BITFURY

# البيتكوين: نظام النقد الإلكتروني القائم على مبدأ النظير للنظير

# Bitcoin: The peer-to-peer electronic cash system

Translation completed in partnership with Mr. Saifedean Ammous,
author of The Bitcoin Standard

GBBC Global Blockchain Business Council

BTA

# البيتكوين: نظام النقد الإلكتروني القائم على مبدأ النظير للنظير

ساتوشي ناكاموتو
satoshin@gmx.com
www.bitcoin.org

**ملخص:** إن وجود نسخةٍ من النظام القائم على مبدأ النظير للنظير مكون من العملات الرقمية سيسمح بإرسال المدفوعات عبر الإنترنت مباشرة من طرفٍ إلى آخر دون المرور بمؤسسة مالية. حيث تقدم التوقيعات الرقمية جزءاً من الحل، ولكنها سيتم فقدان الفوائد الرئيسية إذا ظل هناك حاجة لوجود طرف ثالث موثوق به لمنع الإنفاق المضاعف (Double spending) . نحن نقترح حلاً لمشكلة الإنفاق المضاعف باستخدام شبكة النظير للنظير، حيث تقوم الشبكة بتسجيل الطابع الزمني للتحويلات عن طريق دمجها في سلسلة مستمرة من أنظمة إثبات العمل (Proof-of-work) القائمة على الهاش(Hash) ، فيتم تكوين سجل لا يمكن تغييره دون إعادة تشغيل نظام إثبات العمل. ولا تقتصر الفائدة من السلسلة الأطول على إثبات تسلسل الأحداث التي شهدتها فحسب، بل إنها تقدم دليلاً على أنها جاءت من أكبر كمية من الطاقة من وحدات المعالجة المركزية. وطالما يتم التحكم في غالبية طاقة وحدة المعالجة المركزية من خلال العُقَد التي لا تتعاون في مهاجمة الشبكة، فتكوّن سلسلةً أسرع لتتفوق بها على المهاجمين. وتتطلب الشبكة ذاتها الحد الأدنى من الهيكل التنظيمي، حيث يتم بث الرسائل على أساس ما يُسمى بالجهد الأكبر(Best effort basis) ، ويمكن للعُقَد أن تغادر الشبكة وتعيد الانضمام إليها حسب الرغبة، مع قبول أطول سلسلة لنظام إثبات العمل كدليل على ما حدث أثناء غيابها.

