基于无证书的多方合同签署协议

曹素珍[①] 王 斐*[①] 郎晓丽[①] 汪 锐[①] 刘雪艳[②]

[①](西北师范大学计算机科学与工程学院 兰州 730070)

[②](西北师范大学数学与统计学院 兰州 730070)

摘 要：线上合同签署在电子商务中日益普及，在互不信任的签署方之间签署一份合同并不是一件简单的事情，各方就合同签署问题提出过许多合同签署协议。其中较多的协议是带有第三方参与的，但是在效率方面并不占优势，且易出现安全问题。现有借助区块链技术取代第三方参与的合同签署协议中，区块链的公开验证对不管是签署方还是待签署合同的敏感信息又发起了挑战。且大多协议针对于双方合同签署，随着签署方数量的增加，协议的通信成本和复杂度都在急剧增加。该文结合现有协议，提出一个高效的多方合同签署协议，协议中通过基于无证书的高效聚合签名方案，用于提高区块链下签署方签名验证效率，在区块链上仅公开签署方的临时密钥以减少系统开销。该协议满足正确性、安全性、公平性、私密性以及高效性。

关键词：聚合可验证签名；隐私保护；区块链；合同签署

中图分类号：TP309 **文献标识码**：A **文章编号**：1009-5896(2019)11-2691-08

DOI: 10.11999/JEIT190166

Multi-party Contract Signing Protocol Based on Certificateless

CAO Suzhen[①] WANG Fei[①] LANG Xiaoli[①] WANG Rui[①] LIU Xueyan[②]

[①](*College of Computer Science and Engineering*, *Northwest Normal University*, *Lanzhou* 730070, *China*)

[②](*College of Mathematics and Statistics*, *Northwest Normal University*, *Lanzhou* 730070, *China*)

Abstract: Online contract signing is becoming more and more popular in e-commerce. It is not easy to sign a contract between two parties who do not trust each other. Many of these protocols involve the participation of third parties, but they are not advantageous in efficiency and prone to security problems. Currently, contract signing agreements with third-party participation are replaced by block chain technology, but the public verification of block chain challenges the sensitive information of both the signer and the contract to be signed. And most of the agreements are for the signing of contracts between the two parties. With the increase of the number of signatories, the communication cost and complexity of the agreements increase sharply. Combined with the existing protocols, this paper proposes an efficient multi-party contract signing protocol. In the protocol, an efficient aggregation signature scheme based on no certificate is used to improve the signature verification efficiency of the signer under the block chain, and only the temporary key of the signer is disclosed on the block chain to reduce the system overhead. The protocol satisfies the requirements of correctness, security, fairness, privacy and high efficiency.

Key words: Aggregation verifiable signature; Privacy protection; Blockchain; Contract signing

1 引言

随着电子商务的不断发展，交易方或合作方通过互联网就某些条款签订合同变得日益普遍。在互不信任的签署方中，尤其是针对多方合同签署时，签署合同时的公平性就显得尤为重要。为解决公平签署的问题，最常见的方法便是加入可信第三方，借助第三方交换签署方的签名信息以达到公平签署的目标。这种方法就要求第三方保持在线状态，所以效率不会太高，尤其是随着签署方的增加易产生系统瓶颈问题。并且现实中第三方并非是完全可信的。随之产生了半可信第三方，在这类协议中，第三方仅在签署方发生争端时出现，用以解决争端问题。但是这类协议通常计算比较复杂，效率也不太高，同时存在信息泄露的风险。而另一类无需第三

收稿日期：2019-03-21；改回日期：2019-06-25；网络出版：2019-07-09
*通信作者： 王斐 wf9211@126.com
基金项目：国家自然科学基金(61662071, 61662069, 61562077)
Foundation Items: The National Natural Science Foundation of China (61662071, 61662069, 61562077)

方参与的协议，通常采用逐步释放待公开的信息[1]，但由于互联网的不同步，合同签署的过程也是异步发生的，因而并未解决公平签署的问题，在通信交互过程中的成本较高，实际应用的可能性较低。

2003年由Al-Riyami等人[2]提出了无证书密码体制。聚合签名自Boneh 等人[3]提出以来受到了广泛的应用，把多个签名压缩聚合成一个签名，降低了存储开销，验证的工作量和对网络传输的要求。文献[4–7]提出了应用于不同场景的无证书聚合方案及改进方案。聚合签名的使用将为多用户参与的方案提供了新的解决思路。

近年来，由于区块链技术的兴起，区块链技术受到更多的关注，去中心化特性被用来很好地解决了第三方问题。Wan等人[8]提出了在区块链环境下，采用时间戳服务器检查在最后期限内提交的加盖时间戳的签名的有效性来保障签署双方的公平性。文献[9]在双方合同签署的过程中嵌入了盲的可验证的加密签名，保护了待签署合同的私密性，防止了合同信息的泄露，但是当出现争议时，由于区块链具有公开透明性，区块链上每一个节点均可以获得链上的签名信息，同样泄露用户的信息。故基于区块链的合同签署协议中必不可少的要解决隐私保护问题，防止信息泄露。文献[10]在文献[9]基础上进行改进，引入了半诚信的第三方添加混币模式，保护了用户的隐私，虽然可用于多方合同签署，但是随着签署方数量的增加，在区块链上的工作量也在增加。Huang等人[11]将合同签署的过程分为链上和链下两部分，链下协商一个门限在链上交换，但该协议随着签署人数的增加验证各方签名的消耗也在急剧增加。文献[12]提出了一种新的合同签署协议，将大量的签名工作借助无证书聚合可验证的签名方案放在区块链下执行，只有聚合签名通过验证方可进入链上阶段。在区块链上各签署方交换秘密值，矿工仅验证签署方是否合法有效。但是该协议中无证书方案基于双线性对在聚合签名验证时，验证开销较大。

