بٹ کوائن: ایک پیر-ٹو-پیر الیکٹرانک کیش سسٹم

سنتوشی ناکاموٹو

satoshin@gmx.com

www.bitcoin.org

ابسٹراکٹھ: الیکٹرانک کیش کا ایک خالصتاً پیئر-ٹو-پیئر ورژن جو کسی بھی مالیاتی ادارے میں جاے بغیر آن لائن ادائیگیوں کو سیدھے ایک پارٹی سے دوسرے پارٹی تک پہکنانے میں مدد کرتا ہے. ڈیجیٹل سگنیچر بھی اس سولیشن کا ایک حصّہ ہے لیکن اسکا سبسے بڑا فائدہ ہمارے ہاتھ سے نکل جایگا اگر ہمیں ڈبل-سپینڈنگ کے لئے کسی بھروسمند تھرڈ-پارٹی کی ضرورت پڑتی ہے. ہم ایک پیئر-ٹو-پیئر نیٹ ورک کے زریعے ڈبل-سپینڈنگ کی مشکل کا حل مہیا کروا رہی ہیں. یہ نیٹ ورک ٹرانزیکشن کو ہیش-بیسڈ پروف-عوف-ورک میں کی ایک انگونگ چین میں ہیش کرکے، پروف-عوف-ورک کو ریڈو کرے بنا نہ چینج ہونے والا ریکارڈ بنا کر ٹرانزیکشن کو ٹائمسٹیمپ کرتا ہے. سبسے لنبی چین نہ صرف ایونٹس کی سیکونس کے پروف کا کام کرتی ہے بلکہ یہ بھی سبوت بنتی ہے کہ یہ سیپویو پاور کے سبسے بڑے پول سے آی ہے. جب تک سیپییو پاور کی مجوریٹے ان نوڈز دورہ کنٹرول کی جاتی ہے جو نیٹورک کو اٹیک کرنے میں کوپریٹ نہیں کرتی، ٹیب تک یہ سبسے لنبی چین بنےگا اور اٹیکرز کو بہت پیچھے چھوڈ دیتے ہیں. اس نیٹ ورک کو بہت ہی کم سٹرکچر کی ضرورت ہوتی ہے. میسیجوں کو بہترین کوششوں کے ساتھ بروڈکاسٹ کیا جاتا ہے اور نوڈز اپنی مرضی سے نیٹورک سے باہر یا اندر ہو سکتی ہیں اور انکے جانے کے بعد نیٹورک میں کیا کچھ ہوا ہے اسکے سبوت کے طور پر سبسے لنبی پروف-عوف-ورک چین کو قبول کر سکتے ہیں.

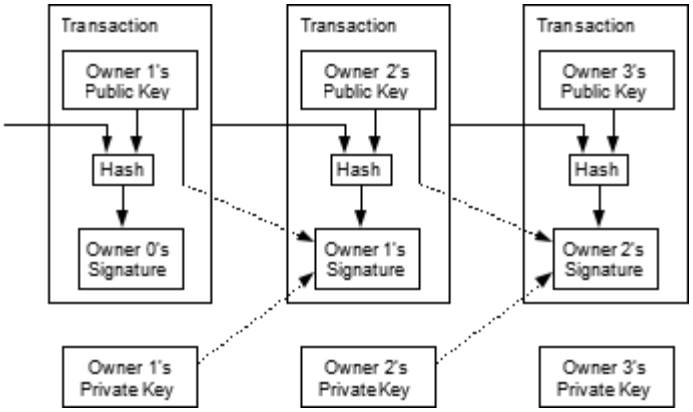
1. تعارف

انٹرنیٹ کے اپر کمرس اب بہت زیادہ ان مالیاتی اداروں کے بھروسے چل رہا ہے جو الیکٹرانک پیمنٹ کرنے کے لئے ایک بھروسےمند تھرڈ-پارٹی کا رول ادا کر رہی ہیں. یہ سسٹم زیادہتر ٹرانزیکشن کے لئے بہت اچھے سے کام کرتا ہے، لیکن ٹرسٹ بیسڈ ماڈل کی موروثی کمزوری سے گرست ہے. پوری طرح سے نون-ریورسیبل ٹرانزیکشن پوری طرح سے ممکن نہیں کیونکہ مالیاتی ادارے ڈسپیوٹس میں میڈیٹ (دخلاندازی) کرنے سے گریز نہیں کر سکتے. اس دخلاندازی کی لاگت سے ٹرانزیکشن کی قیمت بڑھ جاتی ہے جسسے منیمم پریکٹیکل ٹرانزیکشن سائز کم ہو جاتا ہے اور چھوٹی-چھوٹی کیزول ٹرانزیکشنز کی امکان کم ہو جاتی ہے اور نون-ریورسیبل سروسز کے لئے نون-ریورسیبل پیمنٹس کرنے کی کبلیت نہ ہونے کی وجہ سے قیمت بڑھ جاتی ہے. ریورسل کی پوسیبیلیٹی کی وجہ سے بھروسے کی ضرورت بڑھ جاتی ہے. مرچنٹس کو اپنے گراہکوں کے لئے سودھن ہونا چاہئے، اور انہیں انکی ضرورت کے مطابق انفارمیشن دینی چاہئے. کچھ حد تک دھوکھادھڑی کو کالعدم منا جاتا ہے. یہ قیمتیں اور پیمنٹ انسرٹینیٹیز سے فزیکل کرنسی کے استمال سے بچا جا سکتا ہے، لیکن ایسا کوئی مکینظم موجود نہیں ہے جسسے ایک بھروسےمند پارٹی کے بنا کسی کومیونیکشن چینل پر پیمنٹ کی جا سکے.

