

高效安全的可审计盲混币服务方案

乔康, 汤红波, 游伟, 李海涛

(信息工程大学, 河南 郑州 450001)

摘要: 混币服务能够为区块链隐私泄露问题提供解决方案, 但仍然面临效率瓶颈和安全风险, 为进一步提升混币服务的效率和安全防护能力, 提出一种高效安全的可审计盲混币服务方案。该方案首先增加了审计措施, 在传统的混币模型基础上, 增加审计区块链, 以记录用户和混币器行为, 实现可追溯和可问责; 然后利用椭圆曲线算法构造盲签名, 替代现有研究中基于双线性对或 RSA 的盲签名算法; 最后基于可审计的混币模型和新构造的盲签名算法, 提出可审计的盲混币服务协议。仿真分析表明, 所提方案在提供隐私保护的同时, 具有可审计性、抗盗窃攻击等6种安全特性; 在同等安全强度下, 较对比方案, 所提方法能够有效降低计算开销和存储开销。

关键词: 区块链; 可审计; 混币服务; 隐私保护

中图分类号: TP399

文献标识码: A

doi: 10.11959/j.issn.2096-109x.2020043

Efficient and safe auditable mixed-coin service scheme based on blind signature

QIAO Kang, TANG Hongbo, YOU Wei, LI Haitao

Information Engineering University, Zhengzhou 450001, China

Abstract: The mixed-coin service can provide solutions for the privacy problem of blockchain, but it still faces efficiency bottlenecks and security risks. To further improve the efficiency and security protection of the mixed-coin service, an efficient and safe auditable mixed-coin service scheme based on blind signature was proposed. Firstly, this scheme added audit measures. It added an audit blockchain to the traditional mix-coin model to record the behavior of users and mixers, achieving traceability and accountability. Then, this method used elliptic curves algorithm to construct blind signatures instead of blind signature schemes based on bilinear pairs or RSA. Finally, this scheme proposed an auditable blind mix-coin service agreement based on the auditable mix-coin model and the

收稿日期: 2020-01-07; 修回日期: 2020-04-03

通信作者: 乔康, 773441271@qq.com

基金项目: 国家重点研发计划网络空间专项 (2016YFB0801605); 国家自然科学基金创新群体项目 (61521003); 国家自然科学基金 (61801515)

Foundation Items: The National Key R&D Program Cyberspace Special (2016YFB0801605), The National Natural Science Foundation Innovation Group Project (61521003), The National Natural Science Foundation of China (61801515)

论文引用格式: 乔康, 汤洪波, 游伟, 等. 高效安全的可审计盲混币服务方案[J]. 网络与信息安全学报, 2020, 6(4): 23-36.
QIAO K, TANG H B, YOU W, et al. Efficient and safe auditable mixed-coin service scheme based on blind signature[J]. Chinese Journal of Network and Information Security, 2020, 6(4): 23-36.

blind signature algorithm based on elliptic curve. Simulation analysis shows that the proposed scheme has six security features, such as auditability and anti-theft attack, while providing privacy protection. Under the same security intensity, the proposed scheme can effectively reduce the computational overhead and storage overhead.

Key words: blockchain, auditable, mixed-coin, privacy protection

1 引言

区块链作为一种公开的账本系统, 交易信息被收集并详细记录在其中, 任何参与者都可以查询链上信息, 故其面临严重的隐私泄露风险。为保护用户隐私, 区块链提供一定的假名性, 用户可以在本地, 通过一系列密码学变换, 自行生成与身份信息无关的随机地址。这类随机地址(或假名)通常作为交易输入和输出的账号, 虽然与传统账号相比, 具有较好的匿名性, 但只能提供有限的隐私保护。攻击者通过跟踪和分析区块链交易, 配合地址 ID、IP 信息等可以追查到账户和交易的关联性, 进而推测出交易隐私和身份隐私。例如, Dorit 等^[1]通过分析比特币的历史账本, 得出相关交易之间的许多统计特征。Reid 等^[2]证明多个假名地址可以链接到单个用户地址。Koshy 等^[3]介绍了将比特币地址直接映射到 IP 地址的方法。Miller 等^[4]公布了一种发现比特币网络拓扑以及关键节点的技术 AddressProbe。和传统领域不同, 区块链上记录的信息不可删除和篡改, 敏感信息一旦泄露, 无法挽救, 因此, 区块链系统应该更加重视隐私防护问题, 同时迫切需要为区块链用户提供完整的隐私保护服务。

一种直观的区块链隐私保护方法被称为混币。混币是在不改变交易结果的前提下, 通过对交易内容进行混淆, 从而增加攻击难度的隐私保护方法。2013 年, Maxwell 等^[5]提出第一个混币方案 CoinJoin, 通过割裂输入地址和输出地址的联系, 达到保护用户交易隐私的目的。2014 年, Bonneau 等^[6]提出一种改进的混币方案 Mixcoin, 增加审计功能, 监管混币服务器的行为, 防止混币服务器违规操作。2015 年, Valenta 等^[7]在 Mixcoin 的基础上, 提出一种采用盲签名技术的方案 Blindcoin, 以防止混币服务器泄露混币地址的链接关系。2016 年, Heilman 等^[8]采用盲签名和智能合约技术, 通过智能合约提供可自动强制执行执行的协议, 能同时为链上区块链交易和链下区

块链交易提供匿名保护。2018 年, Bao 等^[9]在 Blindcoin 的基础之上, 利用多签名技术对混币方案进一步优化, 他们设计的 Lockcoin 方案不仅能够防止混币服务器建立输入地址和输出地址间的映射关系, 而且能够预防混币服务器篡改输出地址来盗窃用户资金。

在早期的混币方案中, 如 CoinJoin^[5]和 Mixcoin^[6]方案, 为正确完成混币操作且输出到相应的用户地址, 混币服务器了解全部的混币信息, 用户的输入地址和输出地址对于混币服务器而言是透明的, 存在严重的隐私泄露风险。为防御这类隐私泄露风险, 研究者提出采用盲签名技术的改进方案, 确保混币服务器在正常提供混淆服务的同时, 无法建立输入和输出地址之间的关联性, 如 Blindcoin^[7]、Lockcoin^[9]、RSA-Coinmix^[10]方案。这些采用盲签名技术的方案能够有效抵御混币器泄露区块链用户隐私的风险, 但存在计算和存储开销较大的问题。例如, Blindcoin 和 Heilman 采用基于双线性对(bilinear pairing)的盲签名方案, RSA-Coinmix 和 Lockcoin 则采用基于 RSA 算法的盲签名方案。基于双线性对问题构造的签名单位安全强度高, 同等安全强度下, 所需密钥长度短, 但双线性对运算计算开销较大。而 RSA 算法基于大整数因子分解问题, 基于 RSA 算法构造的签名计算开销较小, 但签名的单位安全强度较低, 同等安全强度下, 所需密钥长度较长, 存储开销较大。目前, 计算和存储开销是限制区块链发展的瓶颈, 因此在满足安全强度的条件下, 需要设计更高效节能的盲签名方案。此外, 混币服务器存在盗窃资金的风险, 而用户也存在拖延支付的可能, 任何一方的违规操作, 都会造成安全性和执行效率下降, 因此, 需要设计可审计的混币协议来同时监管混币器和用户两方的行为, 确保混币服务的安全高效。

本文首先在 Blindcoin^[7]方案的基础之上, 设计了包含用户、混币器、审计区块链三类实体的混币服务系统模型; 其次, 利用仿射变换构造盲

签名的方法, 将 Schnorr 盲签名方案在椭圆曲线上进行模拟, 并采用 3 个随机盲化参数, 构造基于椭圆曲线的 Schnorr 强盲签名方案; 最后, 在混币服务系统模型下, 应用所构造的 Schnorr 强盲签名方案, 并采用经济惩罚机制和公共日志审计措施, 提出了高效安全的可审计盲混币服务协议。

2 背景

2.1 区块链的隐私需求

为了保护用户隐私, 区块链需要满足以下要求: ① 交易之间的链接关系不可见或不可被分析; ② 交易内容仅对实际参与者可见。对于私有链或许可链而言, 通过设置访问控制策略, 可以增加非法节点进入区块链系统的难度, 降低数据透明度, 从而提升隐私防护能力。但是, 区块链中的节点通常是个人计算机或手机, 安全防护能力低, 容易遭受攻击者的攻击。此外, 在公有链场景下, 每个用户都可以自由地访问区块链系统, 能够查看并下载全局账本信息, 存在极大的隐私泄露风险。为抵御风险, 文献[11]将区块链的隐私需求划分为身份隐私和交易隐私两方面。

