计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2015.20150596
52(10):2304-2312,2015

基于可信第三方的高效可分电子现金方案

虞郁磊[1] 董晓蕾[2] 曹珍富[2]
[1](上海交通大学计算机科学与工程系 上海 200240)
[2](华东师范大学上海市高可信计算重点实验室 上海 200062)
(ck_yyl@sjtu.edu.cn)

A Trustee-Based and Efficient Divisible E-Cash Scheme

Yu Yulei[1], Dong Xiaolei[2], and Cao Zhenfu[2]
[1](*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240)
[2](*Shanghai Key Laboratory for Trustworthy Computing, East China Normal University, Shanghai* 200062)

Abstract Divisible e-cash systems allow users to purchase a coin of value 2^l and spend it part by part. This system not only need to ensure the anonymity of users, but also can detect double-spending behavior from malicious user. In 2015, Canard presented the first efficient divisible e-cash system in both random oracle model and standard model. In the system, for the coin of value 2^l, the deposit protocol involves up to 2^l pairing operations. When the value of coin is big, the divisible e-cash system will face challenges. If the value is 2^{20}, the system will withstand huge computation pressure; if the value is 2^{30}, it will be a state of collapse. For these potential shortcomings, independent of the work of Canard, we propose a more efficient divisible system based on a trusted third-party, as an improved version of Canard's system. In the scheme, we make use of a trusted third-party, and reduce the number of public parameters and the number of zero-knowledge proof. Especially in the deposit operation, the complexity of deposit protocol is a linear correlation with l, which provides the possibility for solving the problem of large electronic cash.

Key words divisible; e-cash system; trusted third-party; tree-based; commitment

摘 要 可分电子系统满足用户购买价值为 2^l 的电子现金,并能拆分成多块进行消费.在该系统中需要保证用户的匿名性,同时又能查找出那些存在恶意行为的用户.2015 年,Canard 提出了第 1 个在随机预言机和标准模型下安全且高效实用的方案.其中对于提现价值为 2^l 的电子现金,其复杂度是指数级的(需要 2^l 次配对操作).提现操作中需要的高计算量为处理大额电子现金留下了隐患,当金额达到 2^{20} 时,会给系统造成严重的负担;当金额达到 2^{30} 时,系统几乎处于崩溃状态.鉴于以上潜在的不足,在 Canard 的基础上,通过引入可信第三方,提出了一个基于可信第三方的高效电子现金方案,作为对 Canard 方案的改进版.在方案中,结合可信第三方,减少了公共参数的数量和零知识证明的次数.特别是在提现的操作上,其计算复杂度与 l 是成线性相关的,为解决大额电子现金问题提供了可能.

关键词 可分;电子现金系统;可信第三方;树结构;承诺

中图法分类号 TP391

收稿日期:2015-06-17;**修回日期**:2015-08-18

基金项目:国家自然科学基金项目(61321064,61373154,61371083,61411146001)

通信作者:董晓蕾(dongxiaolei@sei.ecnu.edu.cn).

电子现金(e-cash)又被人们称作电子货币或者数字货币,是一种特殊的支付系统.它可以被看做是现实货币的电子化或者数字化,介于其以数字形式存在、不依赖于实物,在交易过程中具有方便、快捷、灵活等特点,现如今已经被广泛应用于各个角落,例如腾讯公司推出的Q币、盛大游戏公司的盛大点券等.其中最著名的是比特币,比特币现已经被一些商家、银行所认可,可以直接参与实物的买卖.

另一方面,电子现金在网络上进行商务活动面临着安全方面的威胁.网络环境决定电子现金容易被截取,容易被恶意分子重复使用,甚至人们可以通过一些技术手段来伪造电子现金.为此,人们迫切寻求一个安全可靠、运行高效的电子现金系统来解决安全隐患.

Chaum[1]在1982年首次提出电子现金方案,利用盲签名来保护用户隐私,但不能找出恶意用户的身份.之后,Stadler等人[2]、Camenisch等人[3]、Frankel等人[4]也各自提出自己的电子现金方案,对电子现金的发展起到了推动作用.Okamoto和Ohta[5]于1991年首次提出了一个可分电子现金系统,利用二叉树的结构表示电子现金,允许用户将电子现金分成任意部分.随后有大量的学者对可分电子现金方案作出改进,但这些方案中,系统执行协议的效率仍然很低,无法应用到实际生活中.

在国内,不少学者专家例如Chen等人[6]对电子现金系统进行过探索,但始终没有产生一个可靠实用的方案.在近期,Canard等人[7]把双线性群结合到电子现金树结构中,将检测双次消费问题转化成判断2个节点是否存在共同叶节点问题,提出了首个在随机预言机和标准模型下安全且高效实用的方案.其方案存在着一个不足:不能很好地解决大额电子现金提现或追踪.当电子现金价值为2^n时,其提现需要2^n次配对操作.当$n=20$时,系统就处于巨大的计算压力之下;当$n=30$时,系统可能已经崩溃.虽然发生双次提现或者双次消费的概率不高,但是一旦发生大金额的双次提现问题,对银行来说这是一个非常严重的问题.这种问题可能发生在银行与银行之间或者下级银行向上级银行之间,一旦发生,追踪活动便刻不容缓,追踪效率就会受到考验.

我们在Canard等人[7]方案的基础上,引入可信第三方,针对提现和追踪的效率问题提出了一个基于可信第三方的电子现金方案.特别是在提现和追踪双次提现方面,计算代价基本与n成线性相关.同时,通过可信第三方的认证作用,减少了原来系统内部交互中存在的零知识证明计算.

1 预备知识

本节主要介绍在我们方案中使用到的双线性群、承诺协议、密码学难题假设以及与可分电子现金相关的一些数据结构.