本文结合已有工作，在文献[12]的基础上提出一个高效的无证书聚合可验证签名方案，并应用于多方合同签署环境中。在多方合同签署的过程中，签署方首先在区块链链下对待签署合同达成一致，分别签署带有共享公钥的签名后聚合统一验证，若聚合签名通过验证之后，方可进入区块链链上阶段，否则退出合同签署协议。在链上，各方首先提交一笔担保金以担保自己的行为，之后各签署方依次公开自己的临时密钥。若公开过程中出现不诚实的签署方，则该签署方失去其保证金用于奖励之前已经诚实的公开自己临时密钥的签署方，晚于该签署方之后的签署方赎回自己的担保金。该协议不管是在链上还是在链下，保证了对签署方而言的公平性和私密性，针对于合同的机密性和不可伪造性，在协议的效率上也有很大的提高。

2 基础知识

2.1 离散对数

离散对数问题(Discrete Logarithm Problem, DLP)[13]：设在生成元为P的大素数为q阶的加法循环群G中，已知元组(P, bP), $b \in Z_q^*$，求解目标b。在任意的概率多项式时间(Probabilistic Polynomial Time, PPT)，算法A成功解决DLP问题的概率$\mathrm{Adv}^{\mathrm{DLP}}(A)$可忽略[13]。其中$\mathrm{Adv}^{\mathrm{DLP}}(A) = \Pr[A(P, bP) = b | b \in Z_q^*]$。

计算性Diffie-Hellan问题 (CDH问题)[14]：设在q阶的加法循环群G中，P为其生成元。已知P, aP, bP求abP问题，其中$a, b \in Z_q^*$。在算法A任意的PPT内，成功解决CDH问题的概率$\mathrm{Adv}^{\mathrm{CDH}}(A)$可忽略[14]。其中$\mathrm{Adv}^{\mathrm{CDH}}(A) = \Pr[A(P, aP, bP) = abP | a, b \in Z_q^*]$。

2.2 安全模型

在无证书签名方案的安全模型中，攻击者可以分为两大类[4]：

A_{I}：模拟为恶意的用户，不知道系统主密钥，但是可以任意替换用户的公钥；

A_{II}：模拟为恶意的KGC，知道系统的主密钥，但是不能替换用户公钥。

证明无证书签名方案的不可伪造性时，可构造挑战者C与攻击者A_{I}和A_{II}之间的游戏。

2.3 安全需求

多方的合同签署过程中需要满足以下安全需求：

(1) 正确性：若参与签署合同的签署者均诚实地执行多方合同签署协议，则合同签署成功，每一个签署者均可以得到其他签署方的普通签名；

(2) 公平性：多方签署合同时，对于每一个签署者都应该保证相对公平，出现不诚实的签署者时应该得到一定的惩罚，而已经诚实的公开自己临时密钥的签署者应该得到补偿；

(3) 私密性：在合同签署的过程中，保证每一个签署者的个人隐私不泄露，所签署的合同信息除了签署者知道外对其他人是保密的；

(4) 可验证性：对于合同签署时的聚合签名可以通过验证，并且具有可提取性，在满足一定条件下，可以恢复出签署者对于合同的普通签名，证明签署的合同有效的；

(5) 不可伪造性：对于签署者签署的签名具有不可伪造性，协商通过后的合同，若一方未签署，其他签署方不可能伪造一个合法的合同。

3　无证书聚合可验证签名方案

3.1　无证书聚合可验证签名方案定义

本方案由以下算法组成：

(1) 系统建立：设置安全参数k, KGC选取主密钥λ，公开系统参数params；

(2) 秘密值生成：$(x_i, \text{params}) \to (x_iP_i)$，选取随机值$x_i$为秘密值，$P_i$为与之对应的公钥；

(3) 部分私钥生成：$(v_i, \text{ID}_i, \lambda, P_i, \text{params}) \to y_i$, v_i为随机值，ID_i为用户身份，y_i为部分私钥；

(4) 用户密钥生成：$(x_i, y_i, \text{ID}_i, \text{params}) \to (\text{pk}_i, \text{sk}_i)$, pk_I和sk_i分别为用户的公私钥；

(5) 临时密钥生成：$(x_i', \text{params}) \to (x_i', P_i')$，选取随机值$x_i'$为用户临时私钥$x_i'$, P_i'为临时公钥；

(6) 共享密钥生成：$(x_1', x_2', \cdots, x_n', P_1', P_2', \cdots, P_n') \to (P_{\text{pub}}, x_{\text{pub}})$, P_{pub}为共享公钥，x_{pub}为共享密钥；

(7) 签名生成：$(M_i, \text{ID}_i, \Delta, r_i, \text{params}) \to \sigma_i'$，消息$M_i$，随机值$r_i$, Δ为状态信息，σ_i'为用户签名信息；

(8) 聚合签名：$(\sigma_1', \sigma_2', \cdots, \sigma_n') \to \sigma$, σ为聚合签名；

(9) 聚合签名验证：$(M_1, M_2, \cdots, M_n, \text{ID}_1\,\text{ID}_2 \cdots \text{ID}_n, \Delta, P_{\text{pub}}, \text{params}, \sigma) \to$true/false。若聚合签名有效，输出true；否则，输出false。

3.2　无证书聚合可验证签名方案具体构造

(1) 系统建立：设安全参数k, KGC 选择素数$q > 2k$阶椭圆曲线加群G, P为其生成元。选择抗碰撞的安全哈希函数$H: \{0,1\}^* \times G \times G \to Z_q^*$, $H_1: G \times \{0,1\}^* \to Z_q^*$，随机选$\lambda \in Z_q^*$作为系统主密钥，系统公钥$P_{\text{T}} = \lambda P$, KGC保存系统主密钥$\lambda$，公开系统参数$\text{params} = \{G, q, P, P_{\text{T}}, H, H_1\}$；

(2) 秘密值生成：用户U_i随机选取$x_i \in Z_q^*$用作秘密值，计算$P_i = x_iP$作为其对应公钥，并将用户的身份ID_i和P_i发送给KGC；

(3) 部分私钥生成：KGC随机选取$v_i \in Z_q^*$，计算$V_i = v_iP$, $h_i = H(\text{ID}_i, P_i, P_{\text{T}})$, $y_i = v_i + \lambda h_i \text{mod} q$, KGC公开$V_i$并将部分私钥$y_i$发送给用户$U_i$；