身份隐私指区块链中的用户信息(包括地址信息和交易内容)和用户真实身份之间的映射关系。在区块链中, 用户可以在本地通过一系列密码学变换, 自行生成与身份信息无关的随机地址。该地址通常作为交易输入和输出的账号, 虽然与传统账号相比, 具有较好的匿名性, 但这类随机地址(或假名)只能提供有限的身份隐私保护。当用户通过随机地址参与区块链交易时, 攻击者通过遍历账本信息, 分析区块链交易的网络传播规律^[12], 配合使用一些行为分析策略^[13], 能够推测出区块链用户的真实身份信息。

交易隐私指区块链交易内容和交易内容背后的敏感信息之间的关联性。公开的交易记录能够反映一些敏感信息, 如用户的购买记录, 能够反映用户的消费水平和购买喜好。此外, 在许多基于区块链的应用中, 如电子医疗记录管理^[14]、匿名的身份验证和授权^[15]等, 区块链上存储的交易记录涉及重要的用户敏感信息。和传统领域不同, 区块链上记录的信息不可删除和篡改, 敏感信息

一旦泄露就无法挽救, 因此区块链系统应该更加重视隐私防护问题。

为提高区块链系统隐私防护的能力, 理想的解决方案应该包括以下几种特征。

1) 匿名性: 用户是唯一知道输入地址和输出地址链接关系的实体。

2) 可扩展性: 该系统具备良好的可扩展性, 能容纳多个用户。

3) 兼容性: 该系统能够兼容现有的区块链系统, 如比特币、以太坊。

4) 可审计性(仅限混币服务): 当协议无法正常执行时, 参与混币的双方都能获取错误操作的证据。

2.2 混币服务

在区块链中, 交易的输入地址和输出地址是可链接的, 通过分析公开账本的内容, 能够推断出一些隐私信息。针对此类攻击, 一种直观的防护方案是借助中间混币器来混淆交易地址的关系。1981年, Chaum 首次提出混币服务的思想^[16], 起初的目的是通过混币器来隐藏参与者的通信内容, 从而实现匿名通信。混币服务的基本思想可表达为

$$K_I(n_0, (K_R(n_1, M), R)) \rightarrow K_R(n_1, M), R \quad (1)$$

式(1)左边代表发送方发送给中间混币器的二次密文, 右边代表中间混币器发送给接收方 R 的一次密文, 中间符号“ $\rightarrow$ ”代表对密文执行的解密操作。假设发送方准备向地址为 R 的接收方发送一条消息 M , 那么发送方首先使用接收方的公钥 K_R 对附加上随机数的消息 (n_1, M) 进行加密, 然后与接收地址 R 打包成一次密文 $(K_R(n_1, M), R)$, 最后用中间混币器的公钥 K_I 对组合包进行加密, 得到二次密文 $K_I(n_0, (K_R(n_1, M), R))$ 。混币器接收到二次密文后, 用自己的私钥 S_I 解密得到 $(n_0, (K_R(n_1, M), R))$, 验证随机数 n_0 后, 将一次密文 $(K_R(n_1, M), R)$ 发送到地址为 R 的用户。用户接收后, 则用自己的私钥 S_R 解密得到 (n_1, M) , 验证随机数 n_1 后, 获得正确的原始消息 M 。

文献[13]介绍了混币服务的基础架构, 如图1所示, 多个输入经中间混币服务器执行混淆操作后依次输出, 为了隐藏输入到达的顺序, 混币器将通过随机排序的方式打乱输出顺序。此外, 为

尽可能减小单个混币器受攻击的风险, 可以将多个混币器连接在一起, 构成级联混币器, 提升防护能力。

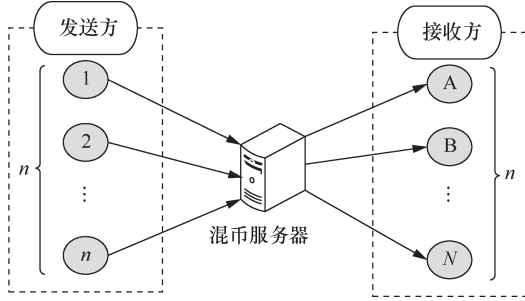


图1 混币服务的基础架构
Figure 1 Infrastructure of mixed-coin service

2.3 盲签名

盲签名是由 Chaum^[17]在 1983 年提出的一种数字签名方式, 其通过允许文档提供者在签名者对文档不可见的情况下, 获得“盲”文档签名的方式, 来为实际文档内容提供隐私。由于文档内容在签名之前对签名者而言是不可见的(或盲化的), 因此盲签名可有效保护签名内容, 在电子选举和数字金融领域已有广泛应用^[18]。文献[19]介绍了一般签名方案和盲签名方案的对比, 如图 2 所示。在图 2(a)所示的一般签名方案中, 签名者根据已知的消息内容产生数字签名。与一般的签名方案相比, 盲签名的过程如图 2(b)所示。签名请求方对消息执行盲化操作, 并将盲消息发送给签名者。签名者签署盲消息并将盲签名返回给请求方, 然后请求方解盲签名以获得原始消息上的签名。

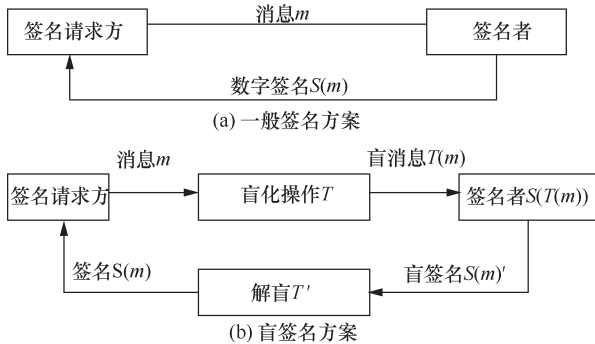


图2 一般签名方案和盲签名方案的对比
Figure 2 Comparison of general signature scheme and blind signature scheme

盲签名可以通过许多公钥加密的方式来实现, 这里介绍 Schnorr 盲签名^[20]的实现方法。假

设 G 是阶为 n 的有限循环群, $g \in G$ 是 G 的生成元, 在 G 中计算离散对数是困难的, 签名者具有公钥 $y = g^x$ 和私钥 x , 假设请求者希望能使用盲签名签署消息 m , 请求者应该与签名者一起执行如下协议。

Step1 签名者随机选择一个整数 $k \in Z_n^*$, 计算 $\tilde{t} = g^k \pmod{n}$, 然后将 $\tilde{t}$ 发送给请求者。

Step2 请求者随机选择两个整数 $\alpha, \beta \in Z_n^*$ 作为盲化因子, 计算 $t = \tilde{t} g^\alpha y^\beta \pmod{n}$, $c = \text{SHA256}(m \parallel t)$, $c' = c - \beta$, 其中, SHA256 是哈希值大小为 256 bit 的哈希函数, c' 是盲消息, 然后将 c' 发送给签名者。

Step3 签名者计算盲签名 $S' = k - c'x \pmod{n}$, 然后将 S' 发送给请求者。

Step4 请求者计算 $S = S' + \alpha \pmod{n}$, 获得消息 m 的签名 (c, S) , 并发送给签名者。

Step5 请求者和签名者都可以通过式(2)验证签名 (c, S) 。

$$c = \text{SHA256}(m \parallel g^S y^c) \quad (2)$$

该方案有效的原因如下。

$$\begin{aligned} g^S y^c &= g^{S'+\alpha} y^{c'+\beta} = g^{S'+\alpha} y^{c'} y^\beta = \\ g^{S'+\alpha} (g^x)^{c'} y^\beta &= g^{S'+c'x} g^\alpha y^\beta = \\ g^k g^\alpha y^\beta &= \tilde{t} g^\alpha y^\beta = t \end{aligned} \quad (3)$$

$$\text{SHA256}(m \parallel g^S y^c) = \text{SHA256}(m \parallel t) = c \quad (4)$$

3 盲混币服务方案

3.1 盲混币服务的系统模型

在 Blindcoin^[7]方案的基础上, 本文设计了盲混币服务的系统模型, 如图 3 所示, 该模型包含 3 个主要实体, 分别是混币服务器 (M, mixer)、用户 (U, user) 和审计区块链 (audit blockchain)。

混币服务器: 混币服务器 M 是混币服务的执行者。混币服务器向用户提供托管地址 R , 用户接收从 U_{in} 输出的待混淆金额 v 。从 U_{in} 至 R 的交易确认后, M 生成另一托管地址 R' , 作为输入地址, 向 U_{out} 转移资金 $v - v\rho$ 。为增加安全性, 降低被攻击风险, 可引入激励机制, 使多个混币器通过竞争的方式获得参与服务的机会。

