بيتكوين: نظام نقدي إلكتروني قائم على مبدأ النظير إلى النظير

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

الملخص: إن إصدار نسخة من نظام نقدي إلكتروني قائم كلياً على مبدأ النظير إلى النظير، من شأنه أن يسمح بإرسال المدفوعات عبر الإنترنت مباشرة من طرف إلى آخر دون المرور عبر مؤسسة مالية. توفر التوقيعات الرقمية جزءًا من الحل، ولكن الفوائد الأساسية تضيع إذا استمرت الحاجة للثقة بطرف ثالث لمنع الإنفاق المزدوج Double Spending. نقترح حلاً لمشكلة الإنفاق المزدوج باستخدام شبكة قائمة على النظير إلى النظير Peer-to-Peer. تقوم الشبكة بختم المعاملات زمنيًا من خلال دمجها في سلسلة مستمرة من إثبات العمل Proof-of-Work القائم على الهاش Hash أو التعمية أحادية الاتجاه، مما يشكل سجلاً لا يمكن تغييره دون إعادة إنتاج إثبات للعمل. لا تعمل السلسلة الأطول كإثبات على تسلسل الأحداث التي شوهدت فحسب، بل تُثبت أيضاً أنها أنتجت من قبل أكبر حوض من قدرة وحدات المعالجة المركزية CPU. طالما أن غالبية القدرة الحسابية تخضع لسيطرة عُقد لا تتعاون لمهاجمة الشبكة، فإنها ستولد أطول سلسلة وتتفوق على المهاجمين. تتطلب الشبكة نفسها بنية بسيطة. يتم بث الرسائل على شبكة البيتكوين وفقًا للطريقة المثلى التي يمكن لكل عُقدة تحقيقها، ويمكن للعُقد مغادرة الشبكة والانضمام إليها متى شاءت، وقبول أطول سلسلة إثبات للعمل كدليل على ما حدث أثناء غيابها.

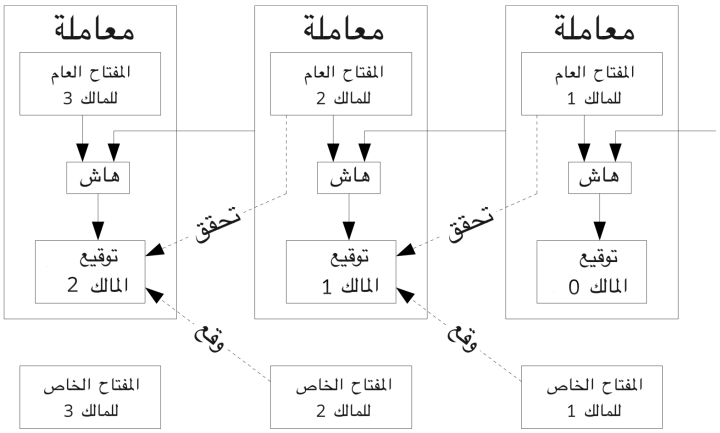
1. المقدمة

لقد أصبحت التجارة على الإنترنت تعتمد بشكل شبه حصري على المؤسسات المالية التي تعمل كأطراف ثالثة تتطلب الثقة لمعالجة المدفوعات الإلكترونية. وفي حين يعمل هذا النظام بشكل جيد بما فيه الكفاية لمعظم المعاملات، فإنه لا يزال يعاني من نقاط الضعف المتأصلة في نموذج الثقة. فعلى سبيل المثال لا يمكن ضمان عدم إلغاء أي معاملة، لأن المؤسسات المالية لا تستطيع تجنب التوسط في النزاعات. تزيد تكلفة الوساطة من تكاليف المعاملات، وتقيد من الحد الأدنى العملي لحجم المعاملات وتمنع إمكانية تنفيذ المعاملات العادية الصغيرة، وهناك تكلفة أوسع في فقدان القدرة على إجراء مدفوعات غير قابلة للإلغاء مقابل خدمات غير قابلة للإلغاء. مع إمكانية الإلغاء، تنتشر الحاجة إلى الثقة. يجب على التجار أن يكونوا حذرين من عملائهم، ومطالبتهم بمعلومات أكثر مما يحتاجون إليه فعلياً. يتم قبول نسبة معينة من الاحتيال على أنها أمر لا مفر منه. إن هذه التكاليف وعدم اليقين في الدفع يمكن تجنبهما باستخدام طرق الدفع نقداً يداً بيد، ولكن لا توجد آلية لإجراء المدفوعات عبر قناة اتصالات بدون وجود طرف ثالث يتطلب الثقة.

