بِسۡمِ ٱللَّهِ ٱلرَّحۡمَٰنِ ٱلرَّحِيمِ

بٹ کوائن: ایک ہمتا بہ ہمتا الیکٹرانک کیش کا نظام

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

Translated in Urdu by Muhammad Safdar Jamal

خلاصہ: الیکٹرانک کیش کا خالصتاً ہم مرتبہ ورژن جو کسی مالیاتی ادارے سے گزرے بغیر آن لائن ادائیگیوں کو براہ راست ایک فریق سے دوسرے فریق کو بھیجنے کی اجازت دے گا۔ ڈیجیٹل دستخط حل کا ایک حصہ فراہم کرتے ہیں، لیکن اگر اب بھی دوہرے اخراجات کو روکنے کے لیے کسی قابل اعتماد تیسرے فریق کی ضرورت محسوس ہو تو اہم فوائد ضائع ہو جاتے ہیں۔ ہم پیئر ٹو پیئر نیٹ ورک کا استعمال کرتے ہوئے دوہرے خرچ کے مسئلے کا حل تجویز کرتے ہیں۔ نیٹ ورک ٹائم اسٹامپ لین دین کو ہیش پر مبنی پروف آف ورک کے جاری سلسلے میں تبدیل کر کے ایک ریکارڈ تشکیل دیتا ہے جسے پروف آف ورک کو دوبارہ کیے بغیر تبدیل نہیں کیا جا سکتا۔ سب سے لمبی زنجیر نہ صرف مشاہدہ کردہ واقعات کے تسلسل کے ثبوت کے طور پر کام کرتی ہے بلکہ اس بات کا ثبوت ہے کہ یہ سی پی یو توانائی کے سب سے بڑے پول سے آئی ہے۔ جب تک سی پی یو توانائی کی اکثریت نوڈز کے زیر قابو ہے جو نیٹ ورک پر حملہ کرنے کے لئے تعاون نہیں کر رہے ہیں، وہ طویل ترین زنجیر پیدا کریں گے اور حملہ آوروں سے آگے نکل جائیں گے۔ نیٹ ورک کو خود کم سے کم ڈھانچے کی ضرورت ہوتی ہے۔ پیغامات بہترین کوشش کی بنیاد پر نشر کیے جاتے ہیں، اور نوڈز اپنی مرضی سے نیٹ ورک چھوڑ سکتے ہیں اور دوبارہ شامل ہو سکتے ہیں، کام کے طویل ترین ثبوت کی زنجیر کو اس بات کے ثبوت کے طور پر قبول کرتے ہیں کہ جب وہ چلے گئے تھے تو کیا ہوا تھا۔

تعارف

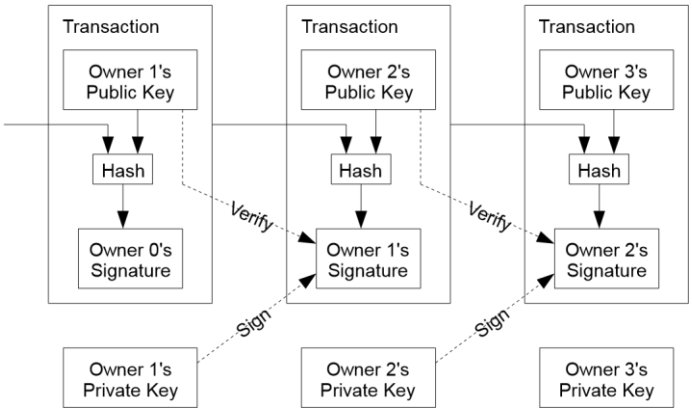
انٹرنیٹ پر تجارت مالیاتی اداروں پر انحصار کرتی ہے جو الیکٹرانک ادائیگیوں کے لیے قابل اعتماد تیسرے فریق کے طور پر کام کرتے ہیں۔ اگرچہ یہ نظام زیادہ تر لین دین کے لیے کافی اچھا کام کرتا ہے، لیکن یہ اب بھی اعتماد پر مبنی ماڈل کی فطری کمزوریوں کا شکار ہے۔ مکمل طور پر غیر معکوس لین دین واقعی ممکن نہیں ہے کیونکہ مالیاتی ادارے تنازعات میں ثالثی سے گریز نہیں کر سکتے۔ ثالثی کی لاگت لین دین کے اخراجات میں اضافہ کرتی ہے، کم سے کم عملی لین دین کے حجم کو محدود کرتی ہے اور چھوٹے معمولی لین دین کے امکانات کو ختم کرتی ہے، اور غیر معکوس خدمات کے لئے غیر معکوس ادائیگیاں کرنے کی صلاحیت کے نقصان میں وسیع تر لاگت ہے۔ الٹ جانے کے امکان کے ساتھ، اعتماد کی ضرورت بھی پھیل جاتی ہے۔ تاجروں کو اپنے گاہکوں سے محتاط رہنا پڑتا ہے، انہیں زیادہ سے زیادہ معلومات کے لیے پریشان کرنا پڑے گا جس کی انہیں دوسری صورت میں ضرورت

