基于区块链的公平合同签署协议

刘 鲁[1,2], 何杰杰[1,2], 田海博[1,2]

[1]中山大学数据科学与计算机学院, 广东省信息安全重点实验室 广州 中国 510006

[2]广西密码学与信息安全重点实验室, 桂林 中国 541004

摘要 一个公平的合同签署协议需要使签名者在协议结束的时候能够同时获取有效的合同、或者无法获取任何有用的信息。大部分的合同签署协议需要一个可信赖的第三方来确保公平性。区块链技术给我们提供了全球账本和计时器。这个计时器在一个新区块出现的时候更新, 每一个区块都可以通过块头中的哈希值标识。如果我们假定这个哈希值是均匀分布的, 就可以使用 Rabin 签名信标来实现一个公平合同签署协议。这些协议不需要改变底层区块链技术的任何操作, 并且不需要提交任何矿工费就可以完成合同签署。

关键词 区块链; 公平性; Rabin 签名
中图法分类号 TP391 **DOI 号** 10.19363/j.cnki.cn10-1380/tn.2018.05.01

Fair Contract Signing Protocols using Block Chain

LIU Lu[1,2], HE Jiejie[1,2], TIAN Haibo[1,2]

[1]School of Data and Computer Science, Sun Yat-sen University; Guangdong Provincial Key Laboratory of Information Security, Guangzhou 510006, China

[2]Guangxi Provincial Key Laboratory of cryptography and Information Security, Guilin 541004, China

Abstract A fair contract signing protocol enables singers to obtain a contract simultaneously or obtain nothing useful at the end of the protocol. Most contract signing protocols need a trusted third party to guarantee the fairness property. The block chain technique gives us a global trustable ledger and a timer. The timer is updated when a new block appears and each block could be identified by the block header's hash value. If we assume the hash values are uniformly distributed, we could take the signature beacon approach of Rabin to make fair contract signing protocols. The protocols need not change anything of the underlying block chain practice, and need not to submit any miner fee to sign a contract.

Key words block chain; fairness; rabin signature

1. 介绍

在商业活动中, 公平性是一项基础的需求。考虑一份纸质合同签署的场景, 我们希望所有的当事人坐在一起签署一份合同并且观看彼此在合同上的签名。在最后一个当事人签名之后, 从法律上来说合同生效。如果当事人尝试签一份电子合同, 那么当事人就无法观看彼此的签名过程。例如, 当 Alice 给 Bob 发送她在合同上的签名, Bob 可能很轻易地就可以延迟回复, 中止合同, 或者滥用 Alice 的签名。解决方法就是设计一个公平合同签署协议来强制要求当事人良好表现。

公平合同签署协议可以根据是否存在一个可信第三方机构(TTP)来进行粗略区分。早期的协议不需要 TTP, 这类协议依赖于密钥或优先级的逐步释放。例如, Alice 和 Bob 将他们自己的密钥划分成 n 块, 一块块的互相交换。在交换周期内, Alice 或 Bob 具有在一块密钥中包含的一个有限的有利条件。这类协议主要是有过早终止问题, 研究人员建议使用 TTP 来解决该问题。一种明显的解决方法是使用一个在线的 TTP, 当事人发送合同的签名给 TTP, TTP 转交当事人的签名。为了避免性能瓶颈, 研究者提出了 TTP 仅在当事人发生争执的时候参与的协议。这类协议被称为最优公平合同签署协议。

通讯作者: 田海博, 博士, 副教授, tianhb@mail.sysu.edu.cn。

本文得到国家重点研发计划(NO.2017YFB0802503)资助, 广西密码学与信息安全重点实验室研究课题(NO.GCIS201711)资助, 广东省自然科学基金(NO.2015A030313133)资助
收稿日期: 2018-01-29; 修改日期: 2018-04-28; 定稿日期: 2018-05-02