إن المطلوب هو بناء نظام دفع إلكتروني يعتمد على الإثبات التشفيري بدلاً من الثقة، مما يسمح لأي طرفين راغبين في التعامل مباشرة مع بعضهما البعض دون الحاجة إلى طرف ثالث موثوق به. إن المعاملات التي يصعب عكسها من الناحية الحسابية ستحمي البائعين من الاحتيال، ويمكن بسهولة تنفيذ آليات ضمان روتينية لحماية المشترين. في هذه الورقة، نقترح حلاً لمشكلة الإنفاق المزدوج باستخدام خادم زمني موزع قائم على النظير إلى النظير لتوليد إثباتات كريبتوجرافية للترتيب الزمني للمعاملات. يظل النظام آمناً طالما أن العُقد النزيهة تتحكم مجتمعةً بقدرة معالجة أكبر من أي مجموعة متعاونة من العُقد المهاجمة.

2. المعاملات

نُعرّف العملة الإلكترونية بأنها سلسلة من التوقيعات الرقمية. ينقل كل مالك العملة إلى المالك التالي من خلال التوقيع رقميًا على هاش المعاملة السابقة والمفتاح العام للمالك التالي وإضافتهما إلى معلومات العملة. يمكن للمستفيد التحقق من صحة التوقيعات للتحقق من سلسلة الملكية.

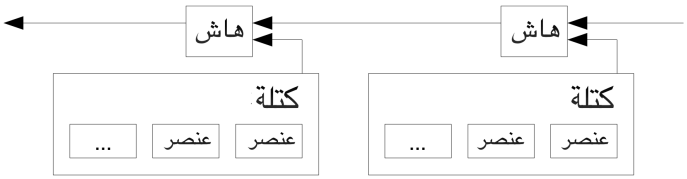


المشكلة بالطبع هي أن المستفيد لا يستطيع التحقق من أن أحد المالكين لم ينفق العملة مرتين. الحل الشائع هو إدخال سلطة مركزية تتطلب الثقة، أو جهة مسؤولة عن صك العملة، للتحقق من كل معاملة بحثًا عن الإنفاق المزدوج. وبعد كل معاملة، يجب إعادة العملة إلى ذات الجهة لإصدارها من جديد، وتكون العملات الصادرة مباشرة من ذات الجهة هي الوحيدة الموثوق أنها لم تُنفق مرتين. المشكلة في هذا الحل هي أن مصير هذا النظام النقدي بالكامل يعتمد على الشركة التي تدير الصك، حيث يتعين على كل معاملة أن تمر عبره، تمامًا مثل البنك.

نحتاج إلى طريقة ليعرف المستفيد الحالي أن المالكين السابقين لم يوقعوا على أي معاملات سابقة لمستفيدين آخرين لأن المعاملة الأولى هي التي تُحتسب، ولا نهتم بمحاولات الإنفاق المزدوج اللاحقة. الطريقة الوحيدة لتأكيد عدم وجود معاملة ما هي أن تكون على علم بجميع المعاملات السابقة. في النموذج القائم على الثقة بمُصدر الصك، تكون هذه الجهة على علم بجميع المعاملات وهي من يقرر أيها وصلت أولاً. لتحقيق ذلك دون الاعتماد على طرف موثوق، يجب الإعلان عن المعاملات علنًا [1]، ونحتاج إلى نظام يتيح للمشاركين الاتفاق على ماضي واحد موحد لترتيب استلام هذه المعاملات. كما يحتاج المستفيد إلى إثبات أنه في وقت إجراء كل معاملة، اتفقت أغلبية العُقد على أنها أول معاملة تم استلامها.

