BITFURY

बिटकॉइन: पियर-टू-पियर इलेक्ट्रॉनिक नकद प्रणाली

Bitcoin: The peer-to-peer electronic cash system

Translation completed in partnership
with the Blockchain Foundation of India

GBBC Global Blockchain Business Council

BTA

बिटकॉइन: पियर-टू-पियर इलेक्ट्रॉनिक नकद प्रणाली

सातोशी नाकामोटो
satoshin@gmx.com
www.bitcoin.org

सार। इलेक्ट्रॉनिक नकद का संपूर्ण रूप से पियर-टू-पियर संस्करण किसी वित्तीय संस्थान के माध्यम से गए बिना एक पक्ष से दूसरे को सीधे ऑनलाइन भुगतान भेजने देगा। डिजिटल हस्ताक्षर आंशिक समाधान प्रदान करते हैं, परंतु यदि दोहरे-व्यय को रोकने के लिए अब भी भरोसेमंद तीसरे पक्ष की ज़रूरत हो तो मुख्य लाभ खो जाते हैं। हम पियर-टू-पियर नेटवर्क का उपयोग करके दोहरे-व्यय की समस्या का समाधान पेश करते हैं। नेटवर्क हैश-आधारित प्रूफ-ऑफ-वर्क की एक चालू श्रृंखला में लेनदेनों को बांटकर, एक ऐसा अभिलेख बनाकर लेनदेनों को टाइमस्टैम्प करता है जिसे प्रूफ-ऑफ-वर्क को दोबारा किए बिना बदला नहीं जा सकता। सबसे लंबी श्रृंखला न केवल देखी गई घटनाओं के अनुक्रम के प्रमाण के रूप में, बल्कि इस बात के प्रमाण के रूप में भी काम करती है कि वह CPU पावर के सबसे बड़े निकाय से आई है। जब तक अधिकांश CPU पावर को ऐसे नोड्स नियंत्रित करते हैं जो नेटवर्क पर हमला करने में सहयोग नहीं देते, तब तक वे सबसे लंबी श्रृंखला बनाएंगे और हमलावरों को पराजित करेंगे। नेटवर्क के लिए न्यूनतम संरचना की आवश्यकता होती है। संदेश श्रेष्ठ प्रयास के आधार पर प्रसारित किए जाते हैं, और नोड्स अपनी अनुपस्थिति में जो हुआ उसके प्रमाण के रूप में प्रूफ-ऑफ-वर्क की सबसे लंबी श्रृंखला को स्वीकार करके, इच्छानुसार नेटवर्क को छोड़ सकते हैं और उसमें फिर से जुड़ सकते हैं।

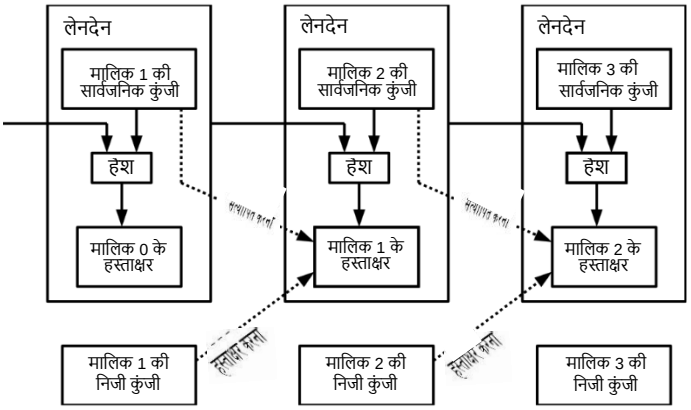
1. प्रस्तावना

इंटरनेट पर व्यापार इलेक्ट्रॉनिक भुगतानों के प्रसंस्करण के लिए भरोसेमंद तीसरे पक्षों के रूप में कार्य करने वाले वित्तीय संस्थानों पर लगभग पूरी तरह से निर्भर रहा है। यद्यपि यह प्रणाली अधिकांश लेनदेनों के लिए काफी अच्छे ढंग से काम करती है, लेकिन इसमें अब भी विश्वास आधारित मॉडल की अंतर्निहित कमज़ोरियां हैं। संपूर्ण रूप से गैर-विपर्ययी (नॉन-रिवर्सिबल) लेनदेन वास्तव में संभव नहीं हैं, क्योंकि वित्तीय संस्थान मध्यस्थता के विवादों से बच नहीं सकते। मध्यस्थता की लागत लेनदेन के व्यावहारिक न्यूनतम आकार को सीमित करके तथा छोटे आकस्मिक लेनदेनों की संभावना को समाप्त करके, लेनदेनों की लागत को बढ़ा देती है, और गैर-विपर्ययी सेवाओं के लिए गैर-विपर्ययी भुगतान करने की क्षमता खोने में व्यापक लागत लगती है। विपर्यय की संभावना के साथ भरोसे की आवश्यकता बढ़ती है। व्यापारियों को उनके ग्राहकों से सावधान रहना पड़ता है, और उनसे अधिक जानकारी मांगकर परेशान करना पड़ता है जिसकी आवश्यकता उन्हें अन्यथा नहीं होती। कुछ प्रतिशत धोखाधड़ी अनिवार्य मानी जाती है। भौतिक मुद्रा का प्रयोग करके इन लागतों और भुगतान की अनिश्चितताओं से निजी रूप से बचा जा सकता है, परंतु किसी भरोसेमंद पक्ष के बिना किसी संचार माध्यम से भुगतान करने के लिए कोई क्रियाविधि नहीं है।

