हुन सक्छ र पूर्ण रूपमा मुद्रास्फीति मुक्त हुन सक्छ।

प्रोत्साहनद्वारा नोडहरूलाई इमानदार रहन प्रोत्साहित गर्न सक्छ। यदि एक लोभी आक्रमणकारीले सबै इमानदार नोडहरू भन्दा बढी CPU पावर भेला गर्न सक्यो भने, उसले यसलाई मानिसहरूलाई ठगी गर्नका लागि आफूले तिरेको भुक्तानीहरू चोर्न वा नयाँ मुद्राहरु उत्पन्न गर्नका लागि प्रयोग गर्न सक्छ। उसलाई आफू अनुकूल नियमहरूद्वारा खेल्नु बढी लाभदायक हुन जान्छ जसमा प्रणाली र आफ्नै सम्पत्तिको वैधतालाई कमजोर पार्नु भन्दा अरू सबैको तुलनामा आफ्नो धेरै नयाँ मुद्रा होस् ।

७. डिस्क स्पेसको पुन: दावी

एक पटक मुद्राको पछिल्लो लेनदेन पर्याप्त ब्लकहरू मुनि गाडिएपछि, डिस्कको ठाउँ बचत गर्न त्यस लेनदेन भन्दा अघि खर्च गरिएका लेनदेनहरूलाई खारेज गर्न सकिन्छ। ब्लकको ह्यासलाई नबिगारिकन यस्तो कार्य गर्न लेनदेनहरू मर्कल ट्री [७][२][५] मा ह्यास गरिन्छ । यसो गर्दा ब्लकको ह्यासमा मूल (रूट) मात्र समावेश गरिएको हुन्छ। त्यसपछि मर्कल ट्रीका हाँगाहरू बन्द गरेर पुरानो ब्लकहरूलाई सानो (कम्प्याक्ट) पार्न सकिन्छ। भित्री ह्यासहरू भण्डार गर्न आवश्यक पर्दैन ।

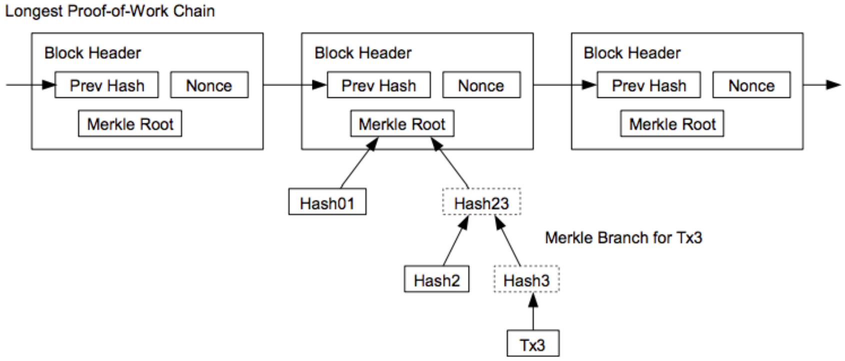


Transactions Hashed in a Merkle Tree — After Pruning Tx0-2 from the Block

कुनै पनि लेनदेन बिनाको ब्लक हेडर लगभग ८० बाइट्सको हुन्छ। यदि हामीले प्रत्येक १० मिनेटमा ब्लकहरू उत्पन्न हुन्छ भन्ने मान्यौ भने, ८० बाइट्स * ६ * २४ * ३६५ = ४.२ एमबी प्रति वर्ष। २००८ मा बिक्री हुने कम्प्यूटर प्रणालीहरू सामान्यतया २ GB को RAM संग बिक्री हुन्छ , र मूरको नियम अनुसार यसको वर्तमान वृद्धि प्रति वर्ष १.२ GB ले हुने भविष्यवाणी छ । त्यसैले ब्लक हेडरहरू मेमोरीमा राख्नै पर्दा पनि भण्डारणमा कुनै समस्या हुँदैन।