包含 TTP 的协议具有的一个基础问题就是 TTP 在实践中是很难实现的。这个问题在比特币区块链技术出现之后似乎是得到了解决。然而, 区块链仅仅提供了一个可信赖的账本和一个计时器, 它距离一个在公平合同签署中的 TTP 还差很远。例如, Asokan 等人的公平交换协议[1]要求他们的 TTP 始终保存一些数据并且需要执行一系列的操作去解决当事人的争执。虽然一个账本可以胜任记录数据的功能, 但是它并没有责任执行一系列的操作来解决当事人之间的争执。或许可以有一个独立的实体仅负责执行这种操作以赚取某种密码货币, 但这个时候, 这个实体已经充当了一个 TTP 的角色。

Rabin[2]已经提出了一种基于签名信标实现合同签署的协议。TTP 每隔固定的时间间隔分发一个随机数的签名和时间戳。我们通过使用区块链技术来完成公平合同签署。

1.1 相关工作

目前已经有了很多包含或不包含 TTP 的合同签署的提议, 简单列举如下。

Blum[3]给出了一种基于 Rabin 加密方案、交换 RSA 密钥参数的协议。大致来看, 只有当一个签名者知道与合同中的 RSA 模块相关的 RSA 密钥参数的时候, 签名者才可以声称这个合同是有效的。然后签名者就可以公平的交换彼此的 RSA 密钥参数, 来使合同签署保持公平。Even[4], Goldreich[5], Okamoto 和 Ohta[6], Stini 和 Mauve[7]也给出了一些不包含 TTP 的简单协议。

在线 TTP 通常涉及到每一次消息传输。Deng 等人[8]提出了一种使用 TTP 来传递电子邮件以及邮件收据的可验证公平的电子邮件协议。Franklin 等人[9]雇用了一个 TTP 来担保每次交易的公平性。Alawi 等人[10]要求 TTP 记录每一个将要被签署的合同。Wan 等人[11]要求一个时间戳服务提供方来为每一个合同产生时间戳。

如果在签名者之间没有产生争执的话, TTP 可能是离线的。Ben-Or 等人[12]介绍了一种离线的 TTP 来解决提前终止问题。这个问题是指如果一个当事人太早地终止协议, 其余的当事人必须无尽地等待。Asokan 等人[1]当争执发生时使用 TTP 来复原加密的签名。Garay 等人[13], Ateniese[14], Wang[15]也用了类似的方式来使用 TTP。Huang 等人[16]雇用了一个 TTP, 当争执发生的时候, TTP 产生一个环签名。

我们接下来介绍两个新近的与区块链技术和合同签署相关的工作。

- Wan 等人[11]提出了一种使用时间戳服务的合同签署协议。大致来看, Alice 和 Bob 协商合同签署的最后期限, 如果两个人均在最后期限之前签名, 那么合同宣称有效。Alice 给合同以及最后期限签名, 并发送签名给 Bob。然后 Bob 给合同、最后期限以及 Alice 的签名进行签名, 并且发送 Bob 的签名给时间戳服务方。时间戳服务方鉴定 Alice 和 Bob 签名的有效性, 然后给合同一个时间戳。假设合同可以被部署在一个比特币的区块上, 那么他们的时间戳服务可以利用区块链技术来实现。
- Kiayias 等人[17]提出了一种使用全局交易账本的多方计算协议。他们的模型包括一个全局的时钟函数和一个账本函数。他们的账本扩展了比特币系统的函数以处理任意的函数。通过这些扩展, 他们给出了一个编译器来将任何一个多方计算协议转化为一个具有公平性和鲁棒性的安全协议。既然一个合同签署协议可以被当作一个特殊的多方安全协议, 那么他们的方法也提供了一种使用扩展的比特币系统来进行合同签署的通用方法。

1.2 贡献

我们提出了一种使用当前的比特币等区块链系统而不需要任何扩展的合同签署协议。比特币系统是一个包括数千个节点的分布式计算系统。该系统的任何修改都需要大部分节点的同意。当前出现的比特币分支系统足以说明这种社区共识是很难达成的。而我们的系统不需要修改任何底层区块链技术, 无需社区的任何共识。

另外, 目前的比特币, 以太坊等系统的代币都具有较高的价格, 这使得矿工费持续升高, 进而使得比特币交易或者以太坊交易必须能够获取远超过矿工费的收益时才会发生。这使得部署在比特币或者以太坊上的应用成为很少有人运行的应用。我们的系统只是把区块链作为随机数和时钟源头, 并不需要生成新的交易, 缴纳矿工费。