ہو گی۔ دھوکہ دہی کا ایک خاص فیصد ناگزیر کے طور پر قبول کیا جاتا ہے۔ کاغذی کرنسی کا استعمال کر کے ذاتی طور پر ان اخراجات اور ادائیگی کی غیر یقینی صورتحال سے بچا جا سکتا ہے، لیکن کسی قابل اعتماد فریق کے بغیر مواصلاتی چینل پر ادائیگی کرنے کا کوئی طریقہ کار موجود نہیں ہے۔

ضرورت اس بات کی ہے کہ اعتماد کی بجائے خفیہ نگاری کے ثبوت پر مبنی الیکٹرانک ادائیگی کا نظام ہو، جس سے کوئی دو رضامند فریق کسی قابل اعتماد تیسرے فریق کی ضرورت کے بغیر براہ راست ایک دوسرے کے ساتھ لین دین کر سکیں۔ ایسے لین دین جو ریورس کرنے کے لئے کمپیوٹیشنل طور پر ناقابل عمل ہیں، فروخت کنندگان کو دھوکہ دہی سے محفوظ رکھیں گے اور خریداروں کے تحفظ کے لئے معمول کے ایسکرو میکانزم کو آسانی سے نافذ کیا جا سکتا ہے۔ اس مقالے میں، ہم لین دین کی ترتیب کا شمارندی ثبوت پیدا کرنے کے لیے ہم مرتبہ سے ہم مرتبہ تقسیم شدہ ٹائم اسٹیمپ سرور کا استعمال کرتے ہوئے دوہرے خرچ کے مسئلے کا حل تجویز کرتے ہیں۔ یہ نظام اس وقت تک محفوظ ہے جب تک ایماندار نوڈز مجموعی طور پر حملہ آور نوڈز کے کسی بھی تعاون کرنے والے گروپ سے زیادہ سی پی یو طاقت کو کنٹرول کرتے ہیں۔

لین دین

ہم ایک الیکٹرانک سکے کو ڈیجیٹل دستخطوں کی زنجیر کے طور پر بیان کرتے ہیں۔ ہر مالک پچھلے لین دین کے ایک ہیش اور اگلے مالک کی عوامی کلید پر ڈیجیٹل طور پر دستخط کر کے اور ان کو سکے کے آخر میں شامل کر کے سکے کو اگلے کو منتقل کرتا ہے۔ ایک وصول کنندہ ملکیت کے سلسلے کی تصدیق کے لیے دستخطوں کی تصدیق کر سکتا ہے۔

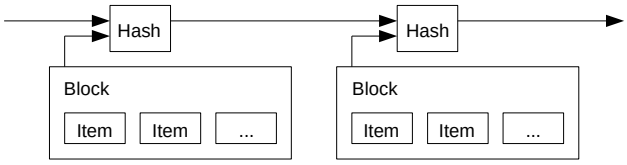


یقیناً مسئلہ یہ ہے کہ وصول کنندہ اس بات کی تصدیق نہیں کر سکتا کہ مالکان میں سے کسی نے سکے کو دوگنا خرچ تو نہیں کیا۔ ایک مشترکہ حل یہ ہے کہ ایک قابل اعتماد مرکزی اتھارٹی، یا ٹکسال متعارف کرایا جائے، جو دوہرے اخراجات کے لئے ہر لین دین کی جانچ پڑتال کرے۔ ہر لین دین کے بعد، ایک نیا سکہ جاری کرنے کے لئے سکہ ٹکسال کو واپس کرنا ضروری ہے، اور صرف ٹکسال سے براہ راست جاری کردہ سکوں پر بھروسہ کیا جاتا ہے تاکہ ان کا دوگنا خرچ نہ کیا جائے۔ اس حل کے ساتھ مسئلہ یہ ہے کہ پورے کرنسی کے نظام کی قسمت کا انحصار اس کمپنی پر ہے جو ٹکسال چلا رہی ہے، ہر لین دین کو بینک کی طرح ان سے گزرنا پڑتا ہے۔

ہمیں وصول کنندہ کے لیے یہ جاننے کے لیے ایک طریقہ کار درکار ہے کہ پچھلے مالکان نے پہلے کسی لین دین پر تو دستخط نہیں کیے۔ ہمارے مقاصد کے لئے، سب سے پہلا لین دین وہ ہے جو شمار ہوتا ہے، لہذا ہمیں بعد میں دوگنا خرچ کرنے کی کوششوں کی پرواہ نہیں ہے۔ لین دین کی عدم موجودگی کی تصدیق کرنے کا واحد طریقہ یہ ہے کہ تمام لین دین سے آگاہ رہیں۔ ٹکسال پر مبنی نظام میں، ٹکسال تمام لین دین سے آگاہ ہوتا تھا اور فیصلہ کرتا تھا کہ کون پہلے آیا۔ کسی قابل اعتماد فریق کے بغیر اسے پورا کرنے کے لئے لین دین کا اعلان عوامی طور پر کیا جانا چاہئے [1]، اور ہمیں شرکاء کے لئے ایک نظام کی ضرورت ہے تاکہ وہ اس ترتیب کی واحد تاریخ پر متفق ہوں جس میں انہیں وصول کیا گیا تھا۔ وصول کنندہ کو اس بات کا ثبوت درکار ہوتا ہے کہ ہر لین دین کے وقت نوڈز کی اکثریت نے اس بات پر اتفاق کیا کہ یہ پہلے وصول کیا گیا تھا۔