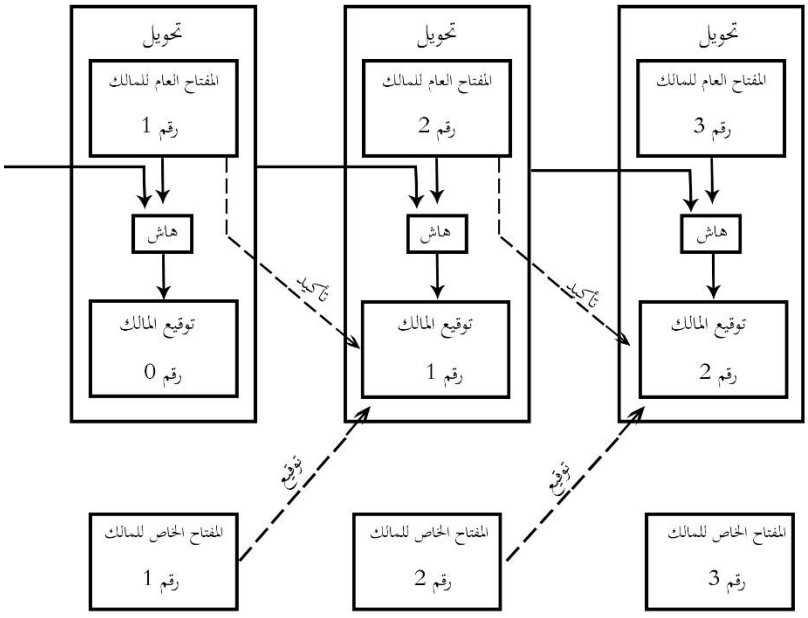
## 1. مقدمة

أصبحت التجارة على الإنترنت تعتمد بشكل شبه حصري تقريباً على المؤسسات المالية التي تعمل كجهات خارجية موثوقة لمعالجة المدفوعات الإلكترونية. وبينما يعمل النظام بشكل جيد لأداء معظم التحويلات، إلا أنه لا زال يعاني من نقاط الضعف التقليدية للنموذج الذي يعتمد على الثقة. فلا يمكن إجراء تحويلات غير قابلة للتراجع بشكل كامل، حيث لا يمكن للمؤسسات المالية تجنب توسط النزاعات. وأما تكلفة الوساطة فإنها تزيد من تكاليف التحويلات، وتقلل الحد الأدنى للحجم العملي للتحويلات، وتحد من إمكانية إجراء تحويلات صغيرة من وقتٍ لآخر، كما أن هناك تكلفة أوسع نطاقاً، تتمثل في فقدان القدرة على تقديم مدفوعات غير قابلة للتراجع للخدمات غير القابلة للرد. فمع وجود الإمكانية للتراجع عن المدفوعات، تزداد الحاجة لانتشار الثقة. فيجب على التجار التعامل بالحذر مع عملائهم، فيضطرون إلى إرهاقهم بطلب قدر يزيد عن حاجتهم من المعلومات. ويتم قبول نسبة معينة من التحويلات المزورة لا يمكن تجنبها. يمكن تجنب هذه التكاليف وحالات عدم التيقن من الدفع بشكل شخصي باستخدام عملة مادية، ولكن لا توجد آلية لسداد المدفوعات عبر قناة تواصل دون الحاجة لوجود طرف ثالث موثوق.

والمطلوب هنا هو نظام للدفع الإلكتروني بحيث يستند إلى دليل مشفر بدلاً من الثقة، مما يسمح لأي طرفين راغبين في التعامل مع بعضهما، بأن يقوما بالتعامل المباشر مع بعضهما البعض دون الحاجة إلى وجود طرف ثالث موثوق به. والتحويلات التي لا يمكن ردها حاسوبياً ستحمي البائعين بشكل عملي من الاحتيال، ويمكن بسهولة تنفيذ آليات الإيداع الروتينية لحماية المشترين. ونقترح في هذه الدراسة حلاً لمشكلة الإنفاق المزدوج باستخدام خادم ذي طابع زمني موزع حسب النظير للنظير لإنشاء دليل حاسوبي على الترتيب الزمني للتحويلات. يبقى النظام آمناً طالما أن العُقَد النزيهة تتحكم بشكل جماعي في طاقة وحدة المعالجة المركزية أكثر من أي مجموعة متعاونة من العقد المهاجمة.

## 2. التحويلات

نحن نعرّف العملة الإلكترونية على أنها سلسلة من التوقيعات الرقمية. يقوم كل مالك بتحويل العملة إلى المالك الذي يليه من خلال التوقيع رقمياً على "هاش" المعاملة السابقة، وعلى المفتاح العمومي للمالك القادم، ثم إضافة هذه البيانات إلى نهاية العملة. يمكن للمستفيد التحقق من التوقيعات للتحقق من سلسلة الملكية.

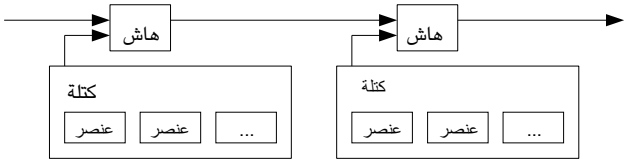



المشكلة بالطبع هي أن المستفيد من عملية السداد لا يمكنه التحقق من أن أحد المالكين لم ينفق العملة بشكل مضاعف. والحل الشائع هو توفير سلطة مركزية موثوق بها، أو جهة لمراقبة العملة، لتقوم بالتحقق من أن كل تحويلٍ لم يتعرض للإنفاق المضاعف. وبعد كل تحويل، يجب أن تعاد العملة إلى جهة مراقبة العملة لإصدار عملة جديدة، ولا يتم الوثوق إلا بالعملات التي تصدر مباشرة من جهة مراقبة العملة على أنها لم تتعرض للإنفاق مرتين. تكمن مشكلة هذا الحل في أن مصير النظام المالي بأكمله يعتمد على الشركة التي تدير جهة مراقبة العملة، إذ يجب أن تمر كل معاملة من خلالها، كما هو الحال تماماً في حالة البنك.

