计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2016.20160413
53(10):2412-2429,2016

基于 DAA-A 的改进可授权电子现金系统

柳 欣[1,2] 张 波[3]
[1](山东青年政治学院信息工程学院 济南 250103)
[2](山东省高校信息安全与智能控制重点实验室(山东青年政治学院) 济南 250103)
[3](济南大学信息科学与工程学院 济南 250022)
(lxonne@163.com)

Improved Endorsed E-Cash System with DAA-A

Liu Xin[1,2] and Zhang Bo[3]
[1](*School of Information Engineering*, *Shandong Youth University of Political Science*, *Jinan* 250103)
[2](*Key Laboratory of Information Security and Intelligent Control in Universities of Shandong* (*Shandong Youth University of Political Science*), *Jinan* 250103)
[3](*School of Information Science and Engineering*, *University of Jinan*, *Jinan* 250022)

Abstract At present, the existing endorsed e-cash system has a low communication efficiency, and its fair exchange protocol employs inefficient cut-and-choose proofs. In addition, the centralized TTP (trusted third party) is vulnerable to denial-of-service attacks. So far, several related fair payment systems have been proposed. Unfortunately, some of them use cut-and-choose proofs, and the others adopt verifiable encryption schemes with security flaw. Inspired by the idea of self-blindable attribute-based credentials, a concrete DAA-A (direct anonymous attestation with attributes) scheme is constructed. Based on the new DAA-A scheme, an improved endorsed e-cash system is proposed, which achieves a high level of exculpability. In order to improve users' computational efficiency in the spending process, the set-membership proof by Arfaoui et al's is adopted, and the efficiency of user's signature of knowledge is also optimized with the technique of pre-computation. In order to bypass the expensive cut-and-choose proof, a new optimistic fair exchange sub-protocol supporting distributed TTPs is provided. Furthermore, if combined with the Golle-Mironov model, the new system also suits for the environment of outsourcing computing. Compared with the previous similar ones, the new system meets several desirable properties simultaneously, i.e., it supports multiple payments, and does not depend on cut-and-choose proofs and allows users to be stateless, etc. What's more, the fair exchange protocol of the new system considers the risk of denial-of-service attacks.

Key words endorsed e-cash; direct anonymous attestation; fair exchange; cut-and-choose proofs; outsourced computation

摘 要 当前,已有的可授权电子现金系统通信效率不高,同时其公平交换子协议要求使用低效的 cut-and-choose 证明技术且集中式的可信第三方(trusted third party, TTP)容易遭受拒绝服务攻击.此外,多个相关的公平支付系统或者要求使用 cut-and-choose 证明技术,或者使用了具有安全性缺陷的可验

收稿日期:2016-06-14;**修回日期**:2016-08-10
基金项目:山东省自然科学基金项目(ZR2015FL023,ZR2014FL011);山东省高等学校科技计划项目(J14LN61);山东青年政治学院博士科研启动经费资助项目(14A007)
This work was supported by the Natural Science Foundation of Shandong Province of China (ZR2015FL023, ZR2014FL011), the Project of Shandong Province Higher Educational Science and Technology Program (J14LN61), and the Doctoral Research Start-up Funding Project of Shandong Youth University of Political Science (14A007).

证加密技术.利用基于属性的自盲化证书系统构造了一个具有属性的直接匿名证明(direct anonymous attestation with attributes, DAA-A)方案,然后基于该方案构造了满足更强可开脱性的可授权电子现金系统.为了提高用户端在支付过程中的运算效率,使用了 Arfaoui 等人的集合关系证明协议,同时利用预计算技术对用户的知识签名进行了效率优化.为了避免执行低效的 cut-and-choose 证明,设计了一个支持分布式 TTP 的乐观公平交换子协议.通过与 Golle-Mironov 模型相结合,新系统可以应用于外包计算领域.与已有同类系统相比,新系统同时满足允许多次支付、无需使用 cut-and-choose 技术和用户无状态性等多个理想性质.此外,新系统的公平交换子协议引入了分布式 TTP,即考虑了拒绝服务攻击的风险.

关键词　可授权电子现金;直接匿名证明;公平交换;cut-and-choose 证明;外包计算

中图法分类号　TP309

在 IEEE S&P 2007 会议上,Camenisch 等人[1]指出标准电子现金系统的缺点是当用户使用某个电子货币(e-coin)购买数字商品时,一旦交易意外终止,用户可能会失去隐私,甚至遭受资金损失.因此,有必要为用户提供一种能以安全且公平的方式实现购买交易的机制.为此,Camenisch 等人[1]提出可授权电子现金(endorsed e-cash)的概念.此类系统允许用户事先向商家提供盲化的电子货币 $coin'$(称为未经授权的货币).只有获得所期望的商品后,用户才会向商家提供一个称为背书(endorsement)的秘密元素 end.该元素可用于将 $coin'$ 转换为得到授权的货币 $coin$.此类系统具有 3 个优点:1)用户可以为同一个 $coin$ 产生多个未经授权的版本(如 $coin'_1$, $coin'_2$,…),且这些版本相互间是不可关联的;2) $coin'$ 本身是匿名的且无法直接使用,因此,即使交易失败,也不会导致用户在隐私和资金方面的损失;3)有利于将"电子货币与数字商品间的公平交换"问题简化为"轻量级的背书元素与数字商品间的公平交换"问题,从而简化了购买交易的设计过程.值得注意的是,可授权电子现金是紧凑电子现金[2]的扩展形式,即一次提取 2^n 个货币,然后每次支付其中一个货币.另一类相关研究是可分电子现金[3],即一次提取价值为 2^n 的货币,然后每次支付其中价值为 $2^l(0\leqslant l\leqslant n)$ 的货币.

直接匿名证明(direct anonymous attestation, DAA)方案[4-6]是当前可信计算技术的一个重要分支.在此类方案中,嵌入了可信平台模块(trusted platform module, TPM)芯片的用户主机平台(Host)可以向验证者证明自己掌握有效的认证令牌.本质上讲,该令牌是由发布权威为 TPM 私钥颁发的数字证书.认证过程的主要运算由 Host 承担,且 TPM 仅执行少量关键性的运算.具有属性的直接匿名证明(direct anonymous attestation with attributes, DAA-A)[5]是对 DAA 方案的扩展,即用户的认证令牌是关于 TPM 私钥 k_0 和用户属性向量 $(k_1,k_2,\cdots,k_n)$ 的数字证书.同时,在认证过程中,用户可以灵活设置由"公开"属性和"隐藏"属性构成的集合,从而符合了隐私保护的最小信息泄露原则.本文认为,可以将 DAA-A 方案作为设计可授权电子现金系统的有效本原.具体原因包括 4 点:

1) 在 DAA-A 方案中,TPM 与 Host 是 2 个不同的实体,只要 TPM 保持诚实,即使在 Host 被攻破的情况下,仍然可以保护用户不受陷害;

2) 可授权电子现金系统的取款和支付协议分别对应于 DAA-A 方案的用户注册和签名协议;

3) DAA-A 方案允许用户有选择性地对个人敏感属性进行隐藏,从而有利于将个人信息的泄露降低至最低程度;

4) DAA-A 方案容易与其他的可信计算技术(如基于 TPM 的群托管协议[7])相结合,从而设计出效率更高的公平交换协议.

我们认为,Camenisch 等人[1]的系统(简称原始系统)存在 2 个主要缺点:1)该系统是基于强 RSA 假设设计的,因此用户在支付过程中的通信效率不高.2)背书元素 end 与数字商品之间的交换是基于 Asokan 等人[8]的乐观公平交换协议实现的,从而继承了后者的 4 个缺点,即:①要求用户在交换过程中执行低效的 cut-and-choose 证明过程;②要求交换双方在发生争执的情况下向可信第三方(trusted third party, TTP)申请仲裁,即必须无条件地信任 TTP;③TTP 本身是集中式的,因此容易遭受拒绝服务攻击;④交换双方都可能成为争议解决子协议的发起者,因此用户必须保持状态[9].

除了原始系统之外,相关的代表性文献总结如下:在文献[10]中,Blanton 提出一个改进的有条件电子现金系统.此类系统将标准电子现金系统的支付

协议替换为有条件的转移协议. 该协议要求付款方与收款方共同参与,且收款方最终将获得一笔有条件的付款,即该付款仅能在某个公开条件 R 得到满足后才能得到兑现. Blanton 的系统[10]与可授权电子现金系统[1]的 2 个主要区别是:1)要求由 TTP 宣布条件 R 是否得到满足;2)允许收款人将所得的有条件的付款转移给他人. 然而,Blanton 的系统效率不高且要求付款方与收款方完全信任 TTP. 在文献[11]中,Chen 等人基于可授权电子现金技术[1]和 Golle-Mironov 外包计算模型[12]提出一个公平的有条件支付系统. 该系统允许用户将运算任务分为多项更小的任务,并将所得任务连同未经授权的电子货币 $coin'$ 外包给志愿的计算机(称为 worker). 每当 worker 完成任务并返回正确的运算结果,用户会对事先支付的 $coin'$ 提供授权. 考虑到双方相互并不信任,该系统还提供了借助 TTP 确保交换过程公平性的机制. 然而,Chen 等人的系统存在 3 个主要缺点:1)底层的可授权电子现金系统是基于 Brands 电子现金系统构造的. 但是,后者的安全性尚未得到完全证明[13],而且不支持多次支付. 因此,用户需要在每次外包运算之前重新执行取款协议. 2)TTP 并非完全可信,即 TTP 有可能会与用户或 worker 进行合谋. 3)该系统的公平交换协议是基于 Ateniese[9]的关于离散对数的可验证加密方案实现的. 需要指出的是,Ateniese 的技术存在缺陷,因为其底层加密方案仅满足较弱的语义安全性[14]. 在文献[15]中,Carbunar 等人提出一个适用于外包运算环境的公平支付系统. 该系统存在 3 个主要缺点:1)底层电子现金系统不满足匿名性,因此无法保护用户的隐私;2)为了验证用户预先支付的电子货币是否有效,用户需要与 worker 执行低效的 cut-and-choose 证明协议;3)并未引入争议解决机制,即未考虑 worker 有可能不向用户提供外包运算结果的情况. 最近,Küpcü[16]提出一个新的公平支付系统. 在外包运算开始之前,用户与 worker 需要事先在同一家银行开设账户. 银行负责对双方账户余额进行托管,且 worker 需要为自己提供错误运算结果的恶意行为支付罚金. 需要指出的是,为了实现对错误结果的检测,该系统要求用户将同一个任务外包给多个 worker,因此显著增加了用户的通信和运算耗费.

本文着重研究了基于 DAA-A 技术设计可授权电子现金系统的问题以及如何利用所得系统解决外包运算环境下的公平支付问题. 具体贡献体现在 4 个方面:

1) 本文提出新的可授权电子现金系统安全模型. 已有的安全模型[1]是基于模拟方式定义的. 新的安全模型是基于游戏方式定义的,且考虑了用户端 Host 被攻破而 TPM 芯片保持诚实的情况,即定义了更强的可开脱性.

2) 本文基于 Ringers 等人[17]的自盲化证书系统具体实现了文献[5]中的 DAA-A 方案框架. 然后,基于所得 DAA-A 方案构造了改进的可授权电子现金系统,使得用户在支付阶段只需执行双线性群 G_1 上的高效运算.

3) 本文提出与可授权电子现金系统紧密结合的公平交换协议. 该协议支持用户利用背书元素 end 购买商家的数字商品 m,且允许双方在发生争执时向分布式的 TTP 申请仲裁. 同时,用户只是争议解决子协议的被动参与方,因此无需保持状态.

4) 本文提出新系统的具体应用实例,即构造了支持用户购买外包运算结果的公平支付系统.

1 预备知识

定义双线性群对(G_1, G_2),使得存在双线性映射 $\hat{e}: G_1 \times G_2 \to G_T$,其中群 G_1, G_2, G_T 的阶数均为素数 p.

1) XKEA(the extended knowledge of exponent)假设[17]. 令 $\mathfrak{A}$ 表示 PPT(probabilistic polynomial time)算法. 假设当以元组 $(K_1, K_2, \cdots, K_m, K_1^a, K_2^a, \cdots, K_m^a) \in G_1^{2m}$ 和 $Q_1, Q_2, \cdots, Q_r \in G_2$(其中,$a \in \mathbb{Z}_p^*$,$m$ 与 r 关于 $|p|$ 满足多项式关系,且 $(K_1, K_2, \cdots, K_m)$ 均不为群 G_1 的单位元)为输入,$\mathfrak{A}$ 能输出数对 $(L, L') \in G_1^2$,则存在 PPT 知识提取器 $\chi_{\mathfrak{A}}$,使得当以 $(K_1, K_2, \cdots, K_m, K_1^a, K_2^a, \cdots, K_m^a) \in G_1^{2m}$ 和 $\mathfrak{A}$ 的随机抛币为输入,$\chi_{\mathfrak{A}}$ 能在压倒性的概率下提取出元素 L 关于底数 $K_1, K_2, \cdots, K_m$ 的离散对数表达式 $c_1, c_2, \cdots, c_m$,满足 $L = K_1^{c_1} \times K_2^{c_2} \times \cdots \times K_m^{c_m}$.

2) XDH(the external Diffie-Hellman)假设[18]. XDH 假设表明,DDH 问题在群 G_1 上是困难的,即给定元组 $(g^x, g^y, g^z) \in G_1^3$,难以对 2 种情况做出区分:①$z = xy$;②$z$ 为 $\mathbb{Z}_p$ 上的随机元素.

3) 属性空间上的签名方案以及基于属性的自盲化证书系统. 最近,Ringers 等人[17]提出一个签名方案,该方案由密钥产生、签名、验证 3 个步骤构成:

Step1. 签名者选取 $Q \in_R G_2$,选取 $a, a_0, a_1, \cdots, a_n, z \in_R \mathbb{Z}_p^*$,设置 $A = Q^a$,$A_0 = Q^{a_0}$,$A_1 = Q^{a_1}$,$A_2, \cdots,$

$A_n=Q^{a_n}$,$Z=Q^z$. 签名者设置公钥为$(Q,A,A_0,A_1,\cdots,A_n,Z)$,私钥为$(a,a_0,a_1,\cdots,a_n,z)$.

Step2. 给定属性向量$(k_0,k_1,\cdots,k_n)$,签名者选取$\kappa\in_R\mathbb{Z}_p^*$,$K\in_R G_1$,计算 $S=K^a$,$S_0=K^{a_0}$,$S_1=K^{a_1},\cdots,S_n=K^{a_n}$,$T=(KS^\kappa\prod_{i=0}^{n}S_i^{k_i})^z$. 最后,输出签名$(\kappa,K,S,S_0,S_1,\cdots,S_n,T)$.

Step3. 给定属性向量$(k_0,k_1,\cdots,k_n)$和对应的签名$(\kappa,K,S,S_0,S_1,\cdots,S_n,T)$,验证者验证是否同时满足:

① $\hat{e}(T,Q)=\hat{e}(C,Z)$,其中 $C=KS^\kappa\prod_{i=0}^{n}S_i^{k_i}$;

② $\hat{e}(S,Q)=\hat{e}(K,A)$;

③ $\hat{e}(S_i,Q)=\hat{e}(K,A_i)$,其中 $i=1,2,\cdots,n$.

此外,Ringers 等人利用上述方案构造了基于属性的自盲化证书系统. 该系统允许用户多次展示关于属性向量$(k_0,k_1,\cdots,k_n)$的证书,并且根据需要定义向验证者揭示的属性集合.