从技术上来说, 假设哈希值的任意一部分是均匀分布的。我们从区块头的哈希值中取出一部分作为随机数, 使用区块的高度和区块的时间戳作为计时器。这个随机数和定时器取代了在 Rabin 的合同签署协议中的签名信标[2]。合同签署的基本思想和 Rabin 协议是相同的。首先给一个可能有效的合同签名, 然后等待一个可信赖的事件来确认或者废除这个合同的合法性。不同之处在于我们的可信赖的事件是发生在区块链系统中一个新块出现的时候。然后我们添加参数来增加签订合同的时间和可能的合

同数量。我们还对不同参数集的预期执行时间进行了估计，并给出了一些实际的结果。

2. 引言

我们的工作仅仅需要区块头的时间戳，区块的高度，区块头的哈希值。我们按照文献[18,19]中惯例来描述这些参数。

2.1 时间戳和区块高度

对于比特币区块链，一个区块头包含 80 字节。前 4 个字节是区块的版本号，指明了要遵循的块验证的规则集。接下来的 32 个字节是前一个区块头的哈希值。这是一个 SHA256(SHA256())的哈希输出。之后的 32 字节也是一个 SHA256(SHA256())的哈希输出，表示该块中包含的所有交易的哈希值的默克尔根。接下来的 4 个字节是时间戳。区块的时间是矿工开始对块头做哈希时的 Unix 时间戳。它要求这个时间戳必须要比前十一个区块的中位时间大或者相等。一个完整的节点将不会接受一个超过两个小时的区块。接下来的两个字段占据了 8 个字节，指明了当前区块的目标，挖矿的目的是生成一个比该目标小的哈希值。

需要注意的是，区块中的时间戳是在本地取样的。当取样结束以后，取样结果不应该小于前十一个区块的中位时间。也就是说，由于这种本地取样的操作，一个区块的时间戳理论上来说是有可能小于前一个区块的时间戳的。如果有一个可信赖的全局时钟，那么这种情况将不会发生，毕竟区块是一个接一个地产生的。这说明了区块链时间戳与全局时钟的不同。

区块的高度指的是在此区块与第一个区块之间所含有的区块数目。第一个区块一般被看作是创世区块，其高度为 0。当两个或者多个矿工几乎在同时生成了自己的区块时，多个区块可以有相同的区块高度。一个人可以使用在“blockchain.info”网站中的“getblockcount”方法来获取最后一个区块的高度。我们仅仅考虑最长的链，与区块的时间戳相比，区块高度更为可靠的提供给我们一个间隔可变的计时器。

2.2 区块头的哈希值

当一个矿工找到了一个小于目标的区块头哈希值时，一个新的区块就诞生了。这个哈希值利用 SHA256(SHA256())算法来计算。我们给出了一个使用#125552 号区块的头部来计算哈希值的 python 代码样例。

```
>>> import hashlib
>>> header_hex = ("01000000"
+"81cd02ab7e569e8bcd9317e2fe99f2de44d49ab2b8851ba4a308000000000000"
+"e320b6c2fffc8d750423db8b1eb942ae710e951ed797f7affc8892b0f1fc122b" +"c7f5d74d" +"f2b9441a"
+"42a14695")
>>> header_bin = header_hex.decode('hex')
>>> hash = hashlib.sha256(hashlib.sha256(header_bin).digest()).digest()
>>> hash.encode('hex_codec')
'1dbd981fe6985776b644b173a4d0385ddc1aa2a829688d1e0000000000000000'
>>> hash[::-1].encode('hex_codec')
'0000000000000001e8d6829a8a21adc5d38d0a473b144b6765798e61f98bd1d'
```

'1dbd981fe6985776b644b173a4d0385ddc1aa2a829688dle0000000000000000'是一个采用小端模式存储的值。它的逆序是一个采用大端模式的值。大部分比特币客户端和网站是用大端模式来显示的区块头部哈希值。在本篇文章中我们也使用大端模式。例如，当我们说在#125552 号区块头部哈希值的最后十位时，我们的意思是指 0100011101，或者是十六进制形式 0x11d。

