密码学报 ISSN 2095-7025 CN 10-1195/TN
Journal of Cryptologic Research,2017,4(2):187–198

E-mail: jcr@cacrnet.org.cn
http://www.jcr.cacrnet.org.cn
Tel/Fax: +86-10-82789618

基于公开区块链的隐私保护公平合同签署协议*

田海博[1,2], 何杰杰[1,2], 付利青[1,2]

1. 中山大学 数据科学与计算机学院, 广州 510006
2. 广东省信息安全技术重点实验室, 广州 510006
通讯作者: 田海博, E-mail: tianhb@mail.sysu.edu.cn

摘 要: 公平合同签署一直都是电子商务中一项重要的密码服务. 现今的技术实践需要一个在线或离线的中心化可信第三方来解决合同签署中所出现的争端. 但是中心化可信第三方具有较高的安全风险, 较易出现单点故障, 受到内部或者外部的攻击. 作为比特币和许多其他数字货币的一项基础技术, 区块链为我们提供了一个无中心的可信第三方, 可以避免中心化可信第三方的问题, 实现全球可信这样一个理想的目标. 然而, 基于区块链设计安全协议需要考虑区块链的公开验证和隐私保护之间的矛盾. 基于区块链的安全协议一般是通过交易来完成协议的安全目标. 而交易是公开的, 区块链中每一个节点都可以读取交易数据, 验证交易数据是否正确. 如果交易数据中涉及隐私的内容, 矛盾就产生了. 特别的, 对于公平合同签署协议, 合同的签署人、内容、数字签名等都可能涉及敏感信息, 往往是不希望公开的. 那么在区块链上设计保护隐私的公平合同签署协议就是困难的. 本文基于传统的可验证加密签名和盲签名思想, 构造了盲的可验证加密签名体制, 可以在公平合同签署这个应用场景中解决隐私保护的问题. 我们基于该体制构造了公平且秘密的合同签署协议, 可以让合同签署人通过区块链完成公平的合同签署, 并且保护与合同相关的隐私内容.

关键词: 区块链; 公平合同签署; 隐私保护; 可验证加密签名
中图法分类号: TP309.7 **文献标识码**: A DOI: 10.13868/j.cnki.jcr.000173

中文引用格式: 田海博, 何杰杰, 付利青. 基于公开区块链的隐私保护公平合同签署协议[J]. 密码学报, 2017, 4(2): 187–198.
英文引用格式: TIAN H B, HE J J, FU L Q. A privacy preserving fair contract signing protocol based on public blcok chains[J]. Journal of Cryptologic Research, 2017, 4(2): 187–198.

A Privacy Preserving Fair Contract Signing Protocol based on Block Chains

TIAN Hai-Bo[1,2], HE Jie-Jie[1,2], FU Li-Qing[1,2]

1. School of Data and Computer Science Sun Yat-sen University, Guangzhou 510006, China
2. Guangdong Provincial Key Laboratory of Information Security, Guangzhou 510006, China
Corresponding author: TIAN Hai-Bo, E-mail: tianhb@mail.sysu.edu.cn

Abstract: Fair Contract signing is always an important cryptographic service for electronic commerce. Current technique practices need an online or offline centralized trusted third party (TTP) to solve disputes appeared in a contract singing process. However, a centralized TTP has high risks. It may suffer the single point of failure, inside or outside attacks. The block chain, as a basic technique of Bitcoin and many other digital currencies, gives

* 基金项目: 国家自然科学基金项目(61672550); 广东省自然科学基金项目(2015A030313133)
收稿日期: 2017-03-10 定稿日期: 2017-04-02

us a decentralized TTP, which could avoid many problems of a centralized one and may achieve the ideal goal of a global TTP. However, the contradiction should be considered between the public verification property of a block chain and the privacy protection requirement of a security protocol when we design such protocols based on block chains. A security protocol based on a block chain means that the protocol needs to fulfill its security goals through some transactions of the block chain. But the transactions are public. Every node of the block chain could read the transactions and verify the correctness of these transactions. If the transaction data involves privacy contents, a contradiction appears. Especially, for a fair contract signing protocol, the signers, contents and signatures of a contract may involve sensitive information and are not intended to be public. So it is hard to design a privacy preserving fair contract signing protocol based on block chains. This paper makes use of a blinded verifiable encrypted signature (VES) scheme based on traditional VES and the idea of blind signature. The new scheme could solve the privacy protection problem in the application scenario of fair contract signing. We give a fair and secret contract signing protocol based on the new scheme, which allows contract signers fulfill the task of fair contract signing based on public chains, and protects the privacy contents related to the contract.
Key words: block chain; fair contract signing; privacy preserving; verifiable encrypted signature

1 引言

在电子商务中, 人们往往通过签署合同来保护自己的权利和经济利益. 现在假设两个签署者 Alice 和 Bob 希望通过互联网签署合同, 以节约差旅等费用, 提高效率. 当 Alice 发送自己在合同上的数字签名给 Bob 后, Bob 根据合同的内容需要, 可能会有意地延迟回复自己的签名, 或者干脆不回复, 直接推脱因为网络的因素没有收到任何签名. 因此, 在互联网中签署合同时, 需要一个公平的合同签署协议, 让合同签署的各个参与者要么同时获得其它参与者的数字签名, 要么谁都不会得到任何有效的数字签名. 目前, 这些合同签署协议根据对可信第三方(trusted third party, TTP)的使用情况大致分为两类. 一类协议不依赖 TTP, 采用逐步释放秘密或优先权方法, 让不公平的程度变得可以容忍. 但是这类协议受过早停止问题的困扰, 会造成诚实参与者不能顺利终止协议的情况. 另外一类协议依赖 TTP. 其中一些协议使用在线的 TTP, 合同签署者发送他们对合同的签名给 TTP, TTP 随后对签名进行可靠的传输, 保证每个参与者都可以得到所应得的数字签名. 另外一些协议使用离线的 TTP, 允许合同签署的参与者在产生纠纷时向 TTP 申述, 否则 TTP 就不用工作, 这些协议通常被称作最优公平合同签署协议.