4) 关于离散对数的可验证加密技术. 假设证明者 P 与验证者 V 共享公开元素 $\mathfrak{g}^m$,且 P 额外掌握秘密元素 m,使得$(\mathfrak{g}^m,m)$满足离散对数关系 R. 假设 Enc 表示 Camenisch-Shoup 公钥加密方案[14]的加密算法,且 pk 表示 TTP 在该方案下的公钥. 在文献[14]中,Camenisch 等人提出关于离散对数的可验证加密技术. 利用该项技术,P 可以向 V 提供 $Enc_{pk}(m)$,并且向 V 证明 2 项事实:①$Enc_{pk}(m)$是利用 pk 产生的关于秘密元素 m 的 Camenisch-Shoup 方案密文;②秘密元素 m 是元素 $\mathfrak{g}^m$ 关于底数 $\mathfrak{g}$ 的离散对数.

5) 基于 TPM 的可验证群托管协议. 假设发送方 P 拥有公开元素 cmt,且 cmt 满足某个性质 R. 在文献[7]中,Tate 等人提出基于 TPM 的可验证群托管协议. 在该协议中,P 向接收方 V 提供关于秘密证据 end 的托管密文 $escrow$,并且能使得后者确信 P 掌握证据 end,满足$(cmt,end)\in R$. 同时,当自己与某个符合特定访问结构 Γ 的恢复代理集合进行合作时,就能得到证据 end;相反,则无法获得有关 end 的任何有用信息. Tate 等人指出,可以利用(k,n)门限技术实现访问结构 Γ,并且提供了该协议在 R 为离散对数关系情况下的具体实例. 此外,Tate 等人强调,若 end 为 P 在公平交换过程中的秘密信息,则可以利用该协议取代低效的 cut-and-choose 证明技术和可验证加密技术. 该协议的显著特点是,P 利用 TPM 芯片产生托管密文 $escrow$,并且使用存储于 TPM 内部受保护区域中的证明身份密钥(attestation identity key)AIK 产生对 $escrow$ 的签名,从而利用 V 对 TPM 芯片本身的信任取代了经典 cut-and-choose 证明技术借助验证双方的多轮交互而实现的统计上的信任.

6) 基于集合关系证明的区间证明协议. 在文献[19]中,Camenisch 等人提出一个基于集合关系证明的区间证明协议. 该协议的基本思想是,假设指定权威的 Boneh-Boyen 签名方案[20]公私钥对为$(Y=Q'^{\xi},\xi)$. 该权威定义公开集合 $\Phi=\{1,2,\cdots,\mathfrak{M}\}$以及对应的 Boneh-Boyen 签名集合 $\Sigma=\{o_1,o_2,\cdots,o_{\mathfrak{M}}\}=\{g^{\frac{1}{\xi+1}},g^{\frac{1}{\xi+2}},\cdots,g^{\frac{1}{\xi+\mathfrak{M}}}\}$. 对于证明者 P 而言,证明"掌握秘密元素 J 且满足 $1\leqslant J\leqslant\mathfrak{M}$"等价于证明"掌握某个 Boneh-Boyen 签名 o_J 且满足 $o_J\in\Sigma$". 最近,Arfaoui 等人[21]提出了该协议的改进形式,使得 P 在证明过程中无需执行对运算.

2　改进的可授权电子现金系统

2.1　设计思想

本文系统的设计思想如下:1)借助基于属性的自盲化证书系统[17]具体实现了 Chen 等人[5]的 DAA-A 方案框架;2)基于所得 DAA-A 方案构造了满足更强可开脱性的可授权电子现金系统. 本文系统的取款和支付协议分别对应于 DAA-A 方案的用户注册和签名协议. 通过取款协议,用户($TPM+Host$)获得银行为属性向量(k_0,k_1,k_2)颁发的证书(称为电子钱包 $wallet$). 其中,k_0 充当 TPM 私钥,属性 k_1 与 k_2 分别用于产生电子货币序列号 $\overline{S}$ 和重复支付等式 $\overline{T}$. 本文系统与原始系统[1]均支持电子钱包的多次支付. 在支付协议中,为了证明电子钱包尚未发生透支,原始系统使用了 Boudot[22]的准确区间证明技术. 为了提高运算效率,本文采用了 Arfaoui 等人[21]的基于集合关系证明的区间证明协议. 本文的公平交换协议是通过对 Avoine 等人[23]的乐观公平交换协议做出修改而得到的. 两者的主要差别体现在 2 点:

1) 在 Avoine 等人的协议中,用户需要在交换(Exchange)子协议中使用 Stadler[24]的公开可验证的秘密分享方案和基于离散对数的可验证加密方案. 然而,Stadler 的可验证加密方案因为使用了仅满足语义安全性的 ElGamal 加密方案而存在安全隐患[14]. 同时,该方案要求用户执行低效的 cut-and-choose 证明. 为了提高用户端运算效率,本文使用了基于 TPM 的可验证群托管协议[7].

2) 在 Avoine 等人的协议中,商家同样需要在恢复(Recovery)子协议中使用 Stadler 的可验证加密方案. 基于上述原因,本文使用了效率更高且满足选择密文攻击(chosen-ciphertext attack, CCA)安全性的 Camenisch-Shoup 可验证加密方案[14].

2.2 本文系统的参与方

本文系统的执行过程涉及以下参与方,即用户 U、银行 B、商家 M 以及 TTP $P_1,P_2,\cdots,P_n$,其中 U 的角色可以划分为 TPM 芯片与主机 $Host$.

2.3 本文系统的符号描述

$param$ 表示系统参数;ID 表示 B 的身份信息;cnt_{ID} 表示 $Host$ 保存的计数器,用于统计 U 与 B 执行取款协议的次数;$DAASeed$ 表示 TPM 内部的秘密种子,AIK 表示 TPM 的证明身份密钥;$cert(AIK)$ 表示可信证书权威(certificate authority, CA)为 AIK 颁发的证书;J 表示用于统计用户电子钱包使用次数的计数器;$wallet$ 表示 U 的电子钱包;$coin'$ 表示 U 向 M 支付的未经授权的电子货币;$coin$ 表示得到授权的电子货币;end 表示用于对 $coin'$ 进行授权处理的背书元素;cmt 表示关于 end 的承诺;$info$ 表示支付交易标识;$\overline{S}$ 与 $\overline{S}'$ 分别表示 $coin$ 的电子货币序列号及其盲化形式;$\overline{T}$ 与 $\overline{T}'$ 分别表示 $coin$ 的重复支付等式及其盲化形式;RL 表示用于存放被废除的 TPM 私钥的公开列表;(k_0,k_1,k_2) 表示 U 掌握的秘密属性,其中 k_0 表示 TPM 私钥,k_1 和 k_2 分别用于计算 $\overline{S}$ 和 $\overline{T}$.

2.4 本文系统的前提假设

1) 假设所有的参与方都利用安全信道进行通信[23].

2) 假设 U 的 TPM 芯片是由诚实制造厂商根据公开标准生产的,且满足 5 个性质:①TPM 本身具备抗篡改性;②TPM 使用的公钥加密算法满足 CCA 安全性;③TPM 使用的数字签名算法在自适应选择消息攻击下满足不可伪造性;④TPM 利用抗碰撞的单向函数产生消息摘要;⑤可信 CA 已经为 TPM 的密钥 AIK 颁发了数字证书 $cert(AIK)$[7].

2.5 本文系统的详细描述

2.5.1 系统建立(Setup)

可信算法执行以下步骤:

Step1. 定义双线性群对 (G_1,G_2),使得存在类型为 3 的映射 $\hat{e}:G_1\times G_2\to G_T$,且群 G_1,G_2,G_T 的阶数均为素数 p;

Step2. 选取 $\overline{S}_0,g,\bar{g},g_1,g_2\in_R G_1$,选取 $Q,Q'\in_R G_2$;

Step3. 定义抗碰撞的散列函数 $H_1:\{0,1\}^*\to\mathbb{Z}_p$,$H_2:\{0,1\}^*\to G_1$,$H_3:\{0,1\}^*\to\mathbb{Z}_p$,$H_4:\{0,1\}^*\to\{0,1\}^{\omega'}$;

Step4. 设置系统参数 $param=(p,G_1,G_2,G_T,\hat{e};\overline{S}_0,g,\bar{g},g_1,g_2;Q,Q';H_1,H_2,H_3,H_4)$.

2.5.2 银行系统建立(Setup for Bank)

B 执行以下步骤:

Step1. 选取 $a,a_0,a_1,a_2,z\in_R\mathbb{Z}_p^*$,设置 $A=Q^a,Z=Q^z,A_i=Q^{a_i},i=0,1,2$;

Step2. 选取 $\xi\in_R\mathbb{Z}_p^*$,设置 $Y=Q'^{\xi}$;

Step3. 定义集合 $\Sigma=\{o_1,o_2,\cdots,o_{\mathfrak{M}}\}=\{g^{\frac{1}{\xi+1}},g^{\frac{1}{\xi+2}},\cdots,g^{\frac{1}{\xi+\mathfrak{M}}}\}$;

Step4. 初始化公开列表 $RL=\{\}$;

Step5. 定义 $sk_B=(a,a_0,a_1,a_2,z,\xi)$,$pk_B=(A,A_0,A_1,A_2,Z,Y)$.

2.5.3 用户系统建立(Setup for Bank)

$U(TPM+Host)$ 执行以下步骤:

Step1. 选取大素数 $\tilde{p}',\tilde{q}'$,满足 $\tilde{p}'\neq\tilde{q}'$,设置 $\tilde{p}=2\tilde{p}'+1,\tilde{q}=2\tilde{q}'+1,\tilde{n}=\tilde{p}\tilde{q}$;

Step2. 选取 $\tilde{g}',\hat{g},\hat{h}\in_R\mathbb{Z}_{\tilde{n}^2}^*$,选取 $x_1,x_2,x_3\in_R\mathbb{Z}_{\lfloor n^2/4\rfloor}$,并且计算 $\tilde{g}=(\tilde{g}')^{2\tilde{n}}\bmod\tilde{n}^2$. 对于 $i=1,2,3$,计算 $\tilde{y}_i=\tilde{g}^{x_i}\bmod\tilde{n}^2$;

Step3. 选取带密钥的散列函数 Hash 以及密钥 hk.

Step4. 设置 $pk_U=(\mathrm{Hash},hk,\tilde{n},\tilde{g},\hat{g},\hat{h},\tilde{y}_1,\tilde{y}_2,\tilde{y}_3)$,$sk_U=(hk,\tilde{n},x_1,x_2,x_3)$.

2.5.4 取款(Withdraw)

为了提取可支付的电子钱包,$U(TPM+Host)$ 与 B 执行以下步骤:

Step1. $Host$ 向 B 发送取款请求 req_{Withdraw},B 选取 $n_B\in_R\mathbb{Z}_p$,$K\in_R G_1$,计算 $S=K^a,S_i=K^{a_i},i=0,1,2$,并且向 $Host$ 发送 (ID,n_B,K,S,S_0,S_1,S_2).

Step2. $Host$ 执行 $cnt_{ID}\leftarrow cnt_{ID}+1$,并且与 TPM 联合产生知识签名 $\pi_1=SPK\{(\kappa',k_0,k_1,k_2):\overline{U}=\overline{S}_0^{k_0}\wedge R=S^{\kappa'}\prod_{i=0}^{2}S_i^{k_i}\}(n_B)$.

Step3. $Host$ 向 B 发送 $\overline{U},R,\pi_1$.

Step4. B 验证 π_1 是否有效,若是,则选取 $\kappa''\in_R\mathbb{Z}_p$,计算 $T=(KS^{\kappa''}R)^z$,并且向 $Host$ 返回 (T,κ'').

Step5. $Host$ 计算 $C=KS^{\kappa''}R=KS^{\kappa}\prod_{i=0}^{2}S_i^{k_i}$,并且验证是否同时满足:1)$\hat{e}(T,Q)=\hat{e}(C,Z)$;2)$\hat{e}(S,Q)=\hat{e}(K,A)$;3)$\hat{e}(S_i,Q)=\hat{e}(K,A_i)$,$i=0,1,2$. 若是,则保存 $wallet=(k_1,k_2,\kappa=\kappa'+\kappa'',K,S,S_0,S_1,S_2,T,J=0)$.

2.5.5　现金分割(Splitcoin)

假设 $U(TPM+Host)$希望向 M 支付一个电子货币,它需要与 M 共同执行以下步骤:

Step1. $Host$ 向 M 发送支付请求 req_{Spend}.

Step2. M 选取 $n_M \in_R \mathbb{Z}_p^*$,并且返回 n_M.

Step3. $Host$ 设置 $\overline{R}=H_1(pk_M \parallel info \parallel n_M)$,计算 $b=H_2(\overline{R})$. $Host$ 选取 $end,\alpha,\beta \in_R \mathbb{Z}_p^*$,计算 $cmt=\bar{g}^{end}$, $\overline{S}'=g_1^{\frac{1}{J+k_1}}g^{end}$, $\overline{T}'=\overline{U}g_2^{\frac{\overline{R}}{J+k_2}}g^{end}$, $\hat{K}=K^{\alpha}$, $\hat{S}=S^{\alpha}$, $\{\hat{S}_i=S_i^{\alpha}\}_{i=0}^{2}$, $\widetilde{C}=C^{-\alpha/\beta}$, $\widetilde{T}=T^{-\alpha/\beta}$. $Host$ 选取 $l,\nu \in_R \mathbb{Z}_p$,计算 $B_1=o_J^l$, $D_1=B_1^{-J}g^l$, $D_2=g_1^J g_2^{\nu}$.

Step4. $Host$ 与 TPM 联合产生知识签名

$$\pi_2 = SPK\{(\beta,\kappa,k_0,k_1,k_2,end,J):\hat{K}^{-1}=\widetilde{C}^{\beta}\hat{S}^{\kappa}\prod_{i=0}^{2}\hat{S}_i^{k_i} \wedge cmt=\bar{g}^{end} \wedge t=b^{k_0} \wedge \overline{S}'=g_1^{\frac{1}{J+k_1}}g^{end} \wedge \overline{T}'=\overline{U}g_2^{\frac{\overline{R}}{J+k_2}}g^{end} \wedge J\in\Phi\}(n_M).$$

$$\Leftrightarrow$$

$$\pi_2 = SPK\{(\beta,\kappa,k_0,k_1,k_2,end,end\times(J+k_1),k_0\times(J+k_2),end\times(J+k_2),J,l,\nu):\hat{K}^{-1}=\widetilde{C}^{\beta}\hat{S}^{\kappa}\prod_{i=0}^{2}\hat{S}_i^{k_i} \wedge cmt=\bar{g}^{end} \wedge t=b^{k_0} \wedge g_1=\overline{S}'^{J}\overline{S}'^{k_1}g^{-end\times(J+k_1)} \wedge 1=cmt^J cmt^{k_1}\bar{g}^{-end\times(J+k_1)} \wedge g_2^{\overline{R}}=\overline{T}'^{J}\overline{T}'^{k_2}\overline{S}_0^{-k_0\times(J+k_2)}g^{-end\times(J+k_2)} \wedge 1=cmt^J cmt^{k_2}\bar{g}^{-end\times(J+k_2)} \wedge 1=t^J t^{k_2}b^{-k_0\times(J+k_2)} \wedge D_1=B_1^{-J}g^l \wedge D_2=g_1^J g_2^{\nu}\}(n_M).$$

π_2 证明了以下事实:①U 掌握由 B 颁发的有效电子钱包;②cmt 为关于背书元素 end 的承诺;③t 为关于 TPM 私钥的承诺;④$\overline{S}'$ 与 $\overline{T}'$ 分别为电子货币序列号和重复支付等式的隐藏形式;⑤用户的电子钱包尚未发生透支.