3. 协议

本节内容中包含 3 个协议，其中最后一个是最灵活的。

3.1 基本协议

Rabin[2]提出了一种包含签名信标的合同签署协议。这个签名信标以固定的时间间隔签名并且广播出一个随机数和一个时间戳。签名者首先声明“如果这个有一个随机数和一个时间戳的合同已经被签名者签署了，并且这个随机数和时间戳也被签名信标签署了，那么合同有效”。然后合同的签名者交换他们自己的随机数，并且对随机数的加和值进行签名。这个随机数的加和值被作为对签名信标下一个签名的随机数的猜测。如果这个猜测是正确的，那么所有合同的签名者就都得到了一份有效的合同。如果猜测是错误的，那么他们重复以上过程直到猜测正确。这个随机数的范围是[1,k]。假设这个时间间隔是Δ，这个协议预期的执行时间就是 $k\Delta$。

在区块链中，每一个区块都有区块高度，区块头的哈希值和本地采样的时间戳。比特币区块链的两个连续的区块之间的时间间隔大约是 10 分钟。这个哈希值是采用密码安全的哈希函数计算出的，使用大端模式表示。哈希值的后$\log_2 k$位应该具有良好的随机性。任何一个人可以使用公开的比特币系统

的 API 来获取最后一个区块的高度, 时间戳以及哈希值。

我们把签名信标的随机数用大端模式的哈希值的最后$\log_2 k$位来代替。同时把签名信标的时间戳用区块的高度和时间戳来代替。然后我们得到了一个基于区块链的签名信标。信标的广播用新区块的生成和广播替换。

合同用类似文献[2]中的方式来签署。Alice 和 Bob 讨论一个合同 C。他们采用了基于区块链的签名信标。信标信息以$M=(r,i,t,f)$的形式出现, 这里的 r 是预测区块头部哈希值的最后$\log_2 k$位, i 和 t 是合同签署时区块的高度和时间戳, f 是一个间隔因子来增加可能的有效区块的数量。大致上来说, Alice 和 Bob 会在一个时间戳为 t, 高度为 i 的区块出现之后开始签署合同。在接下来的 f 个区块中, 如果有一个区块头部哈希值的最后$\log_2 k$位等于 r, 这个合同就生成了。

详细来说, Alice 发送给 Bob 一个 Alice 签署的初步的签名协议PA_A。初步协议PA_A是“如果 Bob 可以给出一个被 Alice 签过名的(C,M), 并且 M 可以链接到区块链的一个部分, 这个部分的起始区块包含时间戳 t 并且此区块的高度是 i, 而且在接下来的 f 个区块中, 存在一个区块的头部哈希值用大端模式表示的时候最后$\log_2 k$位等于 r, Alice 就承认合同 C 是在 M 中提到的 t 时间后签署的有效合同”。Bob 也会发送一个类似的签过名的初步协议PA_B给 Alice 来完成初步阶段。

Alice 和 Bob 执行协议的情况是相同的。Alice 的协议如下:

1. Alice 读出最新区块的高度bh_A,发送PA_A和bh_A给 Bob, 并且打开一个$\Delta/6$的定时器。

2. 在 Alice 收到 Bob 发来的PA_B和区块高度bh_B之后, 她清除之前的定时器。之后她等待高度是$\max(bh_A, bh_B)+1$的区块出现。

3. Alice 读出新区块的时间戳 t 和区块高度 i。然后她读出本地的时间t_0, 计算时间差$t_d = t_0 - t$。如果t_d的绝对值是可接受的, Alice 将任选$r_A \in \{1, \ldots, k\}$, 发送r_A给 Bob, 并且打开一个$\Delta/6$的定时器。否则, Alice 终止。

4. 当 Alice 收到 Bob 发来的r_B, Alice 清除之前的定时器并且计算 $r = r_A + r_B \bmod k$, 生成 $M=(r,i,t,f)$,签名$\mathrm{Sig}_A = \mathrm{Sign}_A(C,M)$。Alice 发送$\mathrm{Sig}_A$给 Bob, 并且打开一个$\Delta/3$的定时器。

