一种匿名可撤销的比特币混淆方案

李雪莲*① 王海玉① 高军涛② 李 伟①
①(西安电子科技大学数学与统计学院 西安 710071)
②(西安电子科技大学通信工程学院 西安 710071)

摘 要：为解决用户在混币过程中无法请求退出的问题，该文提出一种支持用户匿名撤销混币的方案。采用承诺技术将用户和其目的地址进行绑定；当用户请求退出混洗服务时，利用累加器和知识签名对承诺进行零知识证明。最后将撤销用户的混淆输出地址修改为其指定的目的地址。安全性分析表明，该方案基于双离散对数问题和强RSA假设满足退群用户匿名性，且不用修改当前比特币系统即可实施。在$n(n\geq 10)$个诚实用户参与的混淆过程中，方案允许至多$n-2$个用户退出混币操作。

关键词：隐私保护；比特币混淆；可撤销
中图分类号：TN918 **文献标识码**：A **文章编号**：1009-5896(2019)08-1815-08
DOI: 10.11999/JEIT180874

Anonymous Revocation Scheme for Bitcoin Confusion

LI Xuelian① WANG Haiyu① GAO Juntao② LI Wei①
①(*School of Mathematics and Statistics*, *Xidian University*, *Xi'an* 710071, *China*)
②(*School of Telecommunications engineering*, *Xidian University*, *Xi'an* 710071, *China*)

Abstract: In order to solve the problem that users can not request to exit during the bitcoin confusion process, an anonymous revocation scheme for Bitcoin confusion is proposed. The commitment is used to bind the user with its destination address. When the user requests to quit the shuffle service, a zero-knowledge proof of the commitment is made using the accumulator and the signatures of knowledge. Finally, the shuffled output address of the user who quits the service is modified to its destination address. Security analysis shows that the scheme satisfies the anonymity of the user who quits the service based on the double discrete logarithm problem and the strong RSA assumption, and can be implemented without modifying the current bitcoin system. The scheme allows at most $n-2$ users to exit in the confusion process of n ($n\geq 10$) honest users participation.

Key words: Privacy protection; Bitcoin confusion; Revocable

1 引言

比特币系统的假名机制所提供的匿名性难以抵抗交易图谱分析或找零攻击，这使用户的隐私信息如个人总资产等，容易被攻击者获取[1]。按照是否存在第三方(如混币服务器)，已有的提高比特币系统匿名性的方法可以分为基于中心节点的混币机制和去中心化的混币机制[2]。在中心化混淆中，多个用户共同组建一笔交易，该交易包括大量输入输出，通过将用户比特币发送到混币服务器，在现有的用户账户和混币后的新账户之间创建随机的映射关系，使得很难在输入和输出中找出每个人的对应对，从而割裂了输入与输出之间的事实联系。基于这一思想，Maxwell[3]提出CoinJoin方案。通过混淆多个交易发起者的输出地址，以切断交易输入地址与输出地址之间一对一，多对一，多对多等关系。Bonneau等人[4]也引入一个使用可信第三方的混币方案Mixcoin。在上述方案中，尽管外部敌手无法追踪混洗后的资金流，但对混洗服务器而言，用户无法保持其匿名性。为此，Heilman等人[5]提出TumbleBit方案，允许各方通过一个不受信任的中间人Tumbler进行快速，匿名的非区块链支付。这种方法可以很容易地扩展到大型混币群，但存在Tumbler不返还用户资金的风险。同时由于引入了基于盲签公平交换的智能合约和聚合签名技术，TumbleBit需要更高的交易费用。

收稿日期：2018-09-07；改回日期：2018-12-09；网络出版：2019-02-26
*通信作者： 李雪莲 xuelian202@163.com
基金项目：国家重点研发计划(2016YFB0800601)，国家自然科学基金(61303217, 61502372)
Foundation Items: The Nation Key Research and Development Program of China (2016YFB0800601), The National Natural Science Foundation of China (61303217, 61502372)

去中心化的混币机制通常由参与混币的用户合作完成。2014年，Ruffing等人[6]提出CoinShuffle协议，通过使用解密混合网络[7]对地址进行混洗来抵抗内部攻击者。之后他们使用基于DC-net变体的DiceMix改进了该协议，引入CoinShuffle++[8]，将混币协议的通讯轮数优化到4+2f轮，其中f是混币群中恶意用户的个数。但比特币最大交易(100 kb)限制了CoinShuffle++每次只能约混淆538个用户。另外由于缺乏相应恢复机制，用户参与混币后恶意取消将会导致Dos攻击。2015年，Ziegeldorf等人[9]提出CoinParty方案，利用门限椭圆曲线数字签名算法和设置去中心的混币方来实现地址的不可链接性。但它采用的混币方并非交易方，因此当混币方离线时易导致存放在托管地址中的资金无法退还等问题。之后他们更新了CoinParty[10]，采用安全多方计算[11]的方法重构托管地址，同时对不诚实的混币方引入检测算法。此外还存在一些技术，如聚合签名[12]，离链储存[13]等可用于类比特币系统的隐私保护。Miers等人[14]提出一个类比特币方案Zero-Coin，采用基于强RSA假设的累加器[15]和知识签名[16]零知识证明交易方拥有一个有效的序列号，从而保证其匿名性。由于比特币网络尚不支持加密交易，因此无法在当前比特币系统上实施这些方法。

尽管对比特币系统的匿名性研究已经有很多贡献，但在不修改该系统的前提下部署这些协议依然是一个悬而未决的问题。一方面，这些协议的匿名强度更取决于参与混币用户的数目，即匿名集的大小。另一方面用户一旦决定参与混币就无法再退出，这种灵活性和便捷性的缺失限制了现有方案的大规模实际应用。因此，如何在混币操作中实现用户可随时撤销交易，且在撤销时保证用户的匿名性和隐私信息的安全性，成为当前一个亟需解决的问题。