目前, 合同签署协议的 TTP 大多是中心化的. 这些 TTP 通常掌握有合同的一些敏感信息, 例如谁正在签署合同, 合同的数字签名是什么等等, 甚至有一些中心化 TTP 会知道合同文本. 这些信息往往会引起外部攻击者的兴趣, 使得这些 TTP 的安全风险增加. 另外, 中心化 TTP 的员工也可能因为心怀不满或者受到诱惑, 拿敏感信息换取经济利益, 造成内部攻击, 进一步增加了这种 TTP 的安全风险. 另外, 如果中心化 TTP 自身的硬件和软件出现了故障, 可能会造成服务中断. 还有, 如果硬件和软件跟不上业务的规模, 中心化 TTP 会成为协议的性能瓶颈. 最后, 中心化 TTP 的业务范围有限制, 往往只能服务一个区域, 一般做不了全球的 TTP. 总之, 方方面面的原因使得中心化的 TTP 具有较低的可靠性和可信性、较高的安全风险.

区块链技术为人们提供了一种无中心的 TTP. 比特币是区块链技术的典型代表[1], 是一个能够完成交易记账的全球范围内的 TTP. 这种无中心的 TTP 存储的数据都是公开的, 对以获取秘密为目标的外部攻击者而言没有吸引力. 这种 TTP 的节点是动态加入, 动态退出的, 内部攻击需要对共识算法进行颠覆才有可能, 需要的代价是比较大的. 节点的动态性保证了单个节点的故障对全网影响不大. 在性能方面, 无中心的 TTP 正在快速发展, 例如 Luu 等人[2]提出了基于拜占庭协议和节点分割方法的共识算法, 可以使每秒钟处理的交易数量与节点的数量成正比. 总之, 区块链这种无中心的 TTP 具有较高的可靠性和可信性、较低的安全风险.

但是区块链无中心 TTP 在设计安全协议时还存在隐私保护的问题. 以合同签署协议为例, 在通常的合

同签署协议中, 中心化 TTP 在出现争端时可能会提取合同的真实数字签名, 并转交给合同的参与者. 采用无中心的区块链技术之后, 如果不改变技术实现, 每个区块链节点都有责任在出现争端时提取合同的真实数字签名. 这意味着任何人都可以得知一份合同的真实数字签名. 如果合同是敏感的, 这显然是不合适的. 当前, Wan 等人[3]给出了一个合同签署协议, 他们建议由时间戳服务器检查合同签署相关的数字证书和数字签名的合法性, 然后在时间戳服务器中存储合同的哈希值. 他们认为该协议可以通过区块链技术、采用类似比特币的方式实现. Liu 等人[4]通过以太坊的智能合约实现了 Asokan 等人[5]的公平交换协议所需的中心化可信第三方的部分功能. 他们在智能合约中直接存储了交换的数字内容. 这两个新近文献说明当前关于公平合同签署的技术实现并没有特别考虑合同的隐私保护需求.

本文提出了一个新的密码体制, 称为盲的可验证加密签名. 基于该体制, 我们采用比特币交易的形式给出了一个两方的公平合同签署协议. 该协议不会向区块链节点透露签署者的身份、所签署的合同和所生成的数字签名, 同时保证合同签署的双方完成公平的合同签署.

2 区块链

区块链由一组基于点到点网络的节点组成, 各节点通过执行共识算法, 维护数据的一致性. 比特币是区块链的典型代表. 比特币区块包含了区块头部和一些比特币交易, 每一笔交易都有一个唯一标识(TxID), 这个唯一标识就是这笔交易的哈希值. 一个区块上所有交易的标识值构成了该区块默克尔树的叶子结点, 默克尔树的根节点保存在区块头里面, 因此所有交易与区块头部绑定在了一起. 区块头里面的还保存着上一个区块头的哈希值, 这让区块之间组成了一条链. 该链中新的区块通过共识机制来所产生. 比特币节点通过计算候选区块头部的哈希值来产生一个新的区块. 当大部分节点接受这个新区块后, 这个区块便被增加到链上, 其包含的所有交易也会在验证后得到确认.

比特币交易包含一些输入和输出, 如图 1 所示. 图 1 中给出了 T^A 和 T^B 两笔交易, A 和 B 分别是这两个交易的发起者, 他们实际上只是两个公钥. 如果 T^B 的一个输入中包含了 T^A 的唯一标识以及 T^A 的一个输出索引, 那么 T^B 的输入就会和 T^A 的输出相关联. 在图 1 中这种逻辑关联用箭头线来表示. T^B 的输入包括了输入脚本(is), 且此输入脚本和 T^A 的输出脚本(os)相匹配. T^B 的输入脚本中包含了一个签名 Sig_B 以及 B 的公钥, T^B 的签名信息中包含了 T^A 的输出脚本以及 T^B 除输入脚本外的所有内容. 这个签名内容笼统的表示为 $[\mathrm{T}^B]$ [6]. T^A 的输出脚本中包含了 B 公钥的哈希值以及一个签名验证命令. 为了把 T^B 关联到 T^A, 比特币节点把 T^A 输出脚本中的哈希值和 T^B 输入脚本中公钥的哈希值进行比较, 验证 T^B 输入脚本中的签名. T^A 被连接的条件表示为 $\mathrm{os}(\mathrm{body},\delta_B)$, 也就是说, 输出脚本需要一个消息 body 和一个签名来满足验证条件. 这里的 body 显然应该和 $[\mathrm{T}^B]$ 一致. 验证条件表示为 $\mathrm{ver}_B(\mathrm{body},\delta_B)$, 表示用 B 的公钥验证对消息 body 的签名 δ_B 是否正确. 交易的输出还包括一个代表着比特币的值. 图 1 中的符号 C 和 δ_C 表示其它的交易者和签名.