Step5. $Host$ 保存 $end,cmt,coin'=(\overline{S}',\overline{T}',\overline{R},\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\widetilde{C},\widetilde{T},b,t,B_1,D_1,D_2,\pi_2)$. 同时,$Host$ 使 $wallet$ 中的计数器 J 取值增 1.

2.5.6　支付(Spend)

在该协议中,$U(TPM+Host)$向 M 支付未经授权的电子货币,双方的具体步骤如下:

Step1. $Host$ 向 M 发送 $cmt,coin'$.

Step2. M 将 $coin'$ 分离为 $coin'=(\overline{S}',\overline{T}',\overline{R},\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\widetilde{C},\widetilde{T},b,t,B_1,D_1,D_2,\pi_2)$的形式,并且检查 π_2 是否为有效. 此外,M 验证是否同时满足:①$\hat{e}(\widetilde{C},Z)=\hat{e}(\widetilde{T},Q)$;②$\hat{e}(\hat{K},A)=\hat{e}(\hat{S},Q)$;③$\hat{e}(\hat{K},A_i)=\hat{e}(\hat{S}_i,Q),i=0,1,2$;④$\hat{e}(D_1,Q')=\hat{e}(B_1,Y)$.

Step3. 对于 $i=1,2,\cdots,|RL|$,M 验证是否满足 $t\neq b^{k_{i,0}}$,其中 $k_{i,0}\in RL$.

Step4. 若上述检查通过,则 M 保存 $cmt,coin'$.

2.5.7　重构(Reconstruction)

给定 end,M 计算 $\overline{S}=\overline{S}'g^{-end}$, $\overline{T}=\overline{T}'g^{-end}$,从而得到已授权的电子货币 $coin=(\overline{S},\overline{T},\overline{R},\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\widetilde{C},\widetilde{T},b,t,cmt,end,B_1,D_1,D_2,\pi_2)$.

2.5.8　存款(Deposit)

假设 M 获得了 U 支付的 $coin$. 为了实现存款,M 与 B 执行如下步骤:

Step1. M 向 B 发送 $coin$.

Step2. B 将 $coin$ 分离为 $coin=(\overline{S},\overline{T},\overline{R},\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\widetilde{C},\widetilde{T},b,t,cmt,end,B_1,D_1,D_2,\pi_2)$的形式. B 执行以下操作:

Step2.1. 检查 $\overline{S}$ 与 $\overline{R}$ 是否未曾使用过;

Step2.2. 计算 $\overline{S}'=\overline{S}g^{end}$, $\overline{T}'=\overline{T}g^{end}$,并且执行支付协议中的验证操作.

Step3. 若验证通过,则 B 将 $coin$ 作为已支付的电子货币进行保存,并且将相应金额存入 M 的账户,同时,B 向 M 返回确认信息;否则,B 向 M 返回失败信息.

2.5.9　身份追踪(Identify)

假设 B 发现具有相同序列号的电子货币 $coin_1=(\overline{S},\overline{T},\overline{R},\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\widetilde{C},\widetilde{T},b,t,cmt,end,B_1,D_1,D_2,\pi_2)$, $coin_2=(\overline{S},\overline{T}',\overline{R}',\hat{K}',\hat{S}',\hat{S}_0',\hat{S}_1',\hat{S}_2',\widetilde{C}',\widetilde{T}',b',t',cmt',end',B_1',D_1',D_2',\pi_2')$,则判定 U 有重复支付行为. 此时,B 计算 $\overline{U}=(\overline{T}'^{\overline{R}}/\overline{T}^{\overline{R}'})^{(\overline{R}-\overline{R}')^{-1}}$,从而揭示该用户的身份.

2.6　公平交换协议的详细描述

2.6.1　交换(Exchange)

假设 $U(TPM+Host)$希望向 M 购买数字商品 m,且 m 可以编码为 $\mathbb{Z}_\rho$ 上的整数. 为此,M 已经向 U 提供了关于 m 的描述信息 $descr(m)=\mathfrak{g}^m$,其中 $\mathfrak{g}$ 为 ρ 阶循环群 Υ 的生成元,满足 $\rho<\tilde{n}\times 2^{-\omega'-\omega''-1}$[25] 且参数 ω' 与 ω'' 分别用于定义底层 Camenisch-Shoup 可验证加密方案的知识证明协议的挑战空间以及控制该协议的零知识性. 而且,U 已经利用支付协议向 M 提供了 $cmt,coin'$. 对于 $i=1,2,\cdots,n$,假设 TTP 中 P_i 已经公开了自己在某 CCA 安全的加密方案 (Enc,Dec)下的公钥 pk_i. 在当前协议中,U 与 M 将实现关于 end 和 m 的公平交换. 此外,符号 $Time$ 表示交换交易的截止时间,且 $Time_{\max}$ 表示 M 在恢复

子协议中发送给任何参与方的消息总是能在 $Time_{\max}$ 时间内送达.双方具体交换过程如下:

Step1. *Host* 设置条件 $Condition_U=(pk_M, descr(m), Time, pk_U)$,并且向 *TPM* 提供 *end*. *TPM* 执行底层可验证群托管协议[7]中的 SigSender 算法,并且返回 $(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK))$. *TPM* 在 SigSender 算法中的具体操作为:

Step1.1. 产生随机数 r,并将 r 保存在其内部受保护区域.

Step1.2. 计算 $d=(end+r) \bmod p$,并且计算 $\hat{t}=\bar{g}^r$.

Step1.3. 产生 r 在 (k,n)-Shamir 门限方案下的秘密份额 $r_1,r_2,\cdots,r_n$,即选取 $s_1,s_2,\cdots,s_{k-1}\in_R\mathbb{Z}_p^*$,定义 $F(z)=r+\sum_{t=1}^{k-1}s_t z^t$. 对于 $i=1,2,\cdots,n$,计算 $r_i=F(i)$.

Step1.4. 对于 $i=1,2,\cdots,n$,执行以下步骤:

Step1.4.1. 设置 $hCond=H_3(Condition_U)$;

Step1.4.2. 利用 TTP 中 P_i 的公钥 pk_i 产生关于 $hCond$ 和 r_i 的密文 $Enc_{pk_i}(hCond \parallel r_i)$,并且设置 $c_i=(Enc_{pk_i}(hCond \parallel r_i), pk_i, k, n, \hat{t}, hCond)$;

Step1.4.3. 利用 *AIK* 中的私钥 SK_A 产生关于 c_i 的签名 $Sign_{SK_A}(c_i)$,并且设置 $C_i=(c_i, Sign_{SK_A}(c_i))$.

Step1.5. 输出 $(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK))$.

Step2. *Host* 向 M 发送 $(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK)),Condition_U$. M 利用底层可验证群托管协议中的 SigVerifier 算法验证该元组是否有效. M 在 SigVerifier 算法中的具体操作为

Step2.1. 验证由隐私 CA 颁发的证书 $cert(AIK)$ 是否有效;

Step2.2. 对于 $i=1,2,\cdots,n$,将 C_i 分离为 $C_i=(c_i, Sign_{SK_A}(c_i))$ 的形式,并且验证签名 $Sign_{SK_A}(c_i)$ 是否有效;

Step2.3. 计算 $hCond=H_3(Condition_U)$,并且验证 $C_1,C_2,\cdots,C_n$ 中是否包含相同的元组 $(k,n,\hat{t},hCond)$;

Step2.4. 验证是否满足 $\bar{g}^d=cmt\times\hat{t}$;

Step2.5. 若上述验证都通过,则判定元组 $(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK))$ 为有效,并且向 *Host* 发送 m;否则,M 一直等待直至 *Time* 时刻,从而终止当前交换过程.

Step3. 在接收到 m 之后,*Host* 验证是否满足 $\mathfrak{g}^m=descr(m)$,若是,则向 M 发送 *end*;否则,*Host* 一直等待直至 *Time* 时刻,从而终止当前交换过程.

Step4. 在接收到 *end* 之后,M 验证是否满足 $cmt=\bar{g}^{end}$. 若是,则调用 2.5.7 节的重构算法. 否则,即不满足 $cmt=g^{end}$ 或者 M 并未接收到 *end*,则执行以下的恢复(Recovery)子协议.

2.6.2 恢复(Recovery)

在 $Time-Time_{\max}$ 时刻之前,M 与 $P_1,P_2,\cdots,P_n$ 共同执行以下步骤:

Step1. M 利用 U 的加密公钥 pk_U 产生关于 m 的 Camenisch-Shoup 方案密文 $(\tilde{u},\tilde{e},\tilde{v})=(\tilde{g}^{r'} \bmod \tilde{n}^2, \tilde{y}_1^{r'}(1+\tilde{n})^m \bmod \tilde{n}^2, \mathrm{abs}((\tilde{y}_2\tilde{y}_3^{\mathrm{Hash}_{hk}(u,e)})^{r'} \bmod \tilde{n}^2))$,其中 $r'\in_R\mathbb{Z}_{\lfloor\tilde{n}/4\rfloor}$. 函数 $\mathrm{abs}(x)$ 的定义是,若 $x\leqslant\tilde{n}^2/2$,则 $\mathrm{abs}(x)=x$;否则,$\mathrm{abs}(x)=\tilde{n}^2-x$. M 选取 $s\in_R\mathbb{Z}_{\lfloor\tilde{n}/4\rfloor}$,计算 $\hat{t}=\hat{g}^m\hat{h}^s \bmod \tilde{n}^2$,并且产生知识签名 $\pi_3=SPK\{(r',m,s):\tilde{u}^2=\tilde{g}^{2r'} \bmod \tilde{n}^2 \wedge \tilde{e}^2=\tilde{y}_1^{2r'}(1+\tilde{n})^{2m} \bmod \tilde{n}^2 \wedge \tilde{v}^2=\mathrm{abs}(\tilde{y}_2\tilde{y}_3^{\mathrm{Hash}_{hk}(u,e)})^{2r'} \bmod \tilde{n}^2 \wedge \mathfrak{g}^m=(\mathfrak{g})^m \wedge \hat{t}=\hat{g}^m\hat{h}^s \bmod \tilde{n}^2 \wedge (-\tilde{n}/2<m<\tilde{n}/2)\}$.

Step2. 对于 $1\leqslant i\leqslant n$,M 向 P_i 发送 $((\tilde{u},\tilde{e},\tilde{v}),\pi_3,(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK)),Condition_U)$.

Step3. 在接收到 $((\tilde{u},\tilde{e},\tilde{v}),\pi_3,(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK)),Condition_U)$ 之后,P_i 执行以下检查:1)π_3 是否为有效;2)SigVerifier$(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK),Condition_U)$ 是否为有效;3)当前时间是否未超过 $Condition_U$ 中包含的过期时间 *Time*. 若上述检查通过,则 P_i 向 M 发送从 C_i 中恢复得到的元素 r_i,同时向 *Host* 发送 $(\tilde{u},\tilde{e},\tilde{v})$.

Step4. 当收集到来自 k 个 TTP 的解密结果 $r_{i_1},r_{i_2},\cdots,r_{i_k}$ 后,M 计算 $\mathrm{Recover}(r_{i_1},r_{i_2},\cdots,r_{i_k})=\sum_{i_t\in Qual}\lambda_{0,i_t}^{Qual}r_{i_t}=r$. 其中,$Qual=\{i_1,i_2,\cdots,i_k\}$ 且 λ_{0,i_t} 为 Shamir 门限方案中的拉格朗日符号. 最后,M 恢复得到 $end=(d-r) \bmod p$.

3 知识签名的具体实现过程

本节介绍知识签名 π_1,π_2 的实现过程. 在具体执行过程中,我们采用文献[6]中的方式实现了 *TPM* 与 *Host* 间的运算任务分配. 此外,π_3 的实现可以直接根据文献[14,25]中的技术得出.

3.1　π_1 的具体实现过程

1) π_1 的产生过程如下:

Step1. *Host* 选取 κ', k_1, $k_2 \in_R \mathbb{Z}_p^*$, 并且向 *TPM* 发送($ID, cnt_{ID}, n_B, S, S_0, S_1, S_2, \kappa', k_1, k_2$).

Step2. *TPM* 计算私钥 $k_0 = H_1(DAASeed \| cnt_{ID} \| ID)$, 计算 $\overline{U} = \overline{S}_0^{k_0}$, $R' = S_0^{k_0}$. *TPM* 选取 $r_{k_0} \in_R \mathbb{Z}_p$, 计算 $R_1 = \overline{S}_0^{r_{k_0}}$, $R_2' = S_0^{r_{k_0}}$. 然后, *TPM* 向 *Host* 发送($\overline{U}, R', R_2'$).

Step3. *Host* 计算 $R = R' S^{\kappa'} S_1^{k_1} S_2^{k_2} = S^{\kappa'} \prod_{i=0}^{2} S_i^{k_i}$. *Host* 选取 $r_{\kappa'}, r_{k_1}, r_{k_2} \in_R \mathbb{Z}_p$, 计算 $R_2 = S^{r_{\kappa'}} R_2' S_1^{r_{k_1}} S_2^{r_{k_2}}$. 然后, *Host* 向 *TPM* 发送($R, R_2, r_{\kappa'}, r_{k_1}, r_{k_2}$).

Step4. *TPM* 设置 $c = H_1(S \| S_0 \| S_1 \| S_2 \| \overline{U} \| R \| R_1 \| R_2 \| n_B)$, 并且在 $\mathbb{Z}_p$ 上计算 $s_{k_0} = r_{k_0} + ck_0$, $s_{\kappa'} = r_{\kappa'} + c\kappa'$, $s_{k_1} = r_{k_1} + ck_1$, $s_{k_2} = r_{k_2} + ck_2$. 然后, *TPM* 向 *Host* 发送($c, s_{\kappa'}, s_{k_0}, s_{k_1}, s_{k_2}$).

Step5. *Host* 输出 $\pi_1 = (c, s_{\kappa'}, s_{k_0}, s_{k_1}, s_{k_2})$.

2) π_1 的验证过程如下:

Step1. 将 π_1 分离为 $\pi_1 = (c, s_{\kappa'}, s_{k_0}, s_{k_1}, s_{k_2})$ 的形式;

Step2. 计算 $\hat{R}_1 = \overline{S}_0^{s_{k_0}} \overline{U}^{-c}$, $\hat{R}_2 = S^{s_{\kappa'}} \prod_{i=0}^{2} S_i^{s_{k_i}} R^{-c}$;

Step3. 验证是否满足 $c = H_1(S \| S_0 \| S_1 \| S_2 \| \overline{U} \| R \| \hat{R}_1 \| \hat{R}_2 \| n_B)$.