本文提出一种支持用户在比特币混淆过程中匿名退出的方案(CoinExit)，该方案大大提高了用户混币的灵活度。其中退出算法和验证算法确保了用户不会因为退群操作而泄露隐私，同时也保证了混洗群内的其他用户无法知晓谁退出了混币操作。方案采用罚金机制和n时间锁定交易，当恶意用户发起Dos攻击时，强制执行惩罚和恢复协议，以保证诚实用户的资金得以返还，恶意用户将会受到相应惩罚。安全性分析表明文中方案可以抵抗敌手利用伪造的知识签名冒充群中用户退出混洗的攻击。在每个用户相互不知道其他用户指定的目的地址的前提下，方案支持$n(n \geq 10)$个诚实用户中至多$n-2$个用户退出。

2 可匿名撤销的混币方案

2.1 方案模型

图1是方案的系统模型图。假设有n个用户$P_1, P_2, \cdots, P_n$参与混币操作，其中最后一个用户P_n是诚实的。方案包括5部分：用户的输入地址$I_1, I_2, \cdots, I_n$；用于临时存储用户比特币的托管地址T；目的地址$O_1, O_2, \cdots, O_n$是每个用户事先指定的输出地址；混洗输出列表$O_{\tau_i(1)}, O_{\tau_i(2)}, \cdots, O_{\tau_i(n)}$，用于给用户返回存储在托管地址上的比特币，其中$\tau_i$包含一系列随机排列；承诺列表$c_1, c_2, \cdots, c_n$，用于请求退出混洗的用户进行身份证明。假设$n$个用户的总交易金额为$nx$个比特币。此外，每个用户在混币前需缴纳相同数量的罚金x_1。

2.2 方案构造

假定G_1是群Z_p^*上一个阶为q的子群，其中$p = 2^\omega q + 1$, $\omega \geq 1$, p, q为素数；g_1, g_2是群G_1的生成元；G_2是椭圆曲线secp256k1上的点构成的一个阶为q的交换群，$h \in G_2$是椭圆曲线的生成元；$\mathrm{H}: \{0,1\}^* \to Z_q$是一个抗碰撞的哈希函数。

方案的流程如图2所示。

(1) 初始化Setup:给定安全参数λ，计算累加器参数$N = pq$和种子值$u \in Z_N^*, u \neq 1$。输出系统公

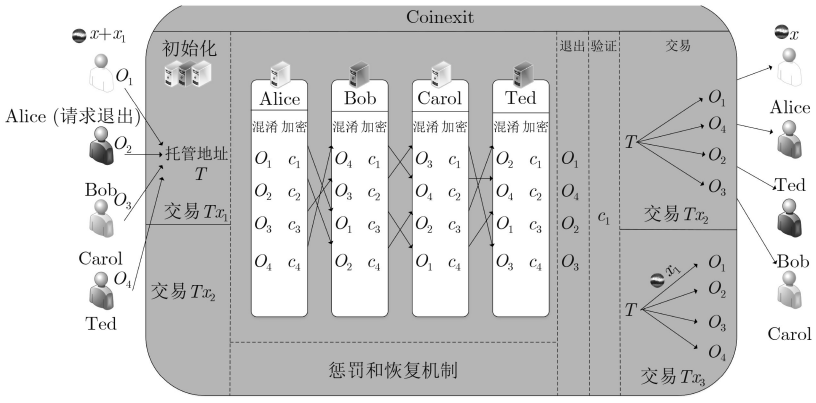

图 1 系统模型

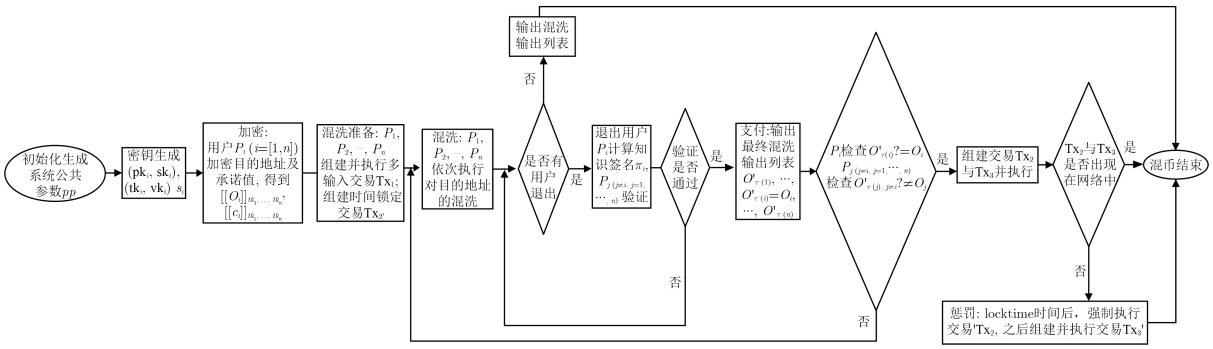

图 2 方案流程

共参数$\mathrm{pp}=(g_1, g_2, N, u, h, H)$。

(2) 密钥生成KeyGen:

(a) 输入系统参数pp，随机选取$0<\mathrm{sk}_i<q-1$，计算公钥$\mathrm{pk}_i=\mathrm{sk}_i\cdot h \bmod q$。算法为每个参与混币的用户$P_i(i=1,2,\cdots,n)$生成签名交易的密钥对$(\mathrm{pk}_i,\mathrm{sk}_i)$及加密密钥对$(\mathrm{tk}_i,\mathrm{vk}_i)$。

(b) 算法采用椭圆曲线上的分布式密钥生成协议(ECDKG)[17]，为每个用户计算对应于托管地址T的私钥的秘密份额s_i。

(3) 加密Encrypt:

(a) 用户P_i向混币群中的其他用户$P_j\,(j\neq i)$广播自己的加密公钥tk_i。

(b) $P_i(i=1,2,\cdots,n)$选择秘密随机数$r_i\in Z_q$，采用Pedersen[18]承诺算法计算承诺值$c_i=g_1{}^{\mathrm{H}(O_i)}g_2{}^{r_i}$。