اس لئے یہاں ہمیں بھروسے کی جگہ پر کرپٹوگرافک پروف پر بیسڈ ایک الیکٹرانک پیمنٹ سسٹم کی ضرورت ہے جو کسی بھی دو پارٹیوں کو سیدھے ٹرانزیکٹ کرنے کی عزازات دے اور جسمیں کسی بھی ٹرسٹڈ پارٹی کی ضرورت نہ ہو. جو ٹرانزیکشنز ریورس نہیں ہو سکتی ہیں ایسی ٹرانزیکشنز سیلرز کو فراڈ سے بچاتی ہیں اور خریداروں کو بھی پروٹیکٹ کرنے کے لئے روٹین ایسکرو میکنزم کو شروع کیا جا سکتا ہے. اس پیپر میں ہم ایک ایسا سولیوشن تجویز کر رہی ہیں جو پیئر-ٹو-پیئر ڈسٹریبیوٹڈ ٹائمسٹیمپ سرور کے زریعے ڈبل-سپینڈنگ مشکل کو حل کریگا اور ٹرانزیکشن کے کرونولوجیکل آرڈر کا کمپوٹیشنل پروف جنریٹ کریگا. یہ سسٹم اس وقت تک سکیور ہے جب تک کہ اماندر نوڈز ملکر کسی بھی اٹیکر نوڈز کے کوپریتنگ گروپ کے مقابلے سیپییو کی زیادہ پاور کو کنٹرول کرتی ہیں.

2. ٹرانزیکشنز

ہم الیکٹرانک کون کی ڈیجیٹل سگنیچرز کی ایک چین کے روپ میں وضاحت کرتے ہیں. ہر ایک مالک پچھلی ٹرانزیکشن کے ہیش کو اور اگلے مالک کی پبلک کی ڈیجیٹلی سائن کرکے اور انہیں کون کے اند پر جوڑ کر اگلے مالک کو یہ کون ٹرانسفر کر دیتا ہے.

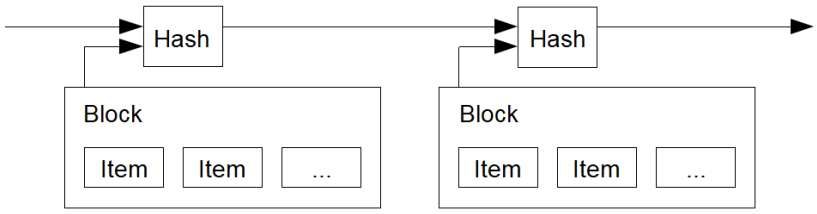


اسمیں مشکل یہ ہے کہ ادائیگی کرنے والا یہ ویریفائی نہیں کر سکتا ہیں کسی مالک نے ڈبل-سپینڈ کیا ہے یا نہیں. اسکا ام سا حل ہے کہ ایک بھروسیمند سینٹرل اتھارٹی، یا منٹ، کو پیش کیا جائے جو ڈبل-سپینڈنگ کو روکنے کے لئے ہر ایک ٹرانزیکشن کو چیک کرے. ہر ایک ٹرانزیکشن کے بعد کوئین منٹ کو واپس ہونا چاہئے اور جو کوئین منٹ سے سیدھے جاری کئے گئے ہیں صف انمیں ہی ڈبل-سپینڈ نہ ہونے کا بھروسہ ہوتا ہے. اس حل کے ساتھ یہ مشکل ہے کہ پورے منی سسٹم کی قسمت منٹ کو چلانے والی کمپنی کے ہاتھ میں ہوتی ہے کیونکہ ایک بنک کی طرح ہر ایک ٹرانزیکشن اسے ہوکر جاتی ہے.

ہمیں ایک ایسا طریقہ چاہئے جو ادائیگی کرنے والے کو یہ بتائے کہ پچھلے مالکوں نے کسی بھی پہلی ٹرانزیکشن کو سائن نہیں کیا ہے. ہمارے مطلب کے لئے سبسے پہلی ٹرانزیکشن وہ ہوتی ہے جسکی گنتی ہوتی ہے، اسلئے ہم ڈبل-سپینڈ کے لئے بعد کی کوششوں پر دھیان نہیں دیتے. اس ڈبل-سپینڈ سے بچنے کا ایک ہی طریقہ یہ ہے کہ ہمیں سبھی ٹرانزیکشنوں کے بارے میں پتا ہو. منٹ بیسڈ ماڈل میں منٹ کو سبھی ٹرانزیکشنوں کے بارے میں پتا ہوتا تھا اور یہ فیصلہ کرتی تھی کہ سبسے پہلے کونسی ٹرانزیکشن پہنچےگی. ایک بر ھوسیمند پارٹی کے بنا اسے پورا کرنے کے لئے ٹرانزیکشن کو پبلیکلی اناونس کرنا پڑیگا [1]، اور ہمیں ایک ایسے سسٹم کی ضرورت ہے جسمیں ہسصدار آرڈر کی سنگل ہسٹری پر متفق ہوں. ادائیگی کرنے والے کو یہ سبوت چاہئے کہ ٹرانزیکشن کے ٹائم پر زیادہتر نوڈز اس بات پر شمت ہیں کہ یہ پہلی بار ریسیو ہو رہی ہے.