3.2　π_2 的具体实现过程

1) π_2 的产生过程如下:

Step1. *Host* 向 *TPM* 发送($ID, cnt_{ID}, \hat{K}, \widetilde{C}, \widetilde{T}, \hat{S}, \hat{S}_0, \hat{S}_1, \hat{S}_2, cmt, b, \overline{S}', \overline{T}', B_1, D_1, D_2, n_M, \beta, \kappa, k_1, k_2, end, J, end \times (J+k_1), end \times (J+k_2), l, \nu$);

Step2. *TPM* 计算私钥 $k_0 = H_1(DAASeed \| cnt_{ID} \| ID)$, 计算 $t = b^{k_0}$. *TPM* 选取 $r_{k_0}, r_{k_0 \times (J+k_2)} \in_R \mathbb{Z}_p$, 计算 $R_1' = \hat{S}_0^{r_{k_0}}$, $R_3 = b^{r_{k_0}}$, $R_6' = \overline{S}_0^{-r_{k_0 \times (J+k_2)}}$, $R_8' = b^{-r_{k_0 \times (J+k_2)}}$. 然后, *TPM* 向 *Host* 发送($t, R_1', R_6', R_8'$);

Step3. *Host* 选取 r_β, r_κ, r_{k_1}, r_{k_2}, r_{end}, r_J, $r_{end \times (J+k_1)}$, $r_{end \times (J+k_2)}$, $r_l, r_\nu \in_R \mathbb{Z}_p$, 计算:

$$R_1 = R_1' \widetilde{C}^{r_\beta} \hat{S}^{r_\kappa} \hat{S}_1^{r_{k_1}} \hat{S}_2^{r_{k_2}} = \widetilde{C}^{r_\beta} \hat{S}^{r_\kappa} \prod_{i=0}^{2} \hat{S}_i^{r_{k_i}},$$

$$R_2 = \bar{g}^{r_{end}},$$

$$R_4 = \overline{S}'^{r_J} \overline{S}'^{r_{k_1}} \bar{g}^{-r_{end \times (J+k_1)}},$$

$$R_5 = cmt^{r_J} cmt^{r_{k_1}} \bar{g}^{-r_{end \times (J+k_1)}},$$

$$R_6 = R_6' \overline{T}'^{r_J} \overline{T}'^{r_{k_2}} \bar{g}^{-r_{end \times (J+k_2)}},$$

$$R_7 = cmt^{r_J} cmt^{r_{k_2}} \bar{g}^{-r_{end \times (J+k_2)}},$$

$$R_8 = t^{r_J} t^{r_{k_2}} R_8',$$

$$R_9 = B_1^{-r_J} g^{r_l},$$

$$R_{10} = g_1^{r_J} g_2^{r_\nu}.$$

然后, *Host* 向 *TPM* 发送($R_1, R_2, R_4, R_5, \cdots, R_{10}, r_\beta, r_\kappa, r_{k_1}, r_{k_2}, r_{end}, r_J, r_{end \times (J+k_1)}, r_{end \times (J+k_2)}, r_l, r_\nu$);

Step4. *TPM* 设置 $c = H_1(\hat{K} \| \widetilde{C} \| \widetilde{T} \| \hat{S} \| \hat{S}_0 \| \hat{S}_1 \| \hat{S}_2 \| cmt \| b \| t \| \overline{S}' \| \overline{T}' \| B_1 \| D_1 \| D_2 \| R_1 \| R_2 \| \cdots \| R_{10} \| n_M)$ 并且在 $\mathbb{Z}_p$ 上计算 $s_\beta = r_\beta + c\beta$, $s_\kappa = r_\kappa + c\kappa$, $s_{k_0} = r_{k_0} + ck_0$, $s_{k_1} = r_{k_1} + ck_1$, $s_{k_2} = r_{k_2} + ck_2$, $s_{end} = r_{end} + c \times end$, $s_J = r_J + cJ$, $s_{end \times (J+k_1)} = r_{end \times (J+k_1)} + c \times end \times (J+k_1)$, $s_{k_0 \times (J+k_2)} = r_{k_0 \times (J+k_2)} + c \times k_0 \times (J+k_2)$, $s_{end \times (J+k_2)} = r_{end \times (J+k_2)} + c \times end \times (J+k_2)$, $s_l = r_l + cl$, $s_\nu = r_\nu + c\nu$. 然后, *TPM* 向 *Host* 发送($c, s_\beta, s_\kappa, s_{k_0}, s_{k_1}, s_{k_2}, s_{end}, s_J, s_{end \times (J+k_1)}, s_{k_0 \times (J+k_2)}, s_{end \times (J+k_2)}, s_l, s_\nu$);

Step5. *Host* 输出 $\pi_2 = (c, s_\beta, s_\kappa, s_{k_0}, s_{k_1}, s_{k_2}, s_{end}, s_J, s_{end \times (J+k_1)}, s_{k_0 \times (J+k_2)}, s_{end \times (J+k_2)}, s_l, s_\nu)$.

2) π_2 的验证过程如下:

Step1. 将 π_2 分离为 $\pi_2 = (c, s_\beta, s_\kappa, s_{k_0}, s_{k_1}, s_{k_2}, s_{end}, s_J, s_{end \times (J+k_1)}, s_{k_0 \times (J+k_2)}, s_{end \times (J+k_2)}, s_l, s_\nu)$ 的形式;

Step2. 计算:

$$\hat{R}_1 = \widetilde{C}^{s_\beta} \hat{S}^{s_\kappa} \prod_{i=0}^{2} \hat{S}_i^{s_{k_i}} \hat{K}^c,$$

$$\hat{R}_2 = \bar{g}^{s_{end}} cmt^{-c},$$

$$\hat{R}_3 = b^{s_{k_0}} t^{-c},$$

$$\hat{R}_4 = \overline{S}'^{s_J} \overline{S}'^{s_{k_1}} \bar{g}^{-s_{end \times (J+k_1)}} g_1^{-c},$$

$$\hat{R}_5 = cmt^{s_J} cmt^{s_{k_1}} \bar{g}^{-s_{end \times (J+k_1)}},$$

$$\hat{R}_6 = \overline{T}'^{s_J} \overline{T}'^{s_{k_2}} \overline{S}_0^{-s_{k_0 \times (J+k_2)}} g^{-s_{end \times (J+k_2)}} (g_2^R)^{-c},$$

$$\hat{R}_7 = cmt^{s_J} cmt^{s_{k_2}} \bar{g}^{-s_{end \times (J+k_2)}},$$

$$\hat{R}_8 = t^{s_J} t^{s_{k_2}} b^{-s_{k_0 \times (J+k_2)}},$$

$$\hat{R}_9 = B_1^{-s_J} g^{s_l} D_1^{-c},$$

$$\hat{R}_{10} = g_1^{s_J} g_2^{s_\nu} D_2^{-c};$$

Step3. 验证是否满足 $c = H_1(\hat{K} \| \widetilde{C} \| \widetilde{T} \| \hat{S} \| \hat{S}_0 \| \hat{S}_1 \| \hat{S}_2 \| cmt \| b \| t \| \overline{S}' \| \overline{T}' \| B_1 \| D_1 \| D_2 \| \hat{R}_1 \| \hat{R}_2 \| \cdots \| \hat{R}_{10} \| n_M)$.

3.3　π_2 实现过程的效率优化

为了进一步提高 π_2 实现过程的效率, 可以采用文献[6]的方式将 π_2 的实现过程划分为预计算阶段和在线计算阶段. 优化后的 π_2 实现过程如下:

1) 在预计算阶段

Host 产生并且向 *TPM* 发送($ID, cnt_{ID}, \hat{K}, \widetilde{C},$

$\widetilde{T},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,cmt,\overline{S}',B_1,D_1,D_2,\beta,\kappa,k_1,k_2,end,J,end\times(J+k_1),end\times(J+k_2),l,\nu)$. *TPM* 计算并且向 *Host* 返回 R_1',R_6'. *Host* 进一步计算 $R_1,R_2,R_4,R_5,R_7,R_9,R_{10}$.

2) 在线计算阶段

Step1. 在接收到 M 提供的挑战 n_M 之后,*Host* 计算 $b,\overline{T}'$,并且向 *TPM* 发送$(b,\overline{T}',n_M)$;

Step2. *TPM* 计算私钥 k_0,并且计算 t,R_3,R_8'. 然后,*TPM* 向 *Host* 发送(t,R_8');

Step3. *Host* 计算 R_6,R_8,并且向 *TPM* 发送$(R_1,R_2,R_4,R_5,\cdots,R_{10},r_\beta,r_\kappa,r_{k_1},r_{k_2},r_{end},r_J,r_{end\times(J+k_1)},r_{end\times(J+k_2)},r_l,r_\nu)$;

Step4. *TPM* 计算并且向 *Host* 发送$(c,s_\beta,s_\kappa,s_{k_0},s_{k_1},s_{k_2},s_{end},s_J,s_{end\times(J+k_1)},s_{k_0\times(J+k_2)},s_{end\times(J+k_2)},s_l,s_\nu)$;

Step5. *Host* 输出$\pi_2=(c,s_\beta,s_\kappa,s_{k_0},s_{k_1},s_{k_2},s_{end},s_J,s_{end\times(J+k_1)},s_{k_0\times(J+k_2)},s_{end\times(J+k_2)},s_l,s_\nu)$.

引理 1. 在随机预言模型下,可以在多项式时间内从知识签名 π_2 中提取出秘密知识,使得以下等式成立,即

$$\widetilde{C}^{-\beta^*}=\hat{K}\hat{S}^{\kappa^*}\prod_{i=0}^{2}\hat{S}_i^{k_i^*},$$

$$cmt=\bar{g}^{end^*},$$

$$t=b^{k_0^*},$$

$$\overline{S}'=g_1^{\frac{1}{J^*+k_1^*}}g^{end^*},$$

$$\overline{T}'=\overline{S}_0^{k_0^*}g_2^{\frac{\bar{R}}{J^*+k_2^*}}g^{end^*},$$

$$D_1=B_1^{-J^*}g^{l^*},$$

$$D_2=g_1^{J^*}g_2^{\nu^*},$$

$$o_{J^*}=g^{\frac{1}{\xi+J^*}}.$$

同时,$(\kappa^*,\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\hat{T})$构成关于向量$(k_0^*,k_1^*,k_2^*)$的底层方案签名.

证明. 假设 U 在现金分割协议中输出 cmt,$coin'$,且 $coin'$中包含有效的签名$\pi_2=(c,s_\beta,s_\kappa,s_{k_0},s_{k_1},s_{k_2},s_{end},s_J,s_{end\times(J+k_1)},s_{k_0\times(J+k_2)},s_{end\times(J+k_2)},s_l,s_\nu)$. 通过对 U 执行重绕,可以获得另一个签名$\bar{\pi}_2=(\bar{c},\bar{s}_\beta,\bar{s}_\kappa,\bar{s}_{k_0},\bar{s}_{k_1},\bar{s}_{k_2},\bar{s}_{end},\bar{s}_J,\bar{s}_{end\times(J+k_1)},\bar{s}_{k_0\times(J+k_2)},\bar{s}_{end\times(J+k_2)},\bar{s}_l,\bar{s}_\nu)$. 根据 $\pi_2,\bar{\pi}_2$ 的验证过程,可以得出等式

$$\hat{R}_1=\widetilde{C}^{s_\beta}\hat{S}^{s_\kappa}\prod_{i=0}^{2}\hat{S}_i^{s_{k_i}}\hat{K}^c=\widetilde{C}^{s_\beta}\hat{S}^{s_\kappa}\prod_{i=0}^{2}\hat{S}_i^{s_{k_i}}\hat{K}^{\bar{c}},\quad(1)$$

$$\hat{R}_2=\bar{g}^{s_{end}}cmt^{-c}=\bar{g}^{s_{end}}cmt^{-\bar{c}},\quad(2)$$

$$\hat{R}_3=b^{s_{k_0}}t^{-c}=b^{s_{k_0}}t^{-\bar{c}},\quad(3)$$

$$\hat{R}_4=\overline{S}'^{s_J}\overline{S}'^{s_{k_1}}\bar{g}^{-s_{end\times(J+k_1)}}g_1^{-c}=\overline{S}'^{s_J}\overline{S}'^{s_{k_1}}\bar{g}^{-s_{end\times(J+k_1)}}g_1^{-\bar{c}},\quad(4)$$

$$\hat{R}_5=cmt^{s_J}cmt^{s_{k_1}}\bar{g}^{-s_{end\times(J+k_1)}}=cmt^{s_J}cmt^{s_{k_1}}\bar{g}^{-s_{end\times(J+k_1)}},\quad(5)$$

$$\hat{R}_6=\overline{S}_0^{-s_{k_0\times(J+k_2)}}\overline{T}'^{s_J}\overline{T}'^{s_{k_2}}g^{-s_{end\times(J+k_2)}}(g_2^{\bar{R}})^{-c}=\overline{S}_0^{-s_{k_0\times(J+k_2)}}\overline{T}'^{s_J}\overline{T}'^{s_{k_2}}g^{-s_{end\times(J+k_2)}}(g_2^{\bar{R}})^{-\bar{c}},\quad(6)$$

$$\hat{R}_7=cmt^{s_J}cmt^{s_{k_2}}\bar{g}^{-s_{end\times(J+k_2)}}=cmt^{s_J}cmt^{s_{k_2}}\bar{g}^{-s_{end\times(J+k_2)}},\quad(7)$$

$$\hat{R}_8=t^{s_J}t^{s_{k_2}}b^{-s_{k_0\times(J+k_2)}}=t^{s_J}t^{s_{k_2}}b^{-s_{k_0\times(J+k_2)}},\quad(8)$$

$$\hat{R}_9=B_1^{-s_J}g^{s_l}D_1^{-c}=B_1^{-s_J}g^{s_l}D_1^{-\bar{c}},\quad(9)$$

$$\hat{R}_{10}=g_1^{s_J}g_2^{s_\nu}D_2^{-c}=g_1^{s_J}g_2^{s_\nu}D_2^{-\bar{c}}.\quad(10)$$

设置 $\Delta\beta=s_\beta-\bar{s}_\beta,\Delta\kappa=s_\kappa-\bar{s}_\kappa,\{\Delta k_i=s_{k_i}-\bar{s}_{k_i}\}_{i=0}^{2},\Delta end=s_{end}-\bar{s}_{end},\Delta J=s_J-\bar{s}_J,\Delta l=s_l-\bar{s}_l,\Delta\nu=s_\nu-\bar{s}_\nu,\Delta c=c-\bar{c}$. 可以分别从式(1)~(4)、式(9)、式(10)中提取出秘密知识 $\beta^*=\Delta\beta/\Delta c,\kappa^*=\Delta\kappa/\Delta c,\{k_i^*=\Delta k_i/\Delta c\}_{i=0}^{2},end^*=\Delta end/\Delta c,J^*=\Delta J/\Delta c,l^*=\Delta l/\Delta c,\nu^*=\Delta\nu/\Delta c$,使得以下等式成立:

$$\widetilde{C}^{-\beta^*}=\hat{K}\hat{S}^{\kappa^*}\prod_{i=0}^{2}\hat{S}_i^{k_i^*},\quad(11)$$

$$cmt=\bar{g}^{end^*},\quad(12)$$

$$t=b^{k_0^*},\quad(13)$$

$$\overline{S}'=g_1^{\frac{1}{J^*+k_1^*}}g^{end^*},\quad(14)$$

$$\overline{T}'=\overline{S}_0^{k_0^*}g_2^{\frac{\bar{R}}{J^*+k_2^*}}g^{end^*},\quad(15)$$

$$D_1=B_1^{-J^*}g^{l^*},\quad(16)$$

$$D_2=g_1^{J^*}g_2^{\nu^*}.\quad(17)$$

由于 $coin'$满足验证等式 $\hat{e}(\hat{K},A)=\hat{e}(\hat{S},Q)$,得出:

$$\hat{S}=\hat{K}^a.\quad(18)$$

由于 $coin'$满足验证等式 $\hat{e}(\hat{K},A_i)=\hat{e}(\hat{S}_i,Q)(i=0,1,2)$,得出:

$$\hat{S}_i=\hat{K}^{a_i},i=0,1,2.\quad(19)$$

由于 $coin'$满足验证等式 $\hat{e}(\widetilde{C},Z)=\hat{e}(\widetilde{T},Q)$,得出:

$$\widetilde{T}=\widetilde{C}^z.\quad(20)$$

现在,令 $\hat{T}=\widetilde{T}^{-\beta^*}$,由式(11)、式(20)得出:

$$\hat{T}=(\widetilde{C}^z)^{-\beta^*}=(\widetilde{C}^{-\beta^*})^z=(\hat{K}\hat{S}^{\kappa^*}\prod_{i=0}^{2}\hat{S}_i^{k_i^*})^z.\quad(21)$$

于是,根据式(18)~(21)可知,$(\kappa^*,\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\hat{T})$构成关于属性向量$(k_0^*,k_1^*,k_2^*)$的有效的底层方案签名.