3. خادم الختم الزمني

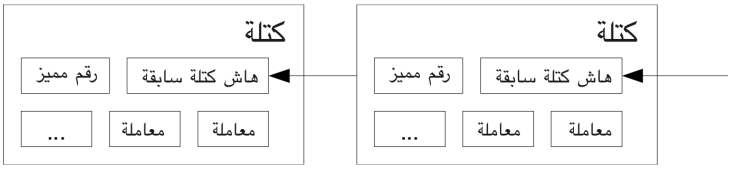
يبدأ الحل الذي نقترحه بخادم للختم الزمني، حيث يقوم بأخذ هاش (بصمة رقمية) لكتلة من البيانات (المعاملات) لتحديد وقتها ثم نشر هذا الهاش على نطاق واسع، كنشرها في جريدة أو في منشور على منتدى مثل يوزنيت Usenet [2-5]. يثبت الختم الزمني أن البيانات كانت موجودة بالفعل في ذلك الوقت، لأنها مضافة ضمن الهاش. سيحتوي كل ختم زمني على معلومات الختم الذي سبقه ضمن الهاش، مما يشكل سلسلة متصلة من الأختام الزمنية التي يقوم كل ختم إضافي فيها بتعزيز ما سبقه.



4. إثبات العمل

لتنفيذ خادم الختم الزمني الموزع على أساس النظير إلى النظير، سنحتاج إلى استخدام نظام إثبات العمل المشابه لنظام هاشكاش Hashcash الذي ابتكره آدم باك[6]، بدلاً من الصحف أو منشورات يوزنيت. يتضمن إثبات العمل البحث عن قيمة معينة، عندما يتم تجزئتها (على سبيل المثال باستخدام خوارزمية SHA-256)، تبدأ شفرة الهاش الناتجة بعدد محدد من البتات الصفرية. يكون متوسط العمل المطلوب أسيًا بالنسبة لعدد البتات الصفرية المطلوبة ويمكن التحقق من صحته بتنفيذ عملية تجزئة واحدة.

بالنسبة لشبكة الختم الزمني الخاصة بنا، ننفذ إثبات العمل عن طريق زيادة تدريجية لرقم مميز nonce في الكتلة حتى يتم العثور على قيمة تمنح هاش الكتلة البتات الصفرية المطلوبة. وبمجرد بذل الجهد الحاسوبي لتحقيق إثبات العمل، لا يمكن تغيير الكتلة دون إعادة بذل العمل من جديد. إعادة بذل العمل. نظرًا لأن الكتل اللاحقة متسلسلة بعدها، فإن العمل لتغيير الكتلة سيتضمن إعادة إنتاج جميع الكتل بعدها.



يحل إثبات العمل أيضًا مشكلة تحديد التمثيل في اتخاذ القرار بالأغلبية. فلو كان التصويت يعتمد على مبدأ "صوت واحد لكل عنوان IP"، لأمكن التلاعب به من قبل أي شخص قادر على تخصيص عناوين IP متعددة. أما في نظام إثبات العمل فإن وحدة معالجة مركزية واحدة تمثل صوتاً واحد. يتم تمثيل قرار الأغلبية بواسطة أطول سلسلة، والتي لديها أكبر جهد إثبات عمل مستثمر فيها. فإذا كانت أغلبية قوة المعالجة تحت سيطرة عُقد نزيهة، فستنمو السلسلة النزيهة بأسرع معدل وتتفوق على أي سلاسل منافسة. لتعديل كتلة سابقة، يجب على المهاجم إعادة إثبات عمل الكتلة وجميع الكتل بعدها ثم اللحاق بعمل العُقد النزيهة وتجاوزه. سنوضح لاحقًا أن احتمالية اللحاق بمهاجم أبطأ تتضاءل بشكل كبير مع إضافة كتل لاحقة.

للتعويض عن زيادة سرعة الأجهزة والاهتمام المتغير بتشغيل العُقد بمرور الوقت، يتم تحديد صعوبة إثبات العمل من خلال متوسط متحرك يستهدف متوسط عدد الكتل في الساعة. إذا كان يتم إنشاؤها بسرعة كبيرة، فستزداد الصعوبة.