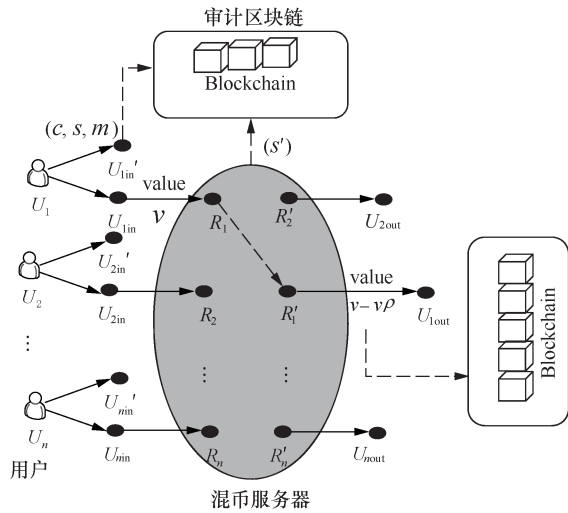


图3 盲混币服务的系统模型
Figure 3 System model of blind mixed-coin service

用户：用户 U 是混币服务的请求者。用户通过混币器，将自己在地址 U_{in} （可能链接到真实身份）上的资金转移到另一地址 U_{out} 。借助混币服务，攻击者很难将 U_{in} 和 U_{out} 联系起来。采用洋葱网络（Tor）^[21]，用户能够通过替代身份 U_{in}' 以匿名的方式发布签名消息到审计区块链。值得注意的是，用户必须确保 U_{in} 和 U_{out} 的绝对独立。

审计区块链：审计区块链是混币服务的监督者，可看作一块只可添加不可更改的公告板，用于第三方验证。审计区块链以区块链的形式分享信息，发送方可通过交易的方式将证据信息记录在区块链上。用户和混币器均可发布区块链消息，并且消息一经发布不可删除。如果用户和混币器中任何一方违反协议，通过审计区块链中包含的公开信息，可证明违规方的行为。

盲混币服务系统模型包含许多参数，具体如表1所示，除了模型图中涉及的参数外，还将在3.3节盲混币服务协议中被大量应用。

3.2 基于椭圆曲线的 Schnorr 强盲签名

采用盲签名技术能够有效抵御混币器泄露区块链用户隐私的风险，但现有的基于盲签名的混币服务方案存在计算和存储开销较大的问题。由于椭圆曲线密码体制的数学基础是椭圆曲线加法群的离散对数问题，与基于大整数因子分解问题的传统公钥密码体制相比，椭圆曲线密码体制的单位安全强度更高，具有密钥长度较短和安全性较高的优势，因此，为降低计算和存储开销，本

文参考文献[22]利用仿射变换构造盲签名的方法，将 Schnorr 盲签名方案扩展到椭圆曲线上。此外，为保证盲签名的安全性，本文采用3个随机盲化参数，构造基于椭圆曲线的 Schnorr 强盲签名方案。假设 G 是阶为 n 的有限循环群， $g \in G$ 是 G 的生成元，在 G 中计算离散对数是困难的，签名者的密钥对是 (Q, d) ， $Q = dG$ ，消息是 m ，Schnorr 强盲签名协议如图4所示。

表1 盲混币服务系统模型参数
Table 1 Model parameters of blind mixed-coin service system

参数	描述
U_{in}	U 的真实地址
U_{out}	U 的目的地址
R	托管地址，对每个用户都是唯一的，由 M 提供给 U
R'	M 用来向 U_{out} 转移资金的托管地址，与 R 无关
U_{in}'	U 的匿名身份，用来向审计区块链发送消息
R_p	审计区块链的地址
P	M 的公钥
d	M 的私钥
Q	U 的公钥
f	U 的私钥
ϖ	M 确认 U 转账交易成功所需的区块数目
nonce	生成不同消息的随机数
v_M	M 预置的押金（ $v_M \gg v_U$ ）
v_U	U 向 M 转移的混币金额
ρ	U 向 M 支付的混币服务费率
t_1	U_{in} 必须在时间 t_1 内向 R 转移 v 资金，才能参与混币
t_2	M 必须在时间 t_2 内，将盲签名 S' 发送到审计区块链
t_3	U_{in}' 必须在 t_3 内，将解盲的签名发送到审计区块链
t_4	R' 必须在时间 t_4 内，将 v 转移到 U_{out}
D	表示一组混币参数 $D = \{t_1, t_2, t_3, t_4, v, \varpi, \rho, v_M\}$

Schnorr 强盲签名协议执行步骤如下。

Step1 签名者随机选择一个整数 $k \in Z_n^*$ ，计算 $R = kG$ ，然后将 R 发送给请求者。

Step2 请求者随机选择3个整数 $\alpha, \beta, \lambda \in Z_n^*$ 作为盲化因子，然后计算 $A = \alpha R + \beta Q + \lambda G = (x, y)$ ， $r = x \pmod n$ 。如果 $r = 0$ ，则 α, β, λ 应该重新选择。请求者计算 $c = \text{SHA256}(m \parallel r)$ ， $c' = \alpha^{-1}(c - \lambda)$ ，

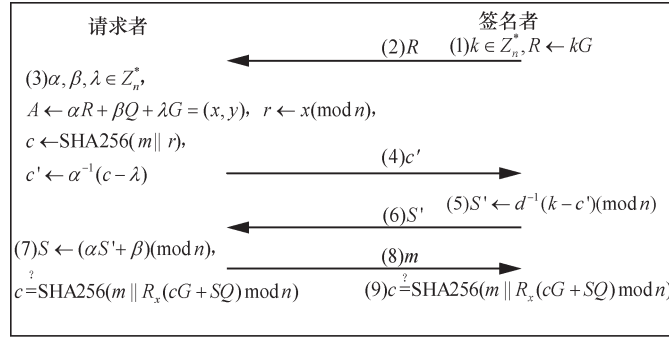


图4 Schnorr 强盲签名协议
Figure 4 Schnorr strong blind signature protocol

其中SHA256是哈希值大小为256 bit的哈希函数， c' 是盲消息，然后将 c' 发送给签名者。

Step3 签名者计算盲签名 $S' = d_A(k - c') \pmod n$ ，其中 $d_A = d^{-1}$ ，然后将 S' 发送给请求者。

Step4 请求者计算 $S = (\alpha S' + \beta) \pmod n$ ，获得消息 m 的签名 (c, S) ，并发送给签名者。

Step5 请求者和签名者都可以通过式(5)验证签名 (c, S) 。

$$c = \text{SHA256}(m \parallel R_x(cG + SQ) \pmod n) \quad (5)$$

可用性证明如下。

$$\begin{aligned} cG + SQ &= (c + (\alpha S' + \beta)d)G = \\ &= (c + (\alpha d^{-1}(k - c') + \beta)d)G = \\ &= (c + \alpha k - \alpha c' + \beta d)G = \\ &= (c + \alpha k - \alpha(\alpha^{-1}(c - \lambda)) + \beta d)G = \\ &= (\alpha k + \lambda + \beta d)G = \alpha R + \beta Q + \lambda G \end{aligned} \quad (6)$$

$$\begin{aligned} \text{SHA256}(m \parallel R_x(cG + SQ) \pmod n) &= \\ \text{SHA256}(m \parallel R_x(\alpha R + \beta Q + \lambda G) \pmod n) &= \\ \text{SHA256}(m \parallel x \pmod n) &= \text{SHA256}(m \parallel r) = c \end{aligned} \quad (7)$$

一般来说，不可伪造性、不可抵赖性、盲性、不可跟踪性4条性质既是设计盲签名所遵循的标准，也是判断盲签名安全性的依据。因此，以这4条性质为标准，对基于椭圆曲线的Schnorr强盲签名方案进行安全分析，该方案具有以下安全特性。

不可伪造性：攻击者（包括签名请求者）想要模拟签名者来签署消息 m ，它可以拦截交互信息 (R, c') ，但它无法获得真实签名者的私钥 d 和随机数 k 。因此，针对本文方案，伪造签名是非常困难的。

不可抵赖性：在该方案中，签名请求者不知道签名者的签名私钥 d 和随机数 k ，因此，签名者对自己签署的消息 S' 无法抵赖。

盲性：在签名者签署消息之前，消息 m 已经被盲化，只有签名请求者才知道盲化因子 α, β, λ 。如果想要从 $c' = \alpha^{-1}(c - \lambda)$ ， $c = \text{SHA256}(m \parallel r)$ 中计算出 m ，需要处理大量的因式分解和离散对数问题，而在前提假设中已经说明 G 中计算离散对数是困难的。因此，签名者无法得知所签署消息的具体内容，具有盲性。

