البيتكوين: نظام نقد الكتروني قائم على مبدأ الند للند

by Satoshi Nakamoto 2008/10/31

مُلخص

نسخة ند إلى ند على نحو محض من النقد الإلكتروني سوف يسمح للمدفوعات عبر الإنترنت أن ترسل مباشرة من طرف إلى آخر دون المرور عبر مؤسسة مالية. التوقيعات الرقمية توفر جزءًا من الحل ، ولكن يتم فقدان الفوائد الرئيسية إذا كان لا يزال مطلوبًا طرف ثالث موثوق به لمنع الإنفاق المزدوج. نقترح حلاً لمشكلة الإنفاق المزدوج باستخدام شبكة الند للند. تقوم هذه الشبكة بعمل ختومات زمنية لكل معاملة عن طريق دمجها في سلسلة مستمرة من إثبات العمل القائم على هاش، مشكلا بذلك سجلا لا يمكن تغييره دون إعادة عمل إثبات العمل. لا تعمل أطول سلسلة كدليل على تسلسل الأحداث التي شهدتها فحسب ، بل إنها دليل على أنها جاءت من أكبرتجمع لقوة وحدة المعالجة المركزية. طالما يتم التحكم في معظم قوة وحدة المعالجة المركزية من خلال العقد الصادقة ، فإنها سوف تولد أطول سلسلة و تتفوق على المهاجمين. الشبكة نفسها تتطلب حد أدنى من البنية. كما يتم بث الرسائل على أساس أفضل جهد ، ويمكن للعقد أن تغادر وتعيد الانضمام إلى الشبكة متى شاءت ، وبقبول أطول سلسلة إثبات للعمل يعتبر كدليل على ما حدث أثناء اختفائها.

المقدمة

أصبحت التجارة على الإنترنت تعتمد بشكل شبه حصري على المؤسسات المالية التي تعمل كجهات خارجية موثوقة لمعالجة المدفوعات الإلكترونية. في حين أن النظام يعمل بشكل جيد بما فيه الكفاية لمعظم المعاملات ، فإنه لا يزال يعاني من نقاط الضعف المتأصلة في نموذج يستند إلى الثقة. المعاملات الغير قابلة للتراجع بالكامل غير ممكنة ، حيث لا يمكن للمؤسسات المالية تجنب التوسط في النزاعات. تكلفة الوساطة تزيد من تكاليف المعاملات ، مما يحد من الحد الأدنى لحجم المعاملات العملية ويقلل من إمكانية إجراء معاملات عارضة صغيرة ، وهناك تكلفة أوسع نطاقاً في خسارة القدرة على تقديم مدفوعات غير قابلة للتراجع للخدمات غير القابلة للإلغاء. ومع إمكانية التراجع، فإن الحاجة إلى الثقة تنتشر. يجب أن يكون التجار حذرين من عملائهم ، وهذا يكلفهم مزيدًا من المعلومات أكثر مما يحتاجون إليه. يتم قبول نسبة معينة من الاحتيال على أنها لا يمكن تجنبها. يمكن تجنب هذه التكاليف ومشاكل الدفع الغير مؤكدة باستخدام عملة مادية شخصيًا ،. ولكن لا توجد آلية لإجراء الدفع عبر قناة اتصالات دون وجود طرف موثوق به

فما هو مطلوب هو نظام دفع إلكتروني مستند إلى دليل تشفير بدلاً من الثقة ، مما يسمح لأي طرفين راغبين بالتعامل مباشرة مع بعضهما البعض دون الحاجة إلى طرف ثالث موثوق به.

إن المعاملات التي لا يمكن إلغائها عمليًا من الناحية الحسابية تحمي البائعين من الاحتيال ، ويمكن بسهولة تنفيذ آليات الضمان الروتينية لحماية المشترين. في هذا البحث ، نقترح حلاً لمشكلة الإنفاق المزدوج باستخدام خادم طابع زمني موزع من ند إلى ند لإنشاء إثبات حاسوبي مرتب زمنيا للمعاملات. النظام آمن طالما أن العقد الصادقة تتحكم بشكل جماعي في طاقة وحدة المعالجة المركزية أكثر من أي مجموعة متعاونة من العقد المهاجمة.