आवश्यकता है विश्वास के बजाय क्रिप्टोग्राफिक प्रमाण पर आधारित इलेक्ट्रॉनिक भुगतान प्रणाली की, जो भरोसेमंद तीसरे पक्ष की ज़रूरत के बिना किन्हीं दो इच्छुक पक्षों को सीधे एक दूसरे के साथ लेनदेन करने दे। ऐसे लेनदेन जिन्हें उलटना संगणन की दृष्टि से अव्यावहारिक होता है वे विक्रेताओं की धोखाधड़ी से रक्षा करेंगे, और ख़रीदारों की रक्षा के लिए नियमित एस्क्रो क्रियाविधियों को आसानी से क्रियान्वित किया जा सकता है। इस पत्र में हम लेनदेनों के कालक्रमिक क्रम का संगणनात्मक प्रमाण उत्पन्न करने के लिए पियर-टू-पियर वितरित टाइमस्टैम्प सर्वर का प्रयोग करके दोहरे-व्यय की समस्या का समाधान पेश करते हैं। यह प्रणाली तब तक सुरक्षित है जब तक ईमानदार नोड्स हमलावर नोड्स के किसी सहयोगी समूह से अधिक CPU पावर को संयुक्त रूप से नियंत्रित करते हैं।

2. लेनदेन

इलेक्ट्रॉनिक सिक्के को हम डिजिटल हस्ताक्षरों की एक श्रृंखला के रूप में परिभाषित करते हैं। प्रत्येक मालिक पिछले लेनदेन के हैश और अगले मालिक की सार्वजनिक कुंजी पर डिजिटल हस्ताक्षर करके एवं इन्हें सिक्के के अंत में जोड़कर अगले मालिक को सिक्का अंतरित करता है। आदाता स्वामित्व की श्रृंखला को सत्यापित करने के लिए हस्ताक्षरों को सत्यापित कर सकता है।

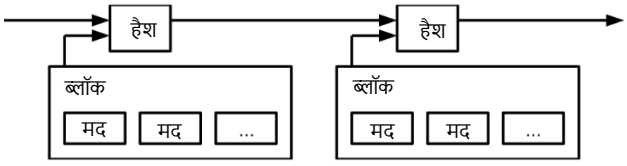


निश्चय ही समस्या यह है कि आदाता इस बात को सत्यापित नहीं कर सकता कि मालिकों में से एक ने सिक्के का दोहरा-व्यय नहीं किया। इसका एक सामान्य समाधान है प्रत्येक लेनदेन की दोहरे-व्यय के लिए जाँच करने वाले भरोसेमंद केन्द्रीय प्राधिकारी, या टकसाल को प्रविष्ट करना। प्रत्येक लेनदेन के बाद, नया सिक्का जारी करने के लिए सिक्के को टकसाल में लौटाया जाना चाहिए, और इस बात पर भरोसा किया जाता है कि केवल टकसाल से सीधे जारी किए गए सिक्कों का दोहरा-व्यय नहीं किया गया। इस समाधान से जुड़ी समस्या यह है कि पूरी धन प्रणाली का परिणाम टकसाल का संचालन करने वाली कंपनी पर निर्भर करता है, क्योंकि एक बैंक की तरह, प्रत्येक लेनदेन को उनसे गुज़रना पड़ता है।