不可跟踪性：签名者接收到消息/签名对 $(m, (c, S))$ 后，通过协议信息 (k, R, c', S') 可以计算 $cG + SQ$ ， $\text{SHA256}(m \parallel R_x(cG + SQ) \pmod n)$ ，由于 $c = \text{SHA256}(m \parallel R_x(cG + SQ) \pmod n)$ ，所以签名者可由消息/签名对 $(m, (c, S))$ 验证签名，即任意一个通过验证的签名可能是经任意一次盲签名产生的。因此，签名者无法判断自己何时签署了何条消息，具有不可跟踪性。

3.3 可审计的盲混币服务协议

本节介绍可审计的盲混币服务协议，该协议是在3.1节介绍的盲混币服务系统模型下，应用3.2节提出的基于椭圆曲线的Schnorr强盲签名方案，并采用经济惩罚机制和审计措施，设计的一种高效安全的服务协议。经济惩罚机制指利用博弈论的思想，通过经济奖惩来制约混币服务器和用户双方的行为。一方面，针对混币服务器，在混币服务器提供混币服务之前，要求混币服务器预置大额诚信押金（远超单次混币金额）作为信用担保。若混币服务器正常提供服务，则在每次混币服务后可获取相应奖励，并且诚信押金可赎

回，若混币服务器执行违规操作，故意延迟或者盗用资金，一经核实，诚信押金将被全部罚没。另一方面，针对用户，混币服务之前用户将混币资金转移到混币服务器，若正常执行流程，则完成混币需求，若故意延迟或恶意发起申请（DoS攻击），一经核实，混币资金将被罚没作为混币服务器的费用。为最大化收益，混币服务器和用户都会遵守协议规则，确保协议的正常执行。审计措施指利用审计区块链不可篡改的特性，记录混币服务器和用户的混币信息，作为审核其行为的可信证据。与现有的方案相比，本文提出的可审计盲混币服务协议，主要的特点是高效性，通过应用基于椭圆曲线的 Schnorr 强盲签名，降低了签名过程的存储开销和计算开销。可审计盲混币服务协议流程如图5所示。

可审计的盲混币服务协议过程具体步骤如下。

Step1 系统初始化。

M 生成密钥对 (P, d) ， U 生成密钥对 (Q, f) ，均公开公钥，并保存私钥。定义一组参

数 $D = \{t_1, t_2, t_3, t_4, v, \varpi, \rho, v_M\}$ ，其中 (t_1, t_2, t_3, t_4) 代表完成不同协议步骤的限定时间， v 表示单次混币金额， ρ 表示 U 向 M 支付的混币服务费， ϖ 表示 M 确认 U 转账交易成功所需的区块数目， v_M 表示 M 向系统预置的押金。定义消息组合 $m = \{U_{out} \| P \| v_U \| nonce\}$ ，其中 U_{out} 表示 U 的地址； P 表示混币器的公钥，用于区分不同混币器； v_U 表示 U 向 M 转移的混币金额 ($v_M \gg v_U$)；nonce 是随机数，用于生成不同的消息。

Step2 混币服务请求。

混币服务开始，用户 U 请求混币服务，向 M 发送请求指令 $\{D, P, v_U\}$ 。

Step3(a) 托管地址分发。

混币器 M 接受请求，发送托管地址。 M 随机选择一个整数 $k \in Z_n^*$ ，计算 $R = kG$ ，然后发送 $\{R, \text{sign}_d(R)\}$ 给 U 。

Step3(b) 混币服务拒绝。

混币器 M 拒绝请求，向 U 发送空消息 $\perp$ 。

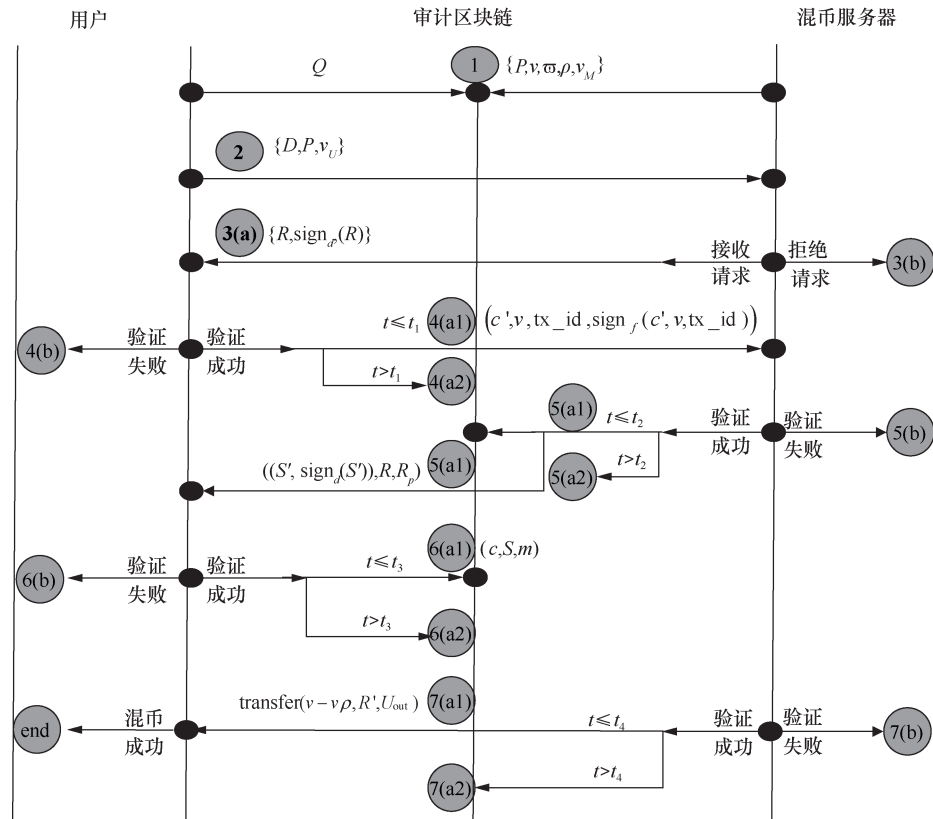


图5 可审计的盲混币服务协议
Figure 5 Auditable blind mixed-coin service protocol

Step4(a1) 盲化。

U 盲化混币消息 m ，并向托管地址 R 转移混币资金。签名 $\text{sign}_d(R)$ 验证成功，在时间 t_1 内，Step(1)中， U 随机选择 3 个整数 $\alpha, \beta, \lambda \in Z_n^*$ 作为盲化因子，然后计算 $A = \alpha R + \beta Q + \lambda G = (x, y)$ ， $r = x(\text{mod } n)$ 。如果 $r = 0$ ，则 α, β, λ 应该重新选择。 U 计算 $c = \text{SHA256}(m \| r)$ ， $c' = \alpha^{-1}(c - \lambda)$ ， c' 是盲消息。Step 2 中， U 从 U_{in} 向 R 转移混币资金，该笔交易记录在区块链上，记为 $\text{transfer}(v, U_{\text{in}}, R)$ ，交易的 ID 号为 tx_id 。Step 1 和 Step 2 完成后， U 发送 $(c', v, \text{tx_id}, \text{sign}_f(c', v, \text{tx_id}))$ 给 M 请求签名。

Step4(a2) t_1 内审核。

如果在时间 t_1 内， U 未能按时向托管地址 R 转移混币资金，则双方终止协议。

Step4(b) 验证。

签名 $\text{sign}_d(R)$ 验证失败，终止协议。

Step5(a1) 盲签名。

混币服务器 M 对盲化后的交易信息 c' 进行盲签名。签名 $\text{sign}_f(c', v, \text{tx_id})$ 和 tx_id 验证成功，在时间 t_2 内， M 计算盲签名 $S' = d^{-1}(k - c')(\text{mod } n)$ ，并将 $(S', \text{sign}_d(S'))$ 发送到 U 和审计区块链。 M 向审计区块链发送盲签名的交易记为 $\text{transfer}((S', \text{sign}_d(S')), R, R_p)$ 。

Step5(a2) t_2 内审核。

如果在时间 t_2 内， M 未能按时将 S' 发送到审计区块链， U 可公开 $\text{transfer}(v, U_{\text{in}}, R)$ 和 $\{R, \text{sign}_d(R)\}$ 作为证据，证明 M 违反协议。 M 的违规操作一经核实，系统将撤销 M 押金赎回请求。