لهذا، نحن بحاجة إلى طريقة يستطيع المستفيد من خلالها معرفة أن المالكين السابقين لم يوقعوا على أي تحويلات سابقة. ولتحقيق أهدافنا، فإن التحويل الأسبق هو الذي سيتم احتسابه، لذا، فإننا لا نهتم بالمحاولات اللاحقة للإنفاق المضاعف. وتتمثل الطريقة الوحيدة لتأكيد غياب تحويل هي معرفة جميع التحويلات التي سبقته. في النموذج القائم على جهة مراقبة العملة، تكون هذه الجهة على علم بجميع التحويلات وتقرر ما يصل أولاً. ولتحقيق ذلك دون وجود طرف موثوق به، يجب الإعلان عن التحويلات بشكل عام [1]، كما أننا بحاجة إلى نظام للمشاركين للموافقة على تاريخ واحد للترتيب الذي تم استلام التحويلات به. ويحتاج المستفيد في وقت كل معاملة، إلى إثبات موافقة أغلبية العقد على أنها المرة الأولى التي يتم استلامها فيها.

## 3. خادم الطابع الزمني

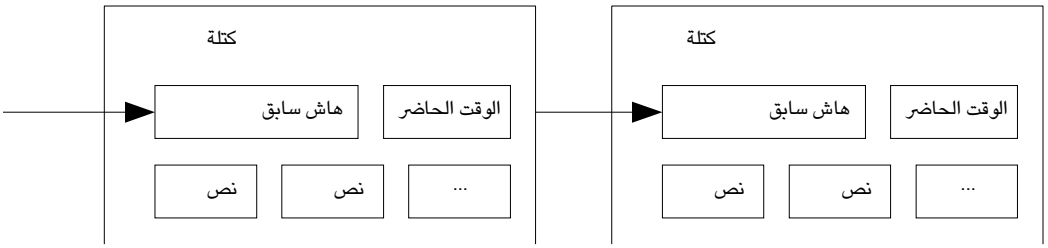
يبدأ الحل الذي نقترحه بخادم الطابع الزمني (Timestamp server). يعمل خادم الطابع الزمني عن طريق وضع هاش على كتلة من العناصر ليتم تحديد زمنها ونشرها على نطاق واسع، كما هو الحال في مقالات الصحف أو موقع Usenet [2-5]. يثبت الطابع الزمني أن البيانات كانت موجودة في ذلك الوقت وبشكل واضح، من أجل الوصول إلى الهاش. يتضمن كل طابع زمني الطابع الزمني الذي يسبقه في علامة هاش الخاصة به، ويشكل سلسلة، مع تأييد كل طابع زمني إضافي لما يسبقه من طوابع زمنية.




## 4. نظام إثبات العمل

لتنفيذ خادم طابع زمني موزّع على أساس النظير للنظير، سنحتاج إلى استخدام نظام إثبات عمل مماثل لنظام Adam Back's Hashcash [6]، بدلاً من مقالات الصحف أو مشاركات Usenet. يتضمن نظام إثبات العمل إجراء فحص لإيجاد قيمة معينة حيث عندما يتم دمجها، مثل SHA-256، تبدأ الهاش بعدد من البتّات الصفرية. يكون متوسط العمل المطلوب هو قيمة أسيّة مضروبة في عدد البتّات الصفرية المطلوبة، ويمكن التحقق منها عن طريق تنفيذ دمج واحد.

بالنسبة لشبكة الطابع الزمني الخاصة بنا، فإننا نقوم بتطبيق نظام إثبات العمل عن طريق زيادة قيمة الرقم في الكتلة حتى يتم العثور على قيمة تعطي لهاش الكتلة البتات الصفرية المطلوبة. بمجرد بذل جهد وحدة معالجة مركزية لتحقيق نظام إثبات العمل، لا يمكن تغيير الكتلة دون إعادة العمل. يشتمل العمل المطلوب لتغيير الكتلة على إعادة تنفيذ كل الكتل التي تليها، بينما يتم وضع الكتل التالية بعد ذلك في سلسلة.