(4) 用户密钥生成：用户U_i计算$h_i = H(\text{ID}_i, P_i, P_{\text{T}})$，验证$y_iP = V_i + h_iP_{\text{T}}$是否成立，成立则合成私钥$\text{sk}_i = x_i + y_i$，公钥$\text{pk}_i = (P_i, V_i)$。否则，终止执行；

(5) 临时密钥生成：用户U_i随机选取x_i'作为临时私钥，计算$P_i' = x_i'P$, P_i'为临时公钥；

(6) 临时公钥承诺：

(a) 用户U_i随机选取辅助值k_i，并对临时公钥进行承诺，得$C_i = \text{Com}(P_i', k_i)$，将$C_i$广播给其他用户；

(b) 当用户收到C_i后，U_i打开承诺，则用户得到临时公钥P_i'；

(7) 共享密钥生成：当所有用户均打开公钥承诺后，可计算共享公钥$P_{\text{pub}} = \sum_{i=1}^{n} P_i'$，与之对应的共享密钥为$x_{\text{pub}} = \sum_{i=1}^{n} x_i'$；

(8) 签名生成：广播消息M，用户U_i对消息M_i进行签名过程如下：

(a) 用户U_i随机选取$r_i \in Z_q^*$，计算$R_i = r_iP$；

(b) 计算$l_i = H_1(R_i, \text{ID}_i \parallel M_i \parallel R_i \parallel \triangle)$, $S_i = r_iP_{\text{pub}} + \text{sk}_i \cdot l_i$；

(c) 用户U_i对消息M_i的签名为$\sigma_i' = (R_i, S_i)$。

(9) 聚合签名：若对n个消息-签名对(M_1, σ_1') $(M_2, \sigma_2') \cdots (M_n, \sigma_n')$聚合签名，计算$R = \sum_{i=1}^{n} R_i$, $S = \sum_{i=1}^{n} S_i$，则聚合签名为$\sigma = (R, S)$；

(10) 聚合签名验证：每一个参与者均可验证聚合签名，输入消息$M_1, M_2, \cdots, M_n$, σ和所需参数，验证者执行：

(a) 计算$h_i = H(\text{ID}_i, P_i, P_{\text{T}})$和$l_i = H_1(R_i, \text{ID}_i \parallel M_i \parallel R_i \parallel \Delta)$，其中$i = 1, 2, \cdots, n$。$h_i$和$l_i$可以预运算；

(b) 判断等式为$S \cdot P = \sum_{i=1}^{n} [R_iP_{\text{pub}} + (P_i + V_i + P_{\text{T}}\, h_i) l_i]$。等式成立，则输出true；否则，输出false。

4　安全性证明

4.1　正确性

计算$h_i = H(\text{ID}_i, P_i, P_{\text{T}})$, $l_i = H_1(R_i, \text{ID}_i \parallel M_i \parallel R_i \parallel \Delta)$和$R = \sum_{i=1}^{n} R_i$，其中$i = 1, 2, \cdots, n$。

首先验证等式$S_iP = R_iP_{\text{pub}} + (P_i + V_i + h_iP_{\text{T}})l_i$是否成立：

$S_iP = (r_iP_{\text{pub}} + \text{sk}_i \cdot l_i)P = r_iP_{\text{pub}}P + (x_i + y_i) \cdot l_iP$
$= R_iP_{\text{pub}} + (x_i + v_i + \lambda h_i) \cdot l_iP = R_iP_{\text{pub}} + (P_i + V_i + h_iP_{\text{T}}) \cdot l_i$，故可验证签名的正确性。

然后验证$S \cdot P = \sum_{i=1}^{n} [R_iP_{\text{pub}} + (P_i + V_i + P_{\text{T}} h_i) l_i]$是否成立：

$S \cdot P = \sum_{i=1}^{n} (r_iP_{\text{pub}} + \text{sk}_i \cdot l_i)P = \sum_{i=1}^{n} [r_iP_{\text{pub}} + (x_i + v_i + \lambda h_i) l_i]P = \sum_{i=1}^{n} [R_iP_{\text{pub}} + (P_i + V_i + P_{\text{T}} h_i) l_i]$，故可验证聚合签名的正确性。

4.2　不可伪造性

定理1　在随机语言模型中，攻击者A_{I}在自适应选择消息攻击下，依赖离散对数困难问题，存在

不可伪造性。若攻击者A_{I}能够以一个不可忽略的PPT内成功伪造一个有效聚合签名，则存在挑战者C能够以不可忽略的优势在多项式时间内解决DLP。

证明 攻击者A_{I}已知元组(P, bP)，伪造有效的聚合签名。挑战者C利用A_{I}求困难问题，解b的值。假定目标用户ID_j。C与A_{I}交互过程如下：

系统初始化阶段 挑战者C建立系统模型，执行系统建立算法，生成系统公开参数$\mathrm{params}=\{G, q, P, P_{\mathrm{T}}, H, H_1\}$和系统主密钥$\lambda$，令系统公钥$P_{\mathrm{T}}=bP$(其中$b\in Z_q^*$)。生成随机数$P_{\mathrm{pub}}\in Z_q^*$作为用户的共享公钥，挑战者$C$保存系统主密钥$\lambda$，将系统公开参数params和$P_{\mathrm{pub}}$发送给敌手$A_{\mathrm{I}}$。

询问阶段 在此阶段中，挑战者C建立维护表$L_H, L_{H1}, L_X, L_Y, L_{\mathrm{SK}}, L_{\mathrm{PK}}$，记录下询问过程。其中表的初始值均为空。询问过程如下：

(1) H询问：当A_{I}进行H询问。首先C检查$L_H(\mathrm{ID}_i, P_i, P_{\mathrm{T}}, h_i)$表，当$L_H$表已包含$h_i$时，将$h_i$返回给$A_{\mathrm{I}}$；当$L_H$表不存在$h_i$时，那么$C$随机选择$h_i\in Z_q^*$，将$h_i$返回给$A_{\mathrm{I}}$并记录到$(\mathrm{ID}, P_i, P_{\mathrm{T}}, h_i)$中；