5. 当 Alice 收到 Bob 发来的Sig_B, 她清除之前的定时器, 等待 f 个区块, 直到有一个区块可以使合同合法。如果没有这样的区块, 那么 Alice 在一个新区块出现的时候从步骤 3 重新开始执行这个协议。

6. 在任何一个步骤中, 如果定时器超时的情况发生, Alice 终止。

注意在比特币系统中的时间戳是由矿工在本地取样得到的, 理论上来说允许一个区块的时间戳比前一个区块的小。因此我们设置一个变量t_d来记录区块时间戳之间的时间差和一个签名者的本地时间。一个签名者应该检验t_d是足够小的。

3.2 延迟起始区块

上述的基础协议隐式地假定区块的间隔足够长以来完成交易。通过在信标信息 M 中增加一个延时因子 d, 我们可以去掉这个假设。大致上来说, Alice 和 Bob 可以在时间$d\Delta$内交换他们的信息。现在 Alice 的初步协议PA_A是“如果如果 Bob 可以生成一个被 Alice 签过名的(C,M), 并且 M 可以链接到区块链的一个部分, 这个部分的起始区块是包含时间戳 t 并且区块高度是 i 的区块的后面第 d 个区块, 而且在接下来的 f 个区块中, 存在一个区块的头部哈希值用大端模式表示的时候最后$\log_2 k$位等于 r, Alice 就承认合同 C 是在 M 中提到的 t 时间后签署的有效合同”。

现在 Alice 的协议如下。

1. Alice 读出最新区块的高度bh_A,发送PA_A和bh_A给 Bob, 并且打开一个$d\Delta/6$的定时器。

2. 在 Alice 收到 Bob 发来的PA_B和区块高度bh_B之后, 她清除之前的定时器。之后她等待高度是$\max(bh_A, bh_B)+1$的区块出现。

3. Alice 读出新区块的时间戳 t 和区块高度 i。然后她读出本地的时间t_0, 计算时间差$t_d = t_0 - t$。如果t_d的绝对值是可接受的, Alice 将任选$r_A \in \{1, \ldots, k\}$, 发送r_A给 Bob, 并且打开一个$d\Delta/6$的定时器。否则, Alice 终止。

4. 当 Alice 收到 Bob 发来的r_B, Alice 清除之前的定时器并且计算 $r = r_A + r_B \bmod k$, 生成 $M=(r,i,t,d,f)$,签名$\mathrm{Sig}_A = \mathrm{Sign}_A(C,M)$。Alice 发送$\mathrm{Sig}_A$给 Bob, 并且打开一个$d\Delta/3$的定时器。

5. 当 Alice 收到 Bob 发来的Sig_B, 她清除之前的定时器, 等待$i+d$个区块。当第$i+d$个区块出现之后, Alice 再等待 f 个区块, 直到有一个区块可以使合同合法。如果没有这样的区块, 那么 Alice 在一个新区块出现的时候从步骤 3 重新开始执行这个协议。

6. 在任何一个步骤中, 如果定时器超时的情况发生, Alice 终止。

3.3 增加成功率

在以上的协议中的参数 f 可以增加生成一个有

效合同的成功率。例如, 如果$k=32, f=16$, 成功率大约是 1/2。然而, 这是有代价的。首先是公平性。如果 Alice 发送一个可能的合同给 Bob 然后 Bob 终止了, Bob 有 1/2 的机会来使合同合法。其次是执行时间。更大的f意味着更长的执行时间。

我们给出了另外一个方法来增加成功率。大致上来说, Alice 和 Bob 可以用不同的随机数连续签署多个可能成功的合同。初步协议和延迟起始区块的版本是相同的。Alice 的协议生成了一个新的参数 b 来指明交换的合同签名的额外数量。在新的协议中采用了一个安全的哈希函数 $hash$。Alice 的协议如下。