1.1 双线性群

设q为素数,G_1,G_2为2个阶为q的乘法循环群.通常称映射$e:G_1\times G_2\rightarrow G_T$为1个双线性映射,如果$e$满足下列性质:

1)双线性.对于任意的$g\in G_1$,$\tilde{g}\in G_2$以及a,$b\in\mathbb{Z}_q$,满足$e(g^a,\tilde{g}^b)=e(g,\tilde{g})^{a\times b}$.

2)非退化性.存在$g\in G_1$,$\tilde{g}\in G_2$,使得$e(g,\tilde{g})\neq 1_{G_T}$.

3)可计算性.存在有效的算法对任意$g\in G_1$,$\tilde{g}\in G_2$计算出$e(g,\tilde{g})$的值.

1.2 计算难题假设

离散对数问题:设$G=\langle g\rangle$,是1个阶为q的有限循环群,给定$y\in G$,求解整数,使得$y=g^x$.一般地,如果认真选择q,这个问题则被认为是难解的,目前没有找到解决它的多项式时间算法.

计算DH假设(文献[8]):设$G=\langle g\rangle$,是1个阶为q的有限循环群.对于给定的三元组(g,g^a,g^b),其中$a,b\in\mathbb{Z}_q^*$,任何多项式时间算法$\mathcal{A}$仅能以可忽略的概率$Pr[\mathcal{A}(g,g^a,g^b)=g^{ab}]$计算$g^{ab}$.

1.3 同态承诺

承诺在密码安全领域被广泛使用,是一个2方参与交互的协议.承诺通常分为2个阶段:承诺阶段和打开阶段.在承诺阶段,承诺方将1个秘密消息用承诺函数计算后发送给接收方;在打开阶段,接收方打开承诺,并证明与第1阶段中被承诺的消息一致.在以上2个阶段过程中,承诺方案应当具备以下2个基本性质:

1)隐藏性.即在承诺被打开以前,任意恶意接收方不能获知关于被承诺秘密的任何信息.

2)绑定性.不诚实的承诺者不能改变承诺的内容,在承诺被打开的时候,被承诺的消息必须和最初的保持一致.

除以上2个基本性质之外,如果承诺函数满足$\mathrm{Com}(m_1+m_2)=\mathrm{Com}(m_1)\times\mathrm{Com}(m_2)$,那么该承诺方案具有同态性.本文中,我们采用Pedersen[9]的承诺方案,其步骤简单描述如下:

1)$\mathrm{Setup}(1^k)$.选取安全参数k作为输入,生成$(k+1)$位长的素数p,使得$p=2q+1$,同时q是素数.随机选择$\mathbb{Z}_q^*$的生成元g和某个群元素h.

2) $\mathrm{Com}(m\in\mathbb{Z}_p^*)$. 为了对消息 m 承诺,选择 1 个随机数 $r\in\mathbb{Z}_p^*$,输出结果为 $c=g^r h^m \bmod p$.

3) $\mathrm{Open}(c,r,m)$. 如果 $c=g^r h^m$,则输出 1;反之,则输出 0.

1.4 可分电子硬币的树结构

经典的电子硬币树形结构化最初由 Okamoto[10] 提出,并广泛采用并改进[11-12]. 在经典的构造中,可分电子硬币可由二叉树结构来定义. 例如在 1 棵 $(l+1)$层二叉树中第 i 层上的节点对应的面值即为 2^{l-i},选中其中某节点作为可分电子硬币,可以拆分成若干子节点进行消费,如图 1 所示:

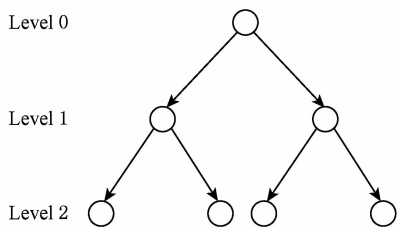


Fig. 1 Basic construction of e-cash.

图 1 经典的电子硬币树结构化

为了保证用户不能透支消费(即花费多于树结构表示的电子硬币面值),在使用过程中应当遵守以下 2 条基本规则:1)如果节点 s 已经被使用,那么节点 s 的祖先节点和子孙节点就不能再使用;2)每个节点不能重复使用.

1.5 双次消费(或重复消费)问题

电子现金毕竟不是实物,而只是一种特殊的数字存在,因此容易被复制,有可能会被恶意分子多次使用,这就是双次消费问题. 双次消费问题是电子现金方案中不得不考虑的问题,其问题本身是由电子现金的特殊背景所决定的. 考虑到可分电子硬币可以对应到树结构中的节点,我们将双次消费问题用树节点间的关系来描述,分 2 种情形:1)节点 s 和节点 s'对应的节点是同一个;2)节点 s 是节点 s'的祖先节点或节点 s 是节点 s'的子孙节点.

1) 基于路径的检测方式. 即通过分析节点路径来检测双次消费,比较直观,检测方需要计算 2 个节点路径信息(至少是二者的相对路径),例如文献[13]中的方案.

2) 基于共同叶节点的检测方式. 即通过分析是否存在共同叶节点来检测双次消费. 较上一种检测方式而言,该思路下,被消费节点的匿名性得到了加强,但如果不知道节点位置信息,就需要对叶节点逐个尝试,这个计算开销通常较大,例如文献[7]中的方案.

2 系统构建

我们在文献[7]方案的基础上,引入了可信第三方,提出了安全、实用、高效的电子现金系统,特别是在提现的协议中大大减少了系统计算代价.

在针对系统和协议时,我们用电子现金来描述. 在针对用树节点来表示具体的电子现金价值时,我们用电子硬币来描述.