(c) $P_i(i=1,2,\cdots,n)$使用加密密钥$\mathrm{tk}_1,\mathrm{tk}_2,\cdots,\mathrm{tk}_n$分层加密自己的目的地址$\mathrm{E}_{\mathrm{tk}_1}(\mathrm{E}_{\mathrm{tk}_2}(,\cdots,\mathrm{E}_{\mathrm{tk}_n}(O_i)))=[[O_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$和承诺值$\mathrm{E}_{\mathrm{tk}_1}(\mathrm{E}_{\mathrm{tk}_2}(,\cdots,\mathrm{E}_{\mathrm{tk}_n}(c_i)))=[[c_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$。之后，$P_i$广播$[[O_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$, $[[c_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$至其他混币用户。

(4) 混洗Shuffle:

(a) 每个用户$P_i(i=1,2,\cdots,n)$将自己输入地址I_i上的比特币$x+x_1$存储在托管地址T上。

① 用户$P_1,P_2,\cdots,P_n$用私钥$\mathrm{sk}_i\,(i=1,2,\cdots,n)$联合签名一笔多输入交易$\mathrm{Tx}_1=\{I_1,I_2,\cdots,I_n\}\xrightarrow{nx+nx_1}T$，表明将输入地址$I_1,I_2,\cdots,I_n$上共$nx+nx_1$个比特币存入托管地址$T$中。之后所有用户联合组建并签名时间锁定交易$\mathrm{Tx}_{2'}=T\xrightarrow{nx}\{I_1,I_2,\cdots,I_n\}$，该交易将在惩罚与恢复阶段执行。

②$P_1,P_2,\cdots,P_n$广播交易Tx_1和$\mathrm{Tx}_{2'}$至比特币网络。

(b) $P_1,P_2,\cdots,P_n$对目的地址进行混洗。

① P_1用私钥vk_1解密$[[O_1]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$, $[[O_2]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$, $\cdots$, $[[O_n]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$，得到$[[O_1]]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $[[O_2]]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\cdots$, $[[O_n]]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$。

② P_1选择τ_1，对解密结果混洗，将混洗结果$\left[\left[O_{\tau_1(1)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\left[\left[O_{\tau_1(2)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\cdots$, $\left[\left[O_{\tau_1(n)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$发送给$P_2$。

③ P_2用vk_2解密$\left[\left[O_{\tau_1(1)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\left[\left[O_{\tau_1(2)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\cdots$, $\left[\left[O_{\tau_1(n)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$并对结果混洗。将混洗结果$\left[\left[O_{\tau_1\tau_2(1)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\left[\left[O_{\tau_1\tau_2(2)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\cdots$, $\left[\left[O_{\tau_1\tau_2(n)}\right]\right]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$发送给$P_3$，依次类推，直至用户$P_n$。最后输出混洗列表$\left\{O_{\tau(1)},O_{\tau(2)},\cdots,O_{\tau(n)}\right\}=\left[\left[O_{\tau_1\tau_2,\cdots,\tau_n(1)}\right]\right]$, $\left[\left[O_{\tau_1\tau_2,\cdots,\tau_n(2)}\right]\right]$, $\cdots$, $\left[\left[O_{\tau_1\tau_2,\cdots,\tau_n(n)}\right]\right]$。对$\tau_i$的选择要求满足$\left[\left[O_{\tau_i(j)}\right]\right]_{\mathrm{tk}_{i+1},\cdots,\mathrm{tk}_n}\neq[[O_j]]_{\mathrm{tk}_{i+1},\cdots,\mathrm{tk}_n}$。

(c) 用户$P_1,P_2,\cdots,P_n$获取承诺列表。

①从用户P_1开始。P_1解密$[[c_1]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$, $[[c_2]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$, $\cdots$, $[[c_n]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$，将解密结果$[[c_1]]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $[[c_2]]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$, $\cdots$, $[[c_n]]_{\mathrm{tk}_2,\cdots,\mathrm{tk}_n}$发送给$P_2$。依次类推，直至最后一个用户$P_n$。最后输出承诺列表$\{c_1,c_2,\cdots,c_n\}$。

② 用户P_n广播$\{c_1,c_2,\cdots,c_n\}$。

(5) 退出Exit:

(a) 假设P_i请求退出混洗操作，P_i计算累加值$A=u^{c_1,c_2,\cdots,c_n} \bmod N$和证据$w=u^{c_1,\cdots,c_{i-1},c_{i+1},\cdots,c_n} \bmod N$。

(b) P_i针对声明R:P_i是混币群中的成员，计算知识签名π，并使用零知识证明协议$\mathrm{PK}_1,\mathrm{PK}_2$向其他用户证明其对$R$的知识签名$(\mathrm{ZKSok}\{R\})$。这里$\pi=\mathrm{ZKSok}\{R\}\left\{(c_i,w_i,r_i):\begin{array}{l}\mathrm{Accverify}\,(N,u,A,w_i,c_i)=1\\ \wedge c_i=g_1{}^{\mathrm{H}(O_i)}g_2{}^{r_i}\end{array}\right\}$。

具体零知识证明过程如下：

$$\mathrm{PK}_1\left\{c_i,w_i,r_i,\mathrm{H}\,(O_i):c_i=g_1{}^{\mathrm{H}(O_i)}g_2{}^{r_i}\right\}\tag{1}$$

式(1)用于证明者使用证据w_i证明c_i是对$\mathrm{H}\,(O_i)$的承诺。假设g,h是公开的系统参数，$l\leq k$是2个安全参数。$\mathrm{H}:\{0,1\}^*\to\{0,1\}^k$是一个抗碰撞的哈希函数。证明者只需证明其拥有正确的c_i和w_i，使得$y_i=g^{c_i}h^{w_i}=g^{g_1{}^{\mathrm{H}(O_i)}g_2{}^{r_i}}h^{w_i}$成立。