1. Alice 读出最新区块的高度bh_A,发送PA_A和bh_A给 Bob, 并且打开一个$d\Delta/2(2b+3)$的定时器。

2. 在 Alice 收到 Bob 发来的PA_B和区块高度bh_B之后, 她清除之前的定时器。之后她等待高度是$\max(bh_A, bh_B)+1$的区块出现。

3. Alice 读出新区块的时间戳 t 和区块高度 i。然后她读出本地的时间t_0, 计算时间差$t_d=t_0-t$。如果t_d的绝对值是可接受的, Alice 将任选$r_A \epsilon \{1,\dots,k\}$, 发送$r_A$给 Bob, 并且打开一个$d\Delta/2(2b+3)$的定时器。否则, Alice 终止。

4. 当 Alice 收到 Bob 发来的r_B, Alice 清除之前的定时器并且计算$r=r_A+r_B \bmod k$, 生成$M=(r,i,t,d,f)$,签名$\mathrm{Sig}_A=\mathrm{Sign}_A(C,M)$。Alice 发送$\mathrm{Sig}_A$给 Bob, 并且打开一个$d\Delta/(2b+3)$的定时器。

5. 当 Alice 收到 Bob 发来的Sig_B, 她清除之前的定时器并且重复下列操作 b 次:

(a) Alice 计算 has$h(r)$来更新 r, 生成$M=(r,i,t,d,f)$,签名$\mathrm{Sig}_A=\mathrm{Sign}_A(C,M)$。Alice 发送$\mathrm{Sig}_A$给 Bob, 并且打开一个$d\Delta/(2b+3)$的定时器。

(b) 当 Alice 收到 Bob 发来的Sig_B, 她清除之前的定时器。

6. 在第$i+d$个区块出现之后, Alice 再等待 f 个区块, 直到有一个区块可以使这$b+1$个合同中的一个合法。如果没有这样的区块, 那么 Alice 在一个新区块出现的时候从步骤 3 重新开始执行这个协议。

7. 在任何一个步骤中, 如果定时器超时的情况发生, Alice 终止。

3.4　协议分析

这个协议假定哈希的输出是均匀分布的。我们接下来按照文献[2]的思路, 分析最后一个协议的安全性。

定理 1. 假设哈希输出是均匀分布的。如果 Alice 和 Bob 遵循他们各自的协议, 在预期执行时间$k(d+f)\Delta/f(b+1)$之后, 每个人都有另外一个人合同的签名。如果 Alice 遵循她的协议, Bob 尝试去欺骗她, Bob 得到 Alice 合同签名并且 Alice 没有 Bob 的签名的概率是f/k。Bob 被欺骗的概率类似。

证明. 假设 Alice 和 Bob 在区块高度为 i 的区块出现的时候开始他们的协议。在区块高度为$i+d$的区块出现之后, 他们的必须已经分别交换了$b+1$个签名Sig_A和Sig_B。对每一组签名对Sig_A和Sig_B, 都存在一个随机数$r \epsilon \{1,\dots,k\}$。因为我们假定哈希值是具有任意性的, 我们可以认为每个 r 都是在集合$\{1,\dots,k\}$上均匀分布的。在$i+d+f$区块出现之后, 存在最多f个区块头。由于假定的随机性, 区块头部哈希值的最后$\log_2 k$位有$(b+1)/k$的机会与$b+1$个随机数中的一个相等。因为我们考虑了 f 个区块头, 所以一个合法合同出现的概率就是$f(b+1)/k$。因此这个协议将在预期执行时间$k(d+f)\Delta=f(b+1)$之内终止。

当协议终止, Alice 和 Bob 可能有最多f个合同是同时合法的。他们可以在这其中选择一个作为他们自己最终的合法合同。如果争执发生了, 也就是说, Alice 否认曾经签署过这份合同, Bob 就可以起诉她并且证明她是签过的。也就是说, Bob 将给出一个初步协议PA_A和签名信息Sig_A。法官需要验证这个签名Sig_A和PA_A中的签名。如果这两个签名都得到了验证, 那么这个法官就去检查区块链, 核实PA_A中的条件。也就是, M 中的时间戳 t 是被包含在区块高度为 M 中的 i 的区块中的, 并且在第$i+d$个区块与第$i+d+f+1$个区块之中存在一个区块头部哈希值的最后$\log_2 k$位等于 M 中的 r 值。如果所有的条件都满足了, 那么 Alice 就是签署过这个合同。