हम एक ऐसा तरीका चाहते हैं जिससे आदाता यह जान सके कि पिछले मालिकों ने पहले के किन्हीं लेनदेनों पर हस्ताक्षर नहीं किए थे। हमारे उद्देश्यों के लिए प्रारंभिक लेनदेन ही महत्वपूर्ण है, इसलिए हम दोहरा-व्यय करने के बाद के प्रयासों की परवाह नहीं करते। लेनदेन की अनुपस्थिति की पुष्टि करने का एकमात्र तरीका है सभी लेनदेनों के बारे में पता होना। टकसाल आधारित मॉडल में, टकसाल को सभी लेनदेनों की जानकारी थी और इस बात का निर्णय करती थी कि कौन-सा लेनदेन पहले प्राप्त हुआ। इसे किसी भरोसेमंद पक्ष के बिना संपादित करने के लिए लेनदेनों को सार्वजनिक रूप से घोषित करना होगा [1], और वे किस क्रम में प्राप्त हुए थे इसके बारे में सिर्फ़ एक इतिहास पर प्रतिभागी सहमति बना सकें इसके लिए हमें एक प्रणाली की ज़रूरत है। आदाता प्रमाण चाहता है कि प्रत्येक लेनदेन के समय अधिकांश नोड्स इस बारे में सहमत थे कि वह पहले प्राप्त हुआ था।

3. टाइमस्टैम्प सर्वर

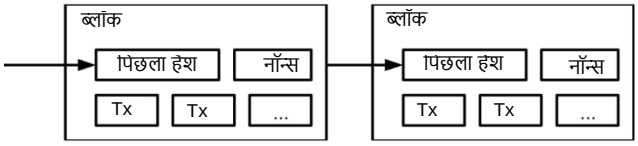
हम जो समाधान पेश करते हैं वह टाइमस्टैम्प से शुरू होता है। टाइमस्टैम्प सर्वर टाइमस्टैम्प की जाने वाली मदों के ब्लॉक के हैश को लेकर तथा उस हैश को व्यापक रूप से प्रकाशित करके कार्य करता है, जैसे कि समाचार पत्र या यूज़नेट पोस्ट में [2-5]। टाइमस्टैम्प प्रमाणित करता है कि हैश में प्रविष्ट होने के लिए डेटा उस समय, स्पष्टत:, अस्तित्व में रहा होगा। प्रत्येक टाइमस्टैम्प इसके हैश में पिछला टाइमस्टैम्प शामिल करके एक श्रृंखला बनाता है, जिसमें प्रत्येक अतिरिक्त टाइमस्टैम्प उसके पहले वाले टाइमस्टैम्पों को प्रबलित करता है।



4. प्रूफ-ऑफ-वर्क

पियर-टू-पियर आधार पर वितरित टाइमस्टैम्प सर्वर को क्रियान्वित करने के लिए हमें समाचार पत्र या यूज़नेट पोस्ट्स के बजाय एडम बैक के हैशकैश [6] जैसी प्रूफ-ऑफ-वर्क प्रणाली का प्रयोग करना होगा। प्रूफ-ऑफ-वर्क में किसी मूल्य के लिए स्कैनिंग करना शामिल होता है जिसे जब हैश किया जाता है, जैसे SHA-256 से, तो हैश अनेक ज़ीरो बिट्स से शुरू होता है। आवश्यक ज़ीरो बिट्स की संख्या के संदर्भ में ज़रूरी औसत कार्य घातीय है और इसे एक हैश को कार्यान्वित करके सत्यापित किया जा सकता है।

हमारे टाइमस्टैम्प नेटवर्क के लिए, हम ब्लॉक के हैश को आवश्यक ज़ीरो बिट्स देने वाला मूल्य न मिल जाए तब तक ब्लॉक में नॉन्स को बढ़ाकर प्रूफ-ऑफ-वर्क को क्रियान्वित करते हैं। एक बार प्रूफ-ऑफ-वर्क को पूरा कराने के लिए CPU के प्रयास को खर्च कर देने के बाद कार्य को दोबारा किए बिना ब्लॉक को बदला नहीं जा सकता। चूँकि बाद वाले ब्लॉक उससे जकड़ दिए जाते हैं, ब्लॉक को बदलने के कार्य में उसके बाद वाले सभी ब्लॉक पर दोबारा कार्य करना शामिल होगा।



प्रूफ-ऑफ-वर्क बहुसंख्या निर्णयन में प्रतिनिधित्व निर्धारित करने की समस्या का भी हल निकालता है। यदि बहुसंख्या वन-IP-एड्रेस-वन-वोट पर आधारित हो, तो उसे कई IPs आवंटित कर सकने वाला कोई व्यक्ति नष्ट कर सकता है। प्रूफ-ऑफ-वर्क मूल रूप से वन-CPU-वन-वोट है। बहुसंख्या निर्णय का प्रतिनिधित्व सबसे लंबी श्रृंखला करती है, जिसमें सबसे अधिक प्रूफ-ऑफ-वर्क प्रयास निवेश किया गया होता है। यदि अधिकांश CPU पावर ईमानदार नोड्स द्वारा नियंत्रित होता है, तो ईमानदार श्रृंखला में सबसे तेज़ गति से वृद्धि होगी और वह किसी प्रतिस्पर्धी श्रृंखला से आगे निकल जाएगी। किसी विगत ब्लॉक को परिवर्तित करने के लिए हमलावर को उस ब्लॉक के और उसके बाद वाले सभी ब्लॉक्स के प्रूफ-ऑफ-वर्क को दोबारा करना होगा और फिर ईमानदार नोड्स के कार्य की बराबरी पर आना होगा और उससे बेहतर कार्य करना होगा। हम आगे दिखाएंगे कि जैसे-जैसे बाद वाले ब्लॉक्स जोड़े जाते हैं वैसे-वैसे धीमे हमलावर की बराबरी पर आने की संभावना घातीय रूप से कम होती जाती है।