المعاملات

نُعَرف أي عملة إلكترونية على أنها سلسلة من التوقيعات الرقمية. يقوم كل مالك بتحويل العملة إلى التالي من خلال التوقيع رقمياً على هاش المعاملة السابقة والمفتاح العمومي للمالك القادم وإضافتهم إلى نهاية العملة. يمكن للمستفيد التحقق من التوقيعات للتأكد من سلسلة الملكية.

المشكلة بالطبع هي أن المستلم لا يمكنه التحقق من أن أحد المالكين لم ينفق العملة مرتين (بشكل مزدوج). الحل الشائع هو تقديم سلطة مركزية موثوق بها أو مصنع صك عملة يقوم بالتحقق من كل معاملة لمنع الإنفاق المزدوج. بعد كل معاملة ، يجب أن تعاد العملة إلى مصنع صك العملة لإصدار عملة جديدة ، ولا يوثق إلا بالعملات التي تصدر مباشرة من مصنع صك العملة للتأكد من عدم إنفاقها أكثر من مرة. تكمن المشكلة في هذا الحل في أن مصير النظام المالي بأكمله يعتمد على الشركة التي تدير مصنع صك العملة ، كل معاملة يجب أن تمر بها ، تمامًا مثل البنك.

نحن بحاجة إلى وسيلة للمستلم أن يعلم أن المالكين السابقين لم يوقعوا على أي معاملات سابقة. هدفنا هو أن تكون المعاملة السابقة هي المعاملة التي يتم احتسابها ، لذا فنحن لا نهتم بالمحاولات اللاحقة للإنفاق المزدوج . الطريقة الوحيدة لتأكيد فقدان معاملة هي أن تكون على علم بجميع المعاملات. في النموذج القائم على مصنع صك العملة ، كان مصنع صك العملة على علم بجميع المعاملات ويمكنه تحديد المعاملات التي حصلت أولاً. لإنجاز هذا دون طرف موثوق به ، يجب الإعلان عن المعاملات بشكل علني [1]، كما نحن بحاجة إلى نظام للمشاركين للاتفاق على تاريخ واحد للترتيب الذي تم به إستلام المعاملات. ويحتاج المستلم إلى إثبات أنه في وقت كل معاملة، وافقت أغلبية العقد على أنها المرة الأولى التي تم استلام هذه المعاملة فيها.

خادم الطابع الزمني

يبدأ الحل الذي نقترحه بخادم الطابع الزمني. يعمل خادم الطابع الزمني على أخذ هاش لكتلة من العناصر ليتم ختمها زمنيا ونشرها على نطاق واسع ، كما هو الحال في الصحف أو بريد المجموعات الأخبارية[2–5]. من الواضح ، ومن أجل الوصول إلى سلسلة هاش ، يثبت الطابع الزمني أن البيانات يجب أن تكون موجودة في ذلك الوقت. يتضمن كل طابع زمني الطابع الزمني السابق له في الهاش ، مشكلا سلسلة ، و كل طابع زمني إضافي يعزز ذلك الذي قبله.

إثبات العمل

لتنفيذ خادم طابع زمني موزع يقوم على مبدأ الند للند، سوف نحتاج إلى نظام إثبات العمل، وهو نظام مشابه ل (Adam Back’s Hashcash) [6] بدلا من الصحف أو بريد المجموعات الأخبارية.

إثبات العمل يشمل البحث عن قيمة هاش، كاستخدام خوارزمية هاش SHA-256، قيمة هاش هذه تبدأ بالعديد من البتات الصفرية. علاقة متوسط العمل المطلوب و عدد البتات الصفرية المطلوبة هي علاقة أسيّة ويمكن التحقق منها عن طريق تنفيذ عملية هاش واحدة

بالنسبة إلى شبكة الطابع الزمني الخاصة بنا ، نقوم بتطبيق إثبات العمل عن طريق زيادة قيمة مؤقتة في

الكتلة حتى يتم العثور على قيمة تعطي هاش الكتلة البتات الصفرية المطلوبة. بمجرد أن يتم إنفاق جهد وحدة المعالجة المركزية لجعله يستوفي متطلبات إثبات العمل ، لا يمكن تغيير الكتلة دون إعادة العمل. كما يتم تقييد الكتل التالية بعد ذلك ، فإن العمل على تغيير الكتلة سيشمل إعادة عمل كل الكتل التي بعدها.