Step5(b) 验证。

签名 $\text{sign}_f(c', v, \text{tx_id})$ 或 tx_id 验证失败，终止协议。

Step6(a1) 解盲。

用户 U 对盲签名内容 S' 记性解盲。签名 $\text{sign}_d(S')$ 验证成功，在时间 t_3 内， U 计算 $S = (\alpha S' + \beta)(\text{mod } n)$ ，得到签名 (c, S) ，并以 U_{in} 的身份将 (c, S, m) 发送到审计区块链。

Step6(a2) t_3 内审核。

如果在时间 t_3 内， U_{in} 未能将 (c, S, m) 发送到审计区块链， M 可公开 $(S', \text{sign}_d(S'))$ ，证明 U 违

反协议。

Step6(b) 验证。

签名 $\text{sign}_d(S')$ 验证失败，终止协议。

Step7(a1) 混币服务完成。

等式 $c = \text{SHA256}(m \| R_x(cG + SQ) \text{mod } n)$ 验证成功，在时间 t_4 内， M 从 R' 向 U_{out} 转移资金，该笔交易记录在区块链上，记为 $\text{transfer}(v - v\rho, R', U_{\text{out}})$ ，混币服务完成。

Step7(a2) t_4 内审核。

如果在时间 t_4 内， M 未能向 U_{out} 转移资金， U 可公开 $\{(c, S, m), S'\}$ ， $\text{transfer}(v, U_{\text{in}}, R)$ ，以及 M 未在 t_4 内完成 $\text{transfer}(v - v\rho, R', U_{\text{out}})$ 作为依据，证明 M 违反协议。

Step7(b) 验证。

等式 $c = \text{SHA256}(m \| R_x(cG + SQ) \text{mod } n)$ 验证失败，终止协议。

4 实验与分析

4.1 强盲签名算法的有效性验证

强盲签名算法实现流程如图 6 所示，主要包括获取全局参数、生成公钥和私钥、获取随机数和盲化因子、盲化、盲签名、解盲、验证这 7 个流程。

盲签名算法的有效性验证实验在配置为 Intel Core i3-3110M @2.40 GHz CPU 和 4 GB 内存的计算机上进行，操作系统为 64 位 Windows7 系统。在 Eclipse 平台上用 Java 语言编程实现基于椭圆曲线的 Schnorr 强盲签名算法，算法实现如图 7 所示。

实验表明，经过盲化、盲签名、解盲、验证后，得到结果为 true，说明本文设计的基于椭圆曲线的 Schnorr 强盲签名算法是可行有效的。

4.2 盲混币服务方案的性能分析

盲混币服务方案的性能分析包括盲签名方案的效率分析和混币服务的开销分析两方面。下面介绍盲签名方案的效率分析，盲签名算法主要分为密钥生成、盲化、盲签名、解盲和验证 5 个步骤，每个步骤都会产生计算开销和存储开销。计算开销以计算时间来衡量，存储开销以密钥长度和盲签名长度来衡量。为保证安全强度，本文设计的基于椭圆曲线的 Schnorr 强盲签名方案采用 256 位密钥长度。为评估本文盲签名算法的计算

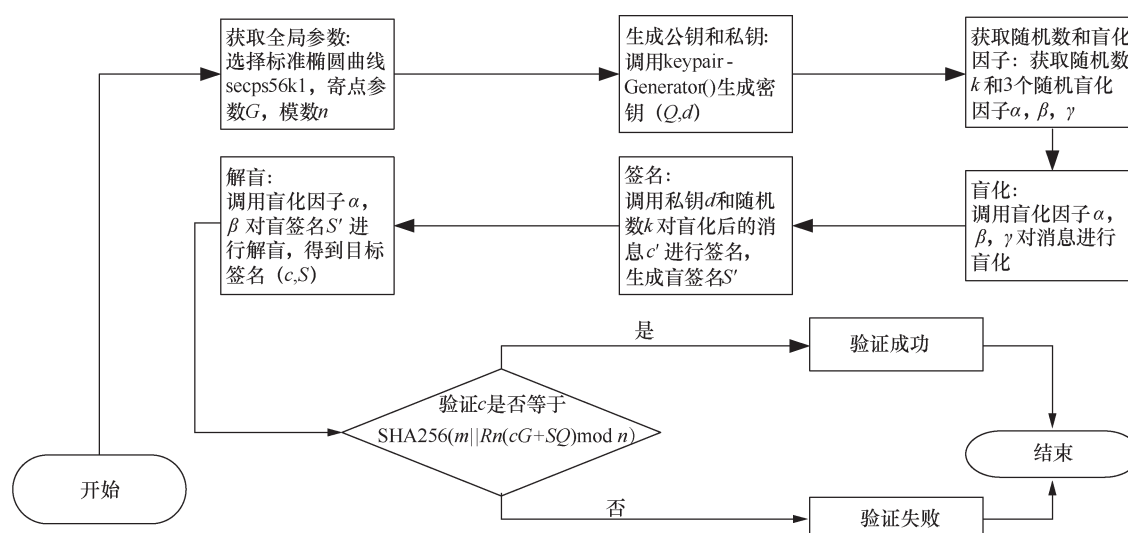


图 6 强盲签名算法实现流程

Figure 6 Strong blind signature algorithm implementation process

```

65 //下面开始正式计算过程
66 // (1) 签名者计算 R=kG
67 ECPoint rPoint = genPoint.multiply(k);
68 盲化 // (2-1) 盲化 用户计算 A=aR+bG+lQ
69 ECPoint aR = rPoint.multiply(alpha);
70 ECPoint bG = genPoint.multiply(beta);
71 ECPoint lQ = pkPoint.multiply(lambda);
72 ECPoint aPoint = aR.add(bG).add(lQ);
73 // (2-2) 计算 r=Rx(A) mod n
74 BigInteger r = getX(aPoint).mod(n);
75 // (2-3) 计算 c=H(m || t) c=H(m || r)
76 byte[] rByte = r.toByteArray();
77 byte[] cByte = hashCombine(data, rByte);
78 BigInteger c = new BigInteger(cByte);
79 // (2-4) 计算 c'-a^(-1) * (c-lambda)
80 BigInteger cPrime = c.add(lambda.negative());
81 BigInteger alphaInv = alpha.modInverse(n);
82 BigInteger cPrime = cPrime.multiply(alphaInv);
83 盲签名 // (3) 盲签名 签名者计算 s'=d^(-1) * (k-c') mod n
84 BigInteger sPrime = k.add(cPrime.negative());
85 BigInteger dm = d.modInverse(n);
86 BigInteger sPrime = sPrime.multiply(dm);
87 解盲 // (4) 解盲 用户计算 s=s'+b. s' 为签名者的盲化
88 BigInteger sPrime = sPrime.multiply(alpha);
89 BigInteger s = sPrime.add(beta);
90 验证 // 签名验证 s=cG+SQ mod n
91 ECPoint cG = genPoint.multiply(c);
92 ECPoint sQ = pkPoint.multiply(s);
93 ECPoint cGsQ = cG.add(sQ);
94 BigInteger cGsQx = getX(cGsQ).mod(n);
95 byte[] cGsQxByte = cGsQx.toByteArray();
96 byte[] testByte = hashCombine(data, cGsQxByte);

```

图 7 强盲签名的算法实现

Figure 7 Algorithm implementation of strong blind signature

效率, 以程序运行时间作为计算时间的实验统计值, 在 4.1 节实验条件下, 分别取用户数目为 10、100、500、1 000、3 000、5 000、7 000、10 000 进行测试, 每次测试重复 10 次取均值, 统计各步骤的运行时间, 统计结果如表 2 所示。

从表 2 的统计结果可以看出各步骤运行时间和用户数目的关系, 为更直观展示其中的关系变化和趋势, 通过 Matlab 绘制分析图, 结果如图 8、图 9 所示。图 8 展示了运行时间占比和用户数目的关系, 当用户数低于 500 时, 密钥生成的运行时间占比最大, 超过 0.5, 随着用户数目的增加, 密钥生成的运行时间占比逐渐减少, 而盲化和验证过程的运行时间占比逐渐增加, 当用户数超过 7 000 时, 密钥生成、盲化和验证的运行时间占比趋于稳定, 都为 0.3 左右。图 9(a)展示了运行总时间和用户数目的关系,