يحل نظام إثبات العمل أيضاً مشكلة تحديد التمثيل في عملية اتخاذ القرار بالأغلبية. إذا كانت الأغلبية مستندة إلى صوت واحد لكل عنوانIP ، فقد يتم العبث بها بواسطة أي شخص قادر على تخصيص العديد من عناوين .IP يستند نظام إثبات العمل على أساس صوت واحد لكل وحدة معالجة مركزية. ويتم تمثيل قرار الأغلبية بأطول سلسلة، والتي تضم أكبر قدر من الجهد المبذول في أنظمة إثبات العمل. إذا تم التحكم بأكبر قدر من طاقة وحدة المعالجة المركزية بواسطة عقد نزيهة، فإن السلسلة النزيهة ستنمو بشكل أسرع وتتخطى أية سلاسل منافسة. لتعديل كتلة سابقة، يجب على المخترق أن يعيد صناعة نظام إثبات عمل ا للكتلة وكل الكتل التي بعدها ثم يلاحق العقد النزيهة ويتفوق على عملها. وسوف نوضح لاحقاً أن احتمالية حدوث ملاحقة من مخترق أبطأ تتضاءل بشكل كبير مع إضافة الكتل التالية.

للتعويض عن زيادة سرعة الأجهزة والاهتمام المتباين بتشغيل العقد بمرور الوقت، يتم تحديد صعوبة نظام إثبات العمل بمتوسط متحرك يستهدف متوسط عدد الكتل في الساعة. أي تزداد الصعوبة بازدياد سرعة إنشاء أنظمة إثبات العمل.

## 5. شبكة الاتصال

خطوات تشغيل الشبكة هي كما يلي:

1. يتم بث التحويلات الجديدة لكل العقد.
2. تجمع كل عقدة التحويلات الجديدة في كتلة.
3. تعمل كل عقدة على العثور على نظام إثبات عمل صارم لكتلتها.
4. عندما تعثر إحدى العقد على نظام إثبات العمل، فإنها تبث الكتلة إلى كل العقد.
5. تقبل العقد الكتلة فقط إذا كانت كل التحويلات الموجودة فيها صالحة ولم يتم إنفاقها بالفعل.
6. تعبر العقد عن قبولها للكتلة من خلال العمل على إنشاء الكتلة التالية في السلسلة، باستخدام هاش الكتلة المقبولة باعتباره هاش سابق.

تعتبر العقد دائماً أطول سلسلة على أنها السلسلة الصحيحة وستستمر في العمل على توسيعها. إذا قامت عقدتان ببث إصدارات مختلفة من الكتلة التالية في نفس الوقت، فقد تتلقى بعض العقد واحدة أو قد تتلقى غيرها أولاً. وهي في هذه الحالة تعمل على أول كتلة تستقبلها، ولكنها تحفظ الفرع الآخر في حال أصبح أكثر طولاً سيتم كسر الترابط بينهما عندما يتم العثور على نظام إثبات العمل التالي حتى يصبح فرع واحد أطول؛ ثم تنتقل العقد التي كانت تعمل في الفرع الآخر إلى الفرع الأطول.

لا تحتاج عمليات بث التحويلات الجديدة بالضرورة إلى الوصول إلى كل العقد. طالما أنها تصل إلى العديد من العقد، فإنها سوف تصل إلى كتلة قبل مرور فترة طويلة. ويمكن لعمليات بث الكتل أن تتحمل الرسائل المفقودة. إذا لم تتلق العقدة كتلة ما، فسوف تطلبها عند استلامها للكتلة التالية وتدرك أنها فقدت واحدة.