कालांतर में हार्डवेयर की गति बढ़ाने और नोड्स के संचालन में परिवर्ती रूचि की क्षतिपूर्ति करने के लिए प्रूफ-ऑफ-वर्क की कठिनाई प्रति घंटा ब्लॉक्स की औसत संख्या को लक्ष्यांकित करने वाला चल औसत निर्धारित करता है। यदि वे बहुत तेज़ी से उत्पन्न होते हैं, तो कठिनाई बढ़ जाती है।

5. नेटवर्क

नेटवर्क का संचालन करने के निम्नलिखित कदम हैं:

1) नए लेनदेन सभी नोड्स पर प्रसारित किए जाते हैं।
2) प्रत्येक नोड नए लेनदेनों को एक ब्लॉक में एकत्र करता है।
3) प्रत्येक नोड अपने ब्लॉक के लिए कठिन प्रूफ-ऑफ-वर्क खोजने पर काम करता है।
4) जब नोड प्रूफ-ऑफ-वर्क खोज लेता है तो वह ब्लॉक को सभी नोड्स पर प्रसारित करता है।

5) **नोड्स तभी ब्लॉक को स्वीकार करते हैं जब उसमें सभी लेनदेन मान्य होते हैं और पहले से ही व्यय नहीं किए गए होते।**
6) **नोड्स स्वीकृत ब्लॉक के हैश को पिछले हैश के रूप में प्रयोग करके श्रृंखला में अगला ब्लॉक बनाने पर कार्य करके उस ब्लॉक की स्वीकृति व्यक्त करते हैं।**

नोड्स हमेशा सबसे लंबी श्रृंखला को सही वाली श्रृंखला समझते हैं और उसका विस्तार करने पर कार्य करते रहते हैं। यदि दो नोड्स अगले ब्लॉक के विभिन्न संस्करणों को एक साथ प्रसारित करें, तो कुछ नोड्स पहले एक या दूसरा प्राप्त कर सकते हैं। इस मामले में, वे उन्हें पहले प्राप्त हुए ब्लॉक पर कार्य करते हैं, लेकिन दूसरी शाखा को सहेज लेते हैं जब वह लंबी बन जाती है। जब अगला प्रूफ-ऑफ-वर्क मिल जाता है और एक शाखा लंबी हो जाती है तब कड़ी टूट जाती है; इसके बाद जो नोड्स दूसरी शाखा पर कार्य कर रहे थे वे लंबी वाली शाखा पर कार्य करने लगते हैं।

नए लेनदेनों के प्रसारणों का सभी नोड्स पर पहुँचना ज़रूरी नहीं है। यदि वे कई नोड्स पर पहुँचते हैं तो वे जल्दी ही किसी ब्लॉक में प्रविष्ट होंगे। ब्लॉक प्रसारण स्थगित संदेशों के प्रति भी सहिष्णु होते हैं। यदि कोई नोड किसी ब्लॉक को प्राप्त नहीं करता, तो अगला ब्लॉक प्राप्त होने पर वह उसके लिए अनुरोध करेगा और उसे समझ में आता है कि एक ब्लॉक छूट गया था।

6. प्रोत्साहन

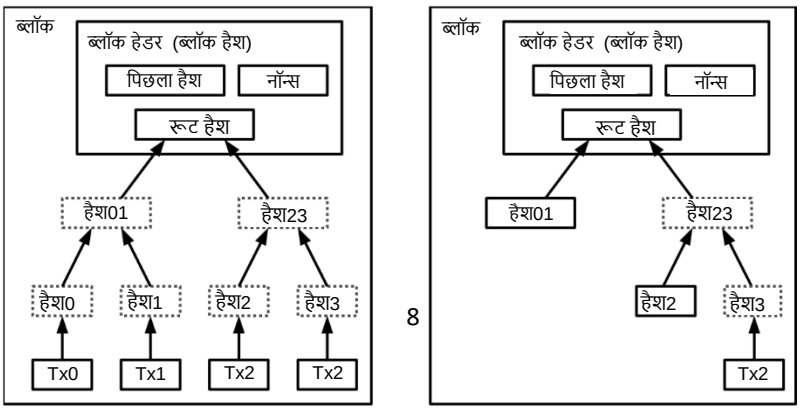
परिपाटी के अनुसार, ब्लॉक में मौजूद पहला लेनदेन एक विशेष लेनदेन होता है जो ब्लॉक निर्माता के स्वामित्व वाला नया सिक्का शुरू करता है। यह नोड्स के लिए नेटवर्क का समर्थन करने हेतु प्रोत्साहन को जोड़ता है, और चूँकि सिक्कों को जारी करने वाला कोई केन्द्रीय प्राधिकारी नहीं है, यह प्रारंभिक रूप से सिक्कों को संचलन में डालने का तरीका प्रदान करता है। अनेक नए सिक्कों को स्थिर रूप से लगातार जोड़ते जाना सोना खनिकों द्वारा सोने को संचलन में डालने के लिए संसाधन खर्च करने के बराबर है।