2.1 系统成员

我们的电子现金系统包含 4 个组成部分:银行、用户、商家和可信第三方. 银行作为整个系统的运营核心,负责和处理用户和商家的各种交易行为,同时向用户出售电子现金,帮助商家进行提现操作;用户向银行购买虚拟货币,然后可用它们向商家购买商品或服务;商家首先需要向银行注册,通过出售虚拟商品或者其他服务来赚取用户的电子现金,通过将电子现金向银行提现来获得收益;可信第三方在系统中起到辅助功能,具有签名消息、认证用户和商家公钥、分担计算压力等作用. 四者之间的相互关系,如图 2 所示:

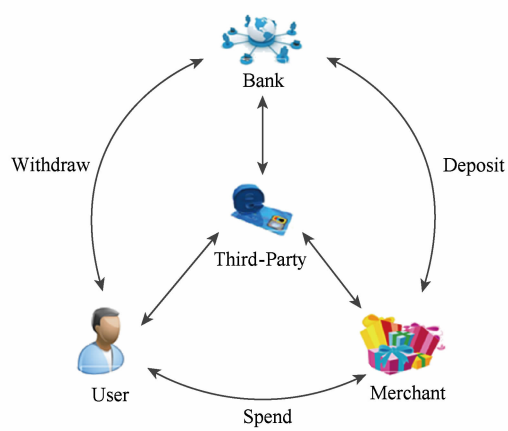


Fig. 2 Participants and interaction.

图 2 系统成员与交互

我们把可信第三方引入到系统中,有以下 3 点理由:

1) 分担平台的计算和存储压力;

2) 例如 PayPal、支付宝等支付平台作为交易的第三方,在现实中被广泛接受;

3) 在以往的方案中(包括 Canard 等人[7]),银行、用户和商家的公钥认证,甚至系统中一些重要的参数生成等都需要可信第三方参与.

2.2 系统算法

我们分别用 $\mathcal{B}$ 代表银行,$\mathcal{U}$ 代表用户,$\mathcal{M}$ 代表

商家,$\mathcal{T}$代表可信第三方.系统算法主要由3类组成:初始化算法、各成员公私钥生成算法和各成员交互算法,简要叙述如下:

1) Setup(1^k,V).该算法为初始化算法由银行运行,安全参数 k 作为输入,生成系统的公共参数(主要为树结构参数).V 作为树结构化电子现金的最大面值.

2) BKeygen( ).该算法为银行公私钥生成算法,由银行运行,生成公私玥($pk_{\mathcal{B}}$,$sk_{\mathcal{B}}$).同时,银行需要对在用户或商家注册登记时,为其构建对应的账户.此外银行需要对商家建立1张列表 L_{Deposit} 来记录被提现的电子硬币.

3) UKeyGen( ).该算法为用户公私钥生成算法,由用户运行,生成用户公私玥($pk_{\mathcal{U}}$,$sk_{\mathcal{U}}$).

4) MKeyGen( ).该算法为商家公私钥生成算法,由商家运行,生成商家公私钥($pk_{\mathcal{M}}$,$sk_{\mathcal{M}}$).

5) TKeyGen( ).该算法为可信第三方公私钥生成算法,由第三方运行,生成公私玥($pk_{\mathcal{T}}$,$sk_{\mathcal{T}}$).

6) Withdraw($\mathcal{U}$($pk_{\mathcal{T}}$, 2^l), $\mathcal{B}$($pk_{\mathcal{U}}$, r, g_s), $\mathcal{T}$($sk_{\mathcal{T}}$)).这是1个用户实现向支付工具购买电子硬币的算法.银行会为用户生成约定好的电子硬币,并用承诺隐藏信息.第三方会对其验证,并签名后发送给用户.其中,2^l 是电子硬币的金额,r 是承诺选择的随机数,g_s 是被承诺的节点特征值.

7) Spend($\mathcal{U}$($sk_{\mathcal{U}}$,C),$\mathcal{B}$($pk_{\mathcal{T}}$,$pk_{\mathcal{M}}$),$\mathcal{M}$($pk_{\mathcal{T}}$)).这是1个由用户、商家和银行参与,以实现电子硬币消费功能的算法.用户将与交易金额等值的电子硬币 C 发送给银行.银行对电子硬币 C 进行验证,如有效则将其发送给商家,同时对用户和商家账户更新.

8) Deposit($\mathcal{M}$($sk_{\mathcal{M}}$,$pk_{\mathcal{T}}$,C,Tag),$\mathcal{B}$($sk_{\mathcal{T}}$,$pk_{\mathcal{M}}$),$\mathcal{T}$($pk_{\mathcal{M}}$,$sk_{\mathcal{T}}$)).这是1个商家、银行和第三方参与并实现商家提现功能的算法.商家为电子硬币 C 添加附加值后发送给第三方,第三方校验并签名,然后交于银行.银行对电子硬币 C 进行有效性和双次提现的检测,如果检测到双次提现则调用 Identify 算法进行追踪.如果检测通过,则接受商家的提现请求.

9) Identify((C,Tag),(C',Tag')).这是1个银行确定恶意商家身份的算法.C 和 C' 是2个在双次提现检测中没有通过的电子硬币,该算法通过对二者的附加值 Tag 和 Tag' 进行分析与运算,得出关于恶意商家公钥的等式,通过对已注册的商家的公钥逐个遍历,找到满足等式的公钥,最终确认恶意商家的身份.

3 系统构造与协议分析

本节具体阐述电子现金的树结构构建,并分析系统主要协议的工作过程与原理.另外,我们采用 DES 加密方案和 RSA 签名方案,在3.2节、3.3节和3.4节中用 $\text{Enc}_{\text{DES}}(\cdot,\cdot)$ 和 $\text{Sign}_{\text{RSA}}(\cdot,\cdot)$ 表示.

3.1 初始化

我们的方案继承了文献[7]中树结构的构造.在初始化阶段生成1个可分电子硬币的树结构,其中节点分为2类:叶节点和内部节点.

结合双线性群,对树结构中的每个节点和每条路径设置相应的特征值.