## 6. الحافز

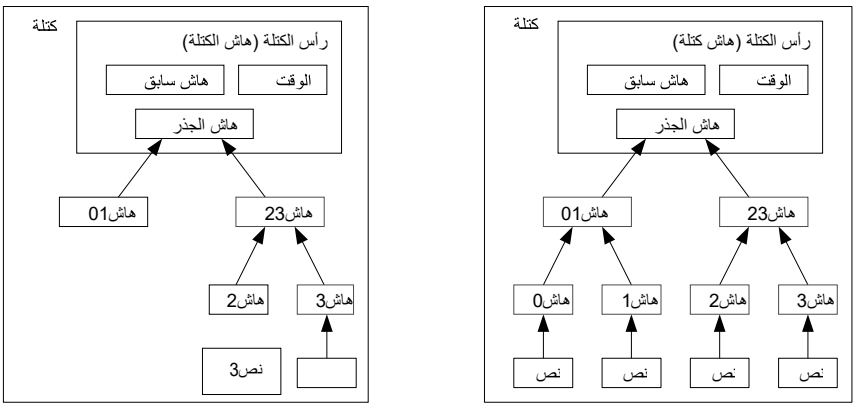
بموجب الاتفاق، فإن التحويل الأول في كتلة ما هو تحويل خاص يبدأ عملة جديدة يملكها صانع الكتلة. ويضيف ذلك حافزاً للعقد لدعم الشبكة، ويوفر طريقة لتوزيع العملات في التداول بشكل مبدئي بما أنه لا توجد سلطة مركزية لإصدارها. إن الإضافة الثابتة لثابت كمية من العملات الجديدة تشبه عمال مناجم الذهب الذين يستهلكون الموارد لإضافة الذهب إلى التداول. وفي حالتنا، يتم استهلاك الوقت والكهرباء الخاص بوحدة المعالجة المركزية.

ويمكن أيضاً أن يتم تمويل الحافز من خلال رسوم التحويلات. إذا كانت قيمة مخرجات التحويل أقل من قيمة مدخلاته، يكون الفرق عبارة عن رسوم تحويل تتم إضافتها إلى قيمة الحافز للكتلة المحتوية على التحويل. بمجرد أن يتم تداول عدد محدد من العملات، يمكن أن يتحول الحافز بالكامل إلى رسوم تحويلات ويكون خالياً من التضخم تماماً.

قد يساعد الحافز على تشجيع العقد لتبقى نزيهة. إذا كان المخترق الجشع قادراً على تجميع المزيد من طاقة وحدة المعالجة المركزية أكثر من جميع العقد النزيهة، فسيتعين عليه الاختيار بين استخدامه للاحتيال على الأشخاص عن طريق سرقة مدفوعاته، أو استخدامه لتكوين عملات جديدة. سيكتشف أنه سيحقق ربحاً أكبر إن التزم بالقواعد التي تكافئه بعملات جديدة أكثر من عملات الآخرين مجتمعة، بدلاً من إفساد النظام وصلاحية ثروته الخاصة.

## 7. استعادة مساحة القرص

بمجرد اختفاء أحدث تحويل لعملة أسفل ما يكفي من الكتل، يمكن التخلص من التحويلات المستنفذة ليتم توفير مساحة على القرص. لفعل ذلك دون كسر هاش الكتلة، يتم تجزئة التحويلات في شجرة ميركل [2] [7] [Merkle Tree] [5]، مع تضمين الجذر فقط في هاش الكتلة. ويمكن بعد ذلك ضغط كتل قديمة عن طريق تقليم أغصان الشجرة. لا يلزم تخزين علامات الهاش الداخلية.