证明者选取$2l$个随机数$d_1,d_2,\cdots,d_l,v_1,v_2,\cdots,v_l$，计算$t_i=g^{g_1{}^{\mathrm{H}(O_i)}g_2{}^{d_i}}h^{v_i}$。对于消息$m$，证明者生成知识

签名$\left(s, f_1, f_2, \cdots, f_l, f_1', \cdots, f_l'\right)$。其中$s=\mathrm{H}\left(m||y_i||g_1||g_2||g||h||\mathrm{H}\left(O_i\right)||t_1||\cdots||t_l\right)$，$||$表示级联运算。若$s[i]=0$($s$化为二进制后的第$i$位)，那么$f_i=d_i$，$f_i'=v_i$；否则，$f_i=d_i-r_i, f_i'=v_i-w_i g_2^{d_i-r_i}$。

证明者发送知识签名给验证者。验证者计算$s'=\mathrm{H}\left(m||y_i||g_1||g_2||g||h||\mathrm{H}\left(O_i\right)||\overline{t_1}||\cdots||\overline{t_l}\right)$，判断$s'$是否等于$s$。若$s'[i]=0$，那么$\overline{t_i}=g^{g_1^{\mathrm{H}(O_i)} g_2^{d_i}} h^{v_i}=t_i$。否则$\overline{t_i}=y_i^{g_2^{f_i}} h^{f_i'}=\left(g^{g_1^{\mathrm{H}(O_i)} g_2^{r_i}} h^{w_i}\right)^{g_2^{d_i-r_i}} h^{v_i-w_i g_2^{d_i-r_i}}=t_i$。

$$\mathrm{PK}_2\left\{c_i, w_i, r_i: \mathrm{Accverify}\left(N, u, A, w_i, c_i\right)=1\right\} \quad (2)$$

式(2)用于证明者证明承诺值$c_i=g_1^{\mathrm{H}(O_i)} g_2^{r_i}$确实被累加到累加值$A$中。为此，证明者首先选取3个随机数$s_1, s_2, s_3 \in_R Z_{\lfloor n/4 \rfloor}$，计算辅助承诺$C_{\mathrm{H}(O_i)}=g_1^{\mathrm{H}(O_i)} g_2^{s_1}$, $C_{w_i}=w_i^{c_i/\mathrm{H}(O_i)} g_2^{s_2}$, $C_n=g_1^{s_2} g_2^{s_3}$。并发送$c_i, C_{\mathrm{H}(O_i)}, C_{w_i}, C_n$给验证者。之后，证明者选取$r_\alpha \in_R \left(-B2^{k'+k''}, \cdots, B2^{k'+k''}\right)$; $r_\gamma, r_\varphi, r_\psi, r_\sigma, r_\xi \in_R Z_q$; $r_\varepsilon, r_\eta, r_\zeta \in_R \left(-\lfloor n/4 \rfloor 2^{k'+k''}, \cdots, \lfloor n/4 \rfloor 2^{k'+k''}\right)$；$r_\beta, r_\delta \in_R \left(-\lfloor n/4 \rfloor q 2^{k'+k''}, \cdots, \lfloor n/4 \rfloor q 2^{k'+k''}\right)$，并计算：

$$\begin{aligned}
&\tau_1=g_1^{r_\alpha} g_2^{r_\varphi} && \tau_2=\left(c_i/g_1\right)^{r_\gamma} g_2^{r_\psi}\\
&\tau_3=\left(g_1 c_i\right)^{r_\sigma} g_2^{r_\xi} && t_1=g_1^{r_\varepsilon} g_2^{r_\zeta}\\
&t_2=g_1^{r_\alpha} g_2^{r_\eta} && t_3=C_{w_i}^{r_\alpha}\left(1/g_2\right)^{r_\beta}\\
&t_4=C_n^{r_\alpha}\left(1/g_2\right)^{r_\delta}\left(1/g_1\right)^{r_\beta}
\end{aligned}$$

验证者随机选择$d \in_R \{0,1\}^k$并发送给证明者。证明者将下述值发送给验证者：

$$\begin{aligned}
&s_\alpha=r_\alpha-d\mathrm{H}\left(O_i\right)\\
&s_\eta=r_\eta-ds_1\\
&s_\varphi=r_\varphi-dr_i \bmod q\\
&s_\beta=r_\beta-ds_2\mathrm{H}\left(O_i\right)\\
&s_\gamma=r_\gamma-d\left(\mathrm{H}\left(O_i\right)-1\right)^{-1} \bmod q\\
&s_\zeta=r_\zeta-ds_3\\
&s_\delta=r_\delta-ds_3\mathrm{H}\left(O_i\right)\\
&s_\psi=r_\psi+dr_i\left(\mathrm{H}\left(O_i\right)-1\right)^{-1} \bmod q\\
&s_\sigma=r_\sigma-d\left(\mathrm{H}\left(O_i\right)+1\right)^{-1} \bmod q\\
&s_\xi=r_\xi+dr_i\left(\mathrm{H}\left(O_i\right)+1\right)^{-1} \bmod q
\end{aligned}$$

验证者验证下述值，若验证通过返回1；否则返回0。

$$\begin{aligned}
&\tau_1=c_i^{d} g_1^{s_\alpha} g_2^{s_\varphi} && \tau_2=g_1^{d}\left(c_i/g_1\right)^{s_\gamma} g_2^{s_\psi}\\
&\tau_3=g_1^{d}\left(g_1 c_i\right)^{s_\sigma} g_2^{s_\xi} && t_1=C_n^{d} g_1^{s_\varepsilon} g_2^{s_\zeta}\\
&t_2=C_{\mathrm{H}(O_i)}^{d} g_1^{s_\alpha} g_2^{s_\eta} && t_3=A^{d} C_{w_i}^{s_\alpha}\left(1/g_2\right)^{s_\beta}\\
&t_4=C_n^{s_\alpha}\left(1/g_2\right)^{s_\delta}\left(1/g_1\right)^{s_\beta}
\end{aligned}$$