प्रोत्साहन के लिए धन लेनदेन शुल्क से भी उपलब्ध कराया जा सकता है। यदि किसी लेनदेन का आउटपुट मूल्य उसके इनपुट मूल्य से कम है, तो इसका अंतर लेनदेन शुल्क है जिसे उस लेनदेन को शामिल करने वाले ब्लॉक के प्रोत्साहन मूल्य में जोड़ दिया जाता है। सिक्कों की एक पूर्वनिर्धारित संख्या संचलन में डाली जाने के बाद प्रोत्साहन पूरी तरह से लेनदेन शुल्क बन सकता है और संपूर्ण रूप से मुद्रास्फीति से मुक्त हो सकता है।

प्रोत्साहन नोड्स को ईमानदार रहने के लिए प्रोत्साहित करने में मददगार बन सकता है। यदि कोई लालची हमलावर सभी ईमानदार नोड्स से अधिक CPU पावर जुटा सकता है, तो उसे उसका उपयोग अपने भुगतान चुराकर लोगों को धोखा देने, या उसका उपयोग नए सिक्के उत्पन्न करने के लिए करने के बीच चुनाव करना होगा। उसे प्रणाली को और अपने धन की मान्यता को नुकसान पहुँचाने के बजाय, नियमों पर चलना ज़्यादा फायदेमंद लगेगा, ऐसे नियम जो उसे अन्य सभी से अधिक नए सिक्के उपलब्ध करवाकर उसका समर्थन करें।

7. डिस्क स्पेस पुनः प्राप्त करना

एक बार सिक्के का नवीनतम लेनदेन काफी ब्लॉक्स में अंतर्विष्ट कर दिया जाता है, तो डिस्क स्पेस बचाने के लिए उसके पहले वाले व्यय किए गए लेनदेनों को हटाया जा सकता है। इसे ब्लॉक के हैश को तोड़े बिना सरल बनाने के लिए, ब्लॉक के हैश में केवल रूट को शामिल करके, लेनदेनों को मर्केल ट्री [7][2][5] में हैश किया जाता है। फिर ट्री की शाखाओं को काटकर पुराने ब्लॉक्स को छोटा किया जा सकता है। भीतरी हैशीस को संग्रहीत करने की आवश्यकता नहीं होती।



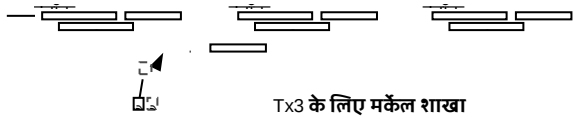
मर्केल ट्री में हैश किए गए लेनदेन Tx0-2 को ब्लॉक में से छाँटने के बाद

कोई भी लेनदेन बिना का ब्लॉक हेडर लगभग 80 बाइट का होगा। यदि हम मान लें कि ब्लॉक्स हर 10 मिनट में उत्पन्न होते हैं, तो 80 बाइट्स * 6 * 24 * 365 = प्रति वर्ष 4.2MB। 2008 में आम तौर पर 2GB RAM वाले कंप्यूटर सिस्टम्स की बिक्री, और मूर के नियम द्वारा की गई प्रति वर्ष 1.2GB की वर्तमान वृद्धि की भविष्यवाणी को देखते हुए ब्लॉक हेडर्स को मेमरी में रखा जाना आवश्यक हो तब भी भंडारण की कोई समस्या नहीं होनी चाहिए।