८. सरलीकृत भुक्तानी प्रमाणिकरण

पूर्ण नेटवर्क नोड संचालन नगरी भुक्तानीहरू प्रमाणित गर्न सम्भव छ। एक प्रयोगकर्ताले सबैभन्दा लामो कार्य-प्रमाणीकरण चेनको ब्लक हेडरहरूको प्रतिलिपि मात्र राख्न आवश्यक छ, जुन उसले प्राप्त गर्न आफूसँग सबैभन्दा लामो चेन छ भनी विश्वस्त नभएसम्म नेटवर्क नोडहरूलाई अनुरोध गर्न सक्छ, र उसले लेनदेनलाई ब्लकमा लिङ्क गर्ने मर्कल शाखा प्राप्त गर्दछ जसले लेनदेनलाई त्यहि लेनदेन टाइमस्ट्याम्प भएको ब्लकमा लिंक गरिदिन्छ। उसले आफ्नै लागि लेनदेन जाँच गर्न सक्दैन, तर यसलाई चेनको एउटा ठाउँमा लिङ्क गरेर, उसले देख्न सक्छ कि नेटवर्क नोडले यसलाई स्वीकार गरेको छ, र नेटवर्कको स्वीकृति यसपछि आउने ब्लकहरूबाट थप पुष्टि हुन्छ।

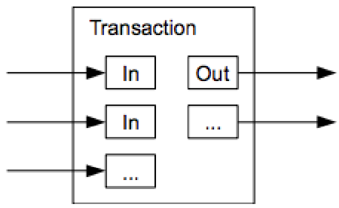


यस प्रकार प्रमाणिकरण तबसम्म भरपर्दो हुन्छ जबसम्म इमानदार नोडहरूले नेटवर्कलाई नियन्त्रण गर्छ, तर यदि आक्रमणकारीद्वारा नियन्त्रित भयो भने नेटवर्क झन् बढी कमजोर हुन्छ। जबकि नेटवर्क नोडहरूले लेनदेनहरू आफैंमा प्रमाणित गर्न सक्छन्, सरलीकृत विधिलाई आक्रमणकारीको बनावटी लेनदेनहरूद्वारा झुक्काउन सकिन्छ जबसम्म आक्रमणकारीले नेटवर्कलाई नियन्त्रित गर्छ। यसबाट बच्ने एउटा रणनीति भनेको जब नेटवर्क नोडहरूले अवैध ब्लक पत्ता लगाउँछन् तब अलर्टहरू स्वीकार गर्नु हो ; यससंगै प्रयोगकर्ताको सफ्टवेयरलाई पूर्ण ब्लक र अलर्ट भएको लेनदेन डाउनलोड गरी असंगतता

पुष्टि गर्न प्रोत्साहन गरिन्छ। बारम्बार भुक्तानीहरू प्राप्त गर्ने व्यवसायहरूले अझ थप स्वतन्त्र सुरक्षा र छिटो प्रमाणीकरण प्राप्त गर्न आफ्नै नोडहरू चलाउन चाहन सक्छन्।

९. मूल्यको संयोजन र विभाजन

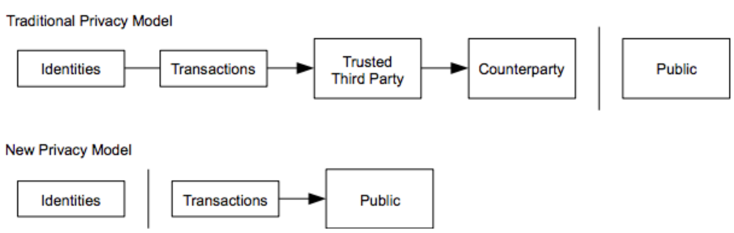
यद्यपि मुद्राहरू एक-एक गरी व्यवस्थापन गर्न सम्भव छ, प्रत्येक मुद्राको (स-सानो एकाइ) स्थानान्तरणका लागि छुट्टा-छुट्टै लेनदेन गर्न असहज हुन्छ ।मूल्य विभाजन र संयोजनको अनुमति प्रधान गर्न लेनदेनले बहु इनपुट र आउटपुटहरू समावेश गर्दछ। सामान्यतया हुने भनेको अघिल्लो ठूलो लेनदेनबाट एकल इनपुट वा साना रकमहरू संयोजन गरिएको बहु इनपुटहरू हुन्, र बढीमा दुईवटा आउटपुटहरू हुन्छन्: एउटा भुक्तानीको लागि, र एउटा यदि कुनै वांकी रकम भएमा प्रेषकलाई फिर्ता दिन।