إثبات العمل يحل أيضًا مشكلة توضيح القرار في عملية اتخاذ القرار بالأغلبية. إذا كانت الأغلبية مستندة على IPمبدأ صوت واحد لكل عنوان واحد ، فقد يتم تخريبها من قبل أي شخص قادر على تخصيص العديد من IP. عناوين و إثبات العمل هو في الأساس صوت واحد لكل وحدة معالجة مركزية واحدة. يتم تمثيل قرار الأغلبية بأطول سلسلة ، والتي لديها أكبر قدر من إثباتات العمل المستثمرة فيها. إذا تم التحكم بأغلبية قوة وحدة المعالجة المركزية بواسطة عقد صادقة ، فإن السلسلة الصادقة ستنمو أسرع وتتفوق على أي سلاسل منافسة. لتعديل كتلة سابقة ، يجب على المهاجم أن يعيد إثبات عمل الكتلة وكل الكتل التي بعدها ثم يلحق ب ويتفوق على عمل العقد الصادقة. سنظهر لاحقًا أن احتمالية اللحاق لمهاجم أبطأ، تتضاءل بشكل كبير مع إضافة الكتل التالية. لتعويض الأثر الناتج عن زيادة سرعة الهاردوير والاهتمام المتباين بتشغيل العقد مع مرور الوقت ، يتم تحديد صعوبة إثبات العمل بمتوسط متحرك يستهدف متوسط عدد الكتل في الساعة. إذا تم إنشاء الكتل بسرعة كبيرة ، تزداد الصعوبة.

الشبكة

خطوات تشغيل الشبكة هي كما يلي:

1. يتم بث المعاملات الجديدة لجميع العقد.
2. كل عقدة تجمع المعاملات الجديدة في كتلة.
3. تعمل كل عقدة على العثور على إثبات عمل صعب لكتلتها.
4. عندما تجد العقدة إثبات عمل، فإنها تبث الكتلة لجميع العقد.
5. العقد تقبل الكتلة فقط إذا كانت جميع المعاملات الموجودة فيها صالحة ولم تنفق بالفعل.
6. تعرب العقد عن قبولها للكتلة من خلال العمل على إنشاء الكتلة التالية في السلسلة ، باستخدام هاش الكتلة المقبولة كهاش سابق.

العقد دائمًا تعتبر أطول سلسلة لتكون السلسلة الصحيحة وستستمر في العمل على توسيعها. إذا قامت عقدتان ببث إصدارات مختلفة من الكتلة التالية في نفس الوقت ، فقد تتلقى بعض العقد واحدة منهما أولاً أو الأخرى. في هذه الحالة ، تعمل على أول واحد يتلقونها ، ولكن يحفظ الفرع الآخر لحالة أن يصبح أطول. سيتم كسر هذه الحالة عندما يتم العثور على إثبات العمل التالي ويصبح فرع واحد أطول ؛ ستنتقل العقد التي كانت تعمل في الفرع الآخر إلى السلسلة الأطول.

بث المعاملات الجديدة لا يحتاج بالضرورة إلى الوصول إلى جميع العقد. طالما أنها تصل إلى العديد من العقد ، فإنها سوف تدخل في كتلة قبل فترة طويلة. بث الكتل هو أيضا متسامح مع فقدان الرسائل. فإذا لم تتلقى العقدة كتلة ما ، فستطلبها عند استلامها للكتلة التالية وتدرك أنها قد فقدتها.

الحافز

حسب الاتفاقية ، فإن المعاملة الأولى في كتلة هي معاملة خاصة تبدأ عملة جديدة يملكها منشئ الكتلة. وهذا

يضيف حافزا للعقد لدعم الشبكة ، ويوفر طريقة لتوزيع القطع النقدية في التداول في البداية ، حيث لا توجد سلطة مركزية لإصدارها. إن الإضافة الثابتة لكمية من العملات الجديدة يشبه عمال مناجم الذهب الذين ينفقون الموارد لإضافة الذهب إلى التداول. في حالتنا ، هو وقت وحدة المعالجة المركزية والكهرباء التي يتم إنفاقها.