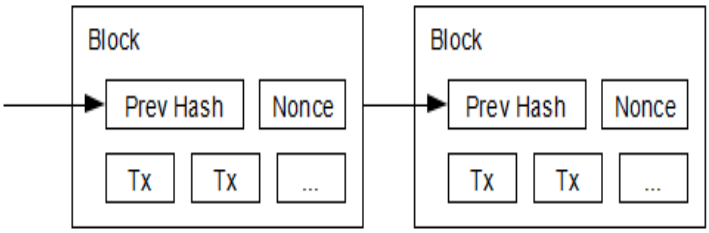
3. ٹائمسٹیمپ سرور

ہم جو سولیوشن لیکر آئے ہیں وہ ایک ٹائمسٹیمپ کے ساتھ شروع ہوتا ہے. ایک ٹائمسٹیمپ سرور ایٹمز کے بلاک کے ایک ہیش کو ٹائمسٹیمپ کرتا ہے اور ہیش کو سبھی تک پبلش کرتا ہے جیسے کہ ایک اخبار یا یوزنیٹ پوسٹ پر ہوتا ہے [2-5]. ٹائمسٹیمپ یہ سابت کرتی ہے کہ ڈیٹا ہیش میں داخل ہونے کے لئے ڈیٹا موجود رہا ہوگا. ہر ایک ٹائمسٹیمپ میں اپنے ہیش کی پچھلی ٹائمسٹیمپ شامل ہوتی ہے جسسے ایک چین بنتی ہے اور ہر ایک ایڈیشنل ٹائمسٹیمپ اپنے سے پہلے والی ٹائمسٹیمپس کو مجبت کرتی ہے.



4. پروف-عوف-ورک

پیئر-ٹو-پیئر کی بنیاد پر ٹائمسٹیمپ سرور کو امپلیمنٹ کرنے کے لئے ہمیں اخبار یا یوزنیٹ کی بجائے ایٹم بیک کے ہیشکش کی طرح ہی پروف-عوف-ورک سسٹم کی ضرورت ہے [6]، پروف-عوف-ورک میں اس ویلیو کے لئے سکیننگ شامل ہوتی ہو جو ایس.ایچ.اے.-256 کے ساتھ ہیش کرنے کے بعد ہیش زیرو بٹس کی گنتی کے ساتھ شروع ہوتا ہے. زیرو بٹز کی گنتی کا ایکسپونینشل جتنے اوسٹن ورک کی ضرورت پڑتی ہے اور اسے سنگل ہیش کو ایگزیکیوٹ کرکے ویریفائی کیا جا سکتا ہے.
ہمارے سٹیمپ نیٹورک کے لئے ہم اس وقت تک بلاک یونٹ میں ایک نونس کو بڑھا کر پروف-عوف-ورک کو امپلیمنٹ کرتے ہیں جب وہ ویلیو حاصل نہیں ہوتی جو بلاک کے ہیش کو اسکے لئے ضروری زیرو بٹس نے دے دے جب سی.پی.یو. کے کام کو پروف-عوف-ورک کو مطمئن کرنے تک بڑھا دیا جاتا ہے تو بلاک کو ورک کو ریڈو کئے بنہ بدلہ نہیں جا سکتا. کیونکہ بعد والے اسکے پیچھے چین میں لگے ہوتے ہیں بلاک کو بدلنے کے لئے ورک کو اسکے پیچھے والے سبھی بلاکس کو ریڈو کرنا پڑیگا.



پروف-عوف-ورک مجوریٹی کے فیصلہ لینے میں ریپرسنٹیشن کو دکھانے کی مشکل کو بھی حل کرتا ہے. اگر مجوریٹی ون-ایپی-اڈریس-ون-ووٹ پر ٹکی ہوئی تو اسے کسی بھی ایسے آدمی دوارہ خراب کیا جا سکتا ہے جو کی ایپیز کو بانٹ سکتا ہے. پروف-عوف-ورک ضروری طور پر ون-سیپیپیو-ون-ووٹ ہوتا ہے. مجوریٹی کا فیصلہ سبسے لمبی چین دوارہ دکھایا جاتا ہے جو جسے سبسے زیادہ پروف-عوف-ورک کوششیں لگی ہوتی ہیں. اگر سیپیپیو پاور کی مجوریٹی کسی امانڈر نوڈ دوارہ کنٹرول ہوتی ہے تو امانڈر چین بہت تیزی سے آگے برھیگی اور کسی بھی مقابلے والی چین کو پیچھے چھوڑ دیگی. کسی پاسٹ بلاک کو بدلنے کے لئے کسی اٹیکر کو بلاک کے پروف-عوف-ورک کو اور اسکے پچھلے سارے بلاکس کو ریڈو کرنا پڑیگا اور فر امانڈر نوڈز کے ورک کو پیچھے چھوڑنا ہوگا. ہم بعد میں بتائینگے کہ اگلے بلاکس کو جوڑنے کے بعد کسی سلو اٹیکر کے آگے نکلنے کے موقہ بہت ہی کم ہو جاتے ہیں.
بڑھتی ہوئی ہارڈویئر سپیڈ اور ایک وقت کے بعد نوڈز کو چلانے میں دلچسپی کے بدلنے کو کمپنسیٹ کرنے لئے پروف-عوف-ورک کی مشکل کو ایک ہر ایک گھنٹے میں بلاک کی اوست گنتی کو ٹارگٹ کرنے والی موونگ ایوریج کے تحت پیش کیا جاتا ہے. اگر وہ تیزی سے بڑھتے ہیں تو مشکل بھی اتنی ہی برھیگی.