(2) $H1$询问：当A_{I}进行$H1$询问。首先C检查$L_{H1}(R_i, ID_i, M_i, \Delta, l_i)$表，当$L_{H1}$表已包含$h1_i$时，将$l_i$返回给$A_{\mathrm{I}}$；当$L_{H1}$表不存在$l_i$时，那么$C$随机选择$l_i\in Z_q^*$，将$l_i$返回给$A_{\mathrm{I}}$并记录到$(R_i, \mathrm{ID}_i, M_i, \Delta, l_i)$中；

(3) 秘密值询问：当A_{I}对于ID_i的秘密值询问时，首先C检查$L_X(\mathrm{ID}_i, P_i, x_i)$表，当$L_X$表已包含$x_i$时，将$x_i$返回给$A_{\mathrm{I}}$；若$\mathrm{ID}_i=\mathrm{ID}_j$，游戏终止；若$\mathrm{ID}_i\neq\mathrm{ID}_j$，$C$随机选择$x_i\in Z_q^*$，计算$P_i=x_iP$，将$x_i$返回给$A_{\mathrm{I}}$并更新$L_X$记录；

(4) 部分私钥询问：当A_{I}对ID_i的部分私钥询问时，首先C检查$L_Y(\mathrm{ID}_i, h_i, V_i, y_i)$表，当$L_Y$表已包含$y_i$时，将$y_i$返回给$A_{\mathrm{I}}$；若$\mathrm{ID}_i\neq\mathrm{ID}_j$，$C$随机选择$y_i\in Z_q^*$，计算，$V_i=y_iP-h_iP_{\mathrm{T}}$，将$y_i$返回给$A_{\mathrm{I}}$并更新$L_Y$记录；若$\mathrm{ID}_i=\mathrm{ID}_j$，$C$随机选择$y_i\in Z_q^*$，计算$V_i=kP$(其中$k\in Z_q^*$为$C$已知随机数)，将$y_i$返回给$A_{\mathrm{I}}$并更新$L_Y$记录；

(5) 私钥生成询问：当A_{I}对ID_i的私钥询问时，C查看$L_{\mathrm{SK}}(\mathrm{ID}_i, x_i, y_i, \mathrm{sk}_i)$表，若$L_{\mathrm{SK}}$表已包含$\mathrm{ID}_i$的记录时，将$sk_i$返回给$A_{\mathrm{I}}$；当$L_{\mathrm{SK}}$表中无$\mathrm{ID}_i$的记录时，$C$执行秘密值询问和部分私钥询问之后，更新$L_{\mathrm{SK}}$表，将私钥$\mathrm{sk}_i=x_i+y_i$返回给$A_{\mathrm{I}}$；

(6) 公钥询问：当A_{I}对ID_i的公钥询问时，首先C检查$L_{\mathrm{PK}}(\mathrm{ID}_i, P_i, V_i)$表，若和$L_{\mathrm{PK}}$表已包含$\mathrm{ID}_i$的记录时，将$(P_i, V_i)$返回给$A_{\mathrm{I}}$；当$L_{\mathrm{PK}}$表中无$\mathrm{ID}_i$的记录时，则执行秘密值询问和部分私钥询问之后，更新L_{PK}表，将(P_i, V_i)返回给A_{I}；

(7) 公钥替换询问：A_{I}可以将ID_i的公钥(P_i, V_i)替换为(P_i', V_i')；

(8) 签名询问：当A_{I}就$(\mathrm{ID}_i, M_i, \Delta)$签名询问时，若$\mathrm{ID}_i\neq\mathrm{ID}_j$，则正常签名，$C$随机选取$r_i\in Z_q^*$，计算$R_i=r_iP$，$S_i=r_iP_{\mathrm{pub}}+\mathrm{sk}_i\cdot l_i$。签名为$\sigma_i'=(R_i, S_i)$。$C$将$\sigma_i'$返回给$A_{\mathrm{I}}$。若$\mathrm{ID}_i=\mathrm{ID}_j$，游戏终止；

(9) 聚合签名询问：当A_{I}就$(\mathrm{ID}_i, M_i, \Delta)$$(1\le i\le n)$聚合签名询问时，若所有用户$\mathrm{ID}_i(1\le i\le n)$都有$\mathrm{ID}_i\neq\mathrm{ID}_j$，则正常签名，$C$从签名$\sigma_i'=(R_i, S_i)$中计算$R=\sum_{i=1}^{n}R_i$，$S=\sum_{i=1}^{n}S_i$，即聚合签名为$\sigma=(R, S)$，$C$将$\sigma$返回给$A_{\mathrm{I}}$。若存在$\mathrm{ID}_i=\mathrm{ID}_j(1\le i\le n)$，游戏终止；

伪造阶段 A_{I}经过有限次数的询问后，输出关于$(\mathrm{ID}_i, M_i, \sigma_i')(1\le i\le n)$的聚合签名$\sigma=(R, S)$，其中至少有1个$\mathrm{ID}_i$未经过秘密值询问，部分私钥询问和私钥生成询问；至少有1个消息M_i未进行签名询问。当所有用户$\mathrm{ID}_i(1\le i\le n)$都有$\mathrm{ID}_i\neq\mathrm{ID}_j$，游戏终止。若至少有1个$\mathrm{ID}_i(1\le i\le n)$存在$\mathrm{ID}_i=\mathrm{ID}_j$，那么$C$在 和$L_D$中查询$\mathrm{ID}_i$对应记录，等式$S\cdot P=\sum_{i=1}^{n}[R_iP_{\mathrm{pub}}+(P_i+V_i+P_{\mathrm{T}}h_i)l_i]$是否成立。成立输出$b=\left\{S-\sum_{i=1,i\neq j}^{n}[r_iP_{\mathrm{pub}}+(x_i+y_i)l_i]-r_jP_{\mathrm{pub}}-(x_j+k)l_j\right\}(h_j\cdot l_j)^{-1}$；否则，$C$未解决困难问题。