ويمكن أيضا أن يتم تمويل الحافز من خلال رسوم الصفقة. إذا كانت قيمة المخرجات للمعاملة أقل من قيمة مدخلاتها ، يكون الفرق عبارة عن رسوم معاملة تتم إضافتها إلى القيمة الحافزة للكتلة المحتوية على الصفقة. بمجرد أن يتم تداول عدد محدد من العملات ، يمكن أن ينتقل الحافز بالكامل إلى رسوم المعاملات ويكون خالٍ من التضخم تمامًا.

قد يساعد الحافز في تشجيع العقد على البقاء صادقة. إذا كان المهاجم الجشع قادرًا على تجميع المزيد من قوة وحدة المعالجة المركزية أكبر من جميع العقد الصادقة ، فسيتعين عليه الاختيار بين استخدامها للاحتيال على الأشخاص عن طريق سرقة و إستعادة مدفوعاته ، أو استخدامها لتوليد عملات معدنية جديدة. سوف يجب عليه أن يجد أن اللعب بالقواعد أكثر ربحية ، مثل القواعد التي تجعله يحصل على قطع نقدية جديدة أكثر من الأشخاص الآخرين مجتمعين ، بدلاً من تقويض النظام وصحة ثروته الخاصة.

استعادة مساحة القرص

بمجرد دفن أحدث معاملة لعملة تحت كتل كافية ، يمكن التخلص من المعاملات المستنفدة قبلها ليتم توفير مساحة على القرص. لتسهيل هذا دون كسر هاش الكتلة ، يتم استخدام هذه المعاملات لتكوين شجرة Merkle فقط الجذر هو الوحيد المتضمن في هاش كتلة.ويمكن بعد ذلك ضغط الكتل القديمة عن طريق فلق أغصان ، الشجرة. ولا يلزم تخزين الهاشات الداخلية.

سيكون رأس أي كتلة بدون معاملات حوالي 80 بايت. إذا افترضنا أن الكتل يتم إنشاؤها كل 10 دقائق ، 80 بايت * 6 * 24 * 365 = 4.2 ميغابايت في السنة. مع أنظمة الكمبيوتر التي تباع عادة مع 2 غيغابايت من ذاكرة الوصول العشوائي اعتبارا من عام 2008 ، ويتوقع قانون مور النمو الحالي ب 1.2 غيغابايت في السنة ، لا ينبغي أن يكون التخزين مشكلة حتى إذا كان يجب الاحتفاظ برؤوس الكتل في الذاكرة.

التحقق المبسط من الدفع

من الممكن التحقق من المدفوعات دون تشغيل عقدة شبكة كاملة. يحتاج المستخدم فقط إلى الاحتفاظ بنسخة من رؤوس الكتل لأطول سلسلة من إثباتات العمل ، والتي يمكن الحصول عليها من خلال الاستعلام عن عقد الشبكة حتى يتم اقتناعه بامتلاك أطول سلسلة ، والحصول على فرع Merkle الذي يربط المعاملة بالكتلة المختمة زمنيا. لا يمكنه التحقق من المعاملة بنفسه ، ولكن عن طريق ربطها بمكان في السلسلة ، يمكنه أن يرى أن هناك عقدة شبكة قد قبلتها ، والكتل التي تمت إضافتها بعد تعمل على مزيد من التأكد من قبول الشبكة لها.

على هذا النحو ، يكون التحقق موثوقًا طالما أن العقد الصادقة تتحكم في الشبكة ، ولكنها أكثر عرضة للخطر إذا تم التغلب على الشبكة من قِبل مهاجم. بينما يمكن لعقد الشبكة التحقق من المعاملات بنفسها ، يمكن أن تنخدع الطريقة المبسطة بواسطة معاملات ملفقة لمهاجم طالما أن المهاجم يمكن أن يستمر في التغلب على

الشبكة. تتمثل إحدى الاستراتيجيات للحماية من ذلك في قبول التنبيهات من عقد الشبكة عند اكتشاف كتلة غير صالحة ، مما يدفع برنامج المستخدم إلى تنزيل الكتلة الكاملة والمعاملات التي تم تنبيهها لتأكيد عدم التناسق. ربما لا تزال الشركات التي تتلقى دفع متكرر ترغب في تشغيل عقدها الخاصة لمزيد من الأمان المستقل والتحقق السريع.