यो कुरा ध्यान दिनुपर्छ कि एउटा लेनदेन धैरै लेनदेनहरूमा निर्भर हुन्छ, र ती लेनदेनहरू अन्य धेरैमा निर्भर हुन्छन् तर यसलाई फिंजाउन (फ्यान-आउट) यहाँ कुनै समस्या छैन। यहाँ लेनदेनको इतिहासको पूर्ण स्वायत्त प्रतिलिपि निकाल्न कहिल्यै आवश्यक पर्दैन।

१०. गोपनीयता

परम्परागत बैंकिङ ढांचाले संलग्न पक्षहरू र विश्वसनीय तेस्रो पक्षहरूलाई सीमित जानकारीको पहुँच दिएर एक स्तरको गोपनीयताको कायम गर्दछ। सबै लेनदेनहरू सार्वजनिक रूपमा प्रसारित गर्नु पर्ने आवश्यकता हुँदा यस परम्परागत बैंकिङ विधि हाम्रो प्रणालीमा राख्न सकिन्न, तर सार्वजनिक कुञ्जीहरू गुमनाम राखेर गोपनीयता अझै पनि कायम राख्न सकिन्छ। सर्वसाधारणहरु कसैले कसैलाई रकम पठाइरहेको देख्न सक्छन्, तर यसमा लेनदेन को-संग जोडिएको छ भन्ने जानकारी हुदैन। यो स्टक एक्सचेन्जहरू द्वारा जारी गरिने जानकारीको स्तरसँग मिल्दोजुल्दो छ, जहाँ व्यक्तिगत व्यापारहरूको समय र आकार ("टेप") सार्वजनिक गरिन्छ, तर व्यापारिक पक्षहरू को थिए भनेर भनिदैन।



अतिरिक्त सुरक्षाको रूपमा प्रत्येक लेनदेनको लागि एउटा नयाँ कुञ्जीको जोडा (निजी र सार्वजनिक) प्रयोग गरिनुपर्छ ताकि लेनदेनहरुको साझा धनिलाई पहिचान हुनबाट जोगाउन सकिन्छ। अझै बहु-इनपुट भएको लेनदेनमा उक्त इनपुटहरू एउटै मालिकको स्वामित्वमा हुने हुँदा उसको पहिचान अनिवार्य रूपमा प्रकट हुन्छ। यसमा जोखिम यो हो कि यदि कुञ्जीको धनि पत्ता लाग्यो भने, त्यस धनिसँग सम्बन्धित अन्य लेनदेनहरू थाहा पाइन सक्छ।

११. गणना

हामी परिदृश्यलाई इमानदार चेन भन्दा छिटो वैकल्पिक चेन उत्पन्न गर्ने प्रयास गर्ने आक्रमणकारीको परिदृश्यलाई विचार गर्छौं। यदि यो पूरा भयो भने, यसले प्रणालीलाई स्वेच्छाचारी परिवर्तनहरूको लागि खुला गर्दैन, जस्तै पातलो हावाबाट मूल्य सिर्जना गर्ने वा आक्रमणकारीसँग कहिल्यै नपर्ने पैसा लिने। नोडहरूले भुक्तानीको रूपमा अवैध लेनदेन स्वीकार गर्दैनन्, र इमानदार नोडहरूले तिनीहरूलाई समावेश भएको ब्लकलाई कहिल्यै स्वीकार गर्दैनन्। आक्रमणकारीले भर्खरै खर्च गरेको पैसा फिर्ता लिन आफ्नै लेनदेनहरू मध्ये एउटा मात्र परिवर्तन गर्ने प्रयास गर्न सक्छ।