现在假设 Alice 遵循了她的协议, 但是 Bob 没有在$d\Delta/(2b+3)$时间之内回复。Alice 将终止协议。假设 Alice 已经收到了 Bob 发来的 j 个签名。然后 Bob 已经收到了 Alice 发来的$j+1$个签名。这 j 个签名对可能可以同时使合同合法。Bob 对于 Alice 来说唯一的有利条件就是一个 Alice 发来的额外的可以使合同合法的签名, 合法的可能性是f/k。有$1-f/k$的概率是 Bob 并没有有利条件。

Alice 和 Bob 在协议中的角色是完全对等的, 因此上述分析也适用于 Bob 被欺骗的概率。

3.5　协议实例

我们给出一些最终版协议的实例。首先, 我们令$d=1, \Delta=10$分钟, $b=0, f=1, k=1024$。然后交换签名的时间间隔大约是 3 分钟。Alice 和 Bob 有 1/1024 的概率需要大约 20 分钟来签署合同。完成协议的预期执行时间是两个星期。注意我们没有考

虑交换他们合法合同的时间, 因为这个交换只发生一次。

然后, 我们令$d = 10, \Delta = 10$分钟, $b = 49, f = 2, k = 1024$。然后交换签名的时间间隔大约是 1 分钟。Alice 和 Bob 有 1/10 的概率需要大约两个小时来签署合同。完成协议的预期执行时间是二十个小时。如果网络延迟是很小的, 我们可以使用更为乐观的参数。例如, $b = 499$, 交换签名的时间间隔大约是 6 秒。理论上来说, 在两个小时以内, 一个合法合同出现的概率是 0.75。

在实践中, 我们使用比特币区块的原始数据来签署合同。当一个合同将要被签名时, 我们令$k = 1024, b = 499, f = 2$。这个随机数r_A和r_B是由 C++中的默认函数“rand()”生成的。我们使用 SHA256()函数来生成不同随机数的链。我们选择一个任意的区块作为起始区块来签署 100 次合同。如果一个随机数在两个区块中都被发现了, 那么有效的合同产生了。我们用不同的起始区块重复 20 次, 结果在表 1 中。当这个起始区块是#80 号区块时, 在 100 协议执行的过程中生成了 81 份有效的合同。当这个起始区块是#180 号区块时, 成功率是 0.64。在这 2000 次执行中签名合同总体的成功率大约是 0.72。

表 1 采用不同的起始区块时有效合同的数量

Table 1 Number of valid contracts with different beginning block

起始区块	有效合同的数量	成功率
10	69	0.69
20	73	0.73
30	72	0.72
40	71	0.71
50	76	0.76
60	75	0.75
70	74	0.74
80	81	0.81
90	77	0.79
100	71	0.71
110	73	0.73
120	71	0.71
130	70	0.70
140	69	0.69
150	71	0.71
160	72	0.72
170	73	0.73
180	64	0.64
190	75	0.75
200	70	0.70

4 结论

我们展示了如何使用当前的比特币区块链系统来设计一个公平合同签署协议。它不需要缴纳矿工费。同样的思路可以很扩展到以太坊等区块链系统中。特别的, 以太坊的区块时间大约是 15 秒, 与比特币的大约 10 分钟相比, 出块速度快了将近 40 倍。这意味着我们的协议效率还有提升的空间。

参考文献

[1] N. Asokan, V. Shoup and M. Waidner, “Optimistic fair exchange of digital signatures,” IEEE Journal on Selected Areas in Communications, vol. 18, no. 4, pp. 593-610, 2000.

[2] M. O. Rabin, “Transaction protection by beacons,” Journal of Computer and System Sciences, vol. 27, no. 2, pp. 256-267, 1983.

[3] M. Blum, “How to exchange (secret) keys,” Proceedings of the fifteenth annual ACM symposium on Theory of computing, Victoria, British Columbia, pp. 440-447, 1983.