ٹائم اسٹامپ سرور

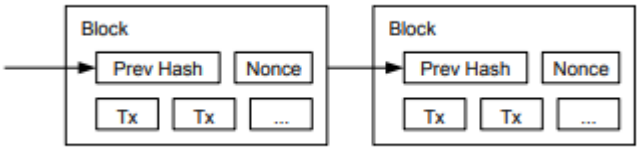
ہم جو حل تجویز کرتے ہیں وہ ٹائم اسٹیمپ سرور سے شروع ہوتا ہے۔ ٹائم اسٹامپ سرور اشیاء کے ایک بلاک کا ہیش لے کر کام کرتا ہے جس پر وقت کی مہر لگائی جاتی ہے اور وسیع پیمانے پر ہیش شائع کیا جاتا ہے، جیسے کہ اخبار یا یوزنیٹ پوسٹ میں [2-5] ٹائم اسٹیمپ یہ ثابت کرتا ہے کہ ہیش میں آنے کے لئے اس وقت اعداد و شمار موجود ہونا ضروری ہے۔ ہر ٹائم اسٹیمپ اپنے ہیش میں پچھلا ٹائم اسٹیمپ شامل کرتا ہے، ایک سلسلہ بناتا ہے اور ہر اضافی ٹائم اسٹیمپ کے ساتھ اس سے پہلے والے کو تقویت ملتی ہے۔



پروف۔آف۔ورک

ایک تقسیم شدہ ٹائم اسٹیمپ سرور کو ہمتا بہ ہمتا کی بنیاد پر نافذ کرنے کے لیے ہمیں اخبار یا یوزنیٹ پوسٹوں کی بجائے آدم بیک کے ہیش کیش (Adam Back's HashCash)[6] کی طرح ایک پروف آف ورک سسٹم استعمال کرنے کی ضرورت ہوگی۔ پروف آف ورک میں ایسی قدر کی اسکیننگ شامل ہوتی ہے جسے ہیش کرنے پر، جیسے SHA-256 کے ساتھ، ہیش کا آغاز متعدد صفر بٹس سے ہوتا ہے۔ مطلوبہ اوسط کام درکار صفر بٹس کی تعداد میں غیر معمولی ہے اور ایک ہی ہیش کو عمل میں لا کر اس کی تصدیق کی جاسکتی ہے۔

ہمارے ٹائم اسٹیمپ نیٹ ورک کے لیے، ہم بلاک میں ایک بار استعمال ہونے والے نمبر کو بڑھا کر پروف آف ورک کو لاگو کرتے ہیں جب تک کہ کوئی قدر نہ مل جائے جو بلاک کے ہیش کو مطلوبہ صفر بٹس فراہم کرتی ہے۔ ایک بار جب سی پی یو کی کوشش کو کام کے ثبوت کو پورا کرنے کے لئے خرچ کیا جائے، تو کام کو دوبارہ کیے بغیر بلاک کو تبدیل نہیں کیا جاسکتا۔ چونکہ بعد کے بلاکس اس کے بعد جکڑے ہوئے ہیں، اس لیے بلاک کو تبدیل کرنے کے کام میں اس کے بعد کے تمام بلاکس کو دوبارہ کرنا ہوگا۔



کام کا ثبوت اکثریتی فیصلہ سازی میں نمائندگی کے تعین کا مسئلہ بھی حل کرتا ہے۔ اگر اکثریت ایک آئی پی پتہ ایک ووٹ پر مبنی ہوتی تو بہت سے آئی پیز مختص کرنے کے قابل کوئی بھی اس کو ختم کر سکتا تھا۔ کام کا ثبوت بنیادی طور پر ایک سی پی یو۔ ایک ووٹ ہے۔ اکثریت کے فیصلے کی نمائندگی سب سے طویل سلسلے سے ہوتی ہے، جس نے کام کرنے کی سب سے بڑی کوشش کی ہوتی ہے۔ اگر سی پی یو طاقت کی اکثریت ایماندار نوڈز کے زیر کنٹرول ہے تو ایماندار زنجیر تیز ترین بڑھے گی اور کسی بھی مسابقتی زنجیر سے آگے نکل جائے گی۔ ماضی کے بلاک میں ترمیم کرنے کے لئے، ایک حملہ آور کو بلاک اور اس کے بعد کے تمام بلاکس کے پروف آف ورک کو دوبارہ کرنا ہوگا اور پھر ایماندار نوڈز کے کام کو پکڑنا اور اس سے آگے نکلنا ہوگا۔ ہم بعد میں ظاہر کریں گے کہ بعد میں بلاکس شامل ہونے کے ساتھ ہی سست حملہ آور کے پکڑنے کا امکان تیزی سے کم ہوتا جاتا ہے۔