इमानदार श्रृंखला (चेन) र एक आक्रमणकारी श्रृंखलाबीचको दौडलाई "द्विपदीय अनियमित चाल" को रूपमा चित्रण गर्न सकिन्छ। इमानदार श्रृंखलाको एक ब्लक विस्तार गर्दै +१ ले नेतृत्व बढाउनुलाई सफलताको घटना मानिन्छ, र असफलताको घटना भनेको आक्रमणकरी श्रृंखलाको एक ब्लक विस्तार गर्दै -१ ले अन्तर घटाउनु हो।

कुनै एक अन्तरबाट आक्रमणकारीले इमानदार श्रृंखलाको नेतृत्व उछिन्ने सम्भाव्यता भनेको "जुवा-खेलाडीको बर्बादी" समस्यासँग मिल्दोजुल्दो छ। मानौं असीमित पैसा (क्रेडिट) भएको एउटा जुवाडे घाटाबाट खेल सुरु गर्छ र घाटा लागेको लगानी पुन: प्राप्ति गर्न असीमित पटक खेल्छ। उसले आफ्नो लगानी कहिले पुन: प्राप्ति गर्न सक्छ भन्ने सम्भावनाको वा भनौ आक्रमणकारीले इमानदार श्रृंखला (चेन) को नेतृत्व उछिन्न सक्ने सम्भावनाको हामी निम्नानुसार गणना गर्न सक्छौं [८]:

p = एक इमानदार नोडले अर्को ब्लक फेला पर्ने सम्भाव्यता

q = आक्रमणकर्ताले अर्को ब्लक फेला पर्ने सम्भाव्यता

qz = आक्रमणकर्ताले z संख्याको ब्लकहरू पछाडिबाट उछिन्ने सम्भाव्यता

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

हाम्रो p > q भन्ने अनुमान अनुसार सम्भाव्यता घातीय (तीव्र) रूपमा घट्छ जब आक्रमणकरीले उछिन्नु पर्ने ब्लकहरूको संख्या बढ्दै जान्छ। जित्ने सम्भावना उसको बिरूद्ध हुँदा यदि उसले सुरुमै भाग्यशाली फड्को मरेर ठुलो अन्तर बनाउन सकेन भने, उसले नेतृत्व लिने सम्भावनाहरू कम हुँदै जान्छ र ऊ झनै पछाडि पर्छ।

नयाँ लेनदेन प्राप्त गर्दा पठाउने (प्रेषक) पक्षले उक्त लेनदेन परिवर्तन गर्न सक्दैन भनेर पर्याप्त रुपमा निश्चित हुन त्यस लेनदेन प्राप्त गर्ने पक्षले कति समय पर्खनु पर्छ भन्नेबारे अब हामी विचार गर्छौं। हामीले प्रेषक एक आक्रमणकारी हो भनी मान्दछौं, जसले प्रापकलाई केही समयको लागि भुक्तान गरेको विश्वास दिलाउन चाहन्छ, अनि केही समय बितेपछि त्यस भुक्तानी आफैलाई फिर्ता गराउछ। यस्तो हुँदा प्रापकलाई सचेत गराइनेछ, तर प्रेषकले ढिलाईको अपेक्षा गर्छ।

लेनदेन प्राप्त गर्ने पक्षले (प्रापकले) हस्ताक्षर गर्नु अघि नयाँ कुञ्जीको जोडा (निजी कुञ्जी र सार्वजनिक कुञ्जी) बनाउँछ र प्रेषकलाई सार्वजनिक कुञ्जी दिन्छ। यसले आक्रमणकरी प्रेषकलाई धेरै अगाडि पुग्न भाग्यशाली नहुँदासम्म लगातार काम गरेर समयभन्दा अगाडि नै ब्लकहरूको श्रृंखला तयार पारेको क्षणमा उक्त लेनदेन कार्यान्वयन गर्नबाट रोक्छ। बेइमान प्रेषकले लेनदेन पठाइसकेपछि समानान्तर श्रृंखलामा गोप्य रूपले काम गर्न थाल्छ जसमा उक्त लेनदेनको वैकल्पिक संस्करण रहेको हुन्छ।