令(G_1,G_2,G_T,e)是素数阶 p 满足双线性群条件的构造,g 为 G_1 的生成元,$\tilde{g}$ 为 G_2 的生成元.设 $g_T=e(g,\tilde{g})$,同时 g_T 是 G_T 的生成元.设 H:$\{0,1\}^*\rightarrow\mathbb{Z}_q$ 是1个抗碰撞的散列函数.

我们定义:内部节点 s 的特征值为 $g_s\in G_1$,叶节点 f 的特征值为 $X_f\in G_T$,从内节点 s 到叶节点 f 的路径 $path$ 的特征值为 $\tilde{g}_{s\mapsto f}\in G_2$.这些特征值通过以下步骤生成:

1) 对每一个内部节点 s,生成随机数 $r_s\in\mathbb{Z}_q$,并计算 $g_s\leftarrow g^{r_s}$;

2) 对每一个叶节点 f,生成随机数 $l_f\in\mathbb{Z}_q$;

3) 如果叶节点 f 是内部节点 s 的子孙,对这样的路径 $path(s\mapsto f)$,计算 $\tilde{g}_{s\mapsto f}=\tilde{g}^{l_f/r_s}$.

综合上述,对于每一个叶节点 f,任意1个其祖先节点 s 都满足 $e(g_s,\tilde{g}_{s\mapsto f})=X_f$,如图3所示:

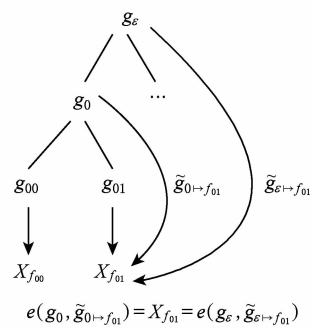


Fig. 3 Construction of e-coin.

图3 电子硬币的树结构

以上参数均有银行生成,包括双线性群(G_1,G_2,G_T,e)以及相应的生成元 g,$\tilde{g}$、散列函数 H,还包括树结构的各项参数:内部节点的特征值$\{g_s$,s 为内部节点$\}$,路径的特征值$\{\tilde{g}_{s\mapsto f}$,s 为内部节点,f 为叶节点,且是 s 的子孙$\}$.其中散列函数未公开,可被商家调用.

3.2 Withdraw 协议

在 Withdraw 协议中主要参与方为用户、银行和第三方.

Withdraw 协议实现用户通过现金购买、向银行购买等值电子现金的功能. 我们以用户选择购买金额 $m=2^l$ 为例(如果用户选的的金额不是 2 的幂次,则可将金额转化成若干个 2 的幂次之和来表示)描述该协议. 银行根据购买金额选择表示相应面值的节点 s,并获得该节点的特征值 g_s. 利用 1.3 节中提到的承诺方案,对特征值 g_s 生成承诺,并将承诺中用到的随机数 r 和 g_s 加密(在银行与第三方之间进行加密解密). 另外,再生成 1 个随机数 x,计算 $t_s \leftarrow g_s^x$ 作为此次购买的标记. 结合上述参数,生成电子硬币. 第三方校验银行生成的电子硬币,确认其价值与用户购买的一致,并对其签名. 完整 Withdraw 协议流程如下:

Withdraw 协议(参与方:用户、银行、第三方).

实现功能:用户向银行购买面值为 $m=2^l$ 的电子现金.

1) 银行操作

① 从树结构上选取面值为 m 的节点 s,并获得其特征值 g_s;

② 承诺加密 $\mathrm{Com}(g_s)=g^{g_s}h^r$;

③ $Enc_{\mathrm{DES}}(g_s,r)$;

④ 生成随机数 $x\in\mathbb{Z}_p^*$,计算 $t_s\leftarrow g_s^x$;

⑤ 生成用户的电子现金,$coin\leftarrow(\mathrm{Com}(g_s),t_s,Enc_{\mathrm{DES}}(g_s,r))$,$UAccount=UAccout\cup\{coin\}$. /* $UAccount$ 表示用户账户 */

2) 第三方操作

① 接受 $coin$,获得 $\mathrm{Com}(g_s)$ 和 t_s,解密获取 g_s 和 r;

② 如果 $\mathrm{Open}(\mathrm{Com}(g_s),r,g_s)=1$,则 $\sigma(coin)=Sign_{\mathrm{RSA}}(sk_T,coin)$;

③ 反之,则协议停止;

④ 将$(coin,\sigma(coin))$发送给用户.

3) 用户操作

① 如果 $Verify(pk_T,\sigma,coin)=1$,接受 $coin$;

② 反之,协议停止.

3.3 Spend 协议

在 Spend 协议中,主要的参与方为用户、银行和商家.

Spend 协议实现用户用电子现金向商家购买商品的功能. 首先用户选择 1 个未使用过的电子硬币 $coin$,发送给银行. 银行将对其做 2 项检测:1)检验该电子硬币是否有效;2)该电子硬币是否确实存在于该用户账号. 如果通过检测,则银行从用户账号的 $coin$ 转到商家账户下,同时将这 $coin$ 发送给商家;商家接收后,如验证无误,就实行交易. 完整 Spend 协议流程如下:

Spend 协议(参与方:用户、商家和银行).

实现功能:根据交易协议,商家获得用户的电子现金.

1) 用户操作

将$(coin,\sigma(coin))$发送给银行.

2) 银行操作

① 接收$(coin,\sigma(coin))$;

② 如果 $Verify(pk_T,\sigma,coin)=1$,接受$(coin,\sigma(coin))$;

③ 反之,协议停止;

④ 如果 $coin\in UAccount$,允许交易发生 ;

⑤ 反之,协议停止;

⑥ $UAccount=UAccount-\{coin\}$;

/* 表示从账户中删除 */

⑦ $MAccount=MAccount\cup\{coin\}$;

/* $MAccount$ 表示商家账户 */

⑧ 将$(coin,\sigma(coin))$发送给商家.