(6) 验证Verify:

(a) 其他用户计算累加值A'，判断$A'=w^{c_i} \bmod N$是否等于A。

(b) 其他用户使用公开参数验证π是对知识R的签名。若验证通过返回1；否则返回0。

(7) 支付Spend:

(a) 用户P_n将混洗输出列表$\left\{O_{\tau(1)}, O_{\tau(2)}, \cdots, O_{\tau(n)}\right\}$中请求退出混洗操作的用户$P_i$的混洗地址$O_{\tau(i)} \neq O_i$与$O_{\tau(l)}=O_i$调换，生成并广播最终混洗输出列表$O_{\tau(1)}', \cdots, O_{\tau(i)}'=O_i, \cdots, O_{\tau(n)}'$。

(b) 所有用户组建交易$\mathrm{Tx}_2=\left\{T \xrightarrow{nx} O_{\tau(1)}', \cdots, O_{\tau(i)}', \cdots, O_{\tau(n)}'\right\}$和交易$\mathrm{Tx}_3=\left\{T \xrightarrow{nx_1} I_1, I_2, \cdots, I_n\right\}$。

(c) 请求退出的用户P_i检查交易Tx_2中对应于自己位置的混洗输出地址$O_{\tau(i)}'?=O_i$。若相等，所有用户重构托管地址的私钥，并签名交易Tx_2与Tx_3；否则重新执行步骤(4)(b)；未请求退出的用户检查Tx_2中对应于自己位置的混洗输出地址$O_{\tau(j),j\neq i}'?=O_j$，若相等，则重新执行步骤(4)(b)。

(d) 广播交易Tx_2与Tx_3，若交易最终写入区块链，则用户P_i成功退出混洗操作。

2.3 惩罚和恢复机制

当恶意用户在步骤(7)(c)中拒绝签名交易Tx_2时，将执行惩罚和恢复机制：在步骤(4)(a)，所有用户协商一个锁定时间段locktime，若存在恶意用户拒绝签名交易，那么超出这个时间段后，交易Tx_2与交易Tx_3将不会出现在比特币网络中，此时暂停执行混币协议，强制执行时间锁定交易$\mathrm{Tx}_{2'}$。当交易$\mathrm{Tx}_{2'}$出现在网络后，群中所有用户组建交易$\mathrm{Tx}_{3'}=\left\{T \xrightarrow{nx_1} I_1, I_2, \cdots, I_n\right\}$，若恶意用户依旧拒绝签名交易$\mathrm{Tx}_{3'}$，那么他将损失存放在托管地址中的罚金$x_1$，但所有用户存放在托管地址上的比特币交易金额$x$不会受到损失。

3 安全性和性能分析

3.1 安全性分析

3.1.1 匿名性分析

(敌手模型)任何来自混币群外的敌手在看到与正常比特币交易不可区分的混币交易Tx_2时，无法根据交易的输出地址$O_{\tau(1)}', O_{\tau(2)}', \cdots, O_{\tau(n)}'$判断出哪一个用户退出了混币。具体而言，通过一系列谕言机的方式塑造敌手破坏协议的能力。

(1) KeyGen (i)：输入询问请求i，谕言机执行KeyGen算法，返回密钥$(\mathrm{pk}_i, \mathrm{tk}_i)$。

(2) Encrypt (pk_i)：输入公钥pk_i，谕言机执行Encrypt算法，返回承诺值c_i，加密结果$[[O_i]]_{\mathrm{tk}_1, \cdots, \mathrm{tk}_n}$, $[[c_i]]_{\mathrm{tk}_1, \cdots, \mathrm{tk}_n}$。

(3) Exit (pk_i, c_i)：输入公钥pk_i和承诺值c_i，谕言机执行Exit算法，返回知识签名π^*。

(4) Spend $\left(\pi^*, [[O_1]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}, \cdots, [[O_n]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}\right)$: 输入加密结果 $[[O_1]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}, [[O_2]]_{\mathrm{tk}_1,\text{-},\mathrm{tk}_n}, \cdots, [[O_n]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}$和知识签名$\pi^*$，谕言机返回混洗输出列表$O_{\tau(1)}', \cdots, O_{\tau(i)}' = O_i, \cdots, O_{\tau(n)}'$。其中$O_i$是退出混洗的用户对应的目的地址。

为证明知识签名的不可伪造性，额外定义模拟器(Simulator)和提取器[16](Extractor)如下：

(1) Simulator (m):使用模拟器初始化算法(Simsetup)产生公共参数及陷门。随机选择一个NP声明x和证据w，当接收到要签名的消息m时，使用x, w和公共参数运行模拟器签名算法(Simsign)生成签名σ。

(2) Extractor (x, σ): 给定对消息m的知识签名σ及陷门信息，提取器输出一个有效的证据w。

定义1　对任意概率多项式时间的敌手$\mathrm{A} = (\mathrm{A}_1, \mathrm{A}_2)$，当不等式(3)成立时，称该方案是匿名的。

$$\left|\Pr\left[\begin{array}{ll} & pp \leftarrow \mathrm{Setup}(1^\lambda);\ \left(m, P_i, I_i, [[O_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}, [[c_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}, i = 1,2,\cdots,n\right) \\ b' = b: & \leftarrow \mathrm{A}_1(\mathrm{pp});\ b \leftarrow [1,n];\ (\pi^*, c^*, O_{\tau(1)}', \cdots, O_{\tau(n)}') \leftarrow \mathrm{Shuffle}, \\ & \mathrm{Exit}, \mathrm{Spend}(m, P_b, I_b, [[O_b]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}, [[c_b]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}); \\ & b' \leftarrow \mathrm{A}_2(\mathrm{pp}, \pi^*, O_{\tau(1)}', \cdots, O_{\tau(n)}') \end{array}\right] - \frac{1}{n}\right| \le \mathrm{negl}(\lambda) \quad (3)$$