5. نیٹورک

نیٹورک کو چلانے کے لئے قدموں کے بارے میں نیچے بتایا گیا ہے:

1) نئی ٹرانزیکشنز کو سبھی نوڈز تک بروڈکاسٹ کیا جاتا ہے.
2) ہر ایک نوڈ ایک بلاک میں نئی ٹرانزیکشنز کو کلیکٹ کرتی ہے.
3) ہر ایک نوڈ اپنے بلاک کے لئے ایک مشکل پروف-عوف-ورک ڈھونڈنے کا کام کرتی ہے.
4) جب ایک نوڈ ایک پروف-عوف ورک کو ڈھونڈ لیتی ہے تو یہ بلاک کو سبھی نوڈز تک بروڈکاسٹ کر دیتی ہے.
5) نوڈز بلاک کو تبھی عزازات دیتی ہیں اگر اسمیں شامل سبھی ٹرانزیکشنز ویلڈ ہوں اور یہ پہلے سے سپینڈ نہ کی گئی ہوں.
6) یہ نوڈز چین میں اگلا بلاک بناکر یہ بتاتی ہیں کہ انہونے بلاک کو عزازات دے دی ہے. یہ عزازات دئے گئے بلاک کے ہیش کو پچھلے ہیش کے طور پر استمال کرتی ہے.

نوڈز ہمیشہ ہی سبسے لنبی چین کو سہی مانتی ہیں اور اسے آگے بڑھاتی ہیں. اگر ٹو نوڈز ایک ہی وقت پر اگلے بلاک کے الگ-اگل ورزن کو بروڈکاسٹ کرتی ہیں تو کچھ کسی اک یا دوسرے بلاک کو پہلے حاصل کر سکتی ہیں. اس کیس میں، وہ اس بلاک پر کام کرینگی جو انھ پہلے حاصل ہوا ہے، لیکن دوسری برانچ کو بھی سیو کرینگی تاکہ یہ لمبی نہ ہو جائے. جب اگلا پروف-

عوف-ورک ملیگا تو یہ ٹائی ٹوٹ جائیگی اور ایک برانچ لمبی ہو جائیگی. فر جو نوڈز دوسری برانچ پر کام کر رہی تھی وہ لمبی والی برانچ پر سوئچ کر لنگی.

ضروری نہیں ہے کہ نئی ٹرانزیکشن بروڈکاسٹ سبھی نوڈز تک پہنچے. جب تک وہ سبھی نوڈز تک پہکنہٹی ہیں وہ بنا کسی انتظار کے ایک بلاک میں داخل ہو جائینگی. بلاک بورڈکاسٹ چھوڑے گئے میسیجوں کو بھی سیو کرتے ہیں. اگر کوئی نوڈ کسی بلاک کو حاصل نہیں کرتی تو اگلے بلاک کو حاصل کرنے پر یہ اسے ایک ریقویسٹ کریگی اور اسے پتا ہوگا کہ اسنے ایک بلاک کو مس کیا ہے.

6. انسنٹو (ترغیب)

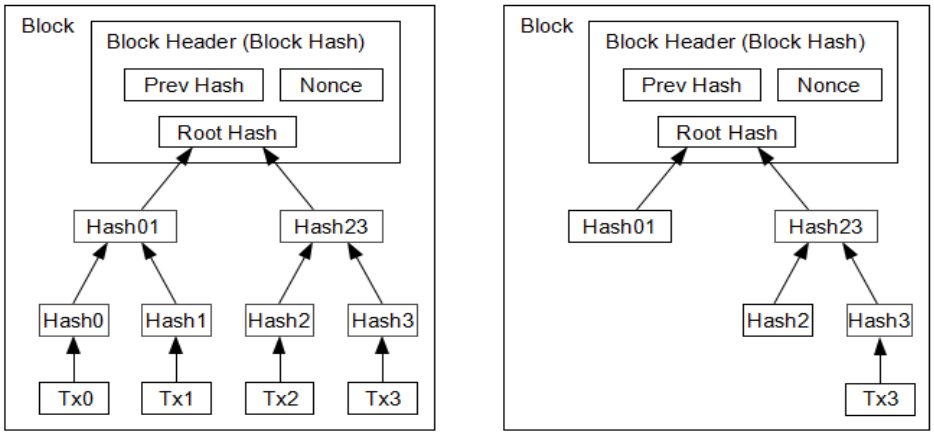
ام طور پر ایک بلاک میں سبسے پہلی ٹرانزیکشن سپیشل ٹرانزیکشن ہوتی ہے جو بلاک کے کریٹر کے کسی نئے کوئین کو شروع کرتی ہے. یہ نیٹورک کو سپورٹ کرنے کے لئے نوڈز کے لئے ایک انسنٹو کو پیش کرتا ہے اور شروع میں سرکولیشن میں کوئینز کو ڈسٹریبیوٹ کرنے ایک طریقہ مہیا کرواتا ہے کیونکہ اسمیں انہیں جاری کرنے کے لئے کوئی بھی سینٹرل اتھارٹی نہیں ہوتی. نئے کوئینز کے امونٹ کے کانسٹنٹ کا مستحکم جوڑ گولڈ مائنز کی تره ہوتا ہے جو ریسورسوں کو سرکولیشن میں گولڈ کو جوڑنے کی اضازٹ دیتا ہے. ہمارے کیس میں، سیپیبو ٹائم اور الیکٹرسٹی کو ایکسپنڈ کیا جاتا ہے.