3) 商家操作

① 如果 $Verify(pk_T,\sigma,coin)=1$,则接受;

② 执行具体交易内容;

③ 反之,则协议停止.

3.4 Deposit 协议

在 Deposit 协议中主要的参与方为商家、银行和第三方.

Deposit 协议实现商家通过电子现金换取真实货币的功能. 首先供货商选择用来提现的电子货币 $coin$(与之前协议中的构造一致),并计算 $v\leftarrow H(info)$,H 是银行公开的散列函数,$info$ 是有关提现的信息,例如数额、时间、商家个人信息等;接着生成附加值 $Tag\leftarrow pk_{\mathcal{M}}^{v}\times t_s^{sk_{\mathcal{M}}}$,其中 t_s 是 Withdraw 协议中对购买的标记,记录在 $coin$ 中;商家将以上参数以及 $coin$ 本身发送给第三方,第三方确认其正确生成,对附加值签名,发送给银行. 银行需要对电子硬币做 2 项验证:1)是否有效,同 Withdraw 的一样;2)是否重复提现.

列表集合 L_{Deposit} 是银行对商家提现过的电子硬币的记录,设 $L_{\mathrm{Deposit}}=\{(g_{s_1},t_{s_1},Tag_1),\cdots,(g_{s_n},t_{s_n},Tag_n)\}$,其集合元素由 3 部分组成:电子硬币对应节点的特征值、购买标记和附加值. 银行收到由第三

方传来的数据 $z=(coin,\sigma(coin),Tag,\sigma(Tag))$,通过解密获取节点特征值、购买标记和附加值,生成三元组 $record_s=(g_s,t_s,Tag\leftarrow pk_{\mathcal{M}}^{v}\times g_s)$,同时银行可以通过 g_s 确定其对应的节点 s. 双次提现检测分以下步骤:

1) 从 L_{Deposit} 中挑选 $t_{s_i}=t_s$ 的三元组,被挑选出的三元组来源于同一次购买.

2) 从挑选结果中选取 1 个三元组,不妨记为 $record_{s'}=(g_{s'},t_{s'},Tag'\leftarrow pk_{\mathcal{M}}^{v'}\times g_{s'})$,银行可以从 $g_{s'}$ 确定其对应的节点 s'.

3) 根据节点 s 和 s' 的位置,判断此次提现行为是否为双次提现(即判断节点 s 是否是 s' 的祖先节点或子孙节点).

4) 重复步骤 2),3),逐个检测从步骤 1)中挑选出来的三元组,直到全部检测通过,或者出现构成双次提现的三元组 $record_{s'}=(g_{s'},t_{s'},Tag')$ 与 $record_s=(g_s,t_s,Tag)$,则进一步进行 Identify 协议.

如果没有检测到双次提现,银行接受商家的提现请求. 从商家账户中扣去相应金额,同时将这次的三元组 $record_s=(g_s,t_s,Tag\leftarrow pk_{\mathcal{M}}^{v}\times g_s^{sk_{\mathcal{M}}})$,添加进 L_{Deposit}. 完整 Deposit 协议流程如下:

Deposit 协议(参与方:商家、银行、第三方).

实现功能:商家用电子现金 $coin$,向银行提现.

1) 商家操作

① $v\leftarrow H(info)$;

② 从 $coin$ 中获得 t_s,$Tag=pk_{\mathcal{M}}^{v}\times t_s^{sk_{\mathcal{M}}}$;

③ 将$(coin,\sigma(coin),Tag)$发送给可信第三方.

2) 可信第三方操作

① 接收$(coin,\sigma(coin),Tag)$,确认 Tag 是否正确生成,如是则继续操作;

② $\sigma(Tag)=Sign_{\text{RSA}}(sk_{\mathcal{T}},Tag)$;

③ 生成 $z\leftarrow(coin,\sigma(coin),Tag,\sigma(Tag))$;

④ 将 z 发送给银行.

3) 银行操作

① 接收 z,验证 $\sigma(coin)$和 $\sigma(Tag)$,如正确无误则继续操作;

② 从 z 中获取 g_s,t_s,Tag,生成 $record_s=(g_s,t_s,Tag)$,并找到 g_s 对应的节点 s;

③ 在列表 $L_{\text{Deposit}}=\{(g_{s_1},t_{s_1},Tag_1),\cdots,(g_{s_n},t_{s_n},Tag_n)\}$中找出满足 $t_{s_i}=t_s$ 的三元组,$0\leqslant i\leqslant n$;

④ 取步骤 3)中 1 个三元组,记为 $record_{s'}=(g_{s'},t_{s'},Tag'\leftarrow pk_{\mathcal{M}}^{v'}\times g_{s'})$,获取 $g_{s'}$,找出 $g_{s'}$ 对应的节点 s';

⑤ 如果通过节点 s 和 s' 的位置,判定 $coin$ 和 $coin'$ 为重复提现,则进行 Identity 协议;反之重复步骤 4),直到遍历完;

⑥ 如没有出现重复提现,则接受商家提现;

⑦ $MAccount=MAccount-\{coin\}$;

⑧ $L_{\text{Deposit}}=L_{\text{Deposit}}\cup\{coin\}$.

3.5 Identify 协议

当检测到双次提现时,Identify 协议将会帮我们找出恶意商家.