دمج وتقسيم القيمة

على الرغم من أنه سيكون من الممكن التعامل مع العملات بشكل فردي ، سيكون من غير العملي إجراء معاملة منفصلة لكل سنت في عملية التحويل. للسماح بتقسيم القيمة ودمجها ، تحتوي المعاملات على العديد من المدخلات والمخرجات. عادةً ما يكون هناك إما إدخال واحد من معاملة سابقة أكبر أو مدخلات متعددة تجمع بين مقادير أصغر ، وفي الغالب مخرجين : أحدهما للدفع ، والآخر للإرجاع ، إن وجد ، إلى المرسل.

وتجدر الإشارة إلى أن المعاملة الواحدة تعتمد على عدة معاملات، وتلك المعاملات تعتمد على عدد أكبر من ذلك ،وهذه ليست مشكلة هنا. لا توجد حاجة مطلقًا إلى استخراج نسخة مستقلة كاملة من سجل المعاملات.

الخصوصية

يحقق النموذج المصرفي التقليدي مستوى من الخصوصية من خلال الحد من الوصول إلى المعلومات إلى الأطراف المعنية والطرف الثالث الموثوق به. إن ضرورة الإعلان عن جميع المعاملات بشكل علني تمنع هذه الطريقة ، ولكن لا يزال من الممكن الحفاظ على الخصوصية من خلال كسر تدفق المعلومات في مكان آخر: عن طريق إبقاء المفاتيح العامة مجهولة الهوية. يمكن للجمهور أن يرى أن شخصًا ما يرسل مبلغًا لشخص آخر ، ولكن بدون معلومات تربط المعاملة بأي شخص. ويشبه ذلك مستوى المعلومات التي تصدرها البورصات ، حيث يتم الإعلان عن وقت المعاملات وحجم المعاملات الفردية ، لكن دون الإخبار عن هوية الشخص.

كجدار ناري إضافي ، يجب استخدام زوج مفاتيح جديد لكل معاملة لمنعهم من الارتباط بمالك مشترك. لا تزال بعض الروابط لا يمكن تجنبها من خلال المعاملات متعددة المدخلات ، والتي تكشف بالضرورة أن مدخلاتها كانت مملوكة لنفس المالك. ويكمن الخطر في أنه في حالة الكشف عن مالك المفتاح ، قد يكشف الارتباط عن معاملات أخرى تخص نفس المالك

العمليات الحسابية

لنفترض سيناريو مهاجم يحاول توليد سلسلة بديلة أسرع من السلسلة الصادقة. حتى إذا تم تحقيق ذلك ، فإنه لن يجر النظام إلى تغييرات عشوائية ، مثل خلق قيمة من فراغ أو أخذ أموال لم تكن مملوكة للمهاجم. لن تقبل العقد أي معاملة غير صالحة كوسيلة للدفع ، ولن تقبل العقد الصادقة أبدًا كتلة تحتوي عليها. يمكن للمهاجم فقط محاولة تغيير أحد معاملاته الخاصة لاسترداد الأموال التي أنفقها مؤخرًا.

يمكن وصف السباق بين السلسلة الصادقة وسلسلة المهاجمين على أنه مسارعشوائي ذو حدين.

الحدث الناجح هو سلسلة صادقة يتم تمديدها بواسطة كتلة واحدة ، مما يزيد من سابقيتها بـ +1 ، وحدث

1- الفشل هو سلسلة المهاجم التي يتم توسيعها بواسطة كتلة واحدة ، مما يقلل الفجوة بمقدار.

إن احتمال لحاق أحد المهاجمين من عجز معين مشابه لمشكلة إفلاس مقامر. لنفترض أن مقامر ذو رصيد غير محدود يبدأ عند عجز وأنه يلعب عددًا غير محدود من التجارب لمحاولة الوصول إلى نقطة التعادل. يمكننا حساب الاحتمال الذي سيصل فيه إلى التعادل ، أو إحتمال أن يدرك المهاجم السلسلة الصادقة ، على النحو التالي:

p = احتمال أن عقدة صادقة تجد الكتلة التالية.
q = احتمال أن يجد المهاجم الكتلة التالية
q_z = احتمال لحاق المهاجم إذا كان متخلفا بمقدار z كتلة

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

وبالنظر إلى افتراضنا بأن $p > q$ ، ينخفض الاحتمال بشكل كبير (بشكل دالة أسية) مع زيادة عدد الكتل التي يتعين على المهاجم اللحاق بها. مع وجود احتمالات ضده ، إذا لم يقم بإندفاع محظوظ إلى الأمام في وقت مبكر ، فإن فرصه تصبح صغيرة بشكل كبير مع تراجعه.