8. सरलीकृत भुगतान सत्यापन

एक संपूर्ण नेटवर्क नोड का संचालन किए बिना भुगतानों को सत्यापित करना संभव है। उपयोगकर्ता को केवल सबसे लंबी प्रूफ-ऑफ-वर्क श्रृंखला के ब्लॉक हेडर्स की एक प्रतिलिपि रखने की आवश्यकता होती है, जिसे वह तब तक नेटवर्क नोड्स से प्रश्न करके प्राप्त कर सकता है जब तक वह आश्वस्त न हो जाए कि उसके पास सबसे लंबी श्रृंखला है, और उस ब्लॉक से लेनदेन को लिंक करती हुई मर्केल शाखा प्राप्त करने की आवश्यकता होती है जिसमें उसे टाइमस्टैम्प किया गया है। वह खुद लेनदेन की जाँच नहीं कर सकता, परंतु उसे श्रृंखला में किसी जगह लिंक करके, वह देख सकता है कि किसी नेटवर्क नोड ने उसे स्वीकार किया है, तथा उसके बाद जोड़े गए ब्लॉक्स इस बात की और भी पुष्टि करते हैं कि नेटवर्क ने उसे स्वीकार कर लिया है।

सबसे लंबी प्रूफ-ऑफ-वर्क श्रृंखला



Tx3 के लिए मर्केल शाखा

इसीलिए, सत्यापन तब तक विश्वसनीय होता है जब तक ईमानदार नोड्स नेटवर्क को नियंत्रित करते हैं, लेकिन यदि हमलावर नेटवर्क को नियंत्रण में लेता है तो अधिक भेद्य होता है। हालाँकि नेटवर्क नोड्स अपने आप लेनदेनों को सत्यापित कर सकते हैं, लेकिन सरलीकृत तरीका तब तक हमलावर के जाली लेनदेनों के धोखे में आ सकता है जब तक हमलावर नेटवर्क को नियंत्रित करना जारी रख सकता है। अमान्य ब्लॉक पाया जाय तब नेटवर्क नोड्स से चेतावनियाँ प्राप्त करना और असंगति की पुष्टि करने के लिए उपयोगकर्ता के सॉफ़्टवेयर को पूरे ब्लॉक और चेतावनी वाले लेनदेनों को डाउनलोड करने को प्रेरित करना इससे बचने की एक रणनीति हो सकती है। बारंबार भुगतान प्राप्त करने वाले व्यवसाय शायद अधिक स्वतंत्र सुरक्षा और अधिक तेज़ सत्यापन के लिए अपने खुद के नोड्स संचालित करना चाहेंगे।

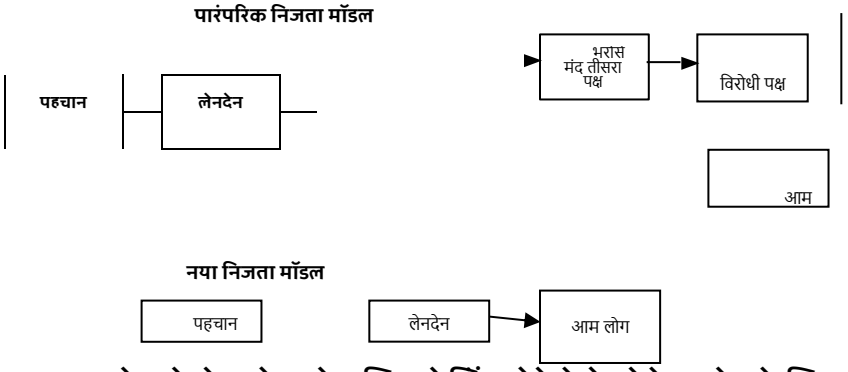
9. मूल्यों को एक करना और विभाजित करना

हालाँकि सिक्कों का प्रबंध व्यक्तिगत रूप से करना संभव होगा, फिर भी अंतरण में प्रत्येक पैसे के लिए एक अलग लेनदेन करना बोझिल हो जाएगा। मूल्यों को विभाजित और एक होने देने के लिए लेनदेनों में अनेक इनपुट और आउटपुट होते हैं। सामान्य रूप से या तो पिछले बड़े लेनदेन से एक इनपुट होगा या छोटी राशियों को मिलाते हुए अनेक इनपुट होंगे, और अधिकांश दो आउटपुट होंगे: एक भुगतान के लिए, और प्रेषक को रेज़गारी, यदि कोई हो तो, लौटाने वाला एक।

इस बात पर ध्यान देना चाहिए कि यहाँ फैन-आउट, जिसमें एक लेनदेन अनेक लेनदेनों पर निर्भर करता है, और वे लेनदेन कई और लेनदेनों पर निर्भर करते हैं, समस्या नहीं है। किसी लेनदेन के इतिहास की पूरी स्टैंडअलोन प्रतिलिपि निकालने की ज़रूरत कभी भी नहीं पड़ती।