计算困难问题过程如下：

$$\begin{aligned}S\cdot P&=\sum_{i=1}^{n}[R_iP_{\mathrm{pub}}+(P_i+V_i+P_{\mathrm{T}}h_i)l_i]\\&=\sum_{i=1,i\neq j}^{n}[r_iP\cdot P_{\mathrm{pub}}+(x_iP+y_iP-h_iP_{\mathrm{T}}+P_{\mathrm{T}}h_i)l_i]\\&\quad+r_jP\cdot P_{\mathrm{pub}}+(x_jP+kP+bPh_j)l_j\end{aligned}\tag{1}$$

$$\begin{aligned}S=&\sum_{i=1,i\neq j}^{n}[r_iP_{\mathrm{pub}}+(x_i+y_i)l_i]+r_jP_{\mathrm{pub}}\\&+(x_j+k)l_j+bh_jl_j\end{aligned}\tag{2}$$

$$\begin{aligned}b=&\left\{S-\sum_{i=1,i\neq j}^{n}[r_iP_{\mathrm{pub}}+(x_i+y_i)l_i]-r_jP_{\mathrm{pub}}\right.\\&\left.-(x_j+k)l_j\right\}(h_j\cdot l_j)^{-1}\end{aligned}\tag{3}$$

综上所述，若A_{I}成功伪造一个聚合签名，则C便可以通过A_{I}求得b，即C成功地解决了DLP问题，而DLP问题是一个困难问题，故本方案具有不可伪造性。 证毕

定理2　在随机语言模型中，攻击者A_{Π}在自适应选择消息攻击下，依赖离散对数困难问题，存在不可伪造性。若攻击者A_{Π}能够以一个不可忽略的PPT内成功伪造一个有效聚合签名，则存在挑战者C能够以不可忽略的优势在多项式时间内解决DLP问题。

证明　攻击者A_{Π}已知元组(P, bP)，伪造有效的聚合签名。挑战者C利用A_{Π}求困难问题，解b的值。假定目标用户ID_j。C与A_{Π}交互过程如下：

系统初始化阶段　挑战者C建立系统模型，执行系统建立算法，生成系统公开参数$\mathrm{params}=\{G, q, P, P_{\mathrm{T}}, H, H_1\}$和系统主密钥$\lambda$，系统公钥$P_{\mathrm{T}}=\lambda P$。生成随机数$P_{\mathrm{pub}}\in Z_q^*$作为用户的共享公钥，挑战者$C$将系统公开参数params，系统主密钥$\lambda$和$P_{\mathrm{pub}}$发送给敌手$A_{\Pi}$。

询问阶段　在此阶段中，挑战者C建立维护表$L_H, L_{H1}, L_X, L_Y, L_{\mathrm{SK}}, L_{\mathrm{PK}}$，记录下询问过程。其中表的初始值均为空。在$A_{\Pi}$询问阶段与$A_{\mathrm{I}}$询问阶段相比较而言，少了公钥替换询问，在部分私钥询问阶段有所不同，其他地方不变。部分私钥询问过程如下：

部分私钥询问：当A_{Π}对ID_i的部分私钥询问时，首先C检查$L_Y(\mathrm{ID}_i, h_i, V_i, y_i)$表，当$L_Y$表已包含$y_i$时，将$y_i$返回给$A_{\Pi}$；若$\mathrm{ID}_i\neq\mathrm{ID}_j$，$C$随机选择$y_i\in Z_q^*$，计算，$V_i=y_iP-h_iP_{\mathrm{T}}$，将$y_i$返回给$A_{\Pi}$并更新$L_Y$记录；若$\mathrm{ID}_i=\mathrm{ID}_j$，$C$随机选择$b\in Z_q^*$，计算$V_i=bP$，将$y_i$返回给$A_{\Pi}$并更新$L_Y$记录。

伪造阶段　A_{Π}经过有限次数的询问后，输出关于$(\mathrm{ID}_i, M_i, \sigma_i')(1\le i\le n)$的聚合签名$\sigma=(R, S)$，其中至少有1个$\mathrm{ID}_i$未经过秘密值询问，部分私钥询问和私钥生成询问；至少有一个消息M_i未进行签名询问。当所有用户$\mathrm{ID}_i(1\le i\le n)$都有$\mathrm{ID}_i\neq\mathrm{ID}_j$，游戏终止。若至少有1个$\mathrm{ID}_i(1\le i\le n)$存在$\mathrm{ID}_i=\mathrm{ID}_j$，那么$C$在$L_H, L_{H1}, L_{\mathrm{SK}}, L_{\mathrm{PK}}$和$L_D$中查询$\mathrm{ID}_i$对应记录，等式$S\cdot P=\sum_{i=1}^{n}[R_iP_{\mathrm{pub}}+(P_i+V_i+P_{\mathrm{T}}h_i)l_i]$是否成立。成立输出$b=\left\{S-\sum_{i=1,i\neq j}^{n}[r_iP_{\mathrm{pub}}+(x_i+y_i)l_i]-r_jP_{\mathrm{pub}}-(x_j+P_{\mathrm{T}}h_j)l_j\right\}l_j^{-1}$；否则，$C$未解决困难问题。

计算困难问题过程如下：

$$\begin{aligned}S\cdot P&=\sum_{i=1}^{n}[R_iP_{\mathrm{pub}}+(P_i+V_i+P_{\mathrm{T}}h_i)l_i]\\&=\sum_{i=1,i\neq j}^{n}[r_iP\cdot P_{\mathrm{pub}}+(x_iP+y_iP-h_iP_{\mathrm{T}}+P_{\mathrm{T}}h_i)l_i\\&\quad+r_jP\cdot P_{\mathrm{pub}}+(x_jP+bP+P_{\mathrm{T}}h_j)l_j\end{aligned}\tag{4}$$

$$\begin{aligned}S=&\sum_{i=1,i\neq j}^{n}[r_iP_{\mathrm{pub}}+(x_i+y_i)l_i]+r_jP_{\mathrm{pub}}\\&+(x_j+P_{\mathrm{T}}h_j)l_j+bl_j\end{aligned}\tag{5}$$

$$\begin{aligned}b=\Bigg\{&S-\sum_{i=1,i\neq j}^{n}[r_iP_{\mathrm{pub}}+(x_i+y_i)l_i]-r_jP_{\mathrm{pub}}\\&-(x_j+P_{\mathrm{T}}h_j)l_j\Bigg\}l_j^{-1}\end{aligned}\tag{6}$$