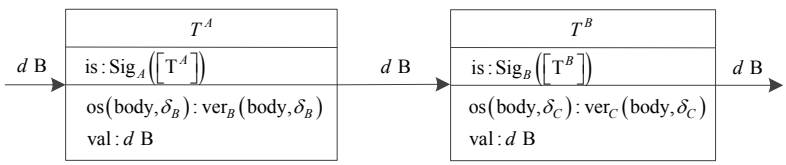


图 1 两个相连接的交易
Figure 1 Two connected transactions

这里有关于区块链的一些假设, 它们是从比特币系统观察出来的, 总结在文献[6,7]中:

- 假设协议参与者通过不安全信道连接;
- 假设协议参与者能访问区块链的内容;
- 假设协议参与者在区块链上公布交易有一个最大延迟 $\max_{BC}$;
- 区块链上被确认的交易是防篡改的.

3 盲的可验证加密签名体制(BVES)

3.1 体制构造

我们基于文献[8]中的 ZSS-VES 方案来设计我们的 BVES 方案, ZSS-VES 方案是基于文献[9]中基础的 ZSS 签名方案. 为了方便读者, 我们把这两个方案放在了附录 A.

一个 BVES 方案由三个参与者构成, 分别是签名者 Alice, 数字签名提取者 Bob 以及验证者 Minter. Alice 和 Bob 知道消息 m, 并给了这个消息一个索引 m_{index}, Bob 想得到 Alice 对消息 m 的签名. Alice 应该具有一个有效证书, 而 Bob 仅需要一个临时的密钥对. Minter 帮助 Bob 验证 Alice 的签名. 该方案由以下八个算法组成:

Setup: 设 G_1 是一个循环加法群, P 为生成元, 大素数 q 作为 G_1 的阶. 设 G_2 是一个循环乘法群, G_2 和 G_1 具有相同的阶. 定义 $e: G_1 \times G_1 \to G_2$ 作为一个双线性对, 并定义两个密码哈希函数 $H: \{0,1\}^* \to Z_q^*$ 以及 $H_1: G_2 \to Z_q^*$.

KeyGen: 随机选择 $x \in_R Z_q^*$ 并计算 $y = xP$. 然后让 y 作为公钥, x 则作为私钥. 此时 Alice 的密钥对表示为 (x_A, y_A), Bob 的密钥对表示为 (x_B, y_B).

Sign: 对于一个消息 m, Alice 计算生成一个普通签名 $\delta_{\text{Alice}} = \dfrac{1}{H(m) + x_A} P$.

Verify: 对于一个消息签名对 $(m, \delta_{\text{Alice}})$, 验证签名的双线性对方程如下:

$$e(H(m)P + y_A, \delta_{\text{Alice}}) = e(P, P) \tag{1}$$

PreSignAgree: 假设 Alice 签署一个消息 m 并发给 Bob, 此消息索引是 m_{index}.

(1) Alice 随机选择 $\alpha \in_R Z_q^*$, 计算 αP 以及 $s = H_1\left(e(y_B, y_A)^{\alpha}\right)$, 然后计算 sy_A 和 $sH(m)$. 随后 Alice 把以下元组发给 Bob:

$$(\alpha P, sy_A, m_{\text{index}}, sH(m)).$$

(2) Bob 计算 $s' = H_1\left(e(y_A, \alpha P)^{x_B}\right)$, 验证 $s'y_A = sy_A$ 以及 $s'H(m) = sH(m)$, 其中 m 由索引 m_{index} 确定. 如果所有的验证都通过的话, Bob 返回 true, 否则返回 false.

BVESSign: Alice 把公钥 y_B, 消息 m 以及 **PreSignAgree** 算法中生成的秘密因子 s 作为输入, 生成 BVES 签名如下:

$$\delta_{\text{BVES}} = \frac{1}{(H(m) + x_A)s} y_B \tag{2}$$

BVESVer: Minter 把四元组 $(\delta_{\text{BVES}}, sH(m), sy_A, y_B)$ 作为输入, 验证以下等式是否成立

$$e(sH(m)P + sy_A, \delta_{\text{BVES}}) = e(P, y_B) \tag{3}$$

BVESExt: Bob 把 BVES 签名 δ_{BVES} 以及 **PreSignAgree** 算法中生成的秘密因子 s' 作为输入，计算 $\frac{s'}{x_B}\delta_{\text{BVES}}$ 来提取普通签名.

注意到由于 BVES 体制是为区块链平台设计的. 因此不能对它像普通的签名方案那样来评估安全性. 例如，有人可能会选择随机的 $\beta,\gamma\in Z_q^*$，然后给 Minter 发送以下四元组以通过 Minter 的验证.

$$\left(\frac{1}{\beta+\gamma}P,\beta,\gamma P,P\right)$$

然而，通过 Minter 的验证并不代表该体制有安全问题. 该体制的不可伪造性在于攻击者不能在四元组的后面三个元素确定的情况下生成第一个元素，即 BVES 签名. 这一点在下面的安全模型中可以更清楚的看到.