5. الشبكة

الخطوات اللازمة لتشغيل الشبكة هي كما يلي:

1) يتم بث المعاملات الجديدة إلى جميع العُقد.
2) تقوم كل عقدة بجمع المعاملات الجديدة في كتلة.
3) تعمل كل عقدة على إيجاد رمز إثبات عمل صعب لكتلتها.
4) عندما تجد عقدة ما رمز إثبات عمل، فإنها تبث الكتلة إلى جميع العُقد.
5) تقبل العُقد الكتلة فقط إذا كانت جميع المعاملات فيها صالحة ولم يتم إنفاقها بالفعل.
6) تعبّر العُقد عن قبولها للكتلة من خلال العمل على إنشاء الكتلة التالية في السلسلة، باستخدام هاش الكتلة المقبولة كهاش سابقة.

تعتبر العُقد دائمًا السلسلة الأطول هي السلسلة الصحيحة وتواصل العمل على تمديدها. وإذا قامت عُقدتان ببث نسختين مختلفتين من الكتلة التالية في نفس الوقت، فقد تتلقى بعض العُقد إحداهما أو الأخرى أولاً. في هذه الحالة، تعمل العُقد على النسخة التي استلمتها أولاً، لكنها تحتفظ بالفرع الآخر احتياطياً في حال أصبح أطول. ويتم حسم التعادل عند اكتشاف إثبات العمل التالي ويصبح أحد الفرعين أطول؛ عندها تتحول العُقد التي كانت تعمل على الفرع الآخر إلى الفرع الأطول.

لا يلزم بالضرورة أن تصل المعاملات الجديدة المُعلن عنها إلى جميع العُقد. فطالما وصلت إلى عدد كبير من العُقد، فستُدرج في كتلة قريباً. كما أن عملية بث تتحمل فقدان بعض الرسائل. فإذا لم تتلقَّ عقدة معينة كتلة ما، ستقوم بطلبها تلقائيًا عند تلقيها الكتلة التالية، مما يتيح لها اكتشاف الفجوة وتعويض الكتلة المفقودة.

6. الحافز

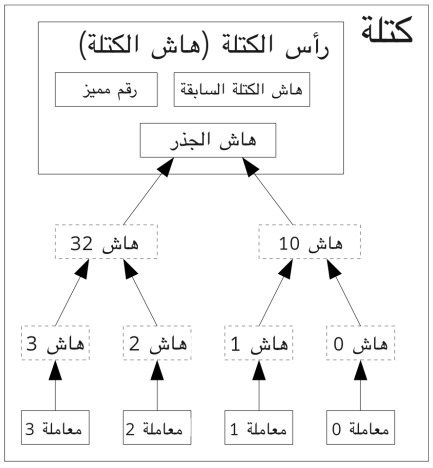
وفقًا للميثاق، فإن أول معاملة في كتلة ما هي معاملة خاصة تُصدر عملة جديدة يتملكها منشئ الكتلة. وهذا يضيف حافزًا للعقد لدعم الشبكة، ويوفر طريقة لتوزيع العملات في البداية للتداول، نظرًا لعدم وجود سلطة مركزية لإصدارها. إن الإضافة الثابتة لكمية ثابتة من العملات الجديدة تشبه معدني الذهب الذين ينفقون الموارد لإضافة الذهب إلى التداول. في حالتنا، يتم إنفاق وقت تشغيل وحدات المعالجة المركزية والكهرباء.

يمكن أيضًا تمويل الحافز برسوم المعاملات. فإذا كانت قيمة مخرجات المعاملة أقل من قيمة مدخلاتها، يُضاف الفرق كرسوم معاملة إلى قيمة الحافز للكتلة التي تحتوي على المعاملة. بمجرد دخول عدد محدد مسبقًا من العملات إلى التداول، يمكن أن يتحول الحافز بالكامل إلى رسوم المعاملات ويكون خاليًا تمامًا من التضخم.