综上所述，若A_{Π}成功伪造一个聚合签名，则C便可以通过A_{Π}求得b，即C成功地解决了DLP问题，而DLP问题是一个困难问题，故本方案具有不可伪造性。　　证毕

5　多方合同签署协议

本协议中，对于所签署的合同，可以是不同版本，也可以是相同版本的合同。协议包括区块链下由签署方签名聚合生成的可验证聚合签名和区块链上签署方临时私钥的交换过程。整个过程中，由于签名过程链下执行，合同的内容和敏感信息不予公布，从而保证了该协议的隐私性，而各个签署方的临时私钥的公开过程在区块链上执行，保证了该协议的公平性。由于区块链上公开的是临时私钥，也保证了用户前向和后向合同签署的安全性，多方合同签署的过程如下。

5.1　区块链链下阶段

本阶段中，各签署方在链下针对已经达成一致的合同信息，通过执证无证书聚合可验证签名方案，生成聚合签名σ，若验证签名有效，则进入区块链链上阶段，否则协议停止。如下：

(1) 各签署方对合同信息M达成一致；

(2) 各签署方执证无证书聚合可验证签名方案，生成聚合签名σ；

(3) 验证签名是否有效，若通过验证进入区块链链上阶段，否则协议停止。

5.2　区块链链上阶段

各签署方已经在链下对合同签署了一个有效的聚合签名，在本阶段包括预备、提交保证金、索得保证金和失败4部分组成。各签署方将公开自己的临时私钥。为了公平的释放或获取签署方的私钥，各签署方首先预存一定数额的保证金，当签署方公开有效的临时私钥后可赎回担保金，一旦有签署方出现不诚实的行为，此签署方将失去所提交的担保金额，并将奖励给其他签署方。具体过程如下：

(1) 预备(preparation)：为链上公开临时私钥做准备。

(a) 各签署方协商释放临时私钥的顺序，假设释放顺序为$U_1, U_2, \cdots, U_n$；

(b) 约定签署方U_1指定价值为dB的交易T_1；

(c) 对于$i \in \{2,3,\cdots,n\}$，签署方U_i指定价值为$(n-1)$dB的交易T_i；

(d) 交易时间锁$t_1 < t_2 < \cdots < t_n$。

(2) 提交保证金(deposit)：各签署方首先预存一定数额的保证金。

签署方U_1指定价值为dB的交易T_1，将T_1作为输入生成deposit交易并全网广播，如图1所示。

对于$i \in \{2,3,\cdots,n\}$，签署方U_i指定价值为$(i-1)$dB的交易T_i，将T_i作为输入生成deposit交易并全网广播，如图2所示。

(3) 索得担保金(claim)：在有效的时间内，各签署方诚实地释放所拥有的临时私钥后可拿回保证金。

(a) 当$i \neq n$时，在有效的时间内，前一个签署方诚实地释放所拥有的临时私钥后，可从下一个签署方处获得dB；

(b) 当$i = n$时，U_n诚实地释放临时私钥后从U_1处获得dB。

(4) 失败(refund)：若出现U_i在有效的时间内未诚实释放临时私钥，则$U_1, U_2, \cdots, U_{i-1}$均可获得相应的补偿。

(a) 当$i = 1$时，U_1在有效的时间内未诚实释放临时私钥，各签署方拿回其担保金，协议结束；

(b) 当$i \neq 1$时，若签署方U_i为首个在有效的时间内未诚实释放临时私钥，则$U_{i+1}, U_{i+2}, \cdots, U_n$赎回各自的担保金，而$U_i$失去所提交的担保金$(i-2)$ dB，这笔金额将奖励给$U_2, U_3, \cdots, U_{i-1}$每人获得dB。由