3.2 安全模型

我们的安全模型基于文献[10]中可验证加密签名(VES)安全模型. 其中，敌手拥有一个签名预言机和一个仲裁预言机. 签名预言机提供 VES，仲裁预言机提供了从 VES 中提取的普通签名.

3.2.1 不可伪造性

不可伪造性可定义为由一个模拟器和一个敌手进行博弈的一场游戏. 在得到固定的密钥参数之后，敌手需要对模拟器选中的消息来进行伪造.

不可伪造性游戏定义如下:

Init: 模拟器运行 **Setup** 算法来生成以下系统参数:

$$(G_1,G_2,e,P,q,H,H_1)$$

然后模拟器运行 **KeyGen** 算法来产生密钥对 (x_A,y_A) 和 (x_B,y_B). 接下来它选择一个消息 m^* 作为目标，并运行 **PreSignAgree** 算法的第一步骤生成 $\left(sH\left(m^*\right),sy_A\right)$. 最后，模拟器把 $\left(sH\left(m^*\right),sy_A,y_B\right)$ 发送给敌手.

SignQueries: 敌手自适应请求至多 q_s 条消息 $\left\{m_1,\cdots,m_{q_s}\right\}$. 模拟器对第 i 次请求返回 BVES 签名 δ_{BVES_i}，$i\in\{1,\cdots,q_s\}$.

ExtQueries: 敌手自适应请求至多 q_b 条签名 $\left\{\delta_{\text{BVES}_1},\cdots,\delta_{\text{BVES}_{q_b}}\right\}$. 模拟器对第 i 次请求返回提取的普通签名，记作 δ_{Alice_i}，$i\in\{1,\cdots,q_b\}$.

Output: 敌手应该输出一个有效的 BVES 签名 δ^*_{BVES}，如果四元组 $\left(\delta^*_{\text{BVES}},sH\left(m^*\right),sy_A,y_B\right)$ 能通过 Minter 采用 **BVESVer** 算法进行的验证，他就赢得胜利. 注意到这个四元组中有三个是作为固定参数发送给敌手的.

我们把敌手赢得游戏胜利的概率直接定义为敌手的优势. 如果在一个 BVES 方案中敌手的优势可以忽略不计的话，那这个 BVES 方案具备不可伪造性.

3.2.2 不可提取性

不可提取性可定义为由一个挑战者和一个敌手进行博弈的一场游戏. 在得到固定密钥和消息参数之后，敌手需要从 BVES 签名中提取出对应的普通签名. 该游戏定义如下:

Init: 挑战者运行 **Setup** 算法来生成以下系统参数:

$$(G_1,G_2,e,P,q,H,H_1)$$

然后挑战者运行 **KeyGen** 算法来产生密钥对 (x_A,y_A) 和 (x_B,y_B). 接下来他选择一个消息 m^* 作为目标，

并运行 **PreSignAgree** 算法的第一步骤来产生对 $\left(sH\left(m^*\right), sy_A\right)$. 接下来运行 **BVESSign** 算法来产生一个 BVES 签名 δ^*_{BVES}, 最后, 挑战者把 $\left(\delta^*_{\text{BVES}}, sH\left(m^*\right), sy_A, y_B\right)$ 发送给敌手.

SignQueries: 敌手自适应请求至多 q_s 条消息 $\left\{m_1, \cdots, m_{q_s}\right\}$. 模拟器对第 i 次请求返回 BVES 签名 δ_{BVES_i}, $i \in \left\{1, \cdots, q_s\right\}$.

ExtQueries: 敌手自适应请求至多 q_b 条签名 $\left\{\delta_{\text{BVES}_1}, \cdots, \delta_{\text{BVES}_{q_b}}\right\}$. 模拟器对第 i 次请求返回提取的普通签名, 记作 δ_{Alice_i}, $i \in \left\{1, \cdots, q_b\right\}$.

Output: 要赢得游戏胜利, 敌手应该输出一个关于公钥 y_A 和消息 m^* 的有效普通签名 δ^*_{Alice}.

我们把敌手赢得游戏胜利的概率直接定义为敌手的优势. 如果在一个BVES方案中敌手的优势可以忽略不计的话, 那这个 BVES 方案具备不可提取性.

3.3 不可伪造性证明

定理 1 如果 ZSS-VES 方案[8]在它们的安全模型下是不可伪造的, 那么本文的 BVES 体制在我们的安全模型下是不可伪造的.

证明: 模拟器扮作 ZSS-VES 方案[8]的一个伪造者. 也就是说, 模拟器让 y_A, y_B 作为输入, 并在多项式时间之内请求 VES 签名和普通签名, 预期伪造一个 VES 签名. 模拟器运行 BVES 敌手的步骤如下:

- 模拟器选择一个随机值 $\alpha \in_R Z^*_q$ 并选择一个消息 m^*, 计算 $s = H_1\left(e\left(P_A, P_B\right)^{\alpha}\right)$, 然后准备以下参数:

$$\left(sH\left(m^*\right), sy_A, y_B\right)$$

- 当敌手想要消息 m_i 的 BVES 签名时, 模拟器向 VES 签名预言机请求消息 m_i 的 VES 签名 δ_{VES_i} 并计算 $\delta_{\text{BVES}_i} = \frac{1}{s}\delta_{\text{VES}_i}$. 然后 δ_{BVES_i} 被返回给敌手.

- 当敌手想要消息 m_i 的 BVES 签名 δ_{BVES_i} 对应的普通签名时, 模拟器向 VES 仲裁预言机请求 $s\left(\delta_{\text{BVES}_i}\right)$ 对应的普通签名 δ_{Alice_i}, 然后把 δ_{Alice_i} 返回给敌手.