由于 $coin'$ 满足验证等式 $\hat{e}(D_1,Q')=\hat{e}(B_1,Y)$,可以得出:

$$D_1=B_1^{\xi}. \tag{22}$$

根据式(17)、式(22)得出:

$$D_1=B_1^{\xi}=B_1^{-J^*}g^{l^*}\Rightarrow B_1^{\xi+J^*}=g^{l^*}. \tag{23}$$

令 $o_{J^*}=B_1^{\frac{1}{l^*}}$(显然 $l^*\neq 0$,否则得出 $J^*=-\xi$,即表明 U 已经掌握了银行私钥 sk_B 中的元素 ξ),得出:

$$o_{J^*}=g^{\frac{1}{\xi+J^*}}. \tag{24}$$

证毕.

引理 2. 在随机预言模型下,可以在不掌握底层秘密知识的条件下模拟产生 $coin'$ 中的知识签名 π_2,且 $\tilde{\pi}_2$ 与真实签名 π_2 是不可分辨的.

证明. 模拟器以元组 $(\overline{K},\overline{S},\overline{S}_0,\overline{S}_1,\overline{S}_2,\overline{C},\overline{T})\in G_1^7$(即关于随机向量 $(\kappa,k_0,k_1,k_2)\in(\mathbb{Z}_p^*)^4$ 的有效底层属性签名方案签名)为输入,并执行以下的模拟步骤:

Step1. 选取 $J\in_R\{1,2,\cdots,\mathfrak{M}\}$,$l\in_R\mathbb{Z}_p^*$,并且设置 $B_1=o_J^l$,$D_1=B_1^{-J}g^l$;

Step2. 选取 $\alpha,\beta\in_R\mathbb{Z}_p^*$,并且设置 $\hat{K}=\overline{K}^{\alpha}$,$\hat{S}=\overline{S}^{\alpha}$,$\{\hat{S}_i=\overline{S}_i^{\alpha}\}_{i=0}^{2}$,$\widetilde{C}=\overline{C}^{\beta}$,$\widetilde{T}=\overline{T}^{\beta}$;

Step3. 选取 $cmt,b,t,\overline{S}',\overline{T}',D_2\in_R G_1$,选取 c,s_β,s_κ,s_{k_0},s_{k_1},s_{k_2},s_{end},s_J,$s_{end\times(J+k_1)}$,$s_{k_0\times(J+k_2)}$,$s_{end\times(J+k_2)}$,s_l,s_ν,$n_M\in_R\mathbb{Z}_p^*$,并且采用 π_2 验证过程中的方式计算 $\hat{R}_1,\hat{R}_2,\cdots,\hat{R}_{10}$;

Step4. 定义 $c=H_1(\hat{K}\parallel\widetilde{C}\parallel\widetilde{T}\parallel\hat{S}\parallel\hat{S}_0\parallel\hat{S}_1\parallel\hat{S}_2\parallel cmt\parallel b\parallel t\parallel\overline{S}'\parallel\overline{T}'\parallel B_1\parallel D_1\parallel D_2\parallel\hat{R}_1\parallel\hat{R}_2\parallel\cdots\parallel\hat{R}_{10}\parallel n_M)$;

Step5. 输出 $\tilde{\pi}_2=(c,s_\beta,s_\kappa,s_{k_0},s_{k_1},s_{k_2},s_{end},s_J,s_{end\times(J+k_1)},s_{k_0\times(J+k_2)},s_{end\times(J+k_2)},s_l,s_\nu)$.

显然,$\tilde{\pi}_2$ 中各元素与真实签名 π_2 中的各元素符合相同分布. 证毕.

4　本文系统的安全性分析

4.1　新的可授权电子现金系统的安全模型

在本节,我们在文献[5]模型基础上定义了新的可授权电子现金系统安全模型. 在该模型中,符号 $\mathfrak{A}$ 与 $\mathfrak{B}$ 分别代表敌手和归约算法;符号 U,B,M 分别代表诚实的用户、银行与商家;符号 U^*,B^*,M^* 分别代表被攻破的用户、银行与商家;符号 $\overline{U}^*$ 代表被部分攻破的用户(即 *Host* 已被攻破,但 *TPM* 芯片仍然保持诚实). 符号 *att* 表示由用户秘密属性构成的集合. 安全的可授权电子现金系统应当同时满足以下性质:

1) 正确性. 首先,U 通过与 B 执行取款协议而获得电子货币 *coin*. 在交易之前,B 利用现金分割协议将 *coin* 转化为 $(end,cmt,coin')$ 的形式,使得 *end* 与 *cmt* 满足单向群同态关系 $cmt=\phi(end)$. 然后,U 利用支付协议向 M 支付 $(cmt,coin')$. 在完成交易后,M 获得 *end*,满足 $cmt=\phi(end)$. 此时,M 可以利用重构算法将 $coin'$ 转换为 *coin*. 此后,当 M 利用 *coin* 与 B 执行存款协议时,该协议一定会取得成功.

2) 平衡性. 假设 $\mathfrak{A}$ 能攻破 U^*,M^* 以及 $\overline{U}^*$. 对于 $\mathfrak{A}$ 而言,它通过与 $\mathfrak{B}$ 执行存款协议而存入的电子货币总数不会超过它通过与 $\mathfrak{B}$ 执行取款协议而提取的电子货币总数. 在实验执行过程中,$\mathfrak{A}$ 充当 U^*,M^*,$\overline{U}^*$(*Host* 部分)的角色,且 $\mathfrak{B}$ 充当 $B,U,\overline{U}^*$(*TPM* 部分)的角色. 详细步骤如下:

① 初始化. $\mathfrak{B}$ 执行系统建立和银行系统建立算法. 然后,$\mathfrak{B}$ 向 $\mathfrak{A}$ 提供 $params,pk_B$.

② 询问. 在该阶段,$\mathfrak{B}$ 为 $\mathfrak{A}$ 模拟预言机 Hash(•),*Withdraw*(•),*Spend*(•)和 *Corrupt*(•)的执行过程:

Ⅰ Hash 询问. 当接收到 $\mathfrak{A}$ 提供的 (i,M),$\mathfrak{B}$ 在散列函数 $H_i(i=1,2)$ 的值域上选取随机元素,并将该元素返回给 $\mathfrak{A}$.

Ⅱ *Withdraw* 询问. 该询问分 2 种情况:

情况 1. 当接收到 $\mathfrak{A}$ 提供的用户身份 $\overline{U}^*$ 和 *att*,$\mathfrak{B}$ 为该用户选取 *TPM* 私钥 k_0,并且向 $\mathfrak{A}$ 提供关于 k_0 和 *att* 的属性证书.

情况 2. 当接收到 $\mathfrak{A}$ 提供的用户身份 U^*,*TPM* 私钥 k_0 和 *att*,$\mathfrak{B}$ 代表 B 与 $\mathfrak{A}$ 执行取款协议,并且将 *TPM* 私钥 k_0 加入列表 *RL*.

Ⅲ *Spend* 询问. 该询问分 2 种情况:

情况 1. 当接收到 $\mathfrak{A}$ 提供的用户身份 U,$\mathfrak{B}$ 代表 U 与 $\mathfrak{A}$ 执行现金分割和支付协议.

情况 2. 当接收到 $\mathfrak{A}$ 提供的用户身份 $\overline{U}^*$ 以及 $\overline{U}^*$ 的 *Host* 部分提供的数据,$\mathfrak{B}$ 代表 $\overline{U}^*$ 的 *TPM* 为 $\mathfrak{A}$ 提供应答.

Ⅳ *Corrupt* 询问. 当接收到 $\mathfrak{A}$ 提供的用户身份 U 或 $\overline{U}^*$,$\mathfrak{B}$ 返回该用户的 *TPM* 私钥 k_0,*att* 和电子钱包,并将 k_0 加入列表 *RL*.

③ 结束. $\mathfrak{A}$ 输出 $coin^*$,并利用 $coin^*$ 与 $\mathfrak{B}$ 执行存款协议. 最终,$\mathfrak{A}$ 获胜的条件是同时满足:Ⅰ $coin^*$ 能通过存款协议的验证过程;Ⅱ 对于每个用户 U_i^*,其 *TPM* 私钥一定属于列表 *RL*;Ⅲ $coin^*$ 并非在 $\mathfrak{B}$ 的帮助下产生的.

3) 匿名性. 假设敌手 $\mathfrak{A}$ 能攻破 B^*,M^* 和 U^*. 对于 $\mathfrak{A}$ 而言,它无法将 U 的取款、现金分割和支付操作关联起来,也无法将 U 执行的 2 次取款操作关联起来. 在实验执行过程中,$\mathfrak{A}$ 充当 B^*,M^* 和 U^* 的角色,且 $\mathfrak{B}$ 充当 U 的角色. 详细步骤如下:

① 初始化. $\mathfrak{B}$ 执行系统建立和银行系统建立算法. 然后,$\mathfrak{B}$ 向 $\mathfrak{A}$ 提供 $params$,pk_B,sk_B.

② 询问阶段 1. 在该阶段,$\mathfrak{B}$ 为 $\mathfrak{A}$ 模拟预言机 Hash(·),$Withdraw$(·),$Spend$(·)和 $Corrupt$(·)的执行过程:

Ⅰ Hash 询问. 模拟过程与平衡性实验相同.

Ⅱ $Withdraw$ 询问. 当接收到 $\mathfrak{A}$ 提供的用户身份 U 和 att,$\mathfrak{B}$ 代表 U 与 $\mathfrak{A}$ 执行取款协议.

Ⅲ $Spend$ 询问. 当接收到 $\mathfrak{A}$ 提供的用户身份 U,$\mathfrak{B}$ 代表 U 与 $\mathfrak{A}$ 执行现金分割和支付协议.

Ⅳ $Corrupt$ 询问. 当接收到 $\mathfrak{A}$ 提供的用户身份 U,$\mathfrak{B}$ 返回该用户的 TPM 私钥 k_0,att 和电子钱包,并将 k_0 加入列表 RL.

③ 挑战. 当 $\mathfrak{A}$ 输出用户下标 j_0^*,$j_1^* \in \{1,2,\cdots,N\}$ 以及 $att_{j_0^*}$,$att_{j_1^*}$,其中 N 表示实验中的最大用户数量. $\mathfrak{B}$ 检查是否同时满足:

Ⅰ $\mathfrak{A}$ 未曾提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 $Corrupt$ 询问;

Ⅱ $\mathfrak{A}$ 未曾利用相同的挑战提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 $Spend$ 询问;

Ⅲ $att_{j_0^*}$,$att_{j_1^*}$ 中包含相同的属性数量.

若是,则 $\mathfrak{B}$ 选取 $b \in_R \{0,1\}$,并且代表用户 $U_{j_b^*}$ 与 $\mathfrak{A}$ 执行现金分割和支付协议.

④ 询问阶段 2. 在该阶段,$\mathfrak{B}$ 执行与询问阶段 1 相同的模拟过程. 同时,不允许 $\mathfrak{A}$ 提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 $Corrupt$ 询问,也不允许 $\mathfrak{A}$ 利用相同的挑战提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 $Spend$ 询问.

⑤ 结束. $\mathfrak{A}$ 输出挑战用户下标 b'. 最终,$\mathfrak{A}$ 获胜的条件是 $b'=b$.

4) 可开脱性. 假设敌手 $\mathfrak{A}$ 能攻破 B^*,M^*,U^* 和 $\overline{U}^*$. 对于 $\mathfrak{A}$ 而言,它仍然无法诬陷诚实用户有重复支付行为. 在实验执行过程中,$\mathfrak{A}$ 充当 B^*,M^*,U^* 和 $\overline{U}^*$($Host$ 部分)的角色,且 $\mathfrak{B}$ 充当 U 与 $\overline{U}^*$(TPM 部分)的角色. 详细步骤如下:

① 初始化. $\mathfrak{B}$ 执行系统建立算法. 然后,$\mathfrak{B}$ 向 $\mathfrak{A}$ 提供 $params$,并且接受后者提供的 pk_B.

② 询问. 在该阶段,$\mathfrak{B}$ 为 $\mathfrak{A}$ 模拟预言机 Hash(·),$Withdraw$(·),$Spend$(·)和 $Corrupt$(·)的执行过程:

Ⅰ Hash 询问. 模拟过程与平衡性实验相同.

Ⅱ $Withdraw$ 询问. 当接收到 $\mathfrak{A}$ 提供的用户身份 $\overline{U}^*$ 和 att,$\mathfrak{B}$ 代表 TPM 与 $\mathfrak{A}$ 执行取款协议.

Ⅲ $Spend$ 询问. 该询问分 2 种情况:

情况 1. 当接收到 $\mathfrak{A}$ 提供的用户身份 U,$\mathfrak{B}$ 代表 U 与 $\mathfrak{A}$ 执行现金分割和支付协议.

情况 2. 当接收到 $\mathfrak{A}$ 提供的用户身份 $\overline{U}^*$ 以及 $\overline{U}^*$ 的 $Host$ 部分提供的数据,$\mathfrak{B}$ 代表 $\overline{U}^*$ 的 TPM 为 $\mathfrak{A}$ 提供应答.

Ⅳ $Corrupt$ 询问. 当接收到 $\mathfrak{A}$ 提供的用户身份 U 或 $\overline{U}^*$,$\mathfrak{B}$ 返回该用户的 TPM 私钥 k_0,att 和电子钱包,并将 k_0 加入列表 RL.

③ 结束. $\mathfrak{A}$ 输出 $coin_1$ 与 $coin_2$. 最终,$\mathfrak{A}$ 的获胜条件是同时满足:

Ⅰ $coin_1$ 与 $coin_2$ 均能通过存款协议的验证过程;

Ⅱ $coin_1$ 与 $coin_2$ 含有相同的货币序列号;

Ⅲ 当利用 $coin_1$ 与 $coin_2$ 执行身份追踪算法时,将揭示某个诚实用户的身份.

定理 1. 在随机预言模型下,只要 XKEA 假设、XDH 假设和离散对数假设成立,则新的可授权电子现金系统满足正确性、平衡性、匿名性和可开脱性.

证明. 假设 $\mathfrak{B}$ 在各安全性质实验之前自行创建列表 HU,CU,RL,S_{wallet},分别用于存放诚实 TPM 的私钥、被攻破的 TPM 的私钥、被废除的 TPM 的私钥以及用户提取的电子钱包.

1) 正确性. 该性质容易观察得出. 限于篇幅,省略了具体过程.