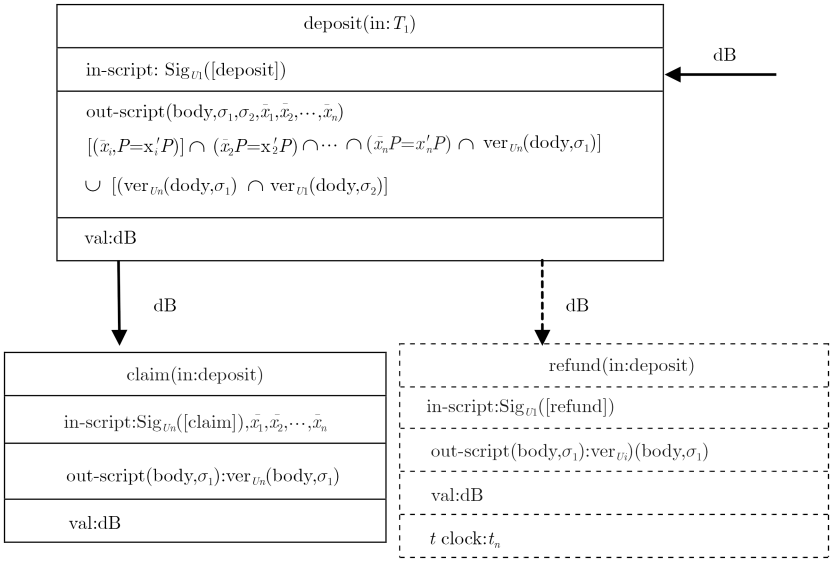

图 1 交易T_1

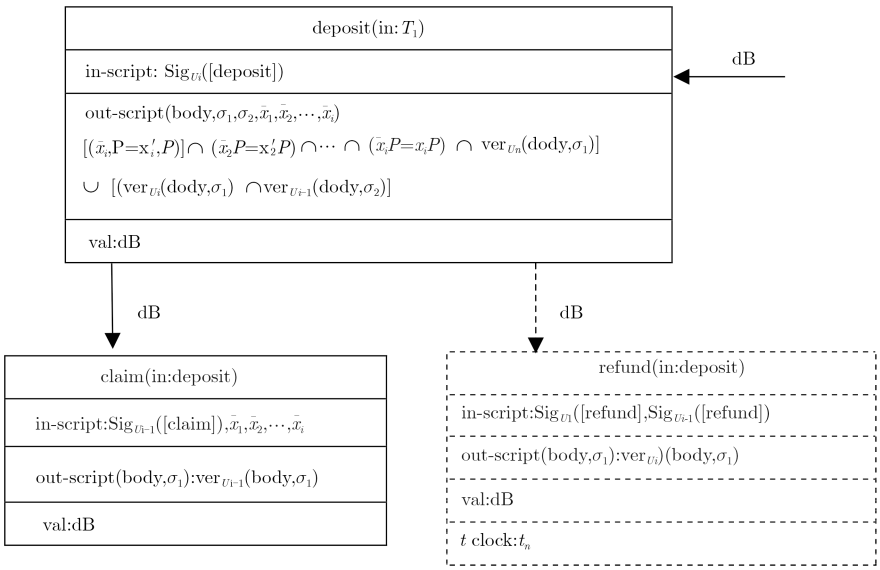

图 2 交易T_i

于未成功共享各签署方的临时私钥，所以U_n未获得U_1的奖励dB，故U_i无需再次奖励U_1。

提取　若各签署方正确有效地在区块链上公开自己的临时私钥x_i'，则各签署方均可获得所有临时私钥，便可计算出共享密钥x_{pub}，由此可从聚合签名中提取出普通签名。

(1) 各签署方计算共享密钥$x_{\text{pub}}=\sum_{i=1}^{n} x_i'$；

(2) 各签署方从σ_i'中提取普通签名$\sigma_i''=(R_i,S_i'')$，$S_i''=S_i-R_i\cdot x_{\text{pub}}$。

6　多方合同签署协议的性能分析

6.1　多方合同签署协议的不透明性

假设攻击者A在不知道共享密钥x_{pub}的情况下，能够从某一签署者的签名$\sigma'=(R_i,S_i)$中提取普通签名$\sigma_i''=(R_i,S_i'')$，可归约为解决CDH困难问题，在已知P, $R_i=r_iP$和$P_{\text{pub}}=x_{\text{pub}}P$时，由$S_i=r_iP_{\text{pub}}+\text{sk}_i\cdot l_i$，得$S_i-S_i''=r_iP_{\text{pub}}=r_i\cdot x_{\text{pub}}P$，攻击者$A$需要计算出$r_i\cdot x_{\text{pub}}P$，而这一结果违背了CDH困难问题。故在本方案中可验证签名具有不透明性。

6.2　多方合同签署协议的可提取性

本方案中的验证者可为签名的任意参与者(合同的签署方)。当所有签署方都诚实地签署合同，并在区块链上公布自己的临时私钥x_i'，则所有的签署方均可计算共享密钥x_{pub}。故任意签署方可以从其他任意签署方签名$\sigma'=(R_i,S_i)$中提取其普通签名$\sigma_i''=(R_i,S_i'')$，即$S_i''=S_i-r_iP_{\text{pub}}=S_i-R_i\cdot x_{\text{pub}}$。

6.3　多方合同签署协议的安全性

在多方合同签署的过程中，系统的安全性由链上和链下同时保证，文中第4节安全性证明已证在区块链链下，各签署方签署聚合签名时的正确性和安全性。在区块链上部分，由区块链的性质保证了签署方所公开的临时密钥真实有效地上传至区块链，保证了链上部分的安全性。而所有签署方公开的仅为临时密钥，所以在合同签署的过程中，同时保证了所有签署者的隐私安全，也不影响其之前签署合同的有效性。

6.4　多方合同签署协议的公平性

多方合同签署的过程中的公平性主要体现在签署方是否是诚实的用户。

(1) 所有签署方均是诚实的用户，则各签署方在链下可以得到关于合同的聚合签名，在链上经过诚实地公开自己的临时私钥，各签署方获得所有参与者的临时私钥，即可得到共享密钥，由共享密钥可从聚合签名中提取普通签名；

(2) 签署方中存在不诚实的用户，若各签署方在链下签名时，存在未能正确签署的签名信息时，则不能通过签名验证，即签署合同失败。若各签署方在链上公开临时私钥时，存在不诚实的行为，未能诚信地公开临时私钥，则首个不诚实者失去所提交的保证金，这笔担保金将补偿给前面已公开临时私钥的签署方，后续签署者赎回担保金。

综上若多方合同签署的过程中，签署方诚实地执行合同签署的全过程，则合同签署成功。若存在不诚实的签署者，则合同签署失败，不诚实者将付出代价用以补偿已公开临时私钥的签署者。

6.5　多方合同签署协议的高效性

分析本协议的效率并与其它合同签署协议作比较，如表1所示。其中，n为签署方的数量，e为一次双线性对运算，s为G中1次标量乘运算，h为1次哈希运算。

表 1　本协议与现有协议比较

协议	签署方	第三方	签名验证次数	签名验证开销
文献[9]	2	√	2	$3e+1s+1h$
文献[10]	n	√	n	$(2n+1)e+ns+nh$
文献[12]	n	–	1	$(4n+1)e+(2n+1)h$
本协议	n	–	1	$(3n+2)s+2nh$

由表1可以看出在这几个合同签署协议中文献[9]是两者在第三方的参与下签署合同，经过两次签名验证。文献[10,12]和本协议均是多方合同签署协议，文献[10]借助了第三方的参与，且验证次数和签署方的数目一致。文献[12]和本协议抛却第三方，直接是签署方直接签署合同，只验证1次聚合签名就可以。但是文献[9,10,12]均采用双线性对运算，而本协议采用离散对数运算。同等条件下耗时较少，运行1次双线性对约是模指数运算的2倍，是点乘运算的20倍[15]。所以本协议大大提高了签名过程中的运算效率。

7　结束语

本文提出一个高效的多方合同签署协议，利用区块链技术替换第三方参与的合同签署协议分为了链下和链上两部分，在保证安全性的前提下，用提高链下聚合签名算法的效率来提高整体合同签署的效率。但该协议理论上还存在其他的优化方式，这是接下来要研究的方向。

参 考 文 献

[1] BLUM M. How to exchange (secret) keys[C]. The Fifteenth Annual ACM Symposium on Theory of Computing, Boston,

USA, 1983: 440–447.