- 当敌手由参数 $\left(sH\left(m^*\right), sy_A, y_B\right)$ 生成了一个伪造的 BVES 签名 δ^*_{BVES} 时, 模拟器生成了一个伪造的 VES 消息签名对 $\left(m^*, s\left(\delta^*_{\text{BVES}}\right)\right)$.

如果满足以下条件, 则此模拟是生效的:

$$H\left(m^*\right) \notin \left\{H\left(m_1\right), \cdots, H\left(m_{q_s}\right)\right\} \tag{4}$$

且

$$H\left(m^*\right) \notin \left\{H\left(m_1\right), \cdots, H\left(m_{q_b}\right)\right\} \tag{5}$$

在这里我们直接使用哈希值代替消息本身, 这样做虽然降低了模拟器的成功概率. 然而, 如果哈希函数具备抗碰撞性, 那这个差别是可忽略的. 现在, 如果一个哈希函数的输出是均匀分布的话, 该仿真至少有 $\left(1 - \frac{q_s + q_b}{q}\right)$ 的概率是有效的. 当模拟有效时, 我们安全模型下的敌手和攻击 ZSS-VES 方案的伪造者有着相同的优势. 假如敌手在对抗 BVES 方案中有优势 Adv_{BVES}, 模拟器就能以至少 $\left(1 - \frac{q_s + q_b}{q}\right)\text{Adv}_{\text{BVES}}$ 的

概率来伪造一个 VES. 由于 ZSS-VES 方案对任意多项式伪造者来说可伪造的概率是可忽略的, 因此这里的 $\mathrm{Adv}_{\mathrm{BVES}}$ 是可忽略的.

3.4 不可提取性证明

定理 2 如果 ZSS-VES 体制[8]在它们的安全模型下具有不可提取性, 那么本文的 BVES 体制在我们的安全模型下也具有不可提取性.

证明: 挑战者扮作 ZSS-VES 方案[8]不可提取性的一个攻击者. 也就是说, 挑战者把 y_A, y_B 以及消息 m 的 δ_{VES} 作为输入, 并预计产生一个关于 y_A, y_B 以及消息 m 的普通签名. 挑战者运行 BVES 敌手步骤如下:

- 挑战者选择一个随机数 $\alpha \in Z_q^*$, 计算 $s = H_1\left(e\left(P_A, P_B\right)^{\alpha}\right)$, 然后准备参数 $\left(sy_A, sH\left(m\right), y_B\right)$.
- 挑战者计算 $\delta_{\mathrm{BVES}} = \frac{1}{s}\delta_{\mathrm{VES}}$ 并把四元组 $\left(\delta_{\mathrm{BVES}}, sy_A, sH\left(m\right), y_B\right)$ 发送给 BVES 敌手作为挑战.
- 当敌手想要消息 m_i 的 BVES 签名时, 挑战者请求消息 m_i 的 VES 签名 δ_{VES_i} 并计算 $\delta_{\mathrm{BVES}_i} = \frac{1}{s}\delta_{\mathrm{VES}_i}$. 然后 δ_{BVES_i} 被返回给敌手.
- 当敌手想要消息 m_i 的 BVES 签名 δ_{BVES_i} 对应的普通签名时, 挑战者向 VES 仲裁预言机请求 $s\left(\delta_{\mathrm{BVES}_i}\right)$ 获得普通签名 $\delta_{\mathrm{Alice}_i}$, 然后把 $\delta_{\mathrm{Alice}_i}$ 返回给敌手.
- 当这个 BVES 敌手生成了一个普通签名 δ_{Alice} 之后, 挑战者则返回 δ_{Alice} 作为此 ZSS-VES 方案的一个普通签名.

如果模拟生效, 则挑战者和 BVES 敌手有相同的成功概率. 和不可伪造性证明进行同样的分析, 可得这里模拟器至少以 $\left(1 - \frac{q_s + q_h}{q}\right)$ 的概率仿真成功. 由于此 ZSS-VES 方案具有不可提取性, 挑战者能成功的概率可忽略不计, 因此 BVES 敌手能成功的概率也可忽略不计.

4 公平合同签署协议

由于比特币系统是区块链技术应用的第一个成功项目, 我们用比特币交易来展示我们的协议. Alice 和 Bob 想公平地签署一个合同, 他们已经协商好了合同 C 且合同的价值大约是 d B. 我们的协议有以下目标:

- 如果 Alice 和 Bob 交换了合同签名, 那么没有人会失去比特币.
- 如果 Alice 和 Bob 没有交换合同的签名, 那么也没有人会失去比特币.
- 如果仅仅是一方获得了合同的签名, 那么这一方将会失去 d B.
- 区块链上的节点并不能推测出所签署的合同的内容以及验证合同签名使用的公钥.

第 65 号比特币改进提议[11]提出了一个 CHECKLOCKTIMEVERIFY 操作, 允许推迟交易中任意输出的有效时间. 我们把该操作简记为 $\mathrm{CH}\left(\mathrm{para}\right)$, 其中 para 的单位是区块高度或者时间. 我们假设 para 的单位是区块高度. 如果一个交易 T^A 的某个输出脚本中包含 $\mathrm{CH}\left(\mathrm{para}\right)$ 操作, 那么与该输出连接的交易 T^B 应该设置了 LOCKTIME 字段, 并且单位也是区块高度. 如果此时 T^A 中参数 para 的值小于等于 T^B 的 LOCKTIME 字段的值, 则该输出条件已满足, $\mathrm{CH}\left(\mathrm{para}\right)$ 返回 true, 任何其它情况都会返回 false.

为了顺利的通过比特币交易完成协议, 我们要求比特币脚本支持以下的双线性对验证操作.