表 2 盲签名方案的计算开销统计

Table 2 Calculation overhead statistics of blind signature scheme

步骤	运行时间/ms							
	U=10	U=100	U=500	U=1 000	U=3 000	U=5 000	U=7 000	U=10 000
密钥生成	733.070 457	607.219 378	1058.532 322	1093.896 64	1 777.432 531	2 620.259 005	3 099.466 574	4 335.089 122
盲化	22.399 536	99.996 325	453.115 392	697.206 227	1 562.697 43	2 751.897 109	3 811.148 79	5 274.156 843
盲签名	0.230 49	0.215 1	0.927 091	1.920 491	6.051 791	6.286 574	9.194 305	12.570 111
解盲	0.023 096	0.074 83	0.257 013	1.063 502	1.028 833	1.664 293	2.385 791	4.632 04
验证	14.771 948	85.232 928	296.170 681	543.565 355	1 382.222 114	2 398.496 338	3 293.995 758	4 763.119 775
总时间	770.495 527	792.738 561	1809.002 499	2 337.652 215	4 729.432 699	7 778.603 319	1 0216.191 22	1 4389.567 89
密钥生成 (平均)	73.307 045	6.072 193	2.117 064	1.093 896	0.5924 77	0.524 051	0.442 78	0.433 508
盲化 (平均)	2.239 953	0.999 963	0.906 23	0.697 206	0.520 899	0.550 379	0.544 449	0.527 415
盲签名 (平均)	0.023 049	0.002 151	0.001 854	0.001 92	0.002 017	0.001 257	0.001 313	0.001 257
解盲 (平均)	0.002 309	0.000 748	0.000 514	0.001 063	0.000 342	0.000 332	0.000 34	0.000 463
验证 (平均)	1.477 194	0.852 329	0.592 341	0.543 565	0.460 74	0.479 699	0.470 57	0.476 311
总时间 (平均)	77.049 552	7.927 385	3.618 004	2.337 652	1.576 477	1.555 72	1.4594 55	1.438 956

随着用户数目的增加,总的运行时间逐渐增加,大致呈线性趋势;图 9(b)展示了平均运行总时间和用户数目的关系,随着用户数目的增加,平均每个用户的运行时间逐渐减少,且当用户数大于 1 000 以后,平均每个用户的运行时间趋于稳定。图 8、图 9 的结果说明本文盲签名方案具有一定的可扩展性,能够支持大规模的用户请求。

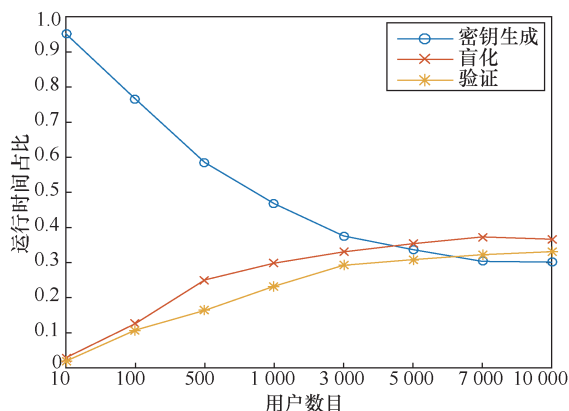
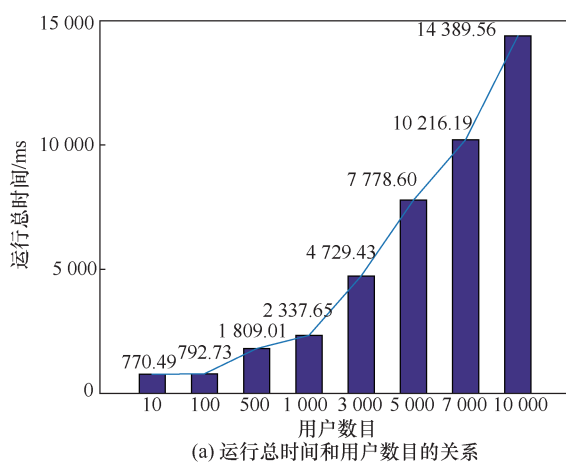
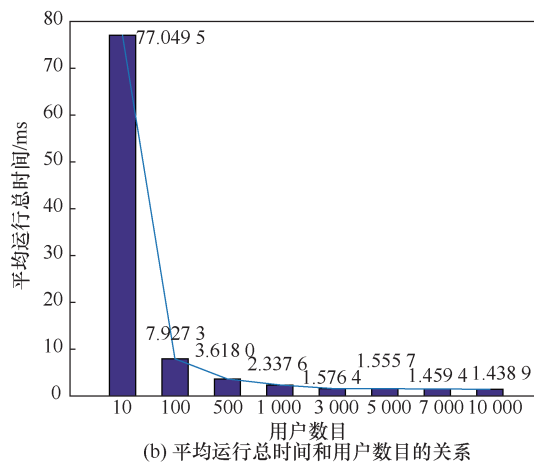


图 8 运行时间占比和用户数目的关系
Figure 8 The relationship between the proportion of running time and the number of users



(a) 运行总时间和用户数目的关系



(b) 平均运行总时间和用户数目的关系

图 9 运行总时间和用户数目的关系
Figure 9 The relationship between the total running time and the number of users requested

表 3 盲签名方案的算法运算过程
Table 3 Algorithm operation process of blind signature scheme

方案	盲化	签名	解盲	验证
本文	$A = \alpha R + \beta Q + \lambda G = (x, y)$ $r = x(\bmod n)$, $S' = d_A(k - c)(\bmod n)$ $c = \text{SHA256}(m \parallel r)$ $c' = \alpha^{-1}(c - \lambda)$		$S = (\alpha S' + \beta)(\bmod n)$	$c = \text{SHA256}(m \parallel R_x(cG + SQ) \bmod n)$
Heilman ^[8]	$\overline{sn} = H(sn)g^r$	$\overline{\sigma} = \overline{sk}$	$\sigma = \overline{\sigma}pk^{-r}$	$e(pk, H(sn)) = e(g, \sigma)$
RSA-Mixing ^[10]	$M = H(Sn)$, $Sn' = Mr^{e_{v_i}}(\bmod n)$	$\delta'(n) = (Sn')^{d_c}(\bmod n)$	$\delta = (\delta'(n))r^{-1}(\bmod n)$	$M = \delta^{e_{v_i}}(\bmod n)$, $H' = H(Sn)$

为进一步评估盲签名方案的效率,选择采用基于双线性对盲签名的 Heilman^[8]方案和基于 RSA 算法盲签名的 RSA-Mixing^[10]方案作为对比,对比结果如表 3 所示。

本文的盲签名算法是基于椭圆曲线上的离散对数问题构造签名密钥,Heilman^[8]方案是基于椭圆曲线上的双线性对问题构造密钥,而双线性对问题是依赖群上的离散对数问题,因此,本质上本文方案和 Heilman^[8]方案都基于椭圆曲线上的离散对数问题构造签名密钥。RSA-Mixing^[10]方案则基于大整数因子分解问题构造密钥。因为基于椭圆曲线上的离散对数问题构造的签单位安全强度高于基于大整数因子分解问题构造的签名,所以在提供同等安全强度时,前者所需密钥长度更短。在本文的对比实验中,本文方案和 Heilman^[8]方案选择 256 bit 的密钥长度, RSA-Mixing^[10]方案则选择 1 024 bit 密钥长度。

为更直观对比各盲签名方案的计算和存储开销,设 P_a 表示双线性对运算, P_m 表示标量乘运算, P_e 表示指数运算, P_c 表示求逆运算, P_d 表示加法

运算, H 表示哈希运算, 将表 3 中各步骤的运算过程进行量化, 得到计算开销和主要数据长度对比结果, 如表 4 所示。其中存储开销为主要数据长度。

取 $n=1\,000$, 代表 10 000 个用户请求盲签名, 在 4.1 节实验条件下, 分别部署 3 个盲签名算法, 每次实验重复 10 次取均值, 得出的对比结果如图 10 所示。

从对比结果可以看出, RSA-Mixing^[10]方案计算时间最长, 且耗时集中在密钥生成阶段, 这是因为该方案基于大整数因子分解问题, 签名的单位安全强度较低, 同等安全强度下, 所需的密钥长度最长, 而密钥长度的增加会导致加解密速度大大降低; Heilman^[8]方案和本文方案计算时间较少, 且耗时主要在盲化、盲签名和验证阶段, 其中盲化和盲签名阶段两方案耗时相近, 验证阶段 Heilman^[8]耗时约为本文方案的 10 倍, 这是由于

在验证阶段, Heilman^[8]方案中双线对运算的计算量较大。此外, Heilman^[8]方案和本文方案的签名单位安全强度较高, 同等安全强度下, 所需的密钥长度较短, 因此存储开销小于 RSA-Mixing^[10]方案。