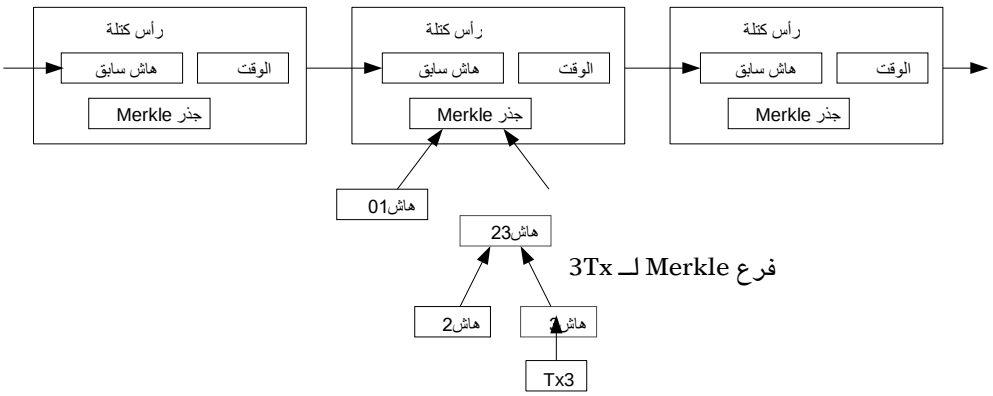

**التحويلات التي تم دمجها في Merkle Tree بعد إخراج 2-0Tx من الكتلة**

يُقدر حجم رأس الكتلة بدون التحويلات حوالي 80 بايت. إذا افترضنا أن الكتل يتم إنشاؤها كل 10 دقائق، 80 بايت * 6 * 24 * 365 = 4.2 ميجابايت في السنة. لا ينبغي أن يكون التخزين مشكلة، حتى إذا كان يجب الاحتفاظ برؤوس الكتل في الذاكرة، مع أنظمة الكمبيوتر التي تباع عادة مع 2 جيجابايت من ذاكرة RAM منذ عام 2008، وتوقُّعِ قانون مور للنمو الحالي بـ 1.2 جيجابايت في السنة.

## 8. التأكيد المبسط للدفع

يمكن التحقق من المدفوعات دون تشغيل عقدة شبكة كاملة. يحتاج المستخدم فقط إلى الاحتفاظ بنسخة من رؤوس الكتل لأطول سلسلة من نظام إثبات العمل، والتي يمكن الحصول عليها من خلال الاستعلام عن عُقَد الشبكة حتى يقتنع بامتلاك أطول سلسلة، ويحصل على فرع شجرة ميركل الذي يربط التحويل بالكتلة التي تم تسجيل طابعه الزمني فيها. لا يمكنه تأكيد التحويل لنفسه، ولكن عن طريق ربطها بمكان في السلسلة، يمكنه أن يرى أن هناك عقدة شبكة قد قبلتها، وأنه قد تم إضافة كتل بعد التأكد من قبول الشبكة لها.

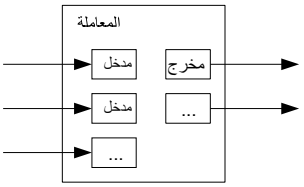
أطول سلسلة لإثبات العمل



على هذا النحو، يكون التحقق موثوقاً طالما أن العقد النزيهة تتحكم في الشبكة، ولكنها تكون أكثر عرضة للخطر إذا تم التغلب على الشبكة من جانب أحد المخترقين. بينما يمكن لعقد الشبكة التحقق من التحويلات بنفسها، يمكن اختراق الطريقة المبسطة بواسطة تحويلات مزورة للمخترقين طالما أنه يمكن للمخترق أن يستمر في التغلب على الشبكة. تتمثل إحدى استراتيجيات الحماية من ذلك في قبول التنبيهات من عقد الشبكة عند اكتشاف كتلة غير صالحة، مما يدفع برنامج المستخدم إلى تحميل الكتلة بالكامل والتحويلات التي تم تنبيهها لتأكيد عدم التناسق. ربما لا تزال الشركات التي تتلقى دفعات متكررة ترغب في إدارة عقدها الخاصة لمزيد من الأمان المستقل والتحقق السريع.

## 9. دمج القيمة وتقسيمها

على الرغم من أنه من الممكن التعامل مع العملات بشكل فردي، سيكون من غير العملي إجراء تحويل منفصل لكل سنت في عملية النقل.، تحتوي التحويلات على العديد من المدخلات والمخرجات، مما يسمح بتقسيم القيمة ودمجها. عادةً ما يكون هناك إما مدخل واحد من تحويل سابق أكبر أو مدخلات متعددة تجمع بين مقادير أصغر، ويوجد مخرجان على الأكثر: أحدهما للدفع، والآخر يعيد المبلغ المتبقي، إن وجد، إلى المرسل.

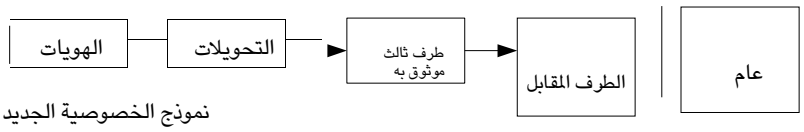



وتجدر الإشارة إلى أن التشعب (Fan-out) – وهي الحالة التي يعتمد التحويل فيها على عدة تحويلات بحيث تعتمد تلك التحويلات على عدد أكبر من التحويلات وهكذا دواليك - لا يمثل مشكلة. ولا توجد حاجة مطلقاً إلى استخراج نسخة مستقلة كاملة من سجل التحويل.