- $\mathrm{BV}\left(\mathrm{para}_1, \mathrm{para}_2, \mathrm{para}_3, \mathrm{para}_4\right)$: 假设 T^A 包含了这个函数, 公开 $\left(\mathrm{para}_2, \mathrm{para}_3, \mathrm{para}_4\right)$. 假定 T^B 和 T^A

相连接, T^B 的输入脚本包含 para_1. 如果下面等式成立, 则此函数返回 true:

$$e\left(\text{para}_2 P+\text{para}_3, \text{para}_1\right)=e\left(P, \text{para}_4\right) \tag{6}$$

我们当然知道目前的比特币系统并不支持双线性对操作. 但是我们可以使用以太坊平台中的智能合约或者超级账本(Hyperledger)平台上的链码来实现这些功能. 我们只是用比特币交易来描述协议, 以清楚的展示协议的逻辑关系、以及协议和区块链系统的关系. 我们在图 2 中展示公平合同签署协议的主要步骤.

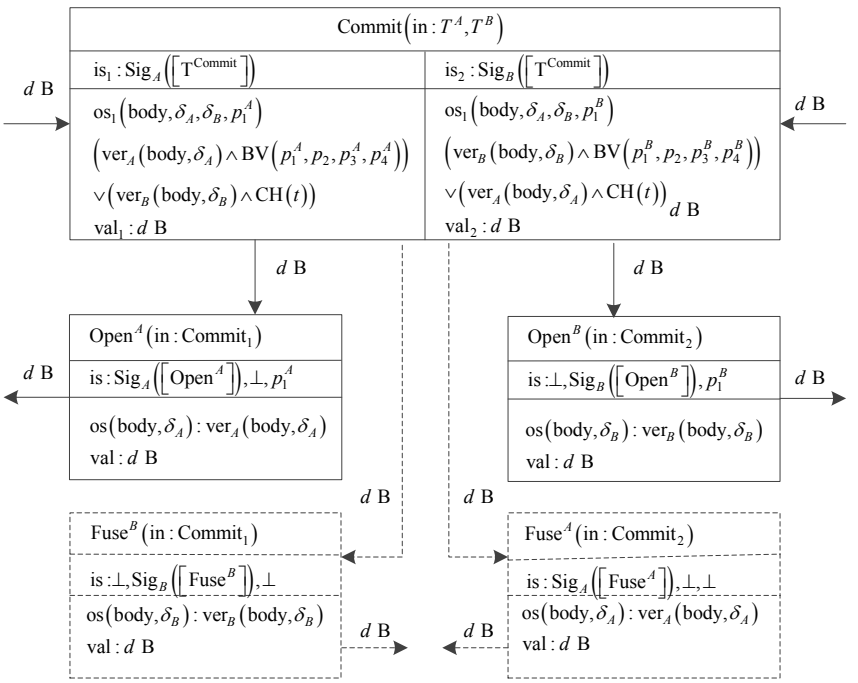


图 2 公平合同签署协议

Figure 2 A fair contract signing protocol

预置条件:

(1) Alice 和 Bob 指定两个交易 T^A 和 T^B, 每一笔都有 d B 的比特币未被花费. Alice 和 Bob 分别有三组密钥对, 一对是比特币密钥, 一对是通过 **KeyGen** 算法生成的临时 BVES 密钥, 最后一对是他们包含在公钥证书中的长期 BVES 密钥. Alice 的长期密钥对是 $\left(x_A, y_A\right)$, 临时密钥对是 $\left(x_A', y_A'\right)$. Bob 的长期密钥对是 $\left(x_B, y_B\right)$, 临时密钥对是 $\left(x_B', y_B'\right)$.

(2) Alice 和 Bob 对合同 C 达成了一致, 并产生了一个合同索引 m_{index}. 他们交换各自的临时 BVES 公钥、他们证书里的长期 BVES 公钥以及他们的比特币公钥. 他们先检查证书的有效性, 如果证书是无效的, 则协议终止, 否则 Alice 运行 **PreSignAgree** 算法和 Bob 来生成一个共享的秘密因子 s. Alice 使用临时 BVES 密钥来生成这个秘密因子. 也就是说, Alice 计算 $s=H_1\left(e\left(y_B', y_A'\right)^\alpha\right)$, Bob 计算 $s=H_1\left(e\left(\alpha P, y_A'\right)^{x_B'}\right)$, 然后他们分别计算 $sH(C)$, sy_A 以及 sy_B.

承诺: Alice 把 T^A 和 T^B 作为输入, 构建 Commit 交易, 如图 2 所示. 其中参数 $p_2=sH(C)$, $p_3^A=sy_A$, $p_4^A=y_B'$, $p_3^B=sy_B$, $p_4^B=y_A'$ 以及 $t=2\max_{BC}$. Alice 对这个交易进行签名然后发送给 Bob. Bob 使用共享的秘

密因子 s、合同 C 以及相关的公钥来核对这个交易的所有参数. 然后 Bob 使用 Alice 的比特币公钥来核对 Alice 对交易的签名. 如果所有核对都通过的话, Bob 对这个交易签名并广播到比特币网络, 否则协议终止.

打开: 如果上述 Commit 交易出现在了比特币区块中, Alice 生成一个 BVES 签名 $\frac{1}{\left(H\left(C\right)+x_A\right)s}y'_B$, 由图 2 中参数 p_1^A 来表示. 然后 Alice 生成图 2 中的 Open^A 交易并广播出去. Bob 也生成一个 BVES 签名 $\frac{1}{\left(H\left(C\right)+x_A\right)s}y'_A$, 由图 2 中参数 p_1^B 表示. 然后 Bob 生成图 2 中的 Open^B 交易并广播出去.