下面参考文献[9]的分析方法介绍混币服务开销分析。混币服务的过程中, 除了签名过程的计算时间开销以外, 耗时最多的是等待交易确定的过程。以本文方案为例, 每一轮成功的混币操作, 需要 U_{in} 至 R 和 R' 至 U_{out} 的两笔资金转移交易, R 至 R_p 和 U_{in}' 至 R_p 的两笔审计区块链消息交易, 以及一笔 M 预付押金的交易, 共 5 笔交易。本文方案和对比方案都采用工作量证明 (PoW) 的共识机制, 因此每笔交易成块上链时间约为 10 min, 等待 ϖ (通常 $\varpi=6$) 区块后交易得到确认。此外, 提供混币服务需要收取一定的费用, 花销和混币金额成比例, 为 $v\rho$ 。各方案混币服务开销对比结果如表 5 所示。

表 4 计算和存储开销的量化比较
Table 4 Quantitative comparison of calculation and storage overhead

运算过程	本文		Heilman ^[8]		RSA-Mixing ^[10]	
	计算开销	主要数据长度/bit	计算开销	主要数据长度/bit	计算开销	主要数据长度/bit
盲化	$n(H+4P_m+P_c+3P_d)$	消息摘要 $c'=256$	$n(H+2P_m+P_e)$	消息摘要 $H(sn)=256$	$n(H+P_m+P_e)$	消息摘要 $H(sn)=256$
签名	$n(P_d+P_m+P_c)$	盲签名 $S'=256$	$n(P_e+P_m)$	盲签名 $\bar{\sigma}=256$	$n(P_e+P_m+P_c)$	盲签名 $\delta'(n)=1\,024$
解盲	$n(P_m+P_d)$	解盲后的签名 $S=256$	$n(2P_m+P_e)$	解盲后的签名 $\sigma=256$	$n(P_m+P_c)$	解盲后的签名 $\delta(n)=1\,024$
验证	$n(H+P_d+2P_m)$	生成验证结果 $c=256$	$n(H+P_d)$	生成验证结果 $e=256$	$n(H+P_e+P_m)$	生成验证结果 $M'=256$

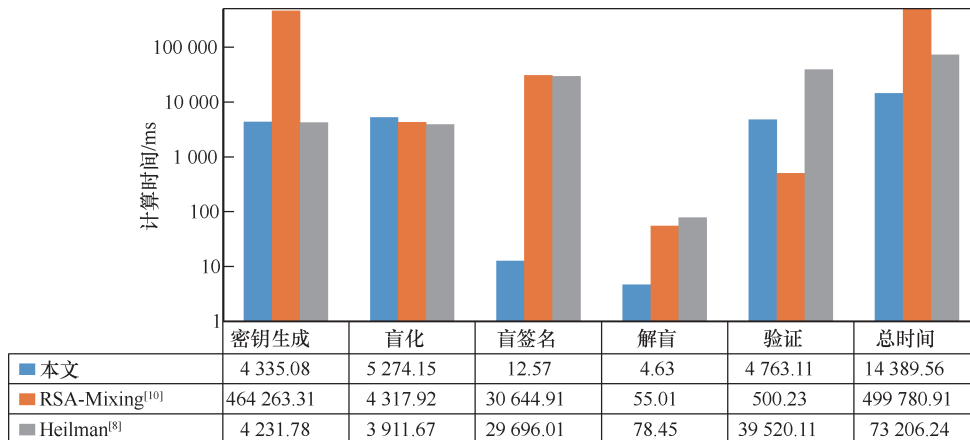


图 10 各盲签名方案计算时间对比
Figure 10 Comparison of computing time of each blind signature scheme

表 5 混币服务开销对比
Table 5 Comparison of mixed-coin service expenses

方案	每轮交易量	轮数	时间	费用	可审计	防盗窃攻击
本文方案	5	1	$10 \cdot \varpi \cdot 5$	$v\rho$	√	√
Lockcoin ^[9]	6	1	$10 \cdot \varpi \cdot 6$	$v\rho$	√	√
Blindcoin ^[7]	4	1	$10 \cdot \varpi \cdot 4$	$v\rho$	√	×
RSA-Coinmix ^[10]	2	1	$10 \cdot \varpi \cdot 2$	$v\rho$	×	×
Blind-mixing ^[20]	2	1	$10 \cdot \varpi \cdot 2$	$v\rho$	×	×

表 5 中所列方案都是基于盲签名的混币服务方案, 在所有对比方案中, RSA-Coinmix^[10]和 Blind-mixing^[19]时间开销最少, 但不可审计, 也不能防止盗窃攻击, 安全性较低。Lockcoin^[9]、Blindcoin^[7]和本文方案都具有可审计性, 且 Blindcoin^[7]时间开销最少, 但 Blindcoin^[7]不能防止盗窃攻击, 存在风险。Lockcoin^[9]和本文方案都具有可审计性, 也能防止盗窃攻击, 但本文方案的时间开销更低, 效率更高。此外, 所有对比方案的混币费用开销相同。

4.3 混币服务方案的特性分析

以理想的区块链隐私匿名解决方案为衡量标准, 讨论本文所提可审计盲混币服务方案, 分析其具有的特性, 包括匿名性、防 DoS 攻击、防盗窃攻击、可审计性、可扩展性和兼容性。

匿名性: 混币服务协议最重要的特性是匿名性, 而匿名程度则以不可链接性和不可追踪性来衡量。

(1) 不可追踪性

混币服务器通过不同的托管地址 R 和 R' , 分别接收和发送混币资金, 确保用户输入地址 U_{in} 和输出地址 U_{out} 在区块链交易中无法相互连接, 因此输入和输出地址是不可追踪的。如图 11 所示, 攻击者知道用户 U_n 的地址, 通过区块链公开账本追踪交易, 可以得到从 U_{min} 到 R_n 和 R_n' 到 U_{out} 的交易, 但攻击者无法将 U_{min} 和 U_{out} 联系起来。

(2) 不可链接性

服务器通过审计区块链获得 U_{in}' 发送的 (c, S, m) , 在成功验证等式 $c = \text{SHA256}(m \parallel R_x(cG + SQ) \bmod n)$ 后, 转移混币资金。由于 U_{in} 与 U_{in}' 无关联性, 且单次混币资金是固定的, 因此服务器无法仅凭金额进行去匿名化, 必须从原始消

息和签名中推断出盲签名, 才能将签名和盲签名联系起来。然而, 要做到这一点, 服务器必须知道 3 个盲化因子 α, β, λ , 而这 3 个盲化因子只有用户知道, 因此本文所提混币协议具有不可连接性。

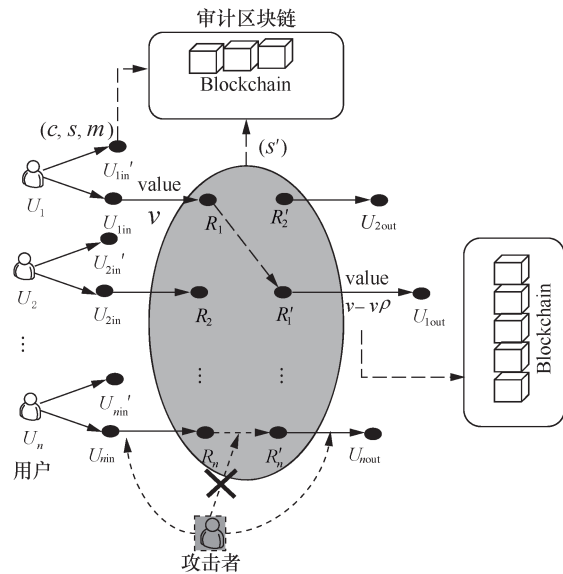


图 11 追踪攻击分析
Figure 11 Tracking attack analysis

防 DoS 攻击: 在许多分布式混币协议中, 攻击者进行 DoS 攻击主要通过用户参与协议到一定程度后, 拒绝转移资金, 导致操作失败。在本文方案中, 每个用户只与混币器交互, 拒绝遵守协议, 不影响其他用户或减慢混币过程。攻击者还可以尝试通过阻止资金交易和审计区块链信息上链的方式, 来进行 DoS 攻击。但这种方式是很难实际操作的, 因为攻击者很难控制大部分的矿池算力。此外, 由于参与混币服务需要收取费用, 针对混币器发动 DoS 攻击, 会给攻击者带来巨大的经济负担。

表6 不同方案的特性比较
Table 6 Comparison of characteristics of different schemes

方案	不可追踪性	不可连接性	防 DoS 攻击	防盗窃攻击	可审计性	可扩展性	兼容性
本文方案	√	√	√	√	√	√	√
Coinjoin ^[5]	√	×	×	√	×	弱	√
Coinshuffle ^[24]	√	弱	√	√	×	弱	√
Valueshuffle ^[25]	√	√	×	√	×	弱	√
ZeroCoin ^[26]	√	√	√	√	×	√	×
Zerocash ^[27]	√	√	√	√	×	√	×
Mixcoin ^[6]	√	弱	√	×	√	√	√
Blindcoin ^[7]	√	弱	√	弱	√	√	√
Blindmixing ^[19]	√	√	×	×	×	√	√
Lockcoin ^[9]	√	√	√	√	√	√	√