ہارڈ ویئر کی رفتار میں اضافے اور وقت کے ساتھ نوڈز چلانے میں بدلتی ہوئی دلچسپی کی تلافی کے لئے، کام کے ثبوت کی مشکل کا تعین ایک متحرک اوسط سے کیا جاتا ہے جو فی گھنٹہ بلاکس کی اوسط تعداد کو نشانہ بناتا ہے۔ اگر وہ بہت تیزی سے پیدا ہوتے ہیں تو مشکل اور بڑھ جاتی ہے۔

نیٹ ورک

نیٹ ورک کو چلانے کے اقدامات درج ذیل ہیں:

1) نئے لین دین کو تمام نوڈز پر نشر کیا جاتا ہے۔
2) ہر نوڈ ایک بلاک میں نئے لین دین جمع کرتا ہے۔
3) ہر نوڈ اپنے بلاک کے لئے کام کا مشکل ثبوت تلاش کرنے پر کام کرتا ہے۔
4) جب نوڈ کو کام کا ثبوت ملتا ہے تو وہ بلاک کو تمام نوڈز پر نشر کرتا ہے۔
5) نوڈز بلاک کو صرف اسی صورت میں قبول کرتے ہیں جب اس میں موجود تمام لین دین جائز ہوں اور پہلے سے خرچ شدہ نہ ہوں۔
6) نوڈز اس سلسلے میں اگلا بلاک بنانے پر کام کرتے ہوئے بلاک کی قبولیت کا اظہار کرتے ہیں، قبول شدہ بلاک کے ہیش کو پچھلی ہیش کے طور پر استعمال کرتے ہیں۔

نوڈز ہمیشہ طویل ترین زنجیر کو صحیح سمجھتے ہیں اور اسے بڑھانے پر کام کرتے رہیں گے۔ اگر دو نوڈز بیک وقت اگلے بلاک کے مختلف ورژن نشر کرتے ہیں، تو کچھ نوڈز ایک یا دوسرے کو پہلے وصول کر سکتے ہیں۔ اس ضمن میں، وہ حاصل کردہ پہلی شاخ پر کام کرتے ہیں، لیکن دوسری شاخ کو اس صورت میں بچاتے ہیں، اگر یہ طویل ہو جاتی ہے۔ جب اگلا پروف آف ورک مل جائے گا اور ایک شاخ لمبی ہو جائے گی تو جوڑ ٹوٹ جائے گا۔ دوسری شاخ پر کام کرنے والے نوڈز پھر طویل شاخ میں تبدیل ہو جائیں گے۔

نئے لین دین کی نشریات کو تمام نوڈز تک پہنچنے کی ضرورت نہیں ہے۔ جب تک وہ بہت سے نوڈز تک پہنچیں گے، وہ بہت پہلے ایک بلاک میں داخل ہو جائیں گے۔ بلاک نشریات بھی پیغامات چھوڑنے کے روادار ہیں۔ اگر نوڈ کو بلاک موصول نہیں ہوتا ہے، تو وہ اس کی درخواست کرے گا جب اسے اگلا بلاک موصول ہو گا تو اسے احساس ہو گا کہ اس نے ایک بلاک کھو دیا ہے۔

ترغیب

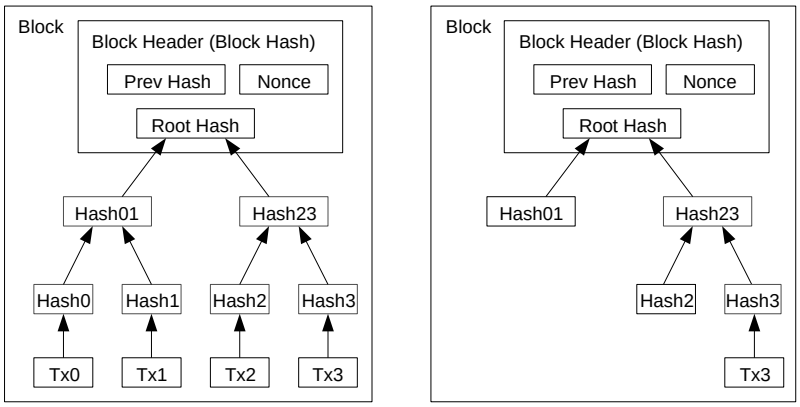
روایت کے مطابق، بلاک میں پہلا لین دین ایک خاص لین دین ہے جو بلاک کے خالق کی ملکیت میں ایک نیا سکہ شروع کرتا ہے۔ اس سے نیٹ ورک کی معاونت کے لئے نوڈز کے لئے ایک ترغیب میں اضافہ ہوتا ہے، اور ابتدائی طور پر سکوں کو گردش میں تقسیم کرنے کا ایک طریقہ فراہم کرتا ہے، کیونکہ ان کو جاری کرنے کا انہیں کوئی مرکزی اختیار نہیں ہے۔ نئے سکوں کی مقدار میں مسلسل اضافہ، سونے کے کان کنوں کا گردش میں سونے کو شامل کرنے کے لیے وسائل خرچ کرنے کے مترادف ہے۔ ہمارے معاملے میں، یہ سی پی یو کا وقت اور بجلی ہے جو خرچ کی جاتی ہے۔