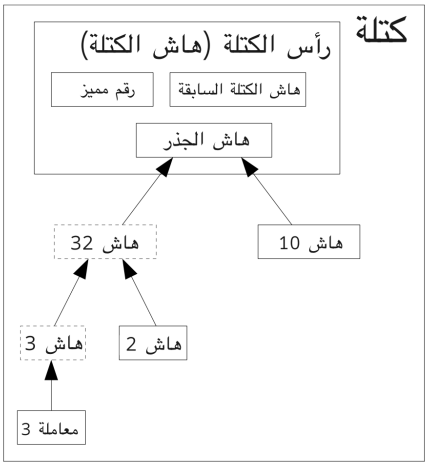
قد يساعد الحافز في تشجيع العُقد على البقاء صادقة. إذا كان المهاجم الجشع قادرًا على تجميع طاقة حاسوبية أكثر من جميع العُقد الصادقة، فسيتعين عليه الاختيار بين استخدامها لخداع الناس عن طريق سرقة (إلغاء) مدفوعاته، أو استخدامها لتوليد عملات جديدة. من المنطقي له أن يجد بأن الالتزام بالقواعد أكثر ربحاً، حيث تمنحه هذه القواعد عملات جديدة تفوق ما يحصل عليه الجميع مجتمعين، بدلاً من تقويض النظام وإضعاف قيمة ثروته الخاصة.

7. استعادة مساحة القرص

بمجرد دفن أحدث معاملة في عملة ما تحت عدد كافي من الكتل، يمكن التخلص من المعاملات المنفقة قبلها لتوفير مساحة القرص. لتسهيل ذلك دون كسر هاش الكتلة، يتم إدخال المعاملات بهاش شجرة ميركل [7][2][5]، مع تضمين الجذر فقط في هاش الكتلة. يمكن بعد ذلك تقليص الكتل القديمة عن طريق إزالة فروع الشجرة. ليس من المستلزم تخزين الهاشات الداخلية.



معاملات يتم دمجها داخل شجرة ميركل

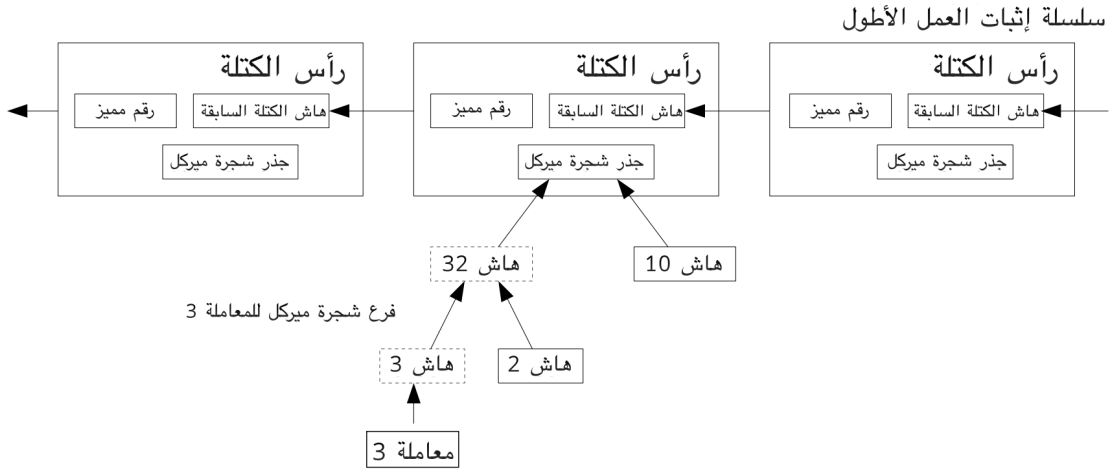


بعد تقليم المعاملات (0 إلى 2) من الكتلة

إن رأس الكتلة الفارغة التي لا تحتوي على معاملات سيكون بحجم 80 بايت تقريبًا. وإذا افترضنا أن الكتل يتم إنشاؤها كل 10 دقائق، فإن 80 بايت * 6 * 24 * 365 = 4.2 ميجا بايت سنويًا. ومع بيع أنظمة الكمبيوتر عادةً بذاكرة عشوائية RAM بحجم 2 جيجابايت اعتبارًا من عام 2008، وتوقع قانون مور نموًا حاليًا بمقدار 1.2 جيجابايت سنويًا، فلا ينبغي أن يكون التخزين مشكلة حتى إذا كان من الضروري الاحتفاظ برؤوس الكتل في الذاكرة.