2) 平衡性. $\mathfrak{B}$ 以底层属性空间上的签名方案公钥 (Q,A,A_0,A_1,A_2,Z) 作为输入. $\mathfrak{B}$ 利用系统建立算法中的方式产生 $params$ 中的剩余参数. $\mathfrak{B}$ 选取 $\xi \in_R \mathbb{Z}_p^*$,设置 $Y=Q'^{\xi}$,并且定义集合 $\Sigma=\{g^{\frac{1}{\xi+1}},g^{\frac{1}{\xi+2}},\cdots,g^{\frac{1}{\xi+2N}}\}$. $\mathfrak{B}$ 初始化列表 RL. $\mathfrak{B}$ 设置 $pk_B=(Q,A,A_0,A_1,A_2,Z)$,$sk_B=(*,*,*,*,*,\xi)$,其中,$*$ 表示未知元素. 最后,$\mathfrak{B}$ 向 $\mathfrak{A}$ 提供 $params$,pk_B. 在当前实验中,$\mathfrak{B}$ 为 $\mathfrak{A}$ 提供以下的预言服务:

① Hash 询问. 当接收到 $\mathfrak{A}$ 提出的关于 $H_i(i=1,2)$ 的散列询问,$\mathfrak{B}$ 分别返回在 $\mathbb{Z}_p$(或 G_1)上选取的随机元素. 同时,$\mathfrak{B}$ 需要确保所提供的回答满足一致性.

② $Withdraw$ 询问. 对该询问的回答分为 2 种情况:

情况 1. 为部分被攻破的用户 $\overline{U}_j^*$ 产生电子钱包. 在接收到底层签名预言机提供的 (K,S,S_0,S_1,S_2) 之后,$\mathfrak{B}$ 将该元组提供给 $\mathfrak{A}$. $\mathfrak{B}$ 选取 $k_{j,0} \in_R \mathbb{Z}_p^*$,并以 TPM 的身份与 $\mathfrak{A}$ 联合产生知识签名 π_1. 在接收到 $\mathfrak{A}$ 提供的 π_1 之后,$\mathfrak{B}$ 对 $\mathfrak{A}$ 执行重绕,提取出秘密知识 κ_j',$k_{j,1}$,$k_{j,2}$. 现在,$\mathfrak{B}$ 向底层签名预言机提出询问 $R=S^{\kappa_j'}\prod_{i=0}^{2}S_{j,i}^{k_{j,i}}$,在获得后者返回的 (T,κ'') 之

后,$\mathfrak{B}$ 向 $\mathfrak{A}$ 返回($T_j=T, k_j''=\kappa''$). 同时,$\mathfrak{B}$ 设置 $HU\leftarrow HU\cup\{k_{j,0}\}$, $S_{\text{wallet}}\leftarrow S_{\text{wallet}}\cup\{((k_{j,0},k_{j,1},k_{j,2}),\kappa_j=\kappa_j'+\kappa'', S_{j,0}=S_0, S_{j,1}=S_1, S_{j,2}=S_2, T_j=T, J_j=0)\}$.

情况 2. 为被攻破的用户 U_j^* 产生电子钱包. 当前情况与情况 1 的区别是:(1) TPM 私钥 $k_{j,0}$ 是由 $\mathfrak{A}$ 选取的;(2) π_1 是由 $\mathfrak{A}$ 产生的;(3) 最终,$\mathfrak{B}$ 设置 $CU\leftarrow CU\cup\{k_{j,0}\}$.

③ *Spend* 询问. 对该询问的回答分为 2 种情况:

情况 1. $\mathfrak{B}$ 以诚实用户 U_j 的身份与 $\mathfrak{A}$ 执行现金分割和支付协议.

情况 2. $\mathfrak{B}$ 以 TPM 的身份与 $\mathfrak{A}$(充当用户 $\overline{U}_j^*$ 的 *Host* 部分)共同执行现金分割协议. 在该过程中,$\mathfrak{B}$ 采用 TPM 在第 3.2 节的方式计算($t, R_1', R_6', R_8', c, s_\beta, s_\kappa, s_{k_0}, s_{k_1}, s_{k_2}, s_{end}, s_J, s_{end\times(J+k_1)}, s_{k_0\times(J+k_2)}, s_{end\times(J+k_2)}, s_l, s_v$),并将这些元素提供给 $\mathfrak{A}$.

④ *Corrupt* 询问. 当 $\mathfrak{A}$ 请求攻破 U_j 或 $\overline{U}_j^*$,$\mathfrak{B}$ 将对应的 TPM 私钥 $k_{j,0}, att=\{k_{j,1},k_{j,2}\}$ 和电子钱包返回给 $\mathfrak{A}$,并且设置 $CU\leftarrow CU\cup\{k_{j,0}\}, RL\leftarrow RL\cup\{k_{j,0}\}$.

最终,$\mathfrak{A}$ 输出伪造的 $coin^*=(\overline{S}^*,\overline{T}^*,\overline{R}^*,\hat{K}^*,\hat{S}^*,\hat{S}_0^*,\hat{S}_1^*,\hat{S}_2^*,\widetilde{C}^*,\widetilde{T}^*,b^*,t^*,cmt^*,end^*,B_1^*,D_1^*,D_2^*,\pi_2^*)$. 现在,$\mathfrak{A}$ 获胜的条件是同时满足:

Ⅰ $coin^*$ 可以通过存款协议的有效性验证过程;

Ⅱ 对于每个 $k_{j,0}\in CU$,满足 $k_{j,0}\in RL$;

Ⅲ $coin^*$ 并非通过向 $\mathfrak{B}$ 提出 *Spend* 询问而产生的.

上述 3 个条件同时成立表明,$\mathfrak{B}$ 能通过对 $\mathfrak{A}$ 执行重绕而从 π_2^* 中提取出秘密元组($k_0^*,k_1^*,k_2^*,\kappa^*,\hat{K}^*,\hat{S}^*,\hat{S}_0^*,\hat{S}_1^*,\hat{S}_2^*,\hat{T}^*$),使得($\kappa^*,\hat{K}^*,\hat{S}^*,\hat{S}_0^*,\hat{S}_1^*,\hat{S}_2^*,\hat{T}^*$)构成关于向量($k_0^*,k_1^*,k_2^*$)的底层方案签名,且 $k_0^*\notin CU$. 这表明,$\mathfrak{B}$ 成功地攻破了底层签名方案的不可伪造性,即违背了 XKEA 假设.

3) 匿名性. $\mathfrak{B}$ 以某个四元组($\bar{u},\bar{v}=\bar{u}^\gamma,\bar{w}=\bar{u}^\delta,\bar{z}$)$\in G_1^4$ 作为输入. $\mathfrak{B}$ 的目标是对以下情况进行分辨,即究竟满足 $\bar{z}=\bar{u}^{\gamma\delta}$,还是 $\bar{z}$ 为 G_1 上的随机元素. $\mathfrak{B}$ 选取诚实用户下标 $j_0,j_1\in_R\{1,2,\cdots,N\}$,其中 N 表示实验中的最大用户数量. $\mathfrak{B}$ 执行系统建立和银行系统建立算法,并且向 $\mathfrak{A}$ 提供 $params, pk_B, sk_B$. 在当前实验中,$\mathfrak{B}$ 为 $\mathfrak{A}$ 提供以下的预言服务:

① Hash 询问. 模拟方式同平衡性实验.

② *Withdraw* 询问. 当接收到 $\mathfrak{A}$ 的询问内容 U_j, att,$\mathfrak{B}$ 与 $\mathfrak{A}$ 执行取款协议. 在该过程中,$\mathfrak{B}$ 充当 U_j 的角色,且 $\mathfrak{A}$ 充当 B^* 的角色. 最后,$\mathfrak{B}$ 设置 $HU\leftarrow HU\cup\{k_{j,0}\}$, $S_{\text{wallet}}\leftarrow S_{\text{wallet}}\cup\{((k_{j,0},k_{j,1},k_{j,2}),\kappa_j,S_{j,0},S_{j,1},S_{j,2},T_j,J_j=0)\}$.

③ *Spend* 询问. 当接收到 $\mathfrak{A}$ 的询问内容 U_j,$\mathfrak{B}$ 分为 3 种情况进行处理:

情况 1. 满足 $j\notin\{j_0,j_1\}$. $\mathfrak{B}$ 利用所掌握的 U_j 的 TPM 私钥和电子钱包诚实地与 $\mathfrak{A}$ 执行现金分割和支付协议.

情况 2. 满足 $j=j_0$. $\mathfrak{B}$ 选取 $r\in_R\mathbb{Z}_p^*$,设置 $b=\bar{u}^r, t=\bar{v}^r$. $\mathfrak{B}$ 丢弃在取款协议中产生的 TPM 私钥. $\mathfrak{B}$ 借助对随机预言机输出的控制能力模拟产生 $cmt, coin'$,并且将它们提供给 $\mathfrak{A}$. $coin'$ 中的知识签名 π_2 表明用户掌握 TPM 私钥 $\log_u\bar{v}$,但 $\mathfrak{B}$ 事实上并不掌握该元素.

情况 3. 满足 $j=j_1$. $\mathfrak{B}$ 选取 $r\in_R\mathbb{Z}_p^*$,设置 $b=\bar{w}^r, t=\bar{z}^r$. 剩余的模拟过程与情况 2 相同.

④ *Corrupt* 询问. 当 $\mathfrak{A}$ 请求攻破 U_j,若 $j\notin\{j_0,j_1\}$,则 $\mathfrak{B}$ 将对应的 TPM 私钥 $k_{j,0}$ 和电子钱包返回给 $\mathfrak{A}$,并且设置 $CU\leftarrow CU\cup\{k_{j,0}\}, RL\leftarrow RL\cup\{k_{j,0}\}$. 否则,$\mathfrak{B}$ 模拟失败.

在挑战阶段,$\mathfrak{A}$ 输出 n_M 以及用户身份 j_0^*, j_1^*. $\mathfrak{B}$ 执行以下检查:

Ⅰ $\mathfrak{A}$ 是否未曾提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 *Corrupt* 询问;

Ⅱ $\mathfrak{A}$ 是否未曾利用相同的挑战提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 *Spend* 询问;

Ⅲ 是否满足 $|att_{j_0^*}|=|att_{j_1^*}|$.

此外,若满足 $\{j_0^*,j_1^*\}\neq\{j_0,j_1\}$,则 $\mathfrak{B}$ 模拟失败;否则,不失一般性地假设,$j_0^*=j_0, j_1^*=j_1$. $\mathfrak{B}$ 选取 $b\in_R\{0,1\}$,并且利用 *Spend* 询问情况 2 或情况 3 中的方式模拟产生 $U_{j_b^*}$ 的输出 $cmt, coin'$. 接下来,$\mathfrak{A}$ 可以在实验中继续向 $\mathfrak{B}$ 提出上述类型的询问,但不允许它提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 *Corrupt* 询问,也不允许它利用同一个挑战提出关于 $U_{j_0^*}$ 或 $U_{j_1^*}$ 的 *Spend* 询问. 最终,$\mathfrak{A}$ 输出猜测结果 b'. 若满足 $b'=b$,则 $\mathfrak{B}$ 判定 $\bar{z}$ 为群 G_1 上的随机元素. 否则,$\mathfrak{B}$ 判定 $\bar{z}=\bar{u}^{\gamma\delta}$ 成立.

4) 可开脱性. $\mathfrak{B}$ 以群 G_1 上的离散对数问题实例$(u,v)\in G_1^2$ 为输入. $\mathfrak{B}$ 的目标是计算离散对数 $\log_u v$. 为此,$\mathfrak{B}$ 设置 $\overline{S}_0=u$. 然后,$\mathfrak{B}$ 执行系统建立算法,从而产生 $params$ 中的剩余参数. $\mathfrak{B}$ 将 $params$ 提供给 $\mathfrak{A}$,并且接受 $\mathfrak{A}$ 提供的 pk_B. $\mathfrak{B}$ 选取 $j^*\in_R\{1,2\cdots,N\}$,其中 N 的含义同匿名性实验. 在当前实验的执行过程中,$\mathfrak{B}$ 为 $\mathfrak{A}$ 提供以下类型的预言服务:

① Hash 询问. 模拟方式同平衡性实验.

② *Withdraw* 询问. 当接收到 $\mathfrak{A}$ 的询问内容 $\overline{U}_j^*, att$,$\mathfrak{B}$ 分 2 种情况进行处理:

情况 1. 满足 $j\neq j^*$,$\mathfrak{B}$ 在取款协议中选取 $k_{j,0}$,并且以诚实方式与 $\mathfrak{A}$ 合作产生知识签名 π_1. 最后,$\mathfrak{B}$ 设置 $HU\leftarrow HU\cup\{k_{j,0}\}$,$S_{\text{wallet}}\leftarrow S_{\text{wallet}}\cup\{((k_{j,0},k_{j,1},k_{j,2}),\kappa_j,S_{j,0},S_{j,1},S_{j,2},T_j,J_j=0)\}$.

情况 2. 满足 $j=j^*$. $\mathfrak{B}$ 设置 $\overline{U}=v$,并且与 $\mathfrak{A}$ 执行取款协议. 与情况 1 的区别是,$\mathfrak{B}$ 并不掌握 *TPM* 私钥 $k_{j^*,0}=\log_{S_0}\overline{U}=\log_u v$. 因此,$\mathfrak{B}$ 采用模拟方式产生知识签名 π_1. 具体方法是:

Ⅰ 当接收到$(ID,cnt_{ID},n_B,S_{j^*},S_{j^*,0},S_{j^*,1},S_{j^*,2},\kappa'_{j^*},k_{j^*,1},k_{j^*,2})$,$\mathfrak{B}$ 设置 $\overline{U}=v$. $\mathfrak{B}$ 选取 $R',R_2\in_R G_1$,并且选取 $s_{\kappa'_{j^*}},s_{k_{j^*,0}},s_{k_{j^*,1}},s_{k_{j^*,2}},c\in_R\mathbb{Z}_p$. $\mathfrak{B}$ 设置 $R_1=\overline{S}_{j^*,0}^{s_{k_{j^*,0}}}\overline{U}^{-c}$,$R_2=S_{j^*}^{s_{\kappa'_{j^*}}}\left(\prod_{i=0}^{2}S_{j^*,i}^{s_{k_{j^*,i}}}\right)(R'S_{j^*}^{\kappa'_{j^*}}S_{j^*,1}^{k_{j^*,1}}S_{j^*,2}^{k_{j^*,2}})^{-c}=S_{j^*}^{s_{\kappa'_{j^*}}}\left(\prod_{i=0}^{2}S_{j^*,i}^{s_{k_{j^*,i}}}\right)R^{-c}$. $\mathfrak{B}$ 定义 $c=H_1(S_{j^*}\|S_{j^*,0}\|S_{j^*,1}\|S_{j^*,2}\|\overline{U}\|R\|R_1\|R_2\|n_B)$. 然后,$\mathfrak{B}$ 向 $\mathfrak{A}$ 返回$(\overline{U},R',R'_2)$;

Ⅱ 当接收到 $\mathfrak{A}$ 提供的$(R,R_2,r_{\kappa'_{j^*}},r_{k_{j^*,1}},r_{k_{j^*,2}})$之后,$\mathfrak{B}$ 返回$(c,s_{\kappa'_{j^*}},s_{k_{j^*,0}},s_{k_{j^*,1}},s_{k_{j^*,2}})$;

Ⅲ $\mathfrak{B}$ 设置 $HU\leftarrow HU\cup\{k_{j^*,0}\}$,$S_{\text{wallet}}\leftarrow S_{\text{wallet}}\cup\{((k_{j^*,0},k_{j^*,1},k_{j^*,2}),\kappa_{j^*},S_{j^*,0},S_{j^*,1},S_{j^*,2},T_{j^*},J_{j^*}=0)\}$,其中 $k_{j^*,0}$ 为未知元素.