लेनदेन प्राप्त गर्ने पक्षले लेनदेन ब्लकमा नथपिँदासम्म र त्यसपछि z संख्याका ब्लकहरू जोडिएसम्म पर्खन्छ। उसलाई आक्रमणकारीले गरेको प्रगतिको सही मात्रा थाहा हुदैन, तर इमानदार ब्लकहरूले प्रति ब्लक लिएको औसत अपेक्षित समयको अनुमान गर्ने हो भने आक्रमणकारीको सम्भावित प्रगति भनेको पोइसन वितरण हुनेछ जसको अपेक्षित मूल्य भनेको :

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

आक्रमणकारीले अहिले पनि उछिन्न सक्ने सम्भाव्यता प्राप्त गर्न हामी उसले गरेको प्रगतिको प्रत्येक मात्राको पोइसन घनत्वलाई उसले त्यस बिन्दुबाट उछिन्न सक्ने सम्भावनाद्वारा गुणन गर्छौं:

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

वितरणको अनन्त पुच्छरलाई संक्षेप गर्नबाट बच्न पुन: व्यवस्थित गर्दै...

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

C कोडमा रूपान्तरण गर्दै...

```
#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;
}
```

केही नतिजाहरू निकाल्दै हेर्दा, हामी z संगै सम्भाव्यताको मुल्य घातीय रूपमा घटेको देख्न सक्छौं।

```
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
```

०.१% भन्दा कम मुल्य को P का लागि समाधान गर्दै...

```
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
```

१२. निष्कर्ष

हामीले विश्वासमा भर नपर्ने विद्युतीय लेनदेनको लागि एक प्रणाली प्रस्ताव गरेका छौं। हामीले डिजिटल हस्ताक्षरले बनेको मुद्राको सामान्य ढाँचाबाट सुरु गर्‍यौं, जसले स्वामित्वको बलियो नियन्त्रण प्रदान गर्दछ, तर दोहोरो खर्च रोक्ने तरिका बिना अपूर्ण छ। यसलाई समाधान गर्न हामीले "कार्य-प्रमाणीकरण" प्रयोग गरेर लेनदेनको सार्वजनिक इतिहास रेकर्ड गर्ने एक पक्ष-पक्षबीचको नेटवर्क प्रस्ताव गरेका छौं जसमा इमानदार नोडहरूले CPU पावर को बहुमत नियन्त्रण गर्दा आक्रमणकारीका लागि स्वेच्छाचारी परिवर्तन गर्ने अवसर द्रुत रूपले गणनात्मक रूपमा अव्यावहारिक हुन्छ।

नेटवर्क आफैमा भएको असंरचित सरलतामा बलियो छ। नोडहरू थोरै समन्वयका साथ एकैचोटि काम गर्छन्। नोडहरूलाई पहिचान गर्न आवश्यक छैन, किनकि लेनदेनको सन्देशहरू कुनै एक विशेष ठाउँमा पठाइएका हुदैनन् र आफ्नो सर्वश्रेष्ठ प्रयासको आधारमा सन्देशहरू प्रसारित गरे पुग्छ ।आफू अनुपस्थित हुदा के भयो भन्ने प्रमाणको रूपमा "कार्य-प्रमाणीकरण"को श्रृंखलालाई स्वीकार गर्दै नोडहरूले आफ्नो इच्छा अनुसार नेटवर्क छोड्न र पुन:

सामेल हुन सक्छन्। मान्य ब्लकहरूको स्वीकृति व्यक्त गर्दै तिनीहरूलाई विस्तार गर्ने काम र अवैध ब्लकहरूलाई अस्वीकार गरेर अस्वीकृति व्यक्त गर्ने काम नोडहरूले आफ्नो CPU शक्तिबाट मतदान गर्छन्। कुनै पनि आवश्यक नियम र प्रोत्साहनहरू यहि सहमतिको संयन्त्रबाट लागू गर्न सकिन्छ।

सन्दर्भहरू

[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.