8. التحقق المبسط من الدفع

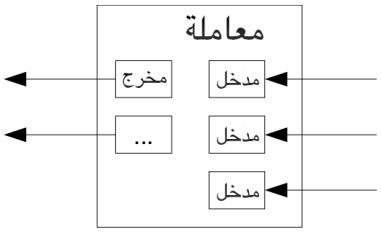
من الممكن التحقق من الدفعات دون تشغيل عقدة شبكة كاملة. يحتاج المستخدم فقط إلى الاحتفاظ بنسخة من رؤوس الكتل لأطول سلسلة إثبات عمل، والتي يمكنه الحصول عليها من خلال الاستعلام من عقد الشبكة حتى يقتنع بأن لديه أطول سلسلة، والحصول على فرع من شجرة ميركل والذي يربط المعاملة بالكتلة التي تم ختمها فيها. لا يمكنه التحقق من المعاملة بنفسه، ولكن من خلال ربطها بمكان في السلسلة، يمكنه معرفة أن عقدة الشبكة قد قبلتها، والكتل المضافة بعد ذلك تؤكد أن الشبكة قد قبلتها.



وعلى هذا النحو، فإن التحقق موثوق به طالما أن العُقد الصادقة تتحكم في الشبكة، ولكنه أكثر عرضة للخطر إذا تغلب المهاجم على الشبكة. في حين أن عقد الشبكة يمكنها التحقق من المعاملات بنفسها، يمكن خداع التحقق المبسط من خلال معاملات ملفقة للمهاجم طالما كان المهاجم قادرًا على الاستمرار في التغلب على الشبكة. تتمثل إحدى الاستراتيجيات للحماية من هذا في قبول التنبيهات من عقد الشبكة عندما تكتشف كتلة غير صالحة، مما يدفع برنامج المستخدم إلى تنزيل الكتلة الكاملة و المعاملات المعنية لتأكيد التناقض. من المحتمل أن ترغب الشركات التي تتلقى مدفوعات متكررة في تشغيل عقدها الخاصة لمزيد من الأمان المستقل والتحقق السريع.

9. الجمع بين القيمة وتقسيمها

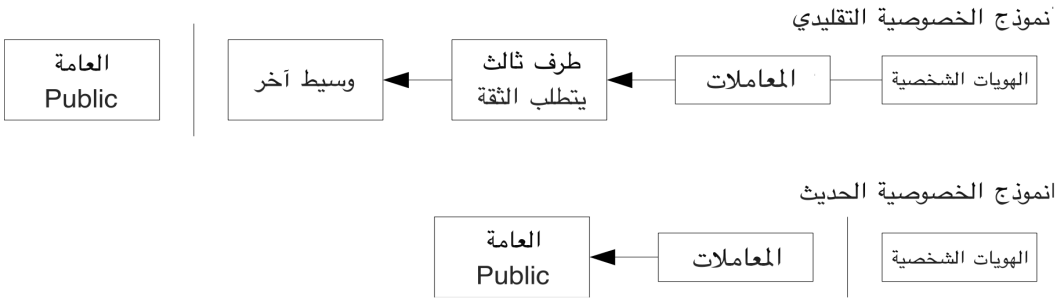
على الرغم من أنه سيكون من الممكن التعامل مع العملات بشكل فردي، إلا أنه سيكون من الصعب إجراء معاملة منفصلة لكل سنت في التحويل. للسماح بتقسيم القيمة ودمجها، تحتوي المعاملات على مدخلات ومخرجات متعددة. عادة سيكون هناك إما مدخل واحد من معاملة سابقة أكبر أو مدخلات متعددة تجمع بين مبالغ أصغر، ومخرجان على الأكثر: واحد للدفع، والآخر يعيد الباقي، إن وجد، إلى المرسل.



يجدر الإشارة إلى أن التفرع الكلي، حيث تعتمد معاملة على عدة معاملات أخرى، وتلك المعاملات تعتمد على العديد غيرها، ليست مشكلة هنا. لا توجد حاجة أبداً لاستخراج نسخة مستقلة وكاملة من تاريخ المعاملة.