③ *Spend* 询问. $\mathfrak{B}$ 分为 2 种情况进行处理:

情况 1. $\mathfrak{B}$ 以诚实用户 U_j 的身份与 $\mathfrak{A}$ 执行现金分割和支付协议. 此时,若满足 $j\neq j^*$,则 $\mathfrak{B}$ 利用所掌握的 *TPM* 私钥和电子钱包以诚实方式产生 $cmt,coin'$,并将它们提供给 $\mathfrak{A}$. 若满足 $j=j^*$,则 $\mathfrak{B}$ 模拟产生 $cmt,coin'$,并将它们提供给 $\mathfrak{A}$.

情况 2. $\mathfrak{B}$ 以诚实 *TPM* 的身份与 $\mathfrak{A}$ 联合执行现金分割和支付协议,且 $\mathfrak{A}$ 将在这 2 个协议中分别充当 *Host* 和 M^* 的角色. 在现金分割协议中,$\mathfrak{B}$ 产生$(t,R'_1,R'_6,R'_8,c,s_\beta,s_\kappa,s_{k_0},s_{k_1},s_{k_2},s_{end},s_J,s_{end\times(J+k_1)},s_{k_0\times(J+k_2)},s_{end\times(J+k_2)},s_l,s_\nu)$,并且将它们提供给 $\mathfrak{A}$. 需要指出的是,若 $j\neq j^*$,则这些元素是 $\mathfrak{B}$ 采用诚实方式产生的. 相反,它们是 $\mathfrak{B}$ 采用模拟方式产生的.

④ *Corrupt* 询问. 当 $\mathfrak{A}$ 请求攻破 U_j,若 $j\neq j^*$,则 $\mathfrak{B}$ 将对应的 *TPM* 私钥 $k_{j,0}$,*att* 和电子钱包返回给 $\mathfrak{A}$,并且设置 $CU\leftarrow CU\cup\{k_{j,0}\}$,$RL\leftarrow RL\cup\{k_{j,0}\}$. 否则,$\mathfrak{B}$ 模拟失败.

最终,$\mathfrak{A}$ 输出$(\overline{U}^*,coin_1,coin_2)$,满足 $coin_1=(\overline{S},\overline{T},\overline{R},\hat{K},\hat{S},\hat{S}_0,\hat{S}_1,\hat{S}_2,\widetilde{C},\widetilde{T},b,t,cmt,end,B_1,D_1,D_2,\pi_2)$,$coin_2=(\overline{S}',\overline{T}',\overline{R}',\hat{K}',\hat{S}',\hat{S}'_0,\hat{S}'_1,\hat{S}'_2,\widetilde{C}',\widetilde{T}',b',t',cmt',end',B'_1,D'_1,D'_2,\pi'_2)$,$\overline{S}=\overline{S}'$,$\overline{U}^*=(\overline{T}'^{\overline{R}}/\overline{T}^{\overline{R}'})^{(\overline{R}-\overline{R}')^{-1}}$. 此时,$\mathfrak{B}$ 通过对 $\mathfrak{A}$ 执行重绕而提取出秘密知识 $k_{j^*,0}=\log_u v$,从而成功求解了群 G_1 上的离散对数问题实例. 证毕.

在定理 2 的证明中,符号 $message_1$ 表示 U 向 M 发送的消息$(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK)),Condition_U)$. 符号 $message_2$ 表示 M 向 U 发送的消息 m. 符号 $message_3$ 表示 U 向 M 发送的消息 *end*. 此外,假设所有的 TTP $P_1,P_2,\cdots,P_n$ 可划分为 4 个集合,即 B_1,B_2,B_3,B_4. 其中,B_1 表示由诚实 TTP 构成的集合. B_2 表示由与 M 合谋的 TTP 构成的集合(即即使当前时间已经超过 *Time*,$P_i\in B_2$ 仍然会在恢复子协议中为 M 提供对 C_i 的解密服务,但不会为 U 提供密文$(\tilde{u},\tilde{e},\tilde{v})$). B_3 表示由与 U 合谋的 TTP 构成的集合(即即使当前时间已经超过 *Time*,当 M 向 $P_i\in B_3$ 发送$((\tilde{u},\tilde{e},\tilde{v}),\pi_3,message_1)$,$P_i$ 会向 U 发送$(\tilde{u},\tilde{e},\tilde{v})$,但不会向 M 提供对 C_i 的解密服务). B_4 表示由拒绝同 M 与 U 中的任何一方合作的 TTP 构成的集合.

定理 2. 假设 k 表示底层 Shamir 门限方案的门限值,在满足 $|B_2|<k\leqslant|B_2|+|B_1|$ 的情况下,本文的交换和恢复子协议满足文献[23]模型要求的完备性、公平性、时效性和保密性.

证明.

1) 完备性. 该性质表明,当 U 与 M 均保持诚实时,U 会在交换协议结束后获得所期望的数字商品 m. 同时,M 会获得 U 的秘密元素 *end*. 观察得出,在 U 与 M 保持诚实的情况下,该性质一定会得到满足. 若由于通信延迟而导致 U 的消息 $message_3$ 无法在 $Time-Time_{\max}$ 之前及时送达,M 将执行恢复子协议. 根据条件 $|B_2|<k\leqslant|B_2|+|B_1|$ 可知,至少有 k 个 $P_i\in(B_2\cup B_1)$ 将与 M 进行合作,且至少有 1 个 $P_i\in(B_3\cup B_1)$ 将与 U 进行合作. 显然,在这种情况下,本文协议仍然满足完备性.

2) 公平性. 该性质表明,当在 U 与 M 中至少有一方保持诚实时,在交换协议结束后,或者是 U 得到 m 且 M 得到 *end*,或者 U 与 M 均未获得有关 m 与 *end* 的任何信息. 公平性的证明过程分 2 种情况:

① U 保持诚实且 M^* 是恶意的. 需要指出的是,由于 U 是诚实的,因此消息 $message_1$ 是采用正确方式产生的. 此时,分为 2 种子情况:

Ⅰ M^* 提供的消息 $message_2$ 是错误的. 此时,U 可以借助验证等式 $\mathfrak{g}^m=descr(m)$ 而发觉这一点,此时它不再向 M^* 发送消息 $message_3$,而是一直等待直至 *Time* 时刻. 若 M^* 选择不运行恢复子协议,则任何参与方都无法获得有关商品 m 的任何有用信息. 显然,此时整个交换过程是公平的. 若 M^* 选择

在 $Time-Time_{\max}$ 时刻之后执行恢复子协议,由于只有 $P_i\in B_2$ 才会为 M^* 提供关于 C_i 的解密服务.根据条件 $|B_2|<k\leqslant|B_2|+|B_1|$ 可知,M^* 将无法获得 *end*.显然,此时整个交换过程仍然是公平的.

Ⅱ 在接收到消息 $message_1$ 之后,M^* 不向 U 返回 m,而是选择在 $Time-Time_{\max}$ 时刻之前执行恢复子协议.此时,U 将获得由 $P_i\in(B_1\cup B_3)$ 提供的关于 m 的 Camenisch-Shoup 方案密文 $(\tilde{u},\tilde{e},\tilde{v})$.在这种情况下,若 M^* 向 $P_i\in(B_1\cup B_3)$ 提供的 π_3 是无效的,则根据前提条件 $|B_2|<k\leqslant|B_2|+|B_1|$ 可知,M^* 仍然无法获得 k 个关于 C_i 的解密结果.同时,由于密文 $(\tilde{u},\tilde{e},\tilde{v})$ 无效,U 同样无法借助 $P_i\in(B_1\cup B_3)$ 获得 m.显然,此时整个交换过程是公平的.相反,若 M^* 向 $P_i\in(B_1\cup B_3)$ 提供的 π_3 是有效的,尽管 M^* 能获得 k 个关于 C_i 的解密结果,进而重构得到 *end*.但是,U 同样会获得由 $P_i\in(B_1\cup B_3)$ 提供的有效密文 $(\tilde{u},\tilde{e},\tilde{v})$,并且通过执行解密操作而获得 m.显然,此时整个交换过程仍然是公平的.

② U^* 是恶意的且 M 是诚实的.此时,分为 2 种子情况:

Ⅰ U^* 提供的消息 $message_1$ 是错误的(或已经丢失).显然,M 能借助底层可验证群托管协议的 SigVerifier 算法发觉这一点,因此它决定不向 U^* 提供 $message_2$.此时,整个交换过程将在 $Time$ 时刻后终止.显然,此时整个交换过程是公平的.

Ⅱ U^* 提供的消息 $message_1$ 是正确的.但是,U^* 提供了错误的消息 $message_3$.借助验证等式 $cmt=\bar{g}^{end}$,M 能发觉 $message_3$ 是错误的,于是执行恢复子协议.根据前提条件 $|B_2|<k\leqslant|B_2|+|B_1|$ 可知,M 能获得由 $P_i\in(B_1\cup B_2)$ 提供的足够多的(即大于等于 k 个)关于 $C_i\in message_1$ 的解密结果,从而利用底层可验证群托管协议的 Recover 算法恢复得到 *end*.同时,U^* 已经在消息 $message_2$ 中获得 m.显然,此时整个交换过程是公平的.另一方面,若 U^* 选择不向 M 提供 $message_3$(或者 $message_3$ 已经丢失).采用相同的分析方法可知,此时整个交换过程仍然是公平的.

3) 时效性.该性质表明,整个交换过程最终一定会终止.由于设置了超时时间 $Time$,因此本文协议满足该性质.

4) 保密性.该性质表明,任何其他的参与方都无法获得有关 m 与 *end* 的任何有用信息.若恢复子协议并未得到执行,则表明信息交换仅发生在 U 与 M 之间.此时,被动参与方 $P_i\in(B_1\cup B_2\cup B_3\cup B_4)$ 无法获得有关 m 与 *end* 的任何有用信息.若恢复子协议得到执行,则某些 $P_i\in(B_1\cup B_2\cup B_3\cup B_4)$ 会获得 $C_i\in message_1$ 的解密结果.尽管 k 个合谋的 P_i 能恢复 r,但由于它们不掌握 d,因此无法恢复 *end*.显然,根据 π_3 的零知识性和底层 Camenisch-Shoup 加密方案的安全性,它们也无法揭示 m.　　证毕.

5　改进的有条件的公平支付系统

在本节,我们基于第 2 节的可授权电子现金系统提出改进的有条件的公平支付系统.与可授权电子现金系统相比,新的支付系统需要增加运算任务产生算法,且该算法使用了 Golle 等人[11-12]的技术.同时,需要对原有的现金分割、支付和公平交换协议做出修改.限于篇幅,我们仅着重描述本节公平支付系统与第 2 节电子现金系统的主要差别:

1) 运算任务产生算法.令 $f:D\rightarrow M$ 表示公开的单向函数.假设用户的目标是计算元素 y 在函数 f 下的前像.*Host* 将 f 的定义域 D 划分为若干个子集 $D_1,D_2,\cdots$.同时,*Host* 将运算任务的截止时间设置为 $Time$.*Host* 选取随机整数 $2m$ 作为检查点(也称 $ringer$[11-12])的总数,然后选取 t 个"真实的"检查点 $x_1,x_2,\cdots,x_t\in_R D_i$,其中 $t\in_R[m+1,2m]$.此外,*Host* 选取 $2m-t$ 个"虚假的"检查点 $\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_{2m-t}$,满足 $\bar{x}_1,\bar{x}_2,\cdots,\bar{x}_{2m-t}\notin D_i$.对于上述的所有 $2m$ 个检查点,*Host* 计算它们在函数 f 下的像 $y_1,y_2,\cdots,y_{2m}$,并且设置 $M_i=\{y,y_1,y_2,\cdots,y_{2m}\}$.

2) 现金分割协议.在该协议中,除了向 $worker_i$ 提供 $cmt,coin'$,*Host* 还需要提供运算任务 $F_i=(f,D_i,M_i)$ 以及辅助程序 S_i.

3) 公平交换协议.假设 $worker_i$ 已经完成指定的运算任务 F_i,并且借助 S_i 产生了问题的解 Sol.具体地,对于每个 $x\in D_i$,$worker_i$ 计算 $f(x)$ 并且向 S_i 提供 $(x,f(x))$.若满足 $f(x)\in M_i$,则 S_i 设置 $Sol\leftarrow Sol\cup\{x\}$.在交换子协议中,当接收到 *Host* 提供的元组 $(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK),Condition_U)$,其中 $Condition_U=(pk_{worker_i},F_i,S_i,Time,pk_U)$,$worker_i$ 执行 SigVerifier 算法.若验证通过,则向 *Host* 提供 Sol.在恢复子协议中,$worker_i$ 在 $Time-Time_{\max}$ 之前向 TTP P_i 发送元组 $(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK),Condition_U,F_i,Sol)$.$P_i$ 执行以下验证:①Sol 是否为正确的运算结果;②SigVerifier$(d,cmt,\hat{t},C_1,C_2,\cdots,C_n,cert(AIK),Condition_U)$ 是否

为有效;③当前时间是否未超过 $Condition_U$ 中包含的过期时间 $Time$. 若通过,则 P_i 向 $worker_i$ 发送从 C_i 中恢复得到的元素 r_i,同时向 $Host$ 发送 Sol.

6 本文系统的性能分析

在本节,我们以实现外包计算环境下的公平支付为应用目标,对本文系统与已有的相关系统[1,10-11,15-16]进行了比较. 首先,我们在表 1 中提供了它们在 7 个重要性质方面的比较结果. 在运算平台方面,本文系统要求部署在可信计算平台上,而其他系统都是针对 PC 机平台设计的. 在文献[1,16]和本文系统中,用户通过执行取款协议而获得可以多次支付的电子钱包,而其他系统则仅支持单次支付. 文献[1]系统的公平交换过程以及文献[10,15]系统的支付过程均要求用户与 worker 执行低效的 cut-and-choose 证明. 为了避免执行该项证明,文献[11,16]系统使用了可验证加密技术,而本文系统则采用了基于 TPM 的群托管技术. 文献[11,15]和本文系统都使用了 Golle-Mironov 外包计算模型,其他系统则未采用该模型. 为了保护用户隐私,多数系统都满足取款和支付过程的匿名性,但文献[15]系统除外. 文献[10,15]系统均未提供用户与 worker 间的公平交换机制,其余系统均借助 TTP 实现了公平交换过程. 然而,仅有本文系统考虑了拒绝服务攻击的风险. 最后,本文系统与文献[11]系统的一个优势是,用户无需在公平交换过程中保持状态,从而有利于在移动设备上进行部署.