保险: 如果在区块链上 Commit 交易所在区块之后、第 t 个区块之前没有出现 Open^B 交易的话, Alice 生成一个 Fuse^A 交易获得 Bob 的比特币作为赔偿. 同样地, 如果没有出现 Open^A 交易, 则 Bob 也能够生成一个 Fuse^B 交易来获得 Alice 的比特币作为赔偿.

提取: 如果在第 t 个区块之前区块链上存在 Open^B 交易, Alice 运行 **BVESExt** 算法从 Bob 的 p_1^B 参数中提取出普通签名, 既 Alice 计算 $\frac{s}{x'_A}P_1^B$. 同样地, 如果存在 Open^A 交易, 则 Bob 从 Alice 的 p_1^A 参数中提取出普通签名, 既 Bob 计算 $\frac{s}{x'_B}P_1^A$.

4.1 安全性分析

定理 3 如果合同双方都遵从了公平合同签署协议, 那他们要么进行了双方数字签名的秘密交换、要么什么也没有交换、要么一方以花费 d B 的代价来获得另外一方的签名.

证明: 根据我们的公平合同签署协议, Alice 和 Bob 能够构造一个 Commit 交易, 该交易并没有分别包含他们的输入脚本. 这为他们提供了一个机会来确认交易的内容. Alice 构造了一个没有输入脚本的 Commit 交易, 并对此交易进行签名然后发送给 Bob. 这个签名使得 Bob 不用改变除了输入脚本之外的其它内容. Bob 可以检查交易中参数的正确性, 当所有参数都是正确的时候, Bob 对此消息签名并广播出去. 因此当一个 Commit 交易出现在比特币区块中, 表示 Alice 和 Bob 已经确认了被签署的合同以及所使用到的参数. 注意签名是由比特币的私钥来生成, 他们并不是合同签名.

在 Commit 交易出现以后, Alice 提供了一个 BVES 签名来构造 Open^A 交易. 这个交易给 Bob 提供了用于交换的签名. 由于此 BVES 方案的不可伪造性, Alice 需要生成一个有效 BVES 签名来通过 Minter 的验证以及拿回自己的比特币. 另外由于此 BVES 方案的不可提取性, 只有 Bob 才能从 Alice 的 BVES 中提取出合同的普通签名. 如果 Bob 构造自己的 Open^B 交易并广播出去, 对于 Alice 也是同样的情况. 因此如果双方都按照协议在第 t 个区块之前生成自己的 Open 交易, Alice 和 Bob 就能成功交换他们在合同上的签名并顺利拿回自己的比特币.

如果在第 t 个区块之后 Open^A 和 Open^B 交易都没有出现的话, 则他们能分别使用 Fuse^A 和 Fuse^B 交易来拿回自己的比特币.

如果在第 t 个区块之后仅有 Open^A 交易出现的话, 则 Bob 能获取到 Alice 的签名. 然而, 由于区块链上不存在 Open^B 交易, 因此 Alice 能通过 Fuse^A 交易来获得 Bob 的比特币. 相反, 如果仅有 Open^B 交易出现的话, Alice 能获取到 Bob 的签名, Bob 则能获得 Alice 的比特币.

4.2 性能评估

此协议预计能在 $2\max_{BC}$ 加上一个缓冲时间之内执行结束. 假设 $\max_{BC}$ 表示 6 个区块. 在比特币网络中产生一个新区块平均需要十分钟, 因此预计执行完此协议的时间大约是 2 个小时. 当然在其它区块链网

络中实现时不一定需要 2 个小时这么长时间.

Alice 应当给 Bob 发送公钥, 构建共享秘密因子的相关素材以及 Commit 交易的内容, Alice 应当广播一个 Open^A 交易或者一个 Fuse^A 交易, 因此这里大致上有三条单播消息和一条广播消息. Bob 应当给 Alice 发送相关公钥以及广播 Commit 交易, 同样地, Bob 也应当广播一个 Open^B 交易或者一个 Fuse^B 交易, 因此这里大致上有一条单播消息和两条广播消息. 我们把通信花费汇总在表 1 中.

表 1 协议通信花费
Table 1 Communication costs of the protocol

参与者	单播	广播
Alice	3	1
Bob	1	2

5 结论

我们给出了一个基于比特币交易的公平合同签署协议. 此协议在交易的输出脚本中使用了新的验证脚本. 我们也给出了一个匹配的盲的可验证加密签名体制, 并对签名体制和安全协议进行了安全性证明.

References

[1] NAKAMOTO S. Bitcoin: A Peer-to-Peer Electronic Cash System[OL]. http://bitcoin.org/bitcoin.pdf, 2008.
[2] LUU L, NARAYANAN V, ZHENG C, et al. A secure sharding protocol for open blockchains[C]. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 17–30.
[3] WAN Z, DENG R H, LEE D. Electronic contract signing without using trusted third party[C]. In: International Conference on Network and System Security. Springer International Publishing, 2015: 386–394.
[4] LIU J, LI W, KARAME G O, et al. Towards fairness of cryptocurrency payments[J]. arXiv preprint arXiv:1609.07256, 2016.
[5] ASOKAN N, SHOUP V, WAIDNER M. Optimistic fair exchange of digital signatures[J]. IEEE Journal on Selected Areas in communications, 2000, 18(4): 593–610.
[6] ANDRYCHOWICZ M, DZIEMBOWSKI S, MALINOWSKI D, et al. Fair two-party computations via bitcoin deposits[C]. In: International Conference on Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2014: 105–121.
[7] KIAYIAS A, ZHOU H S, ZIKAS V. Fair and robust multi-party computation using a global transaction ledger[C]. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 2016: 705–734.
[8] ZHANG F, SAFAVI-NAINI R, SUSILO W. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings[C]. In: International Conference on Cryptology in India. Springer Berlin Heidelberg, 2003: 191–204.
[9] ZHANG F, SAFAVI-NAINI R, SUSILO W. An efficient signature scheme from bilinear pairings and its applications[C]. In: International Workshop on Public Key Cryptography. Springer Berlin Heidelberg, 2004: 277–290.
[10] BONEH D, GENTRY C, LYNN B, et al. Aggregate and verifiably encrypted signatures from bilinear maps[C]. In: International Conference on the Theory and Applications of Cryptographic Techniques. Springer Berlin Heidelberg, 2003: 416–432.
[11] TODD P. Op_checklocktimeverify[OL]. BIP: 65, 2014. https://github.com/bitcoin/bips/blob/master/bip-0065.mediawiki.