9. الخصوصية

إن النموذج المصرفي التقليدي يحقق مستوى من الخصوصية من خلال تقييد الوصول إلى المعلومات على الأطراف المعنية والطرف الثالث الموثوق به. إن ضرورة الإعلان عن جميع المعاملات علناً تمنع هذه الطريقة، ولكن الخصوصية لا تزال ممكنة من خلال قطع تدفق المعلومات في مكان آخر: من خلال إبقاء المفاتيح العامة مجهولة الهوية. يمكن للجمهور أن يرى أن شخصاً ما يرسل مبلغاً إلى شخص آخر، ولكن دون معلومات تربط المعاملة بهوية أي شخص. وهذا يشبه مستوى المعلومات التي تصدرها البورصات، حيث يتم الكشف عن وقت وحجم الصفقات الفردية، أو مايسمى "شريط التداول"، ولكن دون الكشف عن هوية الأطراف.



كجدار حماية إضافي، ينبغي استخدام زوج مفاتيح جديد لكل معاملة لمنع ربطها بمالك مشترك. لكن يظل بعض الربط أمرًا لا مفر منه في المعاملات متعددة المدخلات، والتي تكشف بالضرورة أن مدخلاتها كانت مملوكة من قبل نفس المالك. يكمن الخطر في أنه إذا تم الكشف هوية مالك أحد المفاتيح، فقد يؤدي الربط إلى الكشف عن معاملات أخرى تعود لنفس المالك.

11. الإثباتات الرياضية

لننظر في سيناريو يحاول فيه مهاجم إنشاء سلسلة بديلة أسرع من السلسلة النزيهة. وحتى إذا تم إنجاز ذلك، فإنه لا يجعل النظام مفتوحًا للتغييرات التعسفية، مثل خلق القيمة من الهواء أو أخذ أموال لم تكن ملكًا للمهاجم. لن تقبل العُقد معاملة غير صالحة كدفعة صالحة، ولن تقبل العُقد الصادقة أبدًا كتلة تحتوي معاملات غير صالحة. لا يمكن للمهاجم إلا محاولة تغيير إحدى معاملاته الخاصة لاستعادة الأموال التي أنفقها مؤخرًا.

يمكن وصف السباق بين السلسلة الصادقة وسلسلة المهاجم على أنه عملية عشوائية ثنائية الحد Binomial Random Walk. حدث النجاح هو عندما يتم تمديد السلسلة الصادقة بكتلة واحدة، مما يزيد تقدمها بمقدار +1، وحدث الفشل هو عندما يتم تمديد سلسلة المهاجم بكتلة واحدة، مما يقلل الفجوة بمقدار -1.

تشابه احتمالية أن يلحق المهاجم بالسلسلة الصادقة عند أي عجز معين مع مسألة 'إفلاس المقامر' Gambler's Ruin Problem. لنفترض أن مقامرًا لديه ائتمان غير محدود يبدأ بعجز ويحاول، من خلال عدد غير محدود من التجارب، الوصول إلى نقطة التعادل. يمكننا حساب احتمالية أن يصل إلى التعادل، أو أن يلحق المهاجم بالسلسلة الصادقة، كما يلي[8]:

p = احتمالية أن تجد العُقدة الصادقة الكتلة التالية

q = احتمالية أن يجد المهاجم الكتلة التالية

q_z = احتمال أن يلحق المهاجم بالسلسلة الصادقة عندما يكون متأخراً بـ z كتل

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

وبناءً على افتراضنا بأن $p > q$، فإن الاحتمال ينخفض بشكل كبير مع زيادة عدد الكتل التي يتعين على المهاجم اللحاق بها. ومع وجود احتمالات ضده، إذا لم يقم بالاندفاع المحظوظ إلى الأمام في وقت مبكر، فإن فرصه تصبح ضئيلة للغاية مع تأخره أكثر فأكثر.