现在,假定 $record_{s'}=(g_{s'},t_{s'},Tag')$与 $record_s=(g_s,t_s,Tag)$构成双次消费,即 g_s 对应的节点 s 和 $g_{s'}$ 对应的节点 s' 有公共叶节点,不妨设为 f,根据 4.1 节初始化中的结构性质,我们有等式 $e(g_s,\widetilde{g}_{s\mapsto f})=e(g_{s'},\widetilde{g}_{s'\mapsto f})$. 设 $T=e(Tag,\widetilde{g}_{s\mapsto f})$,$T'=e(Tag',\widetilde{g}_{s'\mapsto f})$,则:

$$T\times T'^{-1}=e(pk_{\mathcal{M}}^{v}\times g_s^{sk_{\mathcal{M}}},\widetilde{g}_{s\mapsto f})\times e(pk_{\mathcal{M}}^{v'}\times g_{s'}^{sk_{\mathcal{M}}},\widetilde{g}_{s'\mapsto f})^{-1}=e(pk_{\mathcal{M}}^{v},\widetilde{g}_{s\mapsto f})\times e(g_s^{sk_{\mathcal{M}}},\widetilde{g}_{s\mapsto f})\times e(pk_{\mathcal{M}}^{-v'},\widetilde{g}_{s'\mapsto f})\times e(g_{s'}^{sk_{\mathcal{M}}},\widetilde{g}_{s'\mapsto f})^{-1}=e(pk_{\mathcal{M}},\widetilde{g}_{s\mapsto f}^{v}\times\widetilde{g}_{s\mapsto f}^{-v'}),$$

其中,$e(g_s^{sk_{\mathcal{M}}},\widetilde{g}_{s\mapsto f})\times e(g_{s'}^{sk_{\mathcal{M}}},\widetilde{g}_{s'\mapsto f})^{-1}=1$ 可以消去. r 和 r' 是由交易相关的信息通过散列函数 H 唯一确定,以及 $\widetilde{g}_{s\mapsto f}$,$\widetilde{g}_{s'\mapsto f}$ 和 $T\times T'^{-1}$ 这些参数银行都可以获取或生成. 剩下的工作就是对注册在案的商家,通过其公钥逐个排查,直到找到符合上述等式的公钥,即我们要找的恶意商家.

3.6 对相关协议的补充

在 Deposit 过程中针对商家发送的 Tag 值是否正确计算,第三方需要用零知识证明:

$$PK(sk_{\mathcal{M}}|Tag=pk_{\mathcal{M}}^{v}\times g_s^{sk_{\mathcal{M}}}).$$

对于可分现金,现实使用时会发生以下的情况:用价值为 m 的电子硬币 $coin$ 去购买价值为 m' 的商品,其中 $m>m'$. 此时,银行需要对该用户做出"找零"操作:从用户账户扣除价值为 m 的电子硬币,同时生成价值为 m' 和 $m-m'$ 的电子硬币 $coin'$ 和 $coin''$,前者发给商家,后者返还给用户. 对于"找零"操作,新生成的电子硬币分配过程,基本上与 Withdraw 协议中一样,但是其购买标记要与原来的一样,即例子中电子硬币 $coin'$ 和 $coin''$ 的购买标记与 $coin$ 相同.

4 安全分析

对于电子现金的系统存在着一些基本安全要求,例如一致性、不可重用性等,接下来我们就这些要求,具体分析方案的安全性.

4.1 一致性

该特性要求银行不会接受恶意用户伪造的电子硬币. 当用户从银行处购买电子硬币,例如获得 $coin \leftarrow (\mathrm{Com}(g_s), t_s, \mathrm{Enc}(g_s, r))$,其中 $t_s \leftarrow g_s^x$. 其中的 DES 机密并未参与到具体交易中,那么恶意用户试图伪造 $coin$,只需伪造其包含的承诺 $\mathrm{Com}(g_s)$ 与购买标记 t_s 即可.

关于承诺,我们设 $\mathrm{Com}(g_s) = g^{g_s} h^r$ 是真实的值,我们设计响应-挑战实验如下:

1) 敌手生成关于 $g_{s'}$ 的承诺;

2) 敌手如果能伪造 $g_{s'}(g_s \neq g_{s'})$,使得 $\mathrm{Com}(g_s)$ 与 $\mathrm{Com}(g_{s'})$ 无法区分,我们则认为敌手获胜.

敌手获胜的概率为

$$Pr[Advwin] = Pr[\mathrm{Com}(g_s) = \mathrm{Com}(g_{s'}) \wedge g_s \neq g_{s'}] = Pr[g^{g_s} h^r = g^{g_{s'}} h^{r'} \wedge g_s \neq g_{s'}],$$

且 $g^{g_s} h^r / h^{r'} = g^{g_{s'}} \bmod p$.

于是问题归结到求解离散对数问题,求解 $g_{s'}$ 的难度等同于求解离散对数问题的难度. 因此,根据难题假设,敌手获胜的概率是可忽略的.

对于购买标记的伪造,也可以归结到离散对数问题,敌手获胜的概率是可忽略的.

4.2 不可重用性

我们的方案中只做到用户的不可重用性,即用户不能消费已经消费过的电子硬币.

用户每次消费结束后,银行都会对用户的账户进行相应的更新. 在用户消费是时候,银行会从用户账户里检测将被消费的电子现金,如果没有找到,则说明该电子现金属于双次重用.

4.3 匿名性

匿名性是电子现金系统的基本要求,即要求在系统正常运行中用户信息不回泄露.

对于用户而言,在交易过程中电子硬币被承诺隐藏. 基于承诺的隐藏性,在承诺没有被打开的情况下,其他人不能从承诺中获取任何内部信息. 因此用户的信息自然也得到了保证.

对商家而言,除了被承诺的电子现金这部分外,电子现金的附加值 Tag 中包含了商家的信息. $Tag \leftarrow pk_{\mathcal{M}}^{v} \times g_s^{sk_{\mathcal{M}}}(v \leftarrow H(info))$. 其中商家私钥 $sk_{\mathcal{M}}$ 是不公开,其他参数都是公开的或者可由公开的算法计算得到. 如果想获得商家私钥 $sk_{\mathcal{M}}$,根据离散对数难题,任何多项式时间算法的地方只能以可忽略概率获取用户私钥信息. 因此方案也确保了商家私钥不泄露.