انسنٹو کو ٹرانزیکشن فیس کے زریعے بھی فنڈ دئے جا سکتے ہیں. اگر کسی ٹرانزیکشن کی اوٹپوٹ ویلیو اسکی ان پٹ ویلیو سے کم ہے تو اسکا یہ فرق ٹرانزیکشن فیس ہوگا جسے ٹرانزیکشن والے بلاک کی انسنٹو ویلیو میں جوڑ دیا جاتا ہے. جب ایک بار پہلے سے تعین نمبر سرکولیشن میں داخل ہوتا ہے تو انسنٹو پورے کو ہی ٹرانزیکشن فیس میں بدل سکتا ہے اور پوری طرح سے مہنگی راحت ہو سکتا ہے.

انسنٹو نوڈز کو امانڈر رہنے کے لئے مدد کر سکتا ہے. اگر کوئی لالچی اٹیکر امانڈر نوڈز کے مقابلے زیادہ سیپیبو پاور حاصل کرنے میں کامیاب ہو جاتا ہے تو وہ اپنی پیمنٹ کو واپس چرا سکتا ہے یا اسے نئے کوئین بنانے میں استمال کر سکتا ہے. اسے قواعد پر رہ کر فائدہ لینا چاہئے، جیسے کہ وہ دوسروں کے مقابلے زیادہ کوئینز حاصل کر سکتا ہے بجائے اسکے کہ وہ پورے سیسیٹم کو اور اپنے خود ک پیسے کو نقصان پہنچائے.

7. ڈسک سپیس کو دوبارہ کلم کرنا

ایک بار جب کسی کوئین میں کوئی نئی ٹرانزیکشن ضروری بلاکس کے تحت چلی جاتی ہے تو اسسے پہلے والی سپینڈ ہو چکی ٹرانزیکشن کو ڈسک سپیس بچانے کے لئے ہٹایا جا سکتا ہے. بلاک کے ہیش کو توڑے بنا اسے پورا کرنے کے لئے ٹرانزیکشنز کو ایک مرکل ٹری میں ہیش کیا جاتا ہے [7] [2] [5], جسکا روٹ بلاک کے ہیش میں ہوگا. اس ٹری کی برانچز کو ہٹاکر پرانے بلاک کو چھوٹا کیا جا سکتا ہے. انٹریور ہیشز کو سٹور کرنے کی ضرورت نہیں ہے.



Transactions Hashed in a Merkle Tree

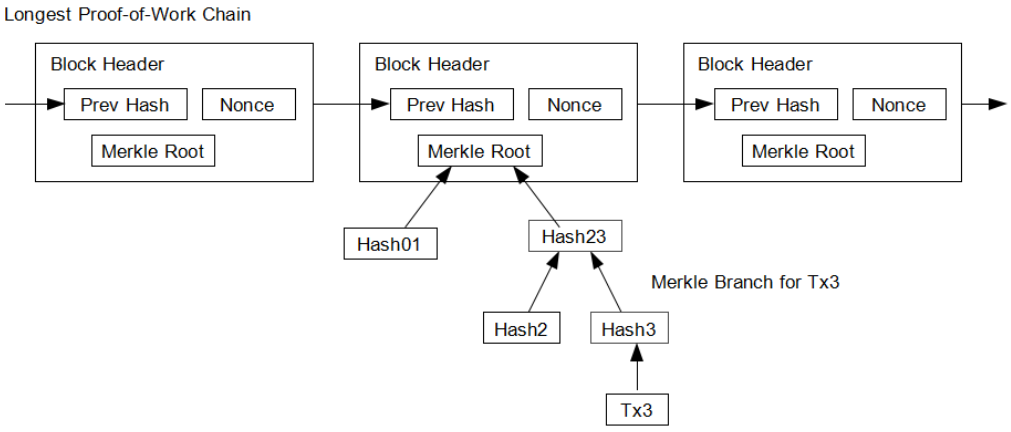
After Pruning Tx0-2 from the Block

بنا کسی ٹرانزیکشن کے ایک بلاک حیڈر 80 بائٹس کا ہوگا. اگر ہم یہ مانتے ہیں کہ بلاک ہر 10 منٹوں میں بنتے ہیں تو 80 بائٹس * 6 * 24 * 365 = ہر سال 4.2 ایمبی. 2008 کے مطابق اگر کسی کمپیوٹر کی 2 جیبی ریم بھی ہے اور مور کا لا ہر سال

1.2جیبی گروتھ کا اندازہ لگاتا ہے تو اسکی سٹوریج کوئی بری مشکل نہیں ہوگی، چاہے بلاک ہیڈرز کو میموری مہ ہی رکھا جائے.