定理1　若离散对数问题(DLP)是困难的，那么本方案是匿名的。

证明　游戏G_0对谕言机的询问获得的响应和在真实协议中一样。游戏G_3则完全隐藏了b的信息。定义Win_i为每个游戏中敌手获胜的概率。

游戏G_0：在该游戏中，挑战者运行Setup算法，生成公共参数pp。一旦接收到挑战询问$\left(m, P_i, I_i, [[O_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}, [[c_i]]_{\mathrm{tk}_1,\cdots,\mathrm{tk}_n}, i = 1,2,\cdots,n\right)$，挑战者随机选择$b \leftarrow [1,n]$，按如下生成$\left(\pi^*, c_b, O_{\tau(1)}', \cdots, O_{\tau(n)}'\right)$。

(1) 随机选择$r_b \in Z_q$，计算$c_b = {g_1}^{\mathrm{H}(O_b)} {g_2}^{r_b}$。

(2) 计算累加值$A = u^{c_1,c_2,\cdots,c_n} \bmod N$，证据$w = u^{c_1,\cdots,c_{b-1},c_{b+1},\cdots,c_n} \bmod N$及知识签名$\pi^*$:

$$\pi^* = \mathrm{ZKSok}\{R\}\left\{(c_b, w_b, r_b): \begin{array}{l} \mathrm{Accverify}(N, u, A, w_b, c_b) = 1 \\ \wedge c_b = {g_1}^{\mathrm{H}(O_b)} {g_2}^{r_b} \end{array}\right\} \quad (4)$$

(3) 随机选择混洗排列$\tau_{(1)}, \tau_{(2)}, \cdots, \tau_{(n)}$，生成最终混洗输出列表$O_{\tau(1)}', O_{\tau(2)}', \cdots, O_{\tau(b)}' \ne O_b, \cdots, O_{\tau(n)}'$。最终敌手输出其猜测$b'$。根据匿名性的定义可得$\Pr[\mathrm{Win}_0] = \Pr[b' = b]$。

游戏G_1: 这个游戏修改了承诺值c_i的计算方式。随机选取$X_i \in Z_q$，计算$c_i = X_i {g_2}^{r_i}$。此时c_i不再和O_i有关。即c_i完全独立于b。因此，根据承诺方案的隐藏性[18]可得$\Pr[\mathrm{Win}_1] = \Pr[\mathrm{Win}_0]$。

游戏G_2: 这个游戏和G_1不同在于知识签名是通过询问Simulator谕言机生成的。当接收到挑战询问，挑战者不再使用w产生签名。根据知识签名的SimExt安全性[16]，$\Pr[\mathrm{Win}_2] - \Pr[\mathrm{Win}_1] \le \mathrm{negl}(\lambda)$。

游戏G_3: 这个游戏修改了随机排列$\tau_1, \tau_2, \cdots, \tau_n$的生成方式。当询问最终混洗输出列表时，挑战者输出$O_{\tau_{(1)}}', \cdots, O_{\tau_{(b)}}' = O_b, \cdots, O_{\tau_{(b)}}'$。由于敌手仅知道目的地址的加密结果，并且事先并没有渠道接收$O_{\tau(1)}, O_{\tau(2)}, \cdots, O_{\tau(n)}$，此时敌手无法通过对比两个混洗列表找出位置不相同的目的地址来判断退群的用户。所以对敌手而言$O_{\tau_{(1)}}', O_{\tau_{(2)}}', \cdots, O_{\tau_{(b)}}' = O_b, \cdots, O_{\tau_{(b)}}'$就是全新的。因此$\Pr[b' = b] = 1/n$。

分析：由于要求挑战者解决的困难问题为至少求出一个$\mathrm{H}(O_i)$，使得$X_i = g^{\mathrm{H}(O_i)}$，因此在游戏$\mathrm{G}_3$中，若敌手能成功地从混洗输出列表中找出退群用户$b'$，那么敌手将其猜测结果$P_{b'}$发送给挑战者等候验证。若猜测正确，挑战者就找到了$X_b$的离散对数并给敌手返回预判结果1。此时，敌手在游戏G_2中总能获胜。因此:$\Pr[\mathrm{Win}_3] - \Pr[\mathrm{Win}_2] \le \mathrm{Adv}_{\mathrm{C}}^{\mathrm{DLP}}(\lambda)$。其中$\mathrm{Adv}_{\mathrm{C}}^{\mathrm{DLP}}(\lambda)$表示挑战者解决DLP困难问题的优势。

对于混币群内的敌手，假设其试图猜测退群用户，该敌手事先没有渠道知道诚实用户的目的地址。文献[10]分析了此时敌手正确猜测输出地址的概率$P_s = \dfrac{1-(1-c/n)^2}{n-c} + \dfrac{(1-c/n)^2}{n-c+N'}$。其中$c$表示$n$个用户中恶意用户的数量。$N' = N/(1-c/n)^2$，$N$是匿名等级。当$n = 100$, $c/n = 10\%$, $N = 900$时，方案[10]中敌手获胜的概率为0.0027。

我们选取混币用户数量$n = 200$, $N = 900$，此时允许混币群中恶意用户最大数量为66。这些恶意用户之间可以相互询问对方指定的目的地址，但无法知道诚实用户的目的地址。利用前文概率公式P_s计算出混币群中敌手正确猜测退群用户的概率为0.004。这个结果适用的最差情况是10个人中有8个人退群。此时剩余两人相互不知道对方的目的地址，并且无法互相询问，因此他们没有依据猜测当

前群中有人退出。即使能够猜测，但不确定自己的混洗输出地址是谁的目的地址，进而他们也不能知道究竟是谁退出。显然$n(n \geq 10)$越大，能够猜测出退群用户的概率越小。