اس ترغیب کو لین دین کی فیس کے ساتھ بھی مالی امداد فراہم کی جاسکتی ہے۔ اگر کسی ٹرانزیکشن کی آؤٹ پٹ ویلیو اس کی ان پٹ ویلیو سے کم ہے، تو فرق ایک ٹرانزیکشن فیس ہے جو ٹرانزیکشن پر مشتمل بلاک کی ترغیبی قیمت میں شامل کی جاتی ہے۔ ایک بار جب سکوں کی پہلے سے طے شدہ تعداد گردش میں داخل ہو جائے تو یہ ترغیب مکمل طور پر ٹرانزیکشن فیس میں منتقل ہو سکتی ہے اور مکمل طور پر افراط زر سے پاک ہو سکتی ہے۔

اس ترغیب سے نوڈز کو ایماندار رہنے کی حوصلہ افزائی کرنے میں مدد مل سکتی ہے۔ اگر کوئی لالچی حملہ آور تمام ایماندار نوڈز سے زیادہ سی پی یو پاور جمع کرنے کے قابل ہے تو اسے اس کا انتخاب کرنا ہو گا کہ وہ اپنی ادائیگیاں واپس چوری کر کے لوگوں کو دھوکہ دینے کے لئے استعمال کرے، یا اسے نئے سکے پیدا کرنے کے لئے استعمال کرے۔ اسطرح اسے اصولوں کے مطابق کھیلنا زیادہ منافع بخش معلوم ہو گا، ایسے اصول جو اس کے لیے سب کے ساتھ مل کر زیادہ نئے سکوں کے ساتھ اس کے حق میں ہوں، بجائے اس کے کہ نظام اور اس کی اپنی دولت کے جواز کو کمزور کیا جائے۔

ڈسک اسپیس کو دوبارہ حاصل کرنا

ایک بار جب کسی سکے میں تازہ ترین لین دین کا فی بلاکس کے نیچے دب جاتا ہے، تو خرچ شدہ لین دین کو ڈسک کی جگہ بچانے کے لیے ختم کرنے سے پہلے ہی ضائع کیا جاسکتا ہے۔ اس کی سہولت کے لیے بلاک کی ہیش کو توڑے بغیر، لین دین کو مرکل ٹری[7][2][5] میں ہیش کیا جاتا ہے، جس میں بلاک کی ہیش میں صرف جڑ شامل ہوتی ہے۔ اس کے بعد پرانے بلاکس کو مرکل ٹری کی شاخوں کو بند کر کے منسلک کیا جاسکتا ہے۔ اندرونی ہیش کو ذخیرہ کرنے کی ضرورت نہیں ہے۔



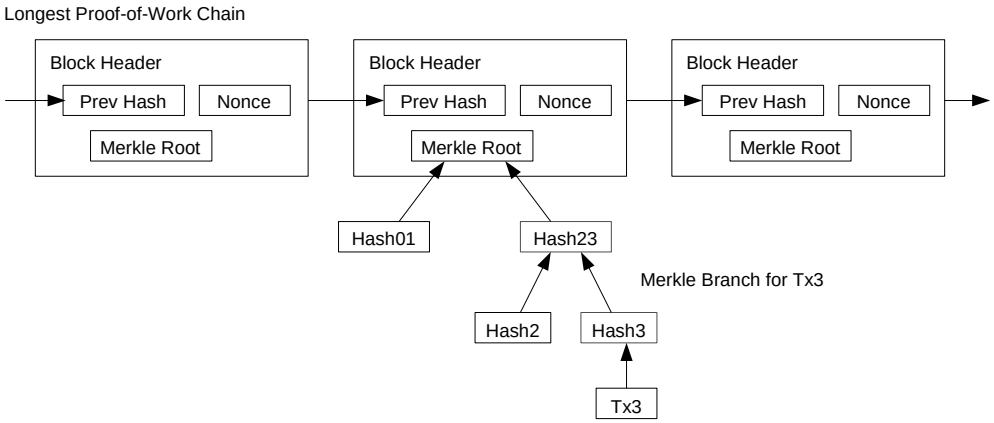
Transactions Hashed in a Merkle Tree

After Pruning Tx0-2 from the Block

بغیر کسی لین دین کے بلاک ہیڈر تقریبا ۸۰ بائٹس کا ہوگا۔ اگر ہم فرض کریں کہ بلاکس ہر 10 منٹ میں، 80 بائٹس *6*24*365=4.2 ایم بی سالانہ پیدا ہوتے ہیں۔ کمپیوٹر سسٹم عام طور پر 2008 تک 2 جی بی ریم کے ساتھ فروخت ہوتا آیا ہے، اور ''مور کا قانون'' 1.2 جی بی سالانہ کی موجودہ نمو کی پیش گوئی کرتا ہے، اس کے مطابق سٹوریج کو کوئی مسئلہ نہیں ہونا چاہئے چاہے بلاک ہیڈرز کو میموری میں رکھنا ضروری ہو۔