8. آسان پیمنٹ وریفیکشن

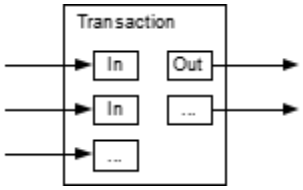
سبھی نیٹورک نوڈ کو چلائے بنا پیمنٹس کو ویریفائی کرنا ممکن ہے. ایک یوزر کو اپنے پاس صرف سبسے لمبے پروف-عوف-ورک چین کے بلاک ہیڈر کی ایک کوپے کو رکھنا ہوگا جسے وہ نیٹورک نوڈز کو اپنی قویری بھیج کر حاصل کر سکتا ہے. اسکے علاوہ یوز مرکل برانچ کو حاصل کرنا ہوگا جو ٹرانزیکشن کے ساتھ لنک کرتا ہے. وہ خود یہ ٹرانزیکشن چیک نہی کر سکتا، لیکن چین میں کسی جگہ پر اسے لنک کرنے کے بعد، وہ دیکھ سکتا ہے کہ نیٹورک نوڈ نے اسے عزازات دے دی ہے اور اسکے بعد جڑنے والے بلاک بھی یہ بتاتے ہیں کہ نیٹورک نے اسے عزازات دے دی ہے.



اس تارہ سے جب تک امانڈر نوڈز نیٹورک کو کنٹرول کر رہی ہیں ٹیب تک وریفیکشن پر بھروسہ ہو سکتا ہے، لیکن اگر کوئی اٹیکر نیٹورک پر زیادہ کبزہ کر لے تو یہ وریفیکشن گلت بھی ہو سکتی ہے. کیوننکہ نیٹورک نوڈز خود اپنے لئے ٹرانزیکشن کو ویریفائی کر سکتی ہیں، لیکن اگر اٹیکر لگاتار نیٹورک کے بڑے حصّے پر کبزہ کرتے ہیں تو اس آسان طریقے کو بھی غلط ٹھہرایا جا سکتا ہے. اسے بچاؤ کے لئے ینویلڈ بلاک کے ملنے پر بلاک دوارہ بھیجے گئے الرٹ کو عزازات دینی چاہئے اور یوزر کو یہ عزازات ہونی چاہئے کہ وہ فل بلاک اور الرٹڈ ٹرانزیکشنز کو دیکھ سکے اور پتا لگا سکے کہاں کوئی گڑبڑ ہوئی ہے. جو بزنس فریکونٹ پیمنٹس حاسیل کرتے ہیں وہ اپنی نوڈز کو خود چلانا چاہینگے تاکہ انھیں زیادہ سیکورٹی اور زیادہ تیز وریفیشن کی عزازات مل سکے.

9. ویلیو کو کمبائن اور سپلٹ کرنا

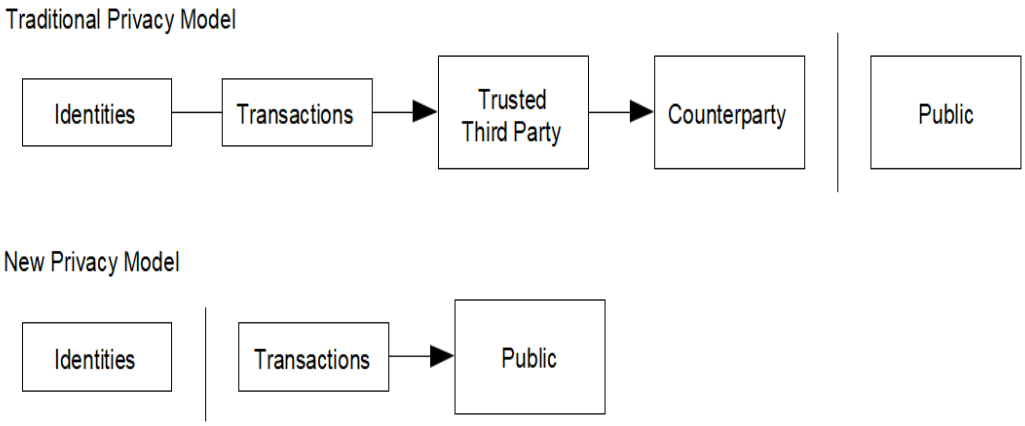
بھلے ہی کوئینز کو انفرادی طور پر ہینڈل کرنا ممکن ہے فر بھی ایک ٹرانسفر میں ہر ایک سینٹ کے لئے ایک الگ ٹرانزیکشن کرنا غلط ہوگا. ویلیو کو سپلٹ اور کمبائن کرنے کے لئے ٹرانزیکشن میں ایک سے زیادہ ان پٹس اور اوٹپوٹس ہونی چاہئے. ام طور پر اسمیں پچھلی بڈی ٹرانزیکشن میں سے ایک ان پٹ ہوگی یا چھوٹے-چھوٹے امونٹ والی ایک سے زیادہ ان پٹس ہونگی، اور زیادہ سے زیادہ ڈو اوٹپوٹس ہونگی: ایک پیمنٹ کے لئے اور ایک بدلاؤ کو، اگر کوئی ہے تو، سینڈر کو واپس بھیجنے کے لئے.



یہ نوٹ کرنا چاہئے کہ فین-اوٹ، جہاں ٹرانزیکشنز کیی ٹرانزیکشن پر منحصر کرتی ہے اور وہ ٹرانزیکشن اور کیی پر پر منحصر ہوتی ہیں، یہاں پر کوئی مشکل نہیں ہے. یہاں کبھی بھی ٹرانزیکشن ہسٹری کی ایک کمپلیٹ اسٹینڈالون کاپی کو نکالنے کی ضرورت نہیں پڑتی ہے.