ننظر الآن إلى المدة التي يحتاجها المتلقي لمعاملة جديدة للانتظار قبل أن يتأكد بشكل كافٍ من أن المرسل لا يستطيع تغيير المعاملة. نفترض أن المرسل هو مهاجم يريد أن يجعل المستلم يعتقد لفترة معينة أنه قد دفع له النقود، ثم يقوم بتغيير وجعة الدفعة إلى نفسه بعد مرور بعض الوقت. سيتم تنبيه المستلم عندما يحدث ذلك، لكن المرسل يأمل أن يكون الأوان قد فات.

يقوم المستلم بإنشاء زوج جديد من المفاتيح ويعطي المفتاح العام للمرسل قبل التوقيع بفترة وجيزة. يمنع هذا المرسل من تحضير سلسلة من الكتل مسبقاً بالعمل عليها باستمرار حتى يحالفه الحظ ويصل إلى تقدم كافٍ، ثم يتم تنفيذ المعاملة في تلك اللحظة. بمجرد إرسال المعاملة، يبدأ المرسل غير الصادق في العمل سراً على سلسلة موازية تحتوي على نسخة بديلة من معاملته.

ينتظر المستلم حتى يتم إضافة المعاملة إلى كتلة وترتبط بعدها z من كتل. لا يعرف المستلم مقدار التقدم الذي حققه المهاجم بالضبط، لكن بافتراض أن الكتل الصادقة قد أخذت الوقت المتوسط المتوقع لكل كتلة، سيكون التقدم المحتمل للمهاجم موزعاً حسب توزيع بواسون Poison Distribution مع القيمة المتوقعة التالية:

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

للحصول على احتمالية أن يتمكن المهاجم من اللحاق بنا الآن، نضرب كثافة بواسون لكل مقدار من التقدم الذي كان من الممكن أن يحرزه في احتمالية أن يتمكن من اللحاق بنا من تلك النقطة:

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

نعيد ترتيب المعادلة لتجنب الحاجة إلى جمع الذيل اللانهائي للتوزيع، مما يبسط الحسابات ويجنب التعامل مع عدد لا نهائي من القيم ذات الاحتمالات الصغيرة

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

تحويل المعادلة إلى كود 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 تحليل النتائج، نلاحظ أن الاحتمالية تتناقص بشكل أُسي مع زيادة z، مما يشير إلى أن الفرصة تتناقص أسياً 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

نجد الحلول لـ P <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

11. الخاتمة

لقد اقترحنا نظامًا للمعاملات الإلكترونية لا تتطلب الاعتماد على الثقة. لقد بدأنا بالإطار المعتاد للعملات التي تعتمد على التوقيعات الرقمية، والتي توفر تحكمًا قويًا في الملكية، ولكنها غير مكتملة بدون طريقة لمنع الإنفاق المزدوج. لحل هذه المشكلة، اقترحنا شبكة قائمة على مبدأ النظير إلى النظير باستخدام إثبات العمل لتسجيل تاريخ عام للمعاملات التي سرعان ما تصبح غير عملية من الناحية الحسابية بالنسبة للمهاجم أن يغيرها إذا كانت العُقد الصادقة تتحكم في غالبية الطاقة الحاسوبية. الشبكة قوية في ببساطتها الغير منظمة. تعمل العُقد كلها في تزامن مع القليل من التنسيق. لا تحتاج العُقد إلى الكشف عن هويتها، حيث لا يتم توجيه الرسائل إلى أي مكان معين ولا تحتاج إلا إلى الاستلام وفقًا لأفضل جهد يمكن لكل عقدة تحقيقه. يمكن للعُقد مغادرة الشبكة والانضمام إليها متى شاءت، وقبول سلسلة إثبات العمل كدليل على ما حدث أثناء غيابها. تصوت العُقد بقوة وحدة المعالجة المركزية الخاصة بها، معبرة عن قبولها للكتل الصالحة من خلال العمل على بناءها ورفض الكتل غير الصالحة من خلال رفض العمل عليها. يمكن فرض أي قواعد وحوافز ضرورية من خلال آلية الإجماع هذه.

12. 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.

ترجمة وإعداد:
باسم العاصي @Bamskki
مراجعة:
عمر محمد @Omer_Saeed
عربي @Arabic_hodl
شكر خاص لمجتمع برق نت t.me/Barqnet
Bitcoin21.io
Bitcoinarabic.org