[2] Al-RIYAMI S S and PATERSON K G. Certificateless public key cryptography[C]. The 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, China, 2003: 452–473.

[3] BONEH D, GENTRY C, LYNN B, *et al*. Aggregate and verifiably encrypted signatures from bilinear maps[C]. 2003 International Conference on the Theory and Applications of Cryptographic Techniques, Warsaw, Poland, 2003: 416–432.

[4] 周彦伟, 杨波, 张文政. 高效可证安全的无证书聚合签名方案[J]. 软件学报, 2015, 26(12): 3204–3214. doi: 10.13328/j.cnki.jos.004830.
ZHOU Yanwei, YANG Bo, and ZHANG Wenzheng. Efficient and provide security certificateless aggregate signature scheme[J]. *Journal of Software*, 2015, 26(12): 3204–3214. doi: 10.13328/j.cnki.jos.004830.

[5] 曹素珍, 郎晓丽, 刘祥震, 等. 可证安全的高效无证书聚合签名方案[J]. 信息网络安全, 2019, 19(1): 42–50. doi: 10.3969/j.issn.1671-1122.2019.01.006.
CAO Suzhen, LANG Xiaoli, LIU Xiangzhen, *et al*. Probably secure and efficient certificateless aggregate signature scheme[J]. *Netinfo Security*, 2019, 19(1): 42–50. doi: 10.3969/j.issn.1671-1122.2019.01.006.

[6] 许芷岩, 吴黎兵, 李莉, 等. 新的无证书广义指定验证者聚合签名方案[J]. 通信学报, 2017, 38(11): 2017220.
XU Zhiyan, WU Libing, LI Li, *et al*. New certificateless aggregate signature scheme with universal designated verifier[J]. *Journal on Communications*, 2017, 38(11): 2017220.

[7] 苏靖枫, 柳菊霞. 不含双线性对的高效无证书聚合签密方案[J]. 计算机应用, 2018, 38(2): 374–378. doi: 10.11772/j.issn.1001-9081.2017081984.
SU Jingfeng and LIU Juxia. Efficient certificateless aggregate signcryption scheme without bilinear pairings[J]. *Journal of Computer Applications*, 2018, 38(2): 374–378. doi: 10.11772/j.issn.1001-9081.2017081984.

[8] WAN Zhiguo, DENG R H, and LEE D. Electronic contract signing without using trusted third party[C]. Proceedings of the 9th International Conference on Network and System Security, New York, USA, 2015: 386–394.

[9] 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.
TIAN Haibo, HE Jiejie, and FU Liqing. A privacy preserving fair contract signing protocol based on block chains[J]. *Journal of Cryptologic Research*, 2017, 4(2): 187–198.

[10] 吴进喜, 高莹, 张宗洋, 等. 基于区块链的多方隐私保护公平合同签署协议[J]. 信息安全学报, 2018, 3(3): 8–16.
WU Jinxi, GAO Ying, ZHANG Zongyang, *et al*. A multi-party privacy preserving fair contract signing protocol based on blockchains[J]. *Journal of Cyber Security*, 2018, 3(3): 8–16.

[11] HUANG Hui, LI K C, and CHEN Xiaofeng. A fair three-party contract singing protocol based on blockchain[C]. The 9th International Symposium on Cyberspace Safety and Security, Xi'an, China, 2017: 72–85.

[12] 高莹, 吴进喜. 基于区块链的高效公平多方合同签署协议[J]. 密码学报, 2018, 5(5): 556–567.
GAO Ying and WU Jinxi. Efficient multi-party fair contract signing protocol based on blockchains[J]. *Journal of Cryptologic Research*, 2018, 5(5): 556–567.

[13] 周彦伟, 杨波, 王青龙. 安全的无双线性映射的无证书签密机制[J]. 软件学报, 2017, 28(10): 2757–2768. doi: 10.13328/j.cnki.jos.005150.
ZHOU Yanwei, YANG Bo, and WANG Qinglong. Secure certificateless signcryption scheme without bilinear pairing[J]. *Journal of Software*, 2017, 28(10): 2757–2768. doi: 10.13328/j.cnki.jos.005150.

[14] 韦性佳, 张京花, 刘增芳, 等. 具有前向安全性质的基于身份的聚合签名方案[J]. 计算机科学, 2018, 45(6A): 387–391.
WEI Xingjia, ZHANG Jinghua, LIU Zengfang, *et al*. Identity based aggregate signature scheme with forward security[J]. *Computer Science*, 2018, 45(6A): 387–391.

[15] 王亚飞, 张睿哲. 强安全无对的无证书签名方案[J]. 通信学报, 2013, 34(2): 94–99. doi: 10.3969/j.issn.1000-436x.2013.02.011.
WANG Yafei and ZHANG Ruizhe. Strongly secure certificateless signature scheme without pairings[J]. *Journal on Communications*, 2013, 34(2): 94–99. doi: 10.3969/j.issn.1000-436x.2013.02.011.

曹素珍：女，1976年生，副教授，研究方向为公钥密码学和软件安全.

王 斐：女，1992年生，硕士生，研究方向为密码学与信息安全.

郎晓丽：女，1993年生，硕士生，研究方向为密码学与信息安全.

汪 锐：男，1991年生，硕士生，研究方向为密码学与信息安全.

刘雪艳：女，1978年生，副教授，研究方向为组密钥协商、密码协议形式化分析.