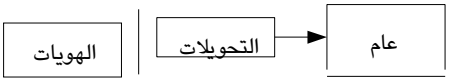
## 10. الخصوصية

يحقق النموذج المصرفي التقليدي مستوىً من الخصوصية من خلال الحد من وصول الأطراف المعنية والطرف الثالث الموثوق للمعلومات. إن ضرورة الإعلان عن كل التحويلات للعامة تعوق هذه الطريقة، ولكن لا يزال من الممكن الحفاظ على الخصوصية من خلال كسر تدفق المعلومات في مكان آخر، اي عن طريق الحفاظ على سرية المفاتيح العامة. ويمكن للعامة أن ترى أن شخصاً ما يرسل مبلغاً لشخص آخر، ولكن دون معلومات تربط التحويل بأي شخص. يشبه ذلك مستوى المعلومات التي تصدرها البورصات، حيث يتم الإعلان عن وقت وحجم الصفقات الفردية - والذي يُعرف باسم "الشريط" - لكن دون الكشف عن الأطراف المعنية.

نموذج الخصوصية التقليدي




نموذج الخصوصية الجديد



يجب استخدام زوج مفاتيح جديد لكل معاملة، باعتباره جدارًا إضافيًا للحماية، وذلك لمنعها من الارتباط بمالك مشترك. لا تزال بعض الروابط لا يمكن تجنبها من خلال تحويلات متعددة المدخلات، والتي تكشف بالضرورة أن مدخلاتها هي للمالك ذاته ويكمن الخطر في أنه في حالة الكشف عن مالك المفتاح، قد يكشف الارتباط عن تحويلات أخرى تعود لمالك ذاته.

## 11. العمليات الحسابية

نحن نضع في عين الاعتبار احتمال المخترق الذي يحاول تكوين سلسلة بديلة أسرع من السلسلة النزيهة. حتى إذا تم تحقيق ذلك، فإنه لا يعرّض النظام لتغييرات تعسفية، مثل تكوين قيمة من لا شيء أو سرقة المخترق لأموال ليست له. ولن تقبل العقد أي تحويل غير صالح كوسيلة للدفع، كما أن العقد النزيهة لن تقبل أي كتلةٍ تحتويها. يمكن للمخترق فقط محاولة تغيير إحدى تحويلاته الخاصة لاسترداد الأموال التي أنفقها مؤخراً.

يمكن وصف السباق بين السلسلة النزيهة وسلسلة المخترقين على أنه مسار عشوائي ذو حدين. الحدث الناجح هو سلسلة نزيهة يتم تمديدها بواسطة كتلة واحدة، مما يزيد من قيادتها بـ +1، والحدث الفاشل هو سلسلة المخترق التي يتم توسيعها بواسطة كتلة واحدة، مما يقلل الفجوة بمقدار -1.

إن احتمال ملاحقة أحد المخترقين من خلال ثغرة ما مشابه لمشكلة إفلاس المقامر .(Gambler's Ruin) لنفترض أن المقامر ذي رصيد الائتمان غير المحدود يبدأ عند عجز ما وأنه يجرب عدداً غير محدود من التجارب لمحاولة الوصول إلى نقطة التعادل. يمكننا حساب احتمال وصوله في أي وقت مضى إلى التعادل، أو أن مخترقاً استطاع اللحاق بالسلسلة النزيهة، على النحو التالي:[8]

p = احتمال أن تجد عقدة نزيهة الكتلة التالية
q = احتمال أن يجد المخترق الكتلة التالية
qz = احتمال أن يستطيع المخترق المجاراة من z كتلة من الخلف

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

وبالنظر إلى افتراضنا بأنp > q ، ينخفض الاحتمال بمعدل أسيّ لأن عدد الكتل التي يتعين على المخترق اللحاق بها يتزايد. مع عدم وجود احتمالات صالحه، فإنه إن لم يحقق تقدمًا محظوظاً للأمام في مرحلة مبكرة، فإن فرصه تتضاءل بشكل كبير مع سقوطه وتراجعه للخلف.