3.1.2 不可伪造性分析

签名询问：当敌手A要求对消息m进行知识签名时，调用Simulator模拟器，用R, w和Simsign算法生成签名π。

假设敌手能成功伪造一个有效的知识签名，该签名包含两个零知识证明：(1)承诺c_i的确被累加到累加器A中。(2)c_i是对$\mathrm{H}(O_i)$的承诺。对于(1)，文献[15]指出如果敌手能攻破该零知识证明协议，那么就可以以至少1/2的概率解决强RSA假设。对于(2)，文献[19]指出若敌手能成功伪造一个对消息m的签名，那么就能以$1-\left(1-2/2^{\lambda(\varepsilon-1)}\right)^k$的概率从对消息$m$的两个不同签名值中计算出$y=g^{a^x}$的双离散对数。当$k=80, \lambda=170, \varepsilon=4/3$时，几乎能以$\Pr=1$的概率计算出双离散对数$x$。据此，在强RSA假设和双离散对数问题下，敌手无法伪造一个有效的知识签名去冒充用户退出混洗操作。

3.2 性能分析

表1分析了方案性能上的优势。其中，主动攻击指敌手伪造知识签名，被动攻击指敌手观察网络中的交易来找出参与混洗的用户。Coinjoin, CoinShuffle, CoinParty, SecureCoin, Mixcoin, CoinShuffle++是传统混币方案。这些方案多采用DC-net及其变体，安全多方计算以增强比特币系统的隐私性和安全性，但均不支持用户在混洗时退出。一种强制的方法就是暂停混洗交易，将存放在托管地址上的所有资金返回到原输入地址。但这对不退出混币操作的用户会造成不必要的损失。本方案相比传统混币协议实现了混币用户的匿名退群功能，且方案采用自行混淆资金，相比集中式混币服务，用户无需信任第三方。即使混币群中存在恶意用户拒绝签名混币交易以发起Dos攻击，方案的错误处理机制也可以保证用户的资金不会被冻结，同时恶意用户会损失其罚金。

表 1 不同方案性能比较

方案	抗主动/被动攻击	退出混洗	兼容比特币系统	惩罚恢复机制	身份隐私	交易金额隐私
Coinjoin[3]	抗被动攻击	×	√	×	√	×
Mixcoin[4]	抗被动攻击	×	√	×	√	×
TumbleBit[5]	抗被动攻击	×	√	√	√	×
CoinShuffle[6]	抗被动攻击	×	√	×	√	×
CoinShuffle++[8]	抗被动攻击	×	√	×	√	×
CoinParty[10]	抗被动攻击	×	√	√	√	×
ZeroCoin[14]	均抗	×	×	×	√	×
SecureCoin[17]	抗被动攻击	×	√	√	√	×
CoinExit	均抗	√	√	√	√	×

3.3 效率分析

表2给出了文献[10]，文献[14]及本文协议CoinExit的执行效率。为便于说明，定义表2中的参数如下。n为混币群中的用户数；$\nu(E)$为执行一次加密的时间；$\nu(m)$为执行一次模乘的时间；$\nu(M)$为执行一次模指数的时间；$\nu(H)$为执行一次普通单向哈希函数的时间；$\nu(R)$为执行一次标准椭圆曲线上点乘的时间。

基于Miracl密码库，我们在个人电脑端对上述协议的运算时长进行了测试，测试结果如表3。选取的累加器中RSA模数N为1024 bit, q为160 bit。哈希函数为安全散列算法SHA256。具体测试环境中，物理机配有酷睿3代I7处理器(主频2.4 GHz)，8G DDR3内存和Windows 7 Ultimate, 64-bit 6.1.7601, Service Pack 1操作系统。编译环境为Microsoft Visual Studio Ultimate 2013版本

表 2 不同方案理论执行时间对比

方案	加密	模乘	模指数	哈希	椭圆曲线上的点乘
CoinParty[10]	$(n^2)_{\nu(E)}$	$(8n)_{\nu(m)}$	$(4n)_{\nu(M)}$	$(4n)_{\nu(H)}$	$(10n)_{\nu(R)}$
ZeroCoin[14]	0	$(9n)_{\nu(m)}$	$(12n)_{\nu(M)}$	$(n)_{\nu(H)}$	0
CoinExit	$(2n^2)_{\nu(E)}$	$(11n)_{\nu(m)}$	$(17n)_{\nu(M)}$	$(2n)_{\nu(H)}$	$(5n)_{\nu(R)}$

表 3 不同方案执行时间对比(ms)

方案	模乘	模指数	哈希	椭圆曲线上的点乘	总运行时间
CoinParty[10]	0.48	1452.48	35.28	26800.00	26288.24
ZeroCoin[14]	0.54	4357.44	8.82	0.00	4366.80
CoinExit	0.66	6173.04	17.64	13400.00	19591.34

12.0.31101.00 Update4。

为了达到协议声称的功能，我们对目的地址进行了承诺，随后分别对目的地址和承诺值进行加密，用于退出混洗的用户零知识证明其身份。同时采用ECDKG算法来生成托管地址的密钥，因此本文方案在加密、模乘和模指数运算上耗时较长。对于3个协议交易阶段和签名交易所耗时间，我们没有计算在内。但计算了知识签名过程中所用的两个零知识证明协议。可以看出，本文方案的运算时间介于文献[10]和文献[14]之间，但在功能上更为丰富。

4　结束语

针对传统混币方案中用户无法在混洗操作时请求退出的问题，提出了支持用户匿名可撤销的混币操作方案。方案综合运用承诺、累加器和知识签名等技术手段，能有效抵抗外部敌手伪造混币用户的知识签名，以及外部敌手观察网络中的交易以找出退出混洗操作的用户的攻击，更好地保护用户隐私。该方案未采用密码学技术加密交易，因此与现行比特币系统兼容。当退出混币操作的用户少于$n-2$时，混币群内的敌手无法正确猜测出谁退出了混洗操作。

参 考 文 献

[1] 秦波, 陈李昌豪, 伍前红, 等. 比特币与法定数字货币[J]. 密码学报, 2017, 4(2): 176–186. doi: 10.13868/j.cnki.jcr.000172.
QIN Bo, CHEN Lichanghao, WU Qianhong, *et al*. Bitcoin and digital fiat currency[J]. *Journal of Cryptologic Research*, 2017, 4(2): 176–186. doi: 10.13868/j.cnki.jcr.000172.