4.4 可追踪性

在系统中正常情况下对用户隐私是保密的,但是如果存在恶意用户,违反规则损害他人利益,我们则要求系统能把恶意用户追踪出来,这便是可追踪性.

在 3.5 节 Identify 协议中我们已经详细介绍了存在双次提现情况下,如何找出恶意商家的方法和过程.

5 性能分析

本文将我们的方案与文献[7]中的方案做了比较,如表 1 和表 2 所示. 文献[7]的方案是目前唯一在随机预言机模型和标准模型下达到安全并且相对高效的方案. 所有的比较均在离线条件下进行,在表 1 中文献[7]的计算次数是以用户的角度计算的.

Table 1 Efficiency Comparison of Schemes

表 1 方案间的效率比较

Scheme	Ref[7]	Ours
Public Parameter	$(n+2)pk+(2^{n+2}+3)\lvert G_1\rvert+1\lvert G_2\rvert+(2^{n+1}-1)\lvert \mathrm{Sign}\rvert$	$2^{n+1}\lvert G_1\rvert+1\lvert \mathrm{Sign}\rvert$
Withdraw Computions	1Sign	1Sign
Size of Coin	$2\lvert p\rvert+1\lvert \mathrm{Sign}\rvert$	$2\lvert p\rvert+1\lvert \mathrm{Com}\rvert+1\lvert \mathrm{Sign}\rvert$
Spend Computions	$NIZK$(2Exp+2Sign)+3Exp+1Sign	1Sign
Deposit Computions	2^n pair	$2n$ Search
Identify Computions	2^n pair	1 pair

关于对表 1 的说明:数据结构都是从用户的角度(包括商家)出发的. 其中电子硬币的最大面值设为 2^n,用户购买(Withdraw)或提现(Deposit)的面值为 2^l. $\lvert G_1\rvert$ 表示群 G_1 上的元素大小;$\lvert G_2\rvert$ 表示群 G_2 上的元素;Exp 表示 1 次指数上的运算;pair 表示 1 次配对计算;Sign 表示 1 次签名操作,表示承诺操作或打开操作. $NIZK$(Exp)表示用零知识证明 1 个指数等式的代价,$NIZK$(pair)表示用零知识证明 1 次配对计算正确的代价,$NIZK$(Sign)表示用零知识证明 1 个有效签名的代价. Search 表示树结构中访问父亲或儿子节点的操作.

公共参数主要与树结构(节点以及路径的特征

值)有关,2 个方案的树结构大部分相同. 前者方案中每个节点对应着(g_s, h_s)对,后者没有生成 h_s.

在实际应用中,我们要求 Withdraw 和 Spend 的操作是高效、快速的,因为这些操作直接影响到用户的体验. 文献[7]和我们的方案都将 Withdraw 和 Spend 操作的计算量定为常数级的. 相对于其他操作而言,零知识证明在实际应用中计算量很大,减少零知识证明能有效提高系统效率. 我们通过引进可信第三方的支付工具,减少了零知识证明的计算.

在 Deposit 协议中,文献[7]中存在着 1 张列表,该列表记录着所有已经被提现的电子硬币的信息. 文献[7]中因为关于电子硬币 *coin* 对应的节点是不可知的,知道的仅仅是其所在层数. 为了验证其是否有效,需要逐个遍历其叶节点. 就每一个叶节点,与文献[7]列表中所有的 *coin* 进行配对比较. 计算复杂度与提现电子硬币对应节点的叶节点数目有关,与列表的大小有关. 我们方案中银行可以解密得到电子硬币对应节点的特征值,进而获取节点的位置. 通过位置信息判断某节点是否为另一个节点祖先或者子孙,只需要至多 $2n$ 次操作.

在 Identify 协议中,还是因为文献[7]中电子硬币 *coin* 对应的节点是不可知的,只能对所有叶节点逐个检测,因此大量计算消耗在了查找共同叶节点上. 我们的方案中,因为知道电子硬币对应节点的位置,只需 1 步即可查找到共同叶节点,减少了大量不必要的计算.

6 结束语

可分电子现金方案中,如何高效、可靠且不泄露用户隐私地进行双次消费检测是 1 个难点. 我们通过引入可信第三方,通过最基本的节点位置信息,即可高效进行双次提现检测;同时继承 Canard 等人[7]方案中通过 2 节点的共同叶节点来追踪恶意用户的方法,并且叶节点的选择是确定有效的,不再是原来的盲目选择. 通过以上 2 点,提高了原方案的效率,特别是在处理大金额的电子现金问题上.

在未来,我们需要考虑在效率不变的情况下,如何将可信第三方从系统中除去. 另外,能否找到一种新的且实现起来高效的结构取代传统的树结构,这一直是一个公开的问题. 这可能会是解决一直以来电子现金方案低效、计算复杂度高的有效办法,值得我们进一步研究.

参 考 文 献

[1] Chaum D. Blind signatures for untraceable payments [C] //Proc of Advances in Cryptology. Berlin: Springer, 1983: 199-203

[2] Stadler M, Piveteau J M, Camenisch J. Fair blind signatures [C] //Proc of Advances in Cryptology (EUROCRYPT'95). Berlin: Springer, 1995: 209-219

[3] Camenisch J, Maurer U, Stadler M. Digital payment systems with passive anonymity-revoking trustees [C] //Proc of Computer Security (ESORICS'96). Berlin: Springer, 1996: 33-43

[4] Frankel J A, Rose A K. Currency crashes in emerging markets: An empirical treatment [J]. Journal of International Economics, 1996, 41(3): 351-366

[5] Okamoto T, Ohta K. Universal electronic cash [C] //Proc of Advances in Cryptology (CRYPTO'91). Berlin: Springer, 1992: 324-337