نحن الآن نأخذ بعين الاعتبار المدة التي تلزم مستلم التحويل الجديد انتظارها قبل أن يكون على يقينٍ كافٍ أن المرسل لا يمكنه تغيير التحويل. لنفترض أن المرسل هو مخترق يريد أن يجعل المستلم يعتقد أنه دفع له لفترة من الوقت، ثم يبدله ليسدد لنفسه بعد مرور بعض الوقت. سيتم تنبيه المستلم عندما يحدث ذلك، ولكن المرسل يأمل في أن يتم ذلك بشكل متأخر.

يقوم المستلم بتكوين زوج جديد من المفاتيح ويعطي المفتاح العمومي إلى المرسل قبل التوقيع بوقت قصير .

يمنع هذا المرسل من إعداد سلسلة من الكتل في وقت مبكر من خلال العمل عليها بشكل مستمر حتى يكون محظوظاً بما فيه الكفاية للتقدم بما يكفي إلى الأمام، ثم تنفيذ الصفقة في تلك اللحظة. بمجرد إرسال التحويل، يبدأ المرسل غير النزيه في العمل سراً في سلسلة موازية تحتوي على نسخة بديلة من تحويله.

ينتظر المستلم حتى تتم إضافة المعاملة إلى كتلة وتم ربط عدد z من الكتل بعدها. وهو لا يعرف المقدار الدقيق للتقدم الذي أحرزه المخترق، ولكن بافتراض أن الكتل النزيهة تأخذ متوسط الوقت المتوقع لكل كتلة، فإن التقدم المحتمل للمخترق سيكون توزيع بواسون (Poisson) بالقيمة المتوقعة:

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

لقياس احتمال إمكانية المخترق من المجاراة الآن، فإننا نضرب كثافة بواسون (Poisson) لكل مقدار من التقدم كان من الممكن أن يحققه من خلال احتمال أن يتمكن من المجاراة من هذه النقطة:

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

وعند إجراء إعادة الترتيب لتجنب جمع الذيل اللانهائي للتوزيع ...

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

جارٍ التحويل إلى شفرة 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;
}
```

عند تشغيل بعض النتائج ، يمكننا ملاحظة انخفاض الاحتمالية أُسّياً مع 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

و يكون الحل عندما يكون P أقل من 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. الخاتمة

لقد اقترحنا نظاماً للتحويلات الإلكترونية دون الاعتماد على الثقة. فقد بدأنا بإطار العمل المعتاد للعملات المكونة من التوقيعات الرقمية، والذي يوفر سيطرة قوية على الملكية، لكنه لا يكتمل من دون وسيلة لمنع الإنفاق المضاعف. لحل هذه المشكلة، فقد اقترحنا شبكة النظير للنظير باستخدام نظام إثبات العمل لتسجيل سجل عام للتحويلات التي سرعان ما يستحيل تغييرها من الناحية الحاسوبية من جانب المخترق، وذلك إذا كانت العُقد النزيهة تتحكم في غالبية طاقة وحدة المعالجة المركزية. وتكمن قوة الشبكة في بساطتها غير المهيكلة. وتعمل كل العقد في وقت واحد مع قدر قليل من التنسيق. حيث ليس من الضروري تحديدها، نظراً لأن الرسائل لا يتم توجيهها إلى أي مكان معين، بل تحتاج فقط إلى تسليمها على أساس أفضل جهد. يمكن للعقد أن تغادر الشبكة وتنضم إليها مجدداً متى تشاء، مع قبول أطول سلسلة نظام إثبات للعمل كدليل على ما حدث أثناء غيابها. وهي تقوم بالتصويت باستخدام طاقة وحدة المعالجة المركزية، تعبيراً

عن موافقتها على الكتل الصالحة من خلال العمل على تمديدها ورفض الكتل غير الصالحة من خلال رفض العمل عليها. يمكن تطبيق أية قواعد وحوافز مطلوبة بواسطة هذه الآلية المتفق عليها.

# المراجع


[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.