[2] KHALILOV M C K and LEVI A. A survey on anonymity and privacy in bitcoin-like digital cash systems[J]. *IEEE Communications Surveys & Tutorials*, 2018, 20(4): 2543–2585. doi: 10.1109/COMST.2018.2818623.

[3] MAXWELL G. CoinJoin: Bitcoin privacy for the real world[EB/OL]. https://en.bitcoin.it/wiki/CoinJoin, 2019.

[4] BONNEAU J, NARAYANAN A, MILLER A, *et al*. Mixcoin: Anonymity for Bitcoin with accountable mixes[C]. The 18th International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 2014: 486–504. doi: 10.1007/978-3-662-45472-5_31.

[5] HEILMAN E, ALSHENIBR L, BALDIMTSI F, *et al*. TumbleBit: An untrusted bitcoin-compatible anonymous payment hub[C]. Network and Distributed System Security Symposium, San Diego, California, 2017. doi: 10.14722/ndss.2017.23086.

[6] RUFFING T, MORENO-SANCHEZ P, and KATE A. CoinShuffle: Practical decentralized coin mixing for bitcoin[C]. The 19th European Symposium on Research in Computer Security, Wroclaw, Poland, 2014: 345–364. doi: 10.1007/978-3-319-11212-1_20.

[7] MEIKLEJOHN S, POMAROLE M, JORDAN G, *et al*. A fistful of bitcoins: Characterizing payments among men with no names[C]. The 2013 Association for Computing Machinery Conference on Internet Measurement Conference, Barcelona, Spain, 2013: 127–140. doi: 10.1145/2504730.2504747.

[8] RUFFING T, MORENO-SANCHEZ P, and KATE A. P2P mixing and unlinkable Bitcoin transactions[C]. Network and Distributed System Security Symposium, San Diego, California, 2017. doi: 10.14722/ndss.2017.23415.

[9] ZIEGELDORF J H, GROSSMANN F, HENZE M, *et al*. CoinParty: Secure multi-party mixing of bitcoins[C]. The 5th Association for Computing Machinery Conference on Data and Application Security and Privacy, San Antonio, USA, 2015: 75–86. doi: 10.1145/2699026.2699100.

[10] ZIEGELDORF J H, MATZUTT R, HENZE M, *et al*. Secure and anonymous decentralized Bitcoin mixing[J]. *Future Generation Computer Systems*, 2018, 80: 448–466. doi: 10.1016/j.future.2016.05.018.

[11] 张卫国, 孙嫚, 陈振华, 等. 空间位置关系的安全多方计算及其应用[J]. 电子与信息学报, 2016, 38(9): 2294–2300. doi: 10.11999/JEIT160102.
ZHANG Weiguo, SUN Man, CHEN Zhenhua, *et al*. Secure multi-party computation of spatial relationship and its application[J]. *Journal of Electronics & Information Technology*, 2016, 38(9): 2294–2300. doi: 10.11999/JEIT160102.

[12] SAXENA A, MISRA J, and DHAR A. Increasing anonymity in Bitcoin[C]. International Conference on Financial Cryptography and Data Security, Christ Church, Barbados, 2014: 122–139. doi: 10.1007/978-3-662-44774-1_9.

[13] CHURYUMOV A. Byteball: A decentralized system for storage and transfer of value[EB/OL]. https://byteball.org/Byteball.pdf, 2018.

[14] MIERS I, GARMAN C, GREEN M, *et al*. Zerocoin: Anonymous distributed E-cash from bitcoin[C]. 2013 IEEE Symposium on Security and Privacy, Berkeley, USA, 2013: 397–411. doi: 10.1109/SP.2013.34.

[15] CAMENISCH J and LYSYANSKAYA A. Dynamic accumulators and application to efficient revocation of anonymous credentials[C]. The 22nd Annual International Cryptology Conference on Advances in Cryptology, California, USA, 2002: 61–76. doi: 10.1007/3-540-45708-9_5.

[16] CHASE M and LYSYANSKAYA A. On signatures of knowledge[C]. Annual International Cryptology Conference

on Advances in Cryptology, Santa Barbara, California, USA, 2006: 78–96. doi: 10.1007/11818175_5.

[17] IBRAHIM M H. SecureCoin: A robust secure and efficient protocol for anonymous Bitcoin ecosystem[J]. *International Journal of Network Security*, 2017, 19(2): 295–312. doi: 10.6633/IJNS.201703.19(2).14.

[18] SUN Shifeng, AU M H, LIU J K, *et al*. RingCT 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero[C]. The 22nd European Symposium on Research in Computer Security, Oslo, Norway, 2017: 456–474. doi: 10.1007/978-3-319-66399-925.

[19] CORRIGAN-GIBBS H, BONEH D, and MAZIÈRES D. Riposte: An anonymous messaging system handling millions of users[C]. IEEE Symposium on Security and Privacy, San Jose, USA, 2015: 321–338. doi: 10.1109/SP.2015.27.

李雪莲：女，1979年生，副教授，研究方向为有限域及其在密码学中的应用.

王海玉：女，1994年生，硕士生，研究方向为分布式信息系统安全，密码货币.

高军涛：男，1979年生，副教授，研究方向为密码学和信息安全，包括区块链的安全性分析.

李 伟：男，1992年生，硕士生，研究方向为物联网及认证加密.