附录

1 ZSS方案

ZSS 方案是一个基础的签名方案. 这里的 Alice 表示一个签名者.

Setup: 设 G_1 是一个循环加法群, P 为生成元, 大素数 q 作为 G_1 的阶. 设 G_2 是一个循环乘法群, G_2 和 G_1 具有相同的阶. 定义 $e: G_1 \times G_1 \to G_2$ 作为一个双线性对, 并定义一个密码哈希函数 $H: \{0,1\}^* \to Z_q^*$.

KeyGen: 随机选择 $x \in_R Z_q^*$ 并计算 $y = xP$. 然后让 y 作为公钥, x 则作为私钥. 此时 Alice 的密钥对表

示为(x_A, y_A).

Sign: 对于一个消息 m , Alice 计算并生成一个普通签名

$$\delta_{\text{Alice}} = \frac{1}{H(m)+x_A}P$$

Verify: 对于一个消息签名对$(m, \delta_{\text{Alice}})$, 验证签名的方程如下:

$$e\left(H(m)P + y_A, \delta_{\text{Alice}}\right) = e(P,P)$$

在群 G_1 中“$k-$CCA 问题”是困难的这个条件下, 他们已经证明了此方案是安全的. $k-$CCA 问题基本来说就是对于一个整数 k 以及 $\alpha \in_R Z_q$, $P \in G_1$,给定以下元组:

$$\left\{P, xP, h_1, \cdots, h_k \in Z_q, \frac{1}{h_1+x}P, \cdots, \frac{1}{h_x+x}P\right\}$$

当 $h \notin \{h_1 \cdots h_k\}$ 的时候, 要求计算出 $\frac{1}{h+x}P$ 的值.

断言 1: 如果存在一个伪造者能对此 ZSS 方案使用自适应选择消息攻击, 那么就会存在一个算法来解决 $k-$CCA 问题.

2 ZSS-VES方案

在他们的方案里有三个参与者, 包括一个签名者 Alice, 一个可信第三方 Bob 以及一个验证者 Charlie.

Setup: 设 G_1 是一个循环加法群, P 为生成元, 大素数 q 作为 G_1 的阶. 设 G_2 是一个循环乘法群, G_2 和 G_1 具有相同的阶. 定义 $e: G_1 \times G_1 \to G_2$ 作为一个双线性对, 并定义一个密码哈希函数 $H: \{0,1\}^* \to Z_q^*$.

KeyGen: 随机选择 $x \in_R Z_q^*$ 并计算 $y = xP$. 然后让 y 作为公钥, x 则作为私钥. 此时 Alice 的密钥对表示为(x_A, y_A), Bob 的密钥对表示为(x_B, y_B).

Sign: 对于一个消息 m , Alice 计算生成一个普通签名

$$\delta_{\text{Alice}} = \frac{1}{H(m)+x_A}P$$

Verify: 对于一个消息签名对$(m, \delta_{\text{Alice}})$, 验证签名的方程如下:

$$e\left(H(m)P + y_A, \delta_{\text{Alice}}\right) = e(P,P)$$

VESSign: Alice 生成一个 VES 签名

$$\delta_{\text{VES}} = \frac{1}{H(m)+x_A}y_B .$$

VESVer: 利用以下元组作为输入:

$$(\delta_{\text{VES}}, m)$$

Charlie 核对下面的等式:

$$e\left(H(m)P + y_A, \delta_{\text{VES}}\right) = e(P, y_B)$$

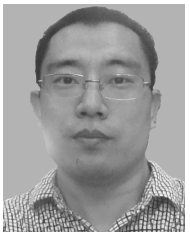
VESExt: Bob 通过计算 $\frac{1}{x_B}\delta_{\text{VES}}$ 来获取到对应的普通签名.

他们已经证明了他们的方案具备不可伪造性以及不可提取性. 也就是说, 攻击者很难去伪造一个 VES 或者是从一个 VES 中提取出普通签名.

断言 2: 如果 ZSS 方案对于存在性伪造是安全的, 那么 ZSS-VES 方案同样对于存在性伪造是安全的.

断言 3: 如果 ZSS 方案对于存在性伪造是安全的且离散对数问题是困难的, 那么 ZSS-VES 方案具有不可提取性.

作者信息



田海博(1979–), 河北衡水人, 博士, 中山大学数据科学与计算机学院副教授. 主要研究领域为安全协议设计与分析.
E-mail: tianhb@mail.sysu.edu.cn

付利青(1992–), 湖南郴州人, 硕士研究生. 主要研究领域为区块链技术及其应用.
E-mail: 15622149705@163.com

何杰杰(1993–), 贵州毕节人, 硕士研究生. 主要研究领域为区块链技术及其应用.
E-mail: hejj9@mail2.sysu.edu.cn