آسان ادائیگی کی تصدیق

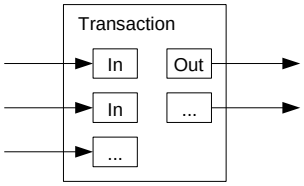
مکمل نیٹ ورک نوڈ چلائے بغیر ادائیگیوں کی تصدیق کرنا ممکن ہے۔ ایک صارف کو صرف اور صرف سب سے طویل پروف آف ورک کی زنجیر کے بلاک ہیڈرز کی ایک کاپی رکھنے کی ضرورت ہے، جسے وہ نیٹ ورک نوڈز سے استفسار کر کے حاصل کر سکتا ہے جب تک کہ اسے یقین نہ ہو جائے کہ اس کے پاس سب سے طویل سلسلہ ہے، اور لین دین کو بلاک سے منسلک کرنے والی مرکل برانچ بھی حاصل کر سکتا ہے۔ جس میں اس وقت پر مہر لگی ہوئی ہے۔ وہ اپنے لیے لین دین کی جانچ نہیں کر سکتا، لیکن اسے زنجیر میں کسی جگہ سے جوڑ کر، وہ دیکھ سکتا ہے کہ نیٹ ورک نوڈ نے اسے قبول کر لیا ہے، اور اس کے بعد مزید بلاکس کا جڑنا بھی اس بات کی تصدیق ہے کہ نیٹ ورک نے اسے قبول کر لیا ہے۔ اس طرح، تصدیق اس وقت تک قابل اعتماد ہے جب تک ایماندار نوڈز نیٹ ورک کو کنٹرول کرتے ہیں، لیکن اگر نیٹ ورک کسی حملہ آور کے زیرِ اثر آ جائے تو یہ زیادہ غیر محفوظ ہے۔ اگرچہ نیٹ ورک نوڈز اپنے لئے لین دین کی تصدیق کر سکتے ہیں، لیکن آسان طریقہ کار کو حملہ آور کے من گھڑت لین دین کے ذریعے اس وقت تک بے وقوف بنایا جا سکتا ہے جب تک حملہ آور نیٹ ورک پر قابو پانا جاری رکھ سکتا ہے۔



اس سے بچاؤ کی ایک حکمت عملی یہ ہو گی کہ جب وہ ناجائز بلاک کا پتہ لگائیں تو نیٹ ورک نوڈز سے الرٹ قبول کریں، جس سے صارف کے سافٹ ویئر کو مکمل بلاک ڈاؤن لوڈ کرنے اور عدم مطابقت کی تصدیق کرنے کے لئے لین دین کو خبر دار کرنے کی ترغیب ملے گی۔ وہ کاروبار جو بار بار ادائیگیاں وصول کرتے ہیں شاید اب بھی زیادہ آزاد سیکیورٹی اور تیز تر تصدیق کے لیے اپنے نوڈز چلانا چاہیں گے۔

قدر کو یکجا کرنا اور تقسیم کرنا

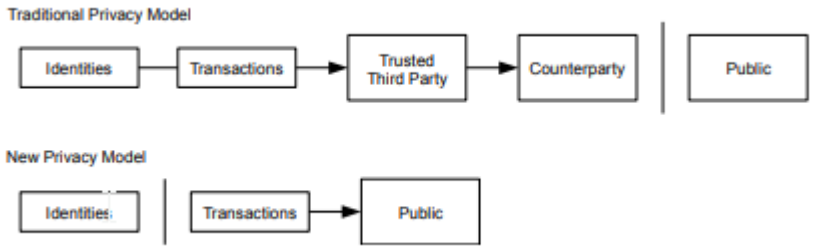
اگرچہ سکوں کو انفرادی طور پر سنبھالنا ممکن ہو گا لیکن منتقلی میں ہر ایک فیصد کے لئے ایک علیحدہ لین دین کرنا مشکل ہو گا۔ قدر کو تقسیم اور یکجا کرنے کی اجازت دینے کے لیے، لین دین میں متعدد ان پٹ اور آؤٹ پٹ ہوتے ہیں۔ عام طور پر یہ یا تو کسی بڑے سابقہ لین دین سے ایک ان پٹ ہو گا یا چھوٹی رقم کو ملا کر متعدد ان پٹ ہوں گے، اور زیادہ سے زیادہ دو آؤٹ پٹ: ایک ادائیگی کے لئے، اور ایک تبدیلی کے لئے، اگر کوئی ہو گا تو بھیجنے والے کو واپس ہو گا۔



واضح رہے کہ فین آؤٹ، جہاں ایک لین دین کا انحصار کئی لین دین پر ہوتا ہے، اور وہ لین دین بہت سی چیزوں پر منحصر ہوتے ہیں، یہاں کوئی مسئلہ نہیں ہے۔ کسی لین دین کی تاریخ کی مکمل اکیلی کاپی نکالنے کی کبھی ضرورت نہیں ہوتی۔