10. निजता

पारंपरिक बैन्किंग मॉडल शामिल पक्षों और भरोसेमंद तीसरे पक्ष की जानकारी तक पहुँच को सीमित करके निजता का स्तर हासिल करता है। सभी लेनदेनों की सार्वजनिक रूप से घोषणा करने की आवश्यकता इस तरीके को असंभव बना देती है, लेकिन दूसरी जगह जानकारी के प्रवाह को तोड़कर फिर भी निजता को बनाए रखा जा सकता है: सार्वजनिक कुंजियों को गुमनाम रखकर। आम लोग देख सकते हैं कि कोई व्यक्ति किसी दूसरे व्यक्ति को धनराशि भेज रहा है, परंतु लेनदेन को किसी व्यक्ति से जोड़ने वाली जानकारी के बिना। यह स्टॉक एक्सचेंजों द्वारा जारी की जानेवाली जानकारी के स्तर के समान है, जिसमें व्यक्तिगत व्यापारों के समय और आकार, "टेप", को सार्वजनिक किया जाता है, परंतु यह बताए बिना कि पक्ष कौन-से थे।



प्रत्येक लेनदेन को साझे मालिक से लिंक होने से रोकने हेतु प्रत्येक के लिए एक अतिरिक्त फायरवॉल के रूप में कुंजी की एक नई जोड़ी उपयोग करनी चाहिए। मल्टी-इनपुट वाले उन लेनदेनों के मामले में कुछ लिन्किंग फिर भी अनिवार्य है, जो अनिवार्य रूप से प्रकट करते हैं कि उनके इनपुट का मालिक एक ही था। इसमें जोखिम यह है कि यदि कुंजी के मालिक को प्रकट किया गया, तो लिन्किंग से उसी मालिक के अन्य लेनदेनों का पता चल सकता है।

11. गणना

हम ईमानदार श्रृंखला से अधिक तेज़ गति से वैकल्पिक श्रृंखला उत्पन्न करने का प्रयास करते हुए एक हमलावर के परिदृश्य के बारे में विचार करते हैं। इसे संपादित किया जाय तो भी यह, प्रणाली को मनमाने बदलावों का शिकार नहीं बनने देता, जैसे पलक झपकते ही मूल्य सृजित करना, या ऐसा धन ऐंठना जो हमलावर का कभी नहीं था। नोड्स भुगतान के रूप में अमान्य लेनदेन को स्वीकार नहीं करेंगे, और ईमानदार नोड्स ऐसे ब्लॉक को कभी भी स्वीकार नहीं करेंगे जिसमें ये लेनदेन हैं। हमलावर उसने हाल ही में व्यय किया हुआ धन वापस लेने के लिए केवल उसके खुद के लेनदेनों में से एक को बदलने का प्रयास कर सकता है।

ईमानदार श्रृंखला और हमलावर श्रृंखला के बीच की रेस का चित्रण बायनॉमियल रैंडम वॉक के रूप में किया जा सकता है। सफल घटना है ईमानदार श्रृंखला की बढ़त को +1 से बढ़ाते हुए उसका एक ब्लॉक से विस्तार होना, और विफल घटना है -1 से अंतर घटाते हुए हमलावर की श्रृंखला का एक ब्लॉक से विस्तार होना।

हमलावर की निर्धारित घाटे से बराबरी पर आने की संभावना किसी जुआरी की बर्बादी की समस्या के

समान है। मान लें कि कोई जुआरी जिसके पास असीमित ऋण है, घाटे से शुरू करता है और संतुलन-स्तर तक पहुँचने का प्रयास करने के लिए संभावित अनगिनत ट्रायल्स खेलता है। वह कभी भी संतुलन-स्तर तक पहुँचेगा, या हमलावर कभी भी ईमानदार श्रृंखला की बराबरी पर आएगा इसकी संभावना की गणना हम निम्नलिखित तरीके से कर सकते हैं [8]:

p = ईमानदार नोड अगला ब्लॉक खोज लेगा इसकी संभावना

q = हमलावर अगला ब्लॉक खोज लेगा इसकी संभावना

q_z = हमलावर कभी भी z ब्लॉक्स पीछे से बराबरी पर आएगा इसकी संभावना

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

हमारी इस धारणा के अनुसार कि p > q है, हमलावर को जिन ब्लॉक्स की बराबरी पर आना है उनकी संख्या में वृद्धि होने के साथ संभावना घातीय रूप से कम हो जाती है। मुश्किलों का सामना करते हुए, यदि वह शुरुआत में ही किस्मत से आगे नहीं बढ़ जाता, तो उसके और पीछे छूट जाने से उसकी संभावनाएं बहुत कम होती जाती हैं।