Table 1 Main Properties Comparison of the Related Systems

表 1 相关系统的主要性质比较

Scheme	Computing Platform	Payment Mechanism	Cut-and-Choose Proof	Golle-Mironov Model	Anonymity	Type of TTP	User's State
Ref[1]	PC	Multiple Payment	Yes	No	Yes	Centralized	Stateful
Ref[10]	PC	Single Payment	Yes	No	Yes	Centralized	
Ref[11]	PC	Single Payment	No	Yes	Yes	Centralized	Stateless
Ref[15]	PC	Single Payment	Yes	Yes	No		
Ref[16]	PC	Multiple Payment	No	No	Yes	Centralized	Stateful
Ours	Trusted Platform	Multiple Payment	No	Yes	Yes	Distributed	Stateless

在表 2 中,我们提供了本文系统与已有系统的通信耗费比较结果,且比较过程中假定以 80 b 安全性为目标. 文献[1,11,15-16]系统是在 p 阶椭圆曲线群 G 和 RSA 群 $\mathbb{Z}_N$ 上构造的,而文献[10]和本文系统是在双线性群 (G_1,G_2) 和 RSA 群 $\mathbb{Z}_N$ 上构造的. 对于文献[1,11,15-16]系统,选取 $|p|=160$ b,且 $\mathbb{Z}_N$ 上的元素长度为 1024 b. 对于文献[10]和本文系统,假设 (G_1,G_2) 是采用 MNT 曲线实现的,群 G_1 上的元素长度为 171 b[26]. 对于文献[1,10-11,15-16]系统,散列函数的输出长度为 160 b. 对于本文系统,散列函数 H_1,H_2,H_3,H_4 的输出长度分别为 160 b,171 b,160 b,512 b. 文献[1,10,15]系统使用了 cut-and-choose 证明技术,根据文献[8]结论,该项证明要求至少迭代执行 40 轮. 需要指出的是,文献[11,15]和

Table 2 Communication Cost Comparison of the Related Systems

表 2 相关系统的通信耗费比较 b

Scheme	Communication Cost of the Withdraw Process		Communication Cost of the Spend Process		Communication Cost of the Fair Exchange Process		Communication Cost of the Reconstruct Process	
	User	Bank	User	Worker	User	Worker	User	Worker
Ref[1]	171	3 014	23 654	0	103 200	352+$\|Sol\|$	0	0
Ref[10]	971	2 041	214 884	109 411			0	0
Ref[11]	673	331	4 514	160	1 195	$\|Sol\|$	0	0
Ref[15]	160	1 184	184 131	40			0	7 584
Ref[16]	171	3 014	23 654×θ	0	4 976×θ	4 496	0	0
Ours	1 462	1 506	5 136	160	3 361	$\|Sol\|$	0	0

本文系统的外包运算过程都使用了 Golle-Mironov 模型,因此要求用户在公平交换过程中向 worker 提供由检查点构成的集合.在表 2 中,我们采用了文献[15]的估算方法,即假设该集合中检查点的数量为 10.为了获得具体的通信耗费估算结果,我们采用了文献[12]中的外包运算任务实例,即给定 80 b 对称加密方案下的密文 CT 和明文 PT,要求寻找与 PT 相匹配的密钥.此外,符号 Sol 表示 worker 提供的外包运算结果,其通信耗费设为 $|Sol|$.正如引言中所述,文献[16]系统要求用户将同一个运算任务外包给多个 worker,因此用户在支付和公平交换过程中的通信耗费需要乘以一个为 $\theta>1$ 的因子.

在表 3 中,我们提供了本文系统与已有系统的运算耗费比较结果.在比较中,我们仅统计代价较高的指数运算和对运算,而且不再区分单指数运算和多指数运算.我们采用如下方式将所有运算都估算为 RSA 群上的指数运算:在使用 MNT 曲线的情况下,1 次群 G_1 和群 G_T 上的指数运算分别相当于 0.1 次和 1 次 RSA 群的指数运算[26].1 次对运算相当于 10 次 RSA 群的指数运算[26].1 次群 G 上的指数运算相当于 0.1 次 RSA 群的指数运算.文献[1,10,15]系统的 cut-and-choose 证明过程和本文系统的公平交换过程都要求使用 CCA 安全的公钥加密系统.在比较过程中,我们统一采用了 RSA-OAEP 加密方案.该方案解密过程的运算耗费相当于 1 次 RSA 群的指数运算,而加密过程高效得多(其效率约为前者的 18～19 倍[27]).此外,本文系统的公平交换过程要求 TPM 利用 AIK 中的私钥部分产生数字签名,我们使用了 RSA 签名方案.同时,TPM 需要利用 (k,n)-Shamir 方案对背书元素 end 进行秘密分享.采用文献[7]的估算方法,我们选取 $n=2$.如上所述,对于文献[16]系统,用户在支付和公平交换过程中的运算耗费需要乘以一个为 θ 的因子.最后,我们在表 3 中用上标 * 和 ** 分别指示本文系统中 TPM 和 Host 的运算耗费.同时,上标 + 指示采用预计算技术后的运算耗费.

Table 3　Computation Cost Comparison of the Related Systems (Number of Exponential Operations)

表 3　相关系统的运算耗费比较(指数运算次数)

Scheme	Communication Cost of the Withdraw Process		Communication Cost of the Spend Process		Communication Cost of the Fair Exchange Process		Communication Cost of the Reconstruct Process	
	User	Bank	User	Worker	User	Worker	User	Worker
Ref[1]	3	4	9.8	9.5	2.2	21.1	0	0.3
Ref[10]	251.1	1.2	28.2	88.6			1	2
Ref[11]	1	0.3	3.2	2.7	1	0	0	0.1
Ref[15]	0.2	1.1	45.1	6.5			0	0
Ref[16]	3	4	$9.8\times\theta$	9.5	$10.2\times\theta$	10.2	0	0.3
Ours	0.4^*,100.3^{**}	0.8	$0.5^*/0.3^{*+}$, $2.2^{**}/0.3^{**+}$	121	2.2^*,0^{**}	0.2	0^*,0^{**}	0.2

Note: * indicates the computation cost of TPM; ** indicates the computation cost of Host; + indicates the computation cost by using the technique of pre-computation.

7　总　　结

针对已有的可授权电子现金系统通信效率不高且要求使用低效 cut-and-choose 证明技术的问题,本文基于 DAA-A 技术提出一个改进的可授权电子现金系统,并利用所得系统解决了外包运算环境下的公平支付问题.与已有同类系统相比,本文系统的优势是可以部署于可信计算平台,而且同时满足允许多次支付、无需使用 cut-and-choose 技术和用户匿名性等实用性质.此外,本文系统的公平交换子协议可以有效抵抗拒绝服务攻击,且无需要求用户保持状态.效率分析表明,在支付和公平交换阶段,本文系统用户端的运算和通信耗费接近于高效的文献[11]系统.在采用预计算技术后,用户端在支付阶段的运算性能显著优于其他系统.

参　考　文　献

[1] Camenisch J, Lysyanskaya A, Meyerovich M. Endorsed e-cash [C] //Proc of IEEE S&P 2007. Piscataway, NJ: IEEE, 2007: 101-115

[2] Camenisch J, Hohenberger S, Lysyanskaya A. Compact e-cash [G] //LNCS 3494: Proc of EUROCRYPT 2005. Berlin: Springer, 2005: 302-321

[3] Yu Yulei, Dong Xiaolei, Cao Zhenfu. A trustee-based and efficient divisible e-cash scheme [J]. Journal of Computer Research and Development, 2015, 52(10): 2304-2312 (in Chinese)
(虞郁磊, 董晓蕾, 曹珍富. 基于可信第三方的高效可分电子现金方案[J]. 计算机研究与发展, 2015, 52(10): 2304-2312)

[4] Yang Bo, Feng Dengguo, Qin Yu, et al. Research on direct anonymous attestation scheme based on trusted mobile platform [J]. Journal of Computer Research and Development, 2014, 51(7): 1436-1445 (in Chinese)
(杨波, 冯登国, 秦宇, 等. 基于可信移动平台的直接匿名证明方案研究[J]. 计算机研究与发展, 2014, 51(7): 1436-1445)

[5] Chen L, Urian R. DAA-A: Direct anonymous attestation with attributes [G] //LNCS 9229: Proc of TRUST 2015. Berlin: Springer, 2015: 228-245

[6] Desmoulins N, Lescuyer R, Sanders O. Direct anonymous attestations with dependent basename opening [G] //LNCS 8813: Proc of CANS 2014. Berlin: Springer, 2014: 206-221

[7] Tate S R, Vishwanathan R. Efficient verifiable escrow and fair exchange with trusted hardware [EB/OL]. (2013-05-29) [2015-11-01]. http://eprint.iacr.org/2013/427

[8] Asokan M, Shoup V, Waidner M. Optimistic fair exchange of digital signatures [J]. IEEE Journal on Selected Areas in Communications, 2000, 18(4): 593-610

[9] Ateniese G. Verifiable encryption of digital signatures and applications [J]. ACM Trans on Information and System Security, 2004, 7(1): 1-20

[10] Blanton M. Improved conditional e-payments [G] //LNCS 5037: Proc of ACNS 2008. Berlin: Springer, 2008: 188-206

[11] Chen Xiaofeng, Li Jin, Susilo W. Efficient fair conditional payments for outsourcing computations [J]. IEEE Trans on Information Forensics and Security, 2012, 7(6): 1687-1694

[12] Golle P, Mironov I. Uncheatable distributed computations [G] //LNCS 2020: Proc of CT-RSA 2001. Berlin: Springer, 2001: 425-440

[13] Baldimts F, Lysyanskaya A. On the security of one-witness blind signature schemes [G] //LNCS 8270: Proc of ASIACRYPT 2013. Berlin: Springer, 2013: 82-99

[14] Camenisch J, Shoup V. Practical verifiable encryption and decryption of discrete logarithms [G] //LNCS 2729: Proc of CRYPTO 2003. Berlin: Springer, 2003: 126-144

[15] Carbunar B, Tripunitara M V. Payments for outsourced computations [J]. IEEE Trans on Parallel and Distributed Systems, 2012, 23(2): 313-320

[16] Küpcü A. Incentivized outsourced computation resistant to malicious contractors [EB/OL]. (2014-12-12) [2016-03-01]. http://eprint.iacr.org/2014/992

[17] Ringers S, Verheul E, Hoepman J H. An effcient self-blindable attribute-based credentials [EB/OL]. [2016-01-03]. https://sietseringers.net/files/abc.pdf

[18] Hwang J Y, Chen L, Cho H S, et al. Short dynamic group signature scheme supporting controllable linkability [J]. IEEE Trans on Information Forensics and Security, 2015, 10(6): 1109-1124

[19] Camenisch J, Chaabouni R, Shelat A. Efficient protocols for set membership and range proofs [G] //LNCS 5350: Proc of ASIACRYPT 2008. Berlin: Springer, 2008: 234-252

[20] Boneh D, Boyen X. Short signatures without random oracles and the SDH assumption in bilinear groups [J]. Journal of Cryptology, 2008, 21(2): 149-177

[21] Arfaoui G, Lalande J F, Traoré J, et al. A practical set-membership proof for privacy-preserving NFC mobile ticketing [C]//Proc of PETS 2015. Berlin: De Gruyter Press, 2015: 25-45

[22] Boudot F. Efficient proofs that a committed number lies in an interval [G] //LNCS 1807: Proc of EUROCRYPT 2000. Berlin: Springer, 2000: 431-444

[23] Avoine G, Vaudena S. Optimistic fair exchange based on publicly verifiable secret sharing [G] //LNCS 3108: Proc of ACISP 2004. Berlin: Springer, 2004: 74-85

[24] Stadler M. Publicly verifiable secret sharing [G] //LNCS 1070: Proc of EUROCRYPT 1996. Berlin: Springer, 1996: 190-199

[25] Liu J K, Tsang P P, Wong D S, et al. Universal custodian-hiding verifiable encryption for discrete logarithms [G] //LNCS 3935: Proc of ICISC 2005. Berlin: Springer, 2006: 389-409

[26] Boyen X. A tapestry of identity-based encryption: Practical frameworks compared [J]. International Journal of Applied Cryptography, 2008, 1(1): 3-21

[27] Lauter K. The advantages of elliptic curve cryptography for wireless security [J]. IEEE Wireless Communications, 2004, 11(1): 62-67

Liu Xin, born in 1978. PhD and associate professor. Member of China Computer Federation. His main research interests include cryptography and information security.

Zhang Bo, born in 1981. PhD and Lecturer. His main research interests include cryptography and information security (zbsdu@126.com).

附录 A. 知识签名 π_3 的具体实现过程

1) π_3 的产生过程如下:

Step1. M 选取 $r_{r'}, r_s \in_R [-\tilde{n} \times 2^{\omega'+\omega''}, \tilde{n} \times 2^{\omega'+\omega''}]$,选取 $r_m \in_R [-\rho \times 2^{\omega'+\omega''}, \rho \times 2^{\omega'+\omega''}]$,计算

$$R_1 = \tilde{g}^{2r_{r'}} \bmod \tilde{n}^2,$$

$$R_2 = \tilde{y}_1^{2r_{r'}} (1+\tilde{n})^{2r_m} \bmod \tilde{n}^2,$$

$$R_3 = (\tilde{y}_2 \tilde{y}_3^{\mathrm{Hash}_{hk}(u,e)})^{2r_{r'}} \bmod \tilde{n}^2,$$

$$R_4 = \mathfrak{g}^{r_m},$$

$$R_5 = \hat{g}^{r_m} \hat{h}^{r_s} \bmod \tilde{n}^2.$$

Step2. M 设置 $c = H_4(\tilde{u} \parallel \tilde{e} \parallel \tilde{v} \parallel \mathfrak{g}^m \parallel \hat{t} \parallel R_1 \parallel R_2 \parallel \cdots \parallel R_5)$.

Step3. M 在 $\mathbb{Z}$ 上计算 $s_{r'} = r_{r'} + cr$, $s_m = r_m + cm$, $s_s = r_s + cs$.

Step4. M 输出 $\pi_3 = (c, s_{r'}, s_m, s_s)$.

2)π_3 的验证过程如下:

Step1. 将 π_3 分离为 $\pi_3 = (c, s_{r'}, s_m, s_s)$的形式.

Step2. 计算

$$\hat{R}_1 = \tilde{g}^{2s_{r'}} \tilde{u}^{-2c} \bmod \tilde{n}^2,$$

$$R_2 = \tilde{y}_1^{2s_{r'}} (1+\tilde{n})^{2s_m} \tilde{e}^{-2c} \bmod \tilde{n}^2,$$

$$\hat{R}_3 = (\tilde{y}_2 \tilde{y}_3^{\mathrm{Hash}_{hk}(u,e)})^{2s_{r'}} \tilde{v}^{-2c} \bmod \tilde{n}^2,$$

$$\hat{R}_4 = \mathfrak{g}^{s_m} (\mathfrak{g}^m)^{-c}, \hat{R}_5 = \hat{g}^{s_m} \hat{h}^{s_s} \hat{t}^{-c} \bmod \tilde{n}^2.$$

Step3. 验证是否满足 $c = H_4(\tilde{u} \parallel \tilde{e} \parallel \tilde{v} \parallel \mathfrak{g}^m \parallel \hat{t} \parallel \hat{R}_1 \parallel \hat{R}_2 \parallel \cdots \parallel \hat{R}_5)$.

Step4. 验证是否满足$(-\tilde{n}/4) \leqslant s_m \leqslant (\tilde{n}/4)$.

《信息安全研究》期刊简介

习近平总书记指出“没有网络安全就没有国家安全,没有信息化就没有现代化”.数字时代信息安全工具的大众化是不可阻挡的历史潮流.大众化的信息安全已经直接影响到我们每个人的利益,信息安全已成为国家、地方区域经济结构优化提升和转型发展的新机遇.在信息安全上升为国家战略、行业迎来崭新发展机遇形势下,《信息安全研究》期刊应时代而生.

《信息安全研究》是由国家发改委主管、国家信息中心主办的中文学术期刊,其宗旨是集中展示和报道国际、国内网络和信息安全研究领域研究成果及最新应用,传播信息安全基础理论和技术策略,服务国家信息安全形势发展需要.所刊登的论文均经过专家严格评审.

《信息安全研究》将于 2015 年 10 月创刊发行.刊期为月刊,每期 96 页,由《信息安全研究》杂志社出版,国内外公开发行.

《信息安全研究》将以研究致以应用,搭建信息安全领域的学术交流平台,愿意和同行业及社会各界建立联系,友好合作,共赢美好未来.欢迎大家积极投稿、赐稿,洽谈合作.

投稿信箱:ris@cei.gov.cn

编辑部联系人:崔先生(185 0008 6481)

马先生(158 1058 2450)