10. پرائیویسی (رازداری)

روایتی بینکنگ ماڈل اپنی پارٹیوں کی اور ٹرسٹڈ تھرڈ پارٹی کو انفارمیشن تک لمیٹڈ پہنچ دیکر پرائیویسی کے لیول کو حاصل کرتا ہے. سبھی ٹرانزیکشنز کو پبلک میں اناونس کرنے کی ضرورت نے اس ماڈل میں بڑی کمی کو پایا ہے، لیکن پرائیویسی کو دوسری جگہ میں انفارمیشن کے فلو کو طوڑ کر بنائے رکھا جا سکتا ہے: پبلک کیز کو بنا کسی نام کے رکھ کر. پبلک یہ دیکھ سکتی ہے کہ کوئی انسان کسی دوسرے انسان کو کوئی امونٹ بھیج رہا ہے، لیکن اسمیں انفارمیشن ٹرانزیکشن کو کسی دوسری ٹرانزیکشن کے ساتھ لنک نہیں کریگی. یہ اسٹاک ایکسچینجوں میں ریلیز کی گئی انفارمیشن کے لیول کی طرح ہی ہوتا ہے، جسمیں انڈیویجول ٹریڈ کا ٹائم اور سائز، "ٹیپ"، پبلک کے پتا چل جاتا ہے لکن پارٹیوں کی کوئی انفارمیشن نہیں ملتی ہے.



ایک اور فائروال کی طرح، ہر ایک ٹرانزیکشن کے لئے ایک نئے کی پیر کو استمال کیا جانا چاہئے تاکہ اسے کسی بھی مالک کے ساتھ لنک نہ کیا جا سکے. کچھ لنکنگ ملٹی-ان پٹ ٹرانزیکشن میں ابھی بھی نزرابداز نہی کیا جا سکتا، جسسے یہ پتا چلتا ہے کہ انکی ان پٹس کے مالک ایک ہی تھے. اسمیں یہ خطرہ ہے کہ کی کے مالک کا پتا چل جاتا ہے، اور لنکنگ اس مالک سے جڑی ہوئی باکی ٹرانزیکشنز کو بھی سامنے لا سکتی ہے.

11. کیلکولیشنز

ہم اس سورت کو بھی کا بھی دھیان رکھتے ہیں جسمیں ایک اٹیکر کسی اماندر چین کے مقابلے زیادہ تیزی سے کوئی الگ چین بنانے کی کوشش کرتا ہے. اگر ایسا ہو بھی جاتا ہے تو اسے سسٹم میں منمرضی کے بدلاؤ نہیں کئے جا سکتے ہیں، جیسے کہ اپنی مرضی سے ویلیو کو بنانا یا اس پیسے کو چرانا جو کبھی اٹیکر کا تھا ہی نہیں. نوڈز پیمنٹ کے طور پر انویلڈ ٹرانزیکشنز کو نہیں مانینگی، اور اماندر نوڈز کبھی بھی ایسی ٹرانزیکشن والے بلاک کو قبول نہیں کرینگی. ایک اٹیکر صرف اپنی کی ہوئی ٹرانزیکشنز کو ہی بدل سکتا ہے تاکہ اسے اپنے وہ پیسے واپس مل سکیں جو اسنے ابھی-ابھی سپینڈ کئے ہیں.

ایک اماندر چین اور ایک اٹیکر چین کے بیچ کی ریس کو بائینومیل رینڈم واک کہا جا سکتا ہے. ایک کامیاب ایونٹ وہ ہوتی ہے جسمیں کہ اماندر چین کو کسی ایک بلاک دوارہ ایکسٹنڈ کیا جائے اور اسکی لیڈ بڑھکر +1 ہو جے، اور ایک نہکامیاب ایونٹ وہ ہوتی ہے جسمیں کہ اٹیکر کی چین کو کسی ایک بلاک دوارہ ایکسٹنڈ کیا جائے اور اسمیں -1 کا گیپ آ جائے.

ایک اٹیکر کی دئے گئے ڈیفیکٹ سے آگے نکلنے کی امکان ایک گیمبلر رن مشکل کی طرح ہوتی ہے. ماں لیں کہ ایک گیمبلر، جسکے پاس بہت سارا پیسا ہے، ایک ڈیفیکٹ سے سٹارٹ کرتا ہے اور بریکیون تک پہنچنے کی کوشش میں وہ کی ٹرائل کھیلتا ہے. ہم یہ امکان کیلکولیٹ کر سکتے ہیں کہ وہ کبھی بھی بریکیون تک نہیں پہنچےگا، یا ایک اٹیکر ایک اماندر چین سے آگے نکل جائیگا، جیسا نیچے بتایا گیا ہے [8]:

p = امکان ایک اماندر نوڈ اگلے بلاک کو ڈھونڈ لیتی ہے

q = امکان اٹیکر اگلے بلاک کو ڈھونڈ لیتا ہے

q_z = امکان اٹیکرپچھلے z بلاکس تک پہنچ جائیگا

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

ہماری مفروضہ ہے کہ p > q ہے، اسسے پروبیبلٹی بہت ہی کم ہو جاتی ہے کیونکہ اٹیکر کو اب زیادہ بلاکس تک پہنچنا پڑیگا. اٹیکر کے آگے مشکلوں کے ساتھ اگر وہ جلدی سے آگے نہیں بڑھتا ہے تو وہ بہت ہی پیچھے رہ جائیگا.
اب ہم یہ دیکھتے ہیں کہ ایک نئی ٹرانزیکشن کے رسیپینٹ کو یہ یکینی بنانے کے لئے کتنا انتظار کرنا پڑیگا کہ سینڈر ٹرانزیکشن کو چینج نہیں کر سکتا ہے. ہم مانتے ہیں کہ سینڈر ایک اٹیکر ہے جو رسیپینٹ کو کچھ دیر کے لئے یہ یکن کروانا چاہتا ہے کے اسنے اسے پیمنٹ دے دی ہے اور فر کچھ ہی وقت کے بعد وہ خود کو پے کر دیتا ہے. ایسا ہونے کی سورت میں ریسیور کو الرٹ کیا جایگا لیکن سینڈر کو لگےگا کہ اب بہت دیر ہو چکی ہے.
رسیور ایک نئی کی جنریٹ کریگا اور سائن کرنے سے تھوڑا وقت پہلے سینڈر کو پبلک کی دیگا. اسسے سینڈر ٹائم سے پہلے بلاکس کی چین نہیں بنا پائیگا اور ٹرانزیکشن محفظ طریقے سے ہو جائیگی. ٹرانزیکشن ہو جانے کے بعد، غیراماندر سینڈر اس پریلال چین پر کام کرنا شروع کرتا ہے جسمیں اسکی ٹرانزیکشن کا ایک الٹرنیٹ ورزن شامل ہے.
رسیپینٹ اس وقت تک انتظار کرتا ہے جب تک کہ اسکی ٹرانزیکشن کسی بلاک میں داخل نہ ہو جائے اور z بلاکس کو اسکے بعد لنک کر دیا جاتا ہے. یوز اٹیکر کی پروگریس کا کوئی اندازہ نہیں ہوتا، لیکن اسے ماں کر اماندر بلاک ہر بلاک کا اوستن ایکسپیکٹڈ ٹائم لیتے ہیں، اٹیکر کی پوٹنشل پروگریس ایکسپکٹڈویلیو کے ساتھ ایک پوازن ڈسٹریبیوشن ہوگا:

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

To get the probability the attacker could still catch up now, we multiply the Poisson density for each amount of progress he could have made by the probability he could catch up from that point:

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

Rearranging to avoid summing the infinite tail of the distribution...

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

Converting to C code...

```
#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;
}
```

Running some results, we can see the probability drop off exponentially with z.

```
q=0.1
z=0     P=1.0000000
z=1     P=0.2045873
z=2     P=0.0509779
z=3     P=0.0131722
z=4     P=0.0034552
z=5     P=0.0009137
z=6     P=0.0002428
z=7     P=0.0000647
z=8     P=0.0000173
z=9     P=0.0000046
z=10    P=0.0000012

q=0.3
z=0     P=1.0000000
z=5     P=0.1773523
z=10    P=0.0416605
z=15    P=0.0101008
z=20    P=0.0024804
z=25    P=0.0006132
z=30    P=0.0001522
z=35    P=0.0000379
z=40    P=0.0000095
z=45    P=0.0000024
z=50    P=0.0000006
```

Solving for P less than 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
```

12. نتیجہ

ہمنے ٹرسٹ پر یکین کئے بغیر الیکٹرانک ٹرانزیکشنز کرنے کے لئے سسٹم کو تجویز کیا ہے. ہمنے ڈیجیٹل سگنیچرز سے بنے کوئینز کے ام فریمورک کو شروع کیا ہے جو اونرشپ پر ایک مضبوط کنٹرول مہیا کرواتا ہے، لیکن یہ ڈبل-سپینڈنگ کو روکنے کے ایک طریقے کے بغیر ادھورا ہے. اسے حل کرنے کے لئے، ہمنے پروف-عوف-ورک کا استمال کرکے ایک پیئر-ٹو-پیئرنیٹورک کو تجویز کیا ہے تاکہ ان ٹرانزیکشنز کی پبلک ہسٹری کو ریکارڈ کیا جا سکے جنہیں چینج کرنا ایک اٹیکر کے لئے کمپوٹیشنلی امپریکٹیکل ہو جاتا، اگر امانڈر نوڈز سیپیپیو پاور کی مجوریٹی کو کنٹرول کرتے ہیں تو. یہ نیتورک اپنی انسٹرکچرڈ سمپلیسٹی میں بہت ہی مجبت ہے. نوڈز تھوڑی سی حصّیداری کے ساتھ ایک بار میں ہی کام کرتی ہیں. انھیں اپنی پہچان بتانے کی ضرورت نہیں ہوتی کیونکہ میسیج کسی بھی خاص جگہ کو نہیں بھیجے جاتے اور انھیں بہترین کوششوں کی بنیاد پر ڈلیور کرنے کی ضرورت ہوتی ہے. نوڈز اپنی مرضی سے نیٹورک میں شامل ہو سکتے ہیں یا اسے چھوڑ سکتے ہیں، اور یہ انکے جانے کے بعد نیٹورک میں جو بھی ہوا ہے اسکے سبوت کے طور پر پروف-عوف-ورک چین کو قبول کر سکتے ہیں. وہ اپنے سیپیپیو پاور کو ووٹ کرتے ہیں، اور ویلڈ بلاکس کو ایکسٹنڈ کرکے انھیں قبول کرتے ہیں اور ینویلڈ بلاک پر کام نہ کرکے انھ نہقبول کرتے ہیں. اس متفقہ طریقہ کار کے ساتھ کسی بھی ضروری اصول اور ترغیبات نافذ کی جا سکتی ہیں.

References

[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 minimaltrust 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, no2, 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 Conferenceon 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 andPrivacy*, IEEE Computer Society, pages 122-133, April 1980.

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