[6] Chen Kai, Zhang Yuqing, Xiao Guozhen. A divisible-cash system based on probabilistic audit [J]. Journal of Computer Research and Development, 2000, 37(6): 752-757 (in Chinese)
(陈凯,张玉清,肖国镇. 基于概率验证的可分电子现金系统[J]. 计算机研究与发展, 2000, 37(6): 752-757)

[7] Canard S, Pointcheval D, Sanders O, et al. Divisible e-cash made practical [C] //Proc of Public-Key Cryptography (PKC 2015). Berlin: Springer, 2015: 77-100

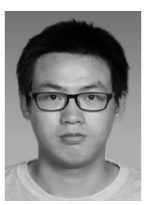
[8] Diffie W, Hellman M E. New directions in cryptography [J]. IEEE Trans on Information Theory, 1976, 22(6): 644-654

[9] Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing [C] //Proc of Advances in Cryptology (CRYPTO'91). Berlin: Springer, 1992: 129-140

[10] Okamoto T. An efficient divisible electronic cash scheme [C] //Proc of Advances in Cryptology (CRYPT0'95). Berlin: Springer, 1995: 438-451

[11] Nakanishi T, Sugiyama Y. Unlinkable divisible electronic cash [C] //Proc of Information Security 2000. Berlin: Springer, 2000: 121-134

[12] Canard S, Gouget A. Divisible e-cash systems can be truly anonymous [C] //Proc of Advances in Cryptology (EUROCRYPT 2007). Berlin: Springer, 2007: 482-497



Yu Yulei, born in 1990. Master candidate in Shanghai Jiao Tong University. His main research interests include cryptography and trusted computing.

Dong Xiaolei, born in 1971. Received her PhD degree in Harbin Institute of Technology. Currently distinguished professor and PhD supervisor in Software Engineering Institute, East China Normal University. Her main research interests include number theory, cryptography and trusted computing.

Cao Zhenfu, born in 1962. Received his PhD degree in Harbin Institute of Technology. Currently distinguished professor and PhD supervisor in Software Engineering Institute, East China Normal University. His main research interests include number theory, cryptography, trusted network and information security.

2013 年《计算机研究与发展》高被引论文 TOP10

排名	论文信息
1	孟小峰，慈祥. 大数据管理:**概念、技术与挑战**[J]. 计算机研究与发展，2013，50(1)：146-169 Meng Xiaofeng and Ci Xiang . Big Data Management: Concepts, Techniques and Challenges [J]. Journal of Computer Research and Development, 2013, 50(1): 146-169
2	傅颖勋，罗圣美，舒继武. **安全云存储系统与关键技术综述**[J]. 计算机研究与发展，2013，50(1)：136-145 Fu Yingxun, Luo Shengmei, and Shu Jiwu. Survey of Secure Cloud Storage System and Key Technologies. Journal of Computer Research and Development, 2013, 50(1): 136-145
3	李建中，刘显敏. **大数据的一个重要方面:数据可用性**[J]. 计算机研究与发展，2013，50(6)：1147-1162 Li Jianzhong and Liu Xianmin. An Important Aspect of Big Data: Data Usability [J]. Journal of Computer Research and Development, 2013, 50(6): 1147-1162
4	刘大有，陈慧灵，齐红，杨博. **时空数据挖掘研究进展**[J]. ，2013，50(2)：225-239 Liu Dayou, Chen Huiling, Qi Hong, and Yang Bo. Advances in Spatiotemporal Data Mining. Journal of Computer Research and Development, 2013, 50(2): 225-239
5	廖彬，于炯，孙华，年梅. **基于存储结构重配置的分布式存储系统节能算法**[J]. 计算机研究与发展，2013，50(1)：3-18 Liao Bin, Yu Jiong, Sun Hua, and Nian Mei. Energy-Efficient Algorithms for Distributed Storage System Based on Data Storage Structure Reconfiguration [J]. Journal of Computer Research and Development, 2013, 50(1): 3-18
6	贾冬艳，张付志. **基于双重邻居选取策略的协同过滤推荐算法**[J]. 计算机研究与发展，2013，50(5)：1076-1084 Jia Dongyan and Zhang Fuzhi. A Collaborative Filtering Recommendation Algorithm Based on Double Neighbor Choosing Strategy [J]. Journal of Computer Research and Development, 2013, 50(5): 1076-1084
7	武建佳，赵伟. **WInternet:从物网到物联网**[J]. 计算机研究与发展，2013，50(6)：1127-1134 Wu Jianjia and Zhao Wei. WInternet: From Net of Things to Internet of Things [J]. Journal of Computer Research and Development, 2013, 50(6): 1127-1134
8	余凯，贾磊，陈雨强，徐伟. **深度学习的昨天、今天和明天**[J]. 计算机研究与发展，2013，50(9)：1799-1804 Yu Kai, Jia Lei, Chen Yuqiang, and Xu Wei. Deep Learning: Yesterday, Today, and Tomorrow [J]. Journal of Computer Research and Development, 2013, 50(9): 1799-1804
9	熊金波，姚志强，马建峰，李凤华，李琦. **基于行为的结构化文档多级访问控制**[J]. 计算机研究与发展，2013，50(7)：1399-1408 Xiong Jinbo, Yao Zhiqiang, Ma Jianfeng, Li Fenghua, and Li Qi. Action-Based Multilevel Access Control for Structured Document [J]. Journal of Computer Research and Development, 2013, 50(7): 1399-1408
10	张振海，李士宁，李志刚，陈昊. **一类基于信息熵的多标签特征选择算法**[J]. 计算机研究与发展，2013，50(6)：1177-1184 Zhang Zhenhai, Li Shining, Li Zhigang, and Chen Hao. Multi-Label Feature Selection Algorithm Based on Information Entropy [J]. Journal of Computer Research and Development, 2013, 50(6): 1177-1184

数据来源:中国知网;统计日期:2014-12-18