防盗窃攻击：由于混币服务器支付了远超过单次混币金额的押金，且协议中包含可审计的问责机制，混币器违规盗用混币资金将得不偿失，从经济角度防止服务器盗窃混币资金。

可审计性：本文方案中使用审计区块链作为审计日志，用户和混币服务器都需要遵守协议，在规定的时间内，执行相应步骤。当某一方违反协议时，第3.3节中的步骤5(a2)、步骤6(a2)、步骤7(a2)可以确保协议被正确问责。

可扩展性：与Blindcoin^[7]方案相似，混币服务器可支撑扩展到更多用户，因为用户只与中心混币器交互，而不与彼此交互。此外，如果一个服务器达到瓶颈，则可以将混币功能负载到几个不同服务器上，所有的服务器均使用相同的加密密钥进行操作。

兼容性：本文方案的混币协议可作为区块链的一种服务，与现有区块链系统兼容性高，如比特币系统。

文献[23]讨论了主流的混币服务方案具备的特性，本文选取了其中几种典型方案^[5-7,24-25]作为比较。此外，虽然在文献[23]中未讨论，但是比较典型^[26-27]和新型^[9-10]的方案，本文也列出一并进行比较，结果如表6所示。

从对比结果可知，与其余方案相比，ZeroCoin^[26]、Zerocash^[27]、Lockcoin^[9]和本文方案具有不可追踪性、不可连接性、防DoS攻击、防盗窃攻击4项安全特性，具备更高的安全防护能力；而ZeroCoin^[26]和Zerocash^[27]缺少可审计性和

兼容性，存在问责困难和部署复杂的问题，Lockcoin^[9]和本文方案则具有全部的6项特性；由4.2节混币服务开销分析可知，相较于Lockcoin^[9]，本文方案开销更低，效率更高。

5 结束语

本文介绍了一种高效安全保护隐私的混币服务方案。该方案不仅能割裂输入地址和输出地址的链接，实现隐私保护的目，具备较低的计算和存储开销，并且拥有良好的安全特性，包括匿名性、防DoS攻击、防盗窃攻击、可审计性等。虽然所提方案具有较高的效率和安全性，但在等待交易的确定过程中，由于采用了工作量证明共识机制，存在交易时延较大的问题，未来可以继续开展对共识机制的优化研究，以提升执行效率。

参考文献：

- [1] DORIT R, SHAMIR A. Quantitative analysis of the full bitcoin transaction graph[C]//International Conference on Financial Cryptography and Data Security. 2013: 6-24.
- [2] REID F, HARRIGAN M. An analysis of anonymity in the bitcoin system[M]//Security and privacy in social networks. 2013: 197-223.
- [3] KOSHY P, KOSHY D, MCDANIEL P. An analysis of anonymity in bitcoin using P2P network traffic[C]//International Conference on Financial Cryptography and Data Security. 2014: 469-485.
- [4] MILLER A, LITTON J, PACHULSKI A, et al. Discovering bitcoin's public topology and influential nodes[R]. 2015.
- [5] MAXWELL G. CoinJoin: bitcoin privacy for the real world[C]//Post on Bitcoin Forum. 2013.

- [6] BONNEAU J, NARAYANAN A, MILLER A, ET al. Mixcoin: anonymity for bitcoin with accountable mixes[C]//International Conference on Financial Cryptography and Data Security. 2014: 486-504.
- [7] VALENTA L, ROWAN B. Blindcoin: blinded, accountable mixes for bitcoin[C]//International Conference on Financial Cryptography and Data Security. 2015: 112-126.
- [8] HEILMAN E, BALDIMTSI F, GOLDBERG S. Blindly signed contracts: anonymous on-blockchain and off-blockchain bitcoin transactions[C]//International Conference on Financial Cryptography and Data Security. 2016: 43-60.
- [9] BAO Z, WANG B, ZHANG Y, et al. Lockcoin: a secure and privacy-preserving mix service for bitcoin anonymity[J]. International Journal of Information Security, 2018, 19(3): 311-321.
- [10] 吴文栋. 基于盲签名技术的比特币混币系统设计与实现[D]. 深圳: 深圳大学, 2015.
- WU W D. Bitcoin mix system design based on partial blind signature[D]. Shenzhen: Shenzhen University, 2015.
- [11] FENG Q, HE D, ZEADALLY S, et al. A survey on privacy protection in blockchain system[J]. Journal of Network and Computer Applications, 2019, 126: 45-58.
- [12] BIRYUKOV A, KHOVRATOVICH D, PUSTOGAROV I. Deanonymisation of clients in Bitcoin P2P network[C]//Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. 2014: 15-29.
- [13] SCHOTT P A. Reference guide to anti-money laundering and combating the financing of terrorism[M]. The World Bank, 2006.
- [14] DUBOVITSKAYA A, XU Z, RYU S, et al. Secure and trustable electronic medical records sharing using blockchain[C]//AMIA Annual Symposium Proceedings. American Medical Informatics Association. 2017: 650.
- [15] YAO Y, CHANG X, MIŠĆ J, et al. BLA: blockchain-assisted lightweight anonymous authentication for distributed vehicular fog services[J]. IEEE Internet of Things Journal, 2019: 3775-3784.
- [16] CHAUM D L. Untraceable electronic mail, return addresses, and digital pseudonyms[J]. Communications of the ACM, 1981, 24(2): 84-90.
- [17] CHAUM D. Blind signatures for untraceable payments[C]//Advances in Cryptology. 1983: 199-203.
- [18] ISLAM S K H, AMIN R, BISWAS G P, et al. Provably secure pairing-free identity-based partially blind signature scheme and its application in online e-cash system[J]. Arabian Journal for Science and Engineering, 2016, 41(8): 3163-3176.
- [19] SHENTU Q C, YU J P. A blind-mixing scheme for Bitcoin based on an elliptic curve cryptography blind digital signature algorithm[J]. Computer Science, 2015.
- [20] CAMENISCH J L, PIVETEAU J M, STADLER M A. Blind signatures based on the discrete logarithm problem[C]//Workshop on the Theory and Application of Cryptographic Techniques. 1994: 428-432.
- [21] DINGLEDINE R, MATHEWSON N, SYVERSON P. Tor: the second-generation onion router[J]. Journal of the Franklin Institute, 2004, 239(2): 135-139.
- [22] 王化群, 张力军, 赵君喜. 基于椭圆曲线的 Schnorr 盲签名[J]. 计算机工程与设计, 2005, 26(007): 1819-1822.
- WANG H Q, ZHANG L J, ZHAO J X. Schnorr blind signature based on elliptic curve[J]. Computer Engineering and Design, 2005, 26(7): 1819-1822.
- [23] CONTI M, KUMAR E S, LAL C, et al. A survey on security and privacy issues of bitcoin[J]. IEEE Communications Surveys & Tutorials, 2018, 20(4): 3416-3452.
- [24] RUFFING T, MORENO-SANCHEZ P, KATE A. Coinshuffle: Practical decentralized coin mixing for bitcoin[C]//European Symposium on Research in Computer Security. 2014: 345-364.
- [25] RUFFING T, MORENO-SANCHEZ P. ValueShuffle: mixing confidential transactions for comprehensive transaction privacy in bitcoin[C]//International Conference on Financial Cryptography and Data Security. 2017: 133-154.
- [26] MIERS I, GARMAN C, GREEN M, et al. Zerocoin: Anonymous distributed e-cash from bitcoin[C]//2013 IEEE Symposium on Security and Privacy. 2013: 397-411.
- [27] SASSON E B, CHIESA A, GARMAN C, et al. Zerocash: Decentralized anonymous payments from bitcoin[C]//2014 IEEE Symposium on Security and Privacy. 2014: 459-474.

[作者简介]



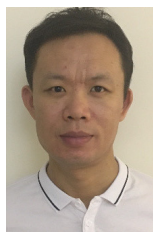
乔康 (1994—), 男, 四川邛崃人, 信息工程大学硕士生, 主要研究方向为移动通信网络安全和区块链技术。



汤红波 (1968—), 男, 湖北孝感人, 信息工程大学教授、博士生导师, 主要研究方向为移动通信网络、新型网络体系结构。



游伟 (1984—), 男, 江西丰城人, 博士, 信息工程大学讲师, 主要研究方向为密码学和移动通信网络。



李海涛 (1982—), 男, 山东泰安人, 信息工程大学讲师, 主要研究方向为网络数据挖掘。