رازداری

روایتی بینکنگ ماڈل اس میں شامل فریقین اور قابل اعتماد تیسرے فریق تک معلومات کی رسائی کو محدود کرنے کی سطح تک رازداری حاصل کرتا ہے۔ تمام لین دین کا عوامی طور پر اعلان کرنے کی ضرورت اس طریقہ کار کو روکتی ہے، لیکن دوسری جگہ معلومات کے بہاؤ کو توڑ کر رازداری کو عوامی چابیاں گمنام رکھ کر برقرار رکھا جاسکتا ہے۔ عوام دیکھ سکتے ہیں کہ کوئی شخص کسی اور کو رقم بھیج رہا ہے، لیکن بغیر کسی معلومات کے لین دین کو کسی سے جوڑ رہا ہے۔ یہ سٹاک ایکسچینجز کی طرف سے جاری کردہ معلومات کی سطح کی طرح ہے، جہاں انفرادی تجارت کا وقت اور حجم یعنی "ٹیپ" کو عام کیا جاتا ہے، لیکن یہ بتائے بغیر کہ فریقین کون تھے۔



ایک اضافی فائروال کے طور پر، ہر لین دین کے لیے ایک نیا کلیدی جوڑا استعمال کیا جانا چاہیے تاکہ انہیں ایک عام مالک سے منسلک ہونے سے بچایا جا سکے۔ ایک سے زیادہ ان پٹ کے لین دین کے ساتھ کچھ جوڑنا اب بھی ناگزیر ہے، جو لازمی طور پر یہ ظاہر کرتی ہے کہ ان کے ان پٹ ایک ہی مالک کی ملکیت تھے۔ اس میں خطرہ یہ ہے کہ اگر کسی کلید کا مالک ظاہر ہوتا ہے، تو اسے جوڑنے سے دوسرے لین دین کا پتہ چلتا ہے جو اسی مالک سے تعلق رکھتے تھے۔

حساب

ہم ایک حملہ آور کے منظر نامے پر غور کرتے ہیں جو ایماندار زنجیر سے زیادہ تیز متبادل سلسلہ پیدا کرنے کی کوشش کر رہا ہے۔ اگر یہ کام مکمل بھی ہو جاتا ہے تو یہ نظام کو من مانی تبدیلیوں کے لیے کھلا نہیں چھوڑتا، جیسے بغیر اطلاع کے قدر پیدا کرنا یا وہ رقم لینا جو حملہ آور کی کبھی تھی ہی نہیں۔ نوڈز کسی بھی ناجائز لین دین کو بطور ادائیگی قبول نہیں کریں گے، اور دیانتدار نوڈز ان پر مشتمل بلاک کو کبھی قبول نہیں کریں گے۔ حملہ آور صرف اپنے ایک لین دین کو تبدیل کرنے کی کوشش کر سکتا ہے تاکہ وہ حال ہی میں خرچ کی گئی رقم واپس لے سکے۔

ایماندار زنجیر اور حملہ آور زنجیر کے درمیان دوڑ کو دو رقمی تصادفی چال کے طور پر نمایاں کیا جاسکتا ہے۔ کامیابی کا واقعہ ایماندارانہ سلسلہ ہے جس کو ایک بلاک تک بڑھایا جاتا ہے، اس کی برتری کو $+1$ سے بڑھاتا ہے، اور ناکامی کا واقعہ حملہ آور کی زنجیر کو ایک بلاک تک بڑھاتے ہوئے، فرق کو -1 تک کم کرتا ہے۔

دیے گئے خسارے سے حملہ آور کے پکڑے جانے کا امکان سٹے باز کے دیوالیہ ہونے کے مسئلے کے مترادف ہے۔ فرض کریں کہ لامحدود کریڈٹ کے ساتھ ایک جواری خسارے سے شروع ہوتا ہے اور ممکنہ طور پر لامحدود باریاں کھیلتا ہے تاکہ نقصان برابر کرنے کی سطح تک پہنچنے کی کوشش کرے۔ ہم اس امکان کا حساب لگا سکتے ہیں کہ وہ کبھی نفع و نقصان نہ ہونے کی سطح تک پہنچ جاتا ہے، یا یہ کہ حملہ آور کبھی بھی ایماندارانہ سلسلہ کو پکڑتا ہے، جیسا کہ مندرجہ ذیل ہے[8]:

$p =$ امکان ہے کہ ایک ایماندار نوڈ اگلا بلاک تلاش کرتا ہے ۔

$q =$ امکان ہے کہ حملہ آور کو اگلا بلاک مل جاتا ہے۔

$q_z =$ امکان ہے کہ حملہ آور کبھی z بلاکس کو پیچھے سے پکڑ لے گا۔

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

ہمارے مفروضے کو دیکھتے ہوئے کہ $p > q$ ، یہ امکان تیزی سے کم ہوتا ہے کیونکہ حملہ آور کو بلاکس کی تعداد میں اضافے کے ساتھ پکڑنا پڑتا ہے۔ اس کے خلاف مشکلات کے ساتھ، اگر وہ جلد ہی خوش قسمتی سے آگے نہیں بڑھتا، تو اس کے مزید پیچھے پڑتے ہی اس کے امکانات معدوم ہو جاتے ہیں۔ اب ہم اس بات پر غور کرتے ہیں کہ نئے لین دین کے وصول کنندہ کو اس بات کا یقین ہونے سے پہلے کہ بھیجنے والا لین دین کو تبدیل نہیں کر سکتا کتنا انتظار کرنے کی ضرورت ہے۔ ہم فرض کرتے ہیں کہ بھیجنے والا ایک حملہ آور ہے جو وصول کنندہ کو یقین دلانا چاہتا ہے کہ اس نے اسے تھوڑی دیر کے لیے ادائیگی کی، پھر کچھ وقت گزرنے کے بعد اسے خود کو واپس کرنے کے لیے اسے تبدیل کر دے۔ ایسا ہونے پر وصول کنندہ کو خبردار کر دیا جائے گا، لیکن بھیجنے والے کو امید ہے کہ بہت دیر ہو چکی ہو گی۔