अब हम इस पर विचार करते हैं कि नए लेनदेन के प्राप्तकर्ता को इस बात का पर्याप्त रूप से विश्वास होने से पहले कितनी देर प्रतिक्षा करनी पड़ेगी कि प्रेषक लेनदेन को बदल नहीं सकता। हम मान लेते हैं कि प्रेषक हमलावर है जो प्राप्तकर्ता को कुछ देर के लिए विश्वास दिलाना चाहता है कि उसने उसे भुगतान किया, और फिर कुछ समय बीत जाने के बाद खुद को भुगतान करने के लिए उसे बदल देता है। जब ऐसा होगा तो प्राप्तकर्ता को चेताया जाएगा, लेकिन प्रेषक उम्मीद करता है कि इसमें बहुत देर हो जाएगी।

प्राप्तकर्ता हस्ताक्षर करने से कुछ समय पहले कुंजी की एक नई जोड़ी जनरेट करता है और सार्वजनिक कुंजी प्रेषक को देता है। यह प्रेषक को, वह काफी आगे बढ जाने में सफल न हो तब तक लगातार कार्य करके समय से पहले ही ब्लॉक्स की श्रृंखला तैयार करने, और फिर उसी क्षण लेनदेन को क्रियान्वित करने से रोकता है। एक बार लेनदेन को भेजने के बाद बेईमान प्रेषक एक ऐसी समानांतर श्रृंखला पर गुप्त रूप से काम करने लगता है जिसमें उसके लेनदेन का वैकल्पिक संस्करण होता है।

प्राप्तकर्ता तब तक प्रतिक्षा करता है जब तक लेनदेन को ब्लॉक में जोड़ा नहीं जाता और उसके बाद z ब्लॉक्स लिंक नहीं किए जाते। वह नहीं जानता कि हमलावर ने ठीक कितनी प्रगति की है, लेकिन यह मानकर कि ईमानदार ब्लॉक्स ने प्रति ब्लॉक औसत अपेक्षित समय लिया, हमलावर की संभावित प्रगति पॉसों वितरण होगी जिसका अपेक्षित मूल्य होगा:

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

p

हमलावर अब भी बराबरी पर आ सकता है इसकी संभावना प्राप्त करने के लिए हम उस बिंदु से वह बराबरी पर आ सकता इसकी संभावना का, उस प्रगति की प्रत्येक मात्रा के पॉसों घनत्व से गुणा करते हैं जो वह कर सकता था:

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

$$q / p \quad ^{z-k} \quad if\ k \le z$$

अपरिमित वितरण तल का जोड़ लगाने से बचने के लिए पुनः व्यवस्था करते हुए…

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

C कोड में रूपान्तरित करते हुए…

```
#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;
}
```

कुछ परिणामों को चलाकर, हम देख सकते हैं कि z के साथ संभावना घातीय रूप से कम होती जाती है।

```
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
```

0.1% से कम P के लिए हल निकालते हुए…

```
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. निष्कर्ष

हमने विश्वास पर निर्भर हुए बिना इलेक्ट्रॉनिक लेनदेनों के लिए एक प्रणाली पेश की है। हमने डिजिटल हस्ताक्षरों से बने सिक्कों के साधारण ढाँचे से शुरुआत की, जो स्वामित्व पर मजबूत नियंत्रण प्रदान करता है, परंतु जो दोहरे-व्यय को रोकने के तरीके के बिना अपूर्ण है। इसका हल निकाने के लिए, हमने लेनदेनों के सार्वजनिक इतिहास को दर्ज करने के लिए प्रूफ-ऑफ-वर्क का उपयोग करके पियर-टू-पियर नेटवर्क पेश किया, जिसे बदलना हमलावर के लिए संगणन की दृष्टि से शीघ्र ही अव्यावहारिक बन जाता है, यदि ईमानदार नोड्स अधिकांश CPU पावर को नियंत्रित करें। नेटवर्क अपनी संरचित सरलता में मजबूत है। थोड़े समन्वय के साथ सभी नोड्स एक ही समय पर काम करते हैं। इनकी पहचान करनी नहीं पड़ती, क्योंकि संदेश किसी खास जगह नहीं भेजे जाते और उन्हें केवल श्रेष्ठ प्रयास के आधार पर पहुँचाना होता है। नोड्स अपनी अनुपस्थिति में जो हुआ उसके प्रमाण के रूप में प्रूफ-ऑफ-वर्क की श्रृंखला को स्वीकार करके, इच्छानुसार नेटवर्क को छोड़ सकते हैं और उसमें फिर से जुड़ सकते हैं। वे मान्य ब्लॉक्स का विस्तार करने पर कार्य करके उनका स्वीकार व्यक्त कर और अमान्य ब्लॉक्स पर कार्य करने से इनकार कर उनका अस्वीकार करके अपने CPU पावर से वोट करते हैं। सहमति की इस क्रियाविधि से किसी भी आवश्यक

नियम और प्रोत्साहन को प्रवर्तित किया जा सकता है।

संदर्भ

[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.