إننا نعتبر الآن المدة التي يحتاج فيها متلقي المعاملة الجديدة إلى الانتظار قبل أن يكون على يقين كاف من أن المرسل لا يمكنه تغيير الصفقة. نفترض أن المرسل هو المهاجم الذي يريد أن يجعل المتلقي يعتقد أنه دفع له لفترة من الوقت ، ثم بدلها لتسديدها لنفسه بعد مرور بعض الوقت . سيتم تنبيه المتلقي عندما يحدث ذلك ، ولكن المرسل يأمل في أن يكون قد فات الأوان.

ينشئ المتلقي زوج مفاتيح جديد ويعطي المفتاح العمومي إلى المرسل قبل التوقيع بوقت قصير.

هذا يمنع المرسل من إعداد سلسلة من الكتل في وقت مبكر من خلال العمل عليها بشكل مستمر حتى يكون محظوظا بما فيه الكفاية للحصول على ما يكفي إلى الأمام ، ثم تنفيذ المعاملة في تلك اللحظة. وبمجرد إرسال المعاملة، يبدأ المرسل غير الأمين العمل سراً في سلسلة موازية تحتوي على نسخة بديلة من معاملته.

من الكتل بعدها. المستلم لا يعرف المقدار z ينتظر المستلم حتى تتم إضافة المعاملة إلى كتلة ويتم ربط الدقيق للتقدم الذي أحرزه المهاجم ، ولكن بافتراض أن الكتل الصادقة تأخذ متوسط الوقت المتوقع لكل كتلة ، فإن التقدم المحتمل للمهاجم سيكون توزيع بويسون مع القيمة المتوقعة:

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

للحصول على احتمال أن المهاجم لا يزال بإمكانه اللحاق الآن ، فإننا نضرب كثافة بويسون لكل مقدار من التقدم الذي كان يمكن أن يحققه باحتمال أن يتمكن من اللحاق من تلك النقطة:

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

إعادة الترتيب لتجنب جمع الذيل اللانهائي للتوزيع:

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

التحويل إلى كود C:

```
#include
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 بشكل كبير (وفقا للدالة الأسية) 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
```

الحل ل P 0.1 أقل من ٪ ...

```
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
```

الإستنتاج

لقد اقترحنا نظامًا للمعاملات الإلكترونية بدون الاعتماد على الثقة. لقد بدأنا بالإطار المعتاد للعملات المصنوعة من التوقيعات الرقمية ، والتي توفر سيطرة قوية على الملكية ، لكنها غير كاملة وبدون وسيلة لمنع الإنفاق المزدوج. لحل هذه المشكلة ، اقترحنا شبكة ند إلى ند باستخدام إثبات العمل لتسجيل سجل عام للمعاملات التي تصبح بسرعة غيرعملية من الناحية الحسابية لمهاجم أن يغيرها إذا كانت العُقد الصادقة تتحكم في غالبية قوة وحدة المعالجة المركزية.

الشبكة قوية في بساطتها غير المنظمة. تعمل العقد في وقت واحد مع قليل من التنسيق. لا يلزم تحديدها ، نظرًا لأن الرسائل لا يتم توجيهها إلى أي مكان معين وتحتاج فقط إلى تسليمها على أساس أفضل جهد. يمكن للعقد المغادرة وإعادة الانضمام إلى الشبكة متى شاءت ، وقبول سلسلة إثبات العمل كدليل على ما حدث أثناء ذهابها. العقد تصوت مع قوة وحدة المعالجة المركزية لها ، معربة عن موافقتها على الكتل الصالحة من خلال العمل على تمديدها ورفض الكتل الغير صالحة من خلال رفض العمل عليها. يمكن تطبيق أي قواعد وحوافز مطلوبة بواسطة آلية التوافق هذه.

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 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.