[4] S. Even, “A protocol for signing contracts,” SIGACT News, vol. 15, no. 1, pp.34-39, 1983.

[5] O. Goldreich, “Simple Protocol for Signing Contracts,” dvances in Cryptology: Proceedings of Crypto 83, California, USA, pp. 133-136, 1984.

[6] T. Okamoto and K. Ohta, “How to simultaneously exchange secrets by general assumptions,” Proceedings of the 2nd ACM Conference on Computer and communications security, VA, USA, pp. 184-192, 1994.

[7] M. Stini and M. Mauve, “Enabling fair offline trading,” Proceedings of the 2009 International Conference on Wireless Communications and Mobile Computing: Connecting the World Wirelessly, NY, USA, pp. 973-978, 2009.

[8] R. H. Deng, L. Gong, A. A. Lazar and W. Wang, “Practical protocols for certified electronic mail,” Journal of Network and Systems Management, vol. 4, no. 3, pp.279-297, 1996.

[9] M. K. Franklin and M. K. Reiter, “Fair exchange with a semi-trusted third party (extended abstract),” Proceedings of the 4th ACM conference on Computer and communications security, Arizona, USA, pp. 1-5, 1997.

[10] A. A. Al-Saggaf and L. Ghouti, “Efficient abuse-free fair contract-signing protocol based on an ordinary crisp commitment scheme,” IET Information Security, vol. 9, no.1, pp. 50-58, 2015.

[11] Z.Wan, R. H. Deng and D. Lee, “Electronic Contract Signing Without Using Trusted Third Party,” Network and System Security: 9th International Conference, NSS 2015, NY, USA, pp.386-394, 2015.

[12] M. Ben-Or, O. Goldreich, S. Micali and R. L. Rivest, “A fair protocol for signing contracts,” IEEE Transactions on Information Theory, vol. 36, no. 1, pp.40-46, 1990.

[13] J. A. Garay, M. Jakobsson and P. MacKenzie, “Abuse-Free Optimistic Contract Signing,” Advances in Cryptology - CRYPTO’ 99: 19th Annual International Cryptology Conference, California, USA, pp. 449-466, 1999.

[14] G. Ateniese, “Verifiable encryption of digital signatures and applications,” ACM Trans. Inf. Syst. Secur., vol. 7, no. 1, pp.1-20, 2004.

[15] G. Wang, “An Abuse-Free Fair Contract-Signing Protocol Based on the RSA Signature,” IEEE Transactions on Information Forensics and Security, vol. 5, no. 1, pp. 158-168, 2010.

[16] Q. Huang, G. Yang, D. S. Wong and W. Susilo, “Efficient Optimistic Fair Exchange Secure in the Multi-user Setting and Chosen-Key Model without Random Oracles,” Topics in Cryptology - CT-RSA 2008: The Cryptographers’ Track at the RSA Conference 2008, CA, USA, pp. 106-120, 2008.

[17] A. Kiayias, H. S. Zhou and V. Zikas, “Fair and Robust Multi-party Computation Using a Global Transaction Ledger,” Advances in Cryptology - EUROCRYPT 2016: 35th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Vienna, Austria, pp. 705-734, 2016.

[18] “Bitcoin developer reference,” https://bitcoin.org/en/developer- reference#blockchain, 25 June. 2016.

[19] “Block hashing algorithm,” https://en.bitcoin.it/wiki/Block hashing algorithm., 25 June. 2016.

刘鲁　于2017在中山大学软件工程专业获得工学学士学位。现在在中山大学软件工程专业攻读工学硕士学位，研究兴趣包括密码协议，机器学习，区块链等。Email: hhdliulu@163.com

田海博　于2006年在西安电子科技大学密码学专业获得博士学位。现任中山大学副教授。研究领域为安全协议设计与分析，近期主要关注区块链技术。Email: tianhb@mail.sysu.edu.cn

何杰杰　于2015年在中央民族大学计算机科学与技术专业获得学士学位 。现在中山大学计算机科学与技术专业攻读硕士学位。研究兴趣包括: 区块链安全协议设计。Email: 414175749@qq.com