وصول کنندہ ایک نیا کلیدی جوڑا تیار کرتا ہے اور دستخط کرنے سے کچھ دیر پہلے بھیجنے والے کو عوامی کلید دیتا ہے۔ یہ مرسل کو مسلسل اس پر کام کر کے وقت سے پہلے بلاکس کی زنجیر تیار کرنے سے روکتا ہے یہاں تک کہ وہ کافی خوش قسمت ہو کے کافی آگے نکل جائے، تاکہ اس لمحے لین دین کو انجام دے سکے۔ ایک بار لین دین بھیجنے کے بعد، بے ایمان مرسل اپنے لین دین کے متبادل ورژن پر مشتمل متوازی زنجیر پر خفیہ طور پر کام کرنا شروع کر دیتا ہے۔

وصول کنندہ اس وقت تک انتظار کرتا ہے جب تک کہ ٹرانزیکشن کو بلاک میں شامل نہیں کیا جاتا اور اس کے بعد Z بلاکس کو جوڑا نہیں جاتا۔ وہ نہیں جانتا کہ حملہ آور نے کتنی پیشرفت کی ہے، لیکن یہ فرض کرتے ہوئے کہ ایماندار بلاکس نے فی بلاک اوسط متوقع وقت لیا، حملہ آور کی ممکنہ پیشرفت متوقع قدر کے ساتھ پوئسین توزیع ہو گی:

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

اس امکان کو حاصل کرنے کے لئے کہ حملہ آور اب بھی پکڑ سکتا ہے، ہم پوئسین کثافت کو ہر اس پیش رفت کی مقدار سے ضرب دیتے ہیں جو وہ اس امکان سے کر سکتا تھا کہ وہ اس مقام سے پکڑ سکتا ہے:

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

تقسیم کی لامحدود دم کا خلاصہ کرنے سے بچنے کے لئے دوبارہ ترتیب دینا؛

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

C کوڈ میں تبدیل کرتے ہوے...

```
#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;
}
```

کچھ نتائج کو چلاتے ہوئے، ہم
z کے ساتھ امکان کو تیزی سے
گرتے ہوئے دیکھ سکتے ہیں۔

```
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 سے کم P کے لئے حل کر رہا ہے...

```
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
```

نتیجہ

ہم نے اعتماد پر انحصار کیے بغیر الیکٹرانک لین دین کے لئے ایک نظام تجویز کیا ہے۔ ہم نے ڈیجیٹل دستخطوں سے بنائے گئے سکوں کے معمول کے فریم ورک کے ساتھ آغاز کیا، جو ملکیت پر مضبوط کنٹرول فراہم کرتا ہے، لیکن دوہرے اخراجات کو روکنے کے طریقے کے بغیر نامکمل ہے۔ اس کے حل کے لئے ہم نے پروف آف ورک کا استعمال کرتے ہوئے ایک پیئر ٹو پیئر نیٹ ورک تجویز کیا تاکہ لین دین کی عوامی تاریخ ریکارڈ کی جا سکے جو کسی حملہ آور کے لئے کمپیوٹیشنل طور پر ناقابل عمل ہو جاتی ہے، اس شرط کے ساتھ اگر ایماندار نوڈز سی پی یو طاقت کی اکثریت کو کنٹرول کرتے ہیں۔ نیٹ ورک اپنی غیر ساختہ سادگی میں مضبوط ہے۔ نوڈز بہت کم ہم آہنگی کے ساتھ ایک ہی وقت میں کام کرتے ہیں۔ ان کی شناخت کی ضرورت نہیں ہے، کیونکہ پیغامات کو کسی خاص جگہ پر نہیں پہنچایا جاتا اور انھیں صرف بہترین کوشش کی بنیاد پر پہنچانے کی ضرورت ہوتی ہے۔ نوڈز اپنی مرضی سے نیٹ ورک چھوڑ کر دوبارہ شامل ہو سکتے ہیں، کام کے ثبوت کے سلسلے کو اس بات کے ثبوت کے طور پر قبول کر سکتے ہیں کہ جب وہ چلے گئے تھے تو کیا ہوا تھا۔ وہ اپنے سی پی یو کی طاقت کے ساتھ ووٹ دیتے ہیں، جائز بلاکس کی توسیع پر کام کر کے ان کی قبولیت کا اظہار کرتے ہیں اور ناجائز بلاکس پر کام کرنے سے انکار کر کے مسترد کرتے ہیں۔ اس متفقہ طریقہ کار کے ساتھ کوئی بھی ضروری اصول اور ترغیبات نافذ کی جا سکتی ہیں۔

حوالہ جات

[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.