可直接计算的高效的可分电子现金系统

刘文远[1],张江霄[1],胡庆华[1],谷秀芝[2]
(1. 燕山大学信息科学与工程学院,河北秦皇岛 066004; 2. 重庆交通大学,重庆 400074)

摘 要: 针对现有的基于可信第三方的可分电子现金存在用户花费电子现金效率低、用户花费时要从根节点一层一层计算的缺点,在无可信第三方的基础上,首次将节点可直接计算与可再分的方法引入到离线可分电子现金中,提出了一种新型的无可信第三方的离线可分电子现金系统. 该协议在基于同一棵二叉树时,用户所能花费的电子现金总额是原来的 n 倍(n 是二叉树层数),由同一电子现金分出的不同节点具有不可链接性,而且花费二叉树上任一节点所做的计算量是一样的,从而提高系统的整体效率.

关键词: 电子现金; 可分性; 不可链接性; 可直接计算
中图分类号: TP393 **文献标识码**: A **文章编号**: 0372-2112 (2009) 02-0367-05

Divisible E-Cash System with Direct Computation and Efficiency

LIU Wen-yuan[1], ZHANG Jiang-xiao[1], HU Qing-hua[1], GU Xiu-zhi[2]
(1. *Information Science and Engineering Institute, Yanshan University, Qinhuangdao, Hebei* 066004, *China*;
2. *Chongqing Jiaotong University, Chongqing*, 400074, *China*)

Abstract: There exist some drawbacks such as low efficiency and computation lay by lay from a root node when the user spends E-cash in the divisible electronic cash (E-cash) based on TTP. Based on without TTP, the concept of direct computation and division again was firstly introduced to off-line divisible E-cash, a new off-line divisible E-cash system without trusted third party was presented. In the new system, the E-cash total amount spent by user was times than before (is the binary tree layer), the different node divided from the same E-cash was unlinkable, moreover the computation of expending on any node was the same, thus we can enhance the overall efficiency of the system.

Key words: e-cash; divisibility; unlinkability; direct computation

1 引言

电子现金支付系统是实现金融电子化的重要支付手段. 由于离线可分电子现金系统无需银行在线验证,从而有较高的系统效率,而且它还有另一个重要的性质—可分性,这样用户就能花费任意面额的电子现金,直到用户所取电子现金的总额.

由于离线可分电子现金所具有的优点,自从 Okamoto 和 Ohta[1]第一次利用二叉树实现了可分电子现金以后,现已有很多人对其进行研究,Okamoto[2]构建了一个可分电子现金,但是同一电子现金分出的电子现金是可链接的. Chan 和 Frankel[3]针对文献[2]构建电子现金的低效率性,对其进行改进,从而提高文献[2]的效率. Nakanishi 和 Shiota[4]真正实现了第一个不可链接的可分电子现金,但是在零知识证明时利用了切割选择技术,因此协议的效率不是很高. 彭冰、洪帆[5]等人在零知识证明签名和强 RSA 问题的基础上构建了基于 K 叉树的可分电子现金,减少了树的层数,从而减少用户计算量. Canard 和 Gouget[6]构建了一个无可信第三方的可分电子现金,但存在银行检验用户是否重复花费时计算量大、协议效率低的缺点. 而且上面的可分电子现金协议都存在用户花费时要从根节点一层一层计算的缺点,这样使得可分电子现金协议的整体效率不高.

本文在参考 Canard 和 Gouge 构建的无可信第三方的可分电子现金后,结合李梦东和杨义先[7]在无可信第三方下对电子现金匿名性控制的分析,构建了一个无可信第三方的可分电子现金协议. 第一次提出节点可直接计算与可再分的方法. 在基于同一棵二叉树时,用户所能花费的电子现金总额是原方案的 n 倍,同时在花费二叉树上任一节点时所做的计算量是一样的,从而提高了协议的效率. 而且从同一电子现金分割而得的不同节点具有不可链接性.

收稿日期:2008-03-11;修回日期:2008-09-25
基金项目:国家科技部高新技术计划(No. 2005EJ000017);国家电子信息发展基金及河北省信息产业发展计划(No. 2005035025);河北省自然科学基金(No. F2005000368)

2 符号描述与二叉树的构建

全文所涉及的有关符号：‖表示串的连接，$a \in_R B$ 表示从集合 B 中随机选择某一元素 a；Z_n^* 表示整数模 n 的乘法群；$h \in_R G$ 表示 h 是从群 G 中随机选择的生成元. $H: \{0,1\}^* \to \{0,1\}^{l_H}$ 为碰撞自由的单向散列函数.

可分的电子现金协议用一棵二叉树来表示，假如所花的金额是 $w = (l+1)2^l$，则对应一个 $l+2$ 层二叉树，有 2^l 个叶子节点，第 $l+2$ 层是一层“死”叶子节点，根节点的面值是 $w_0 = 2^l$，除了“死”叶子节点，其孩子节点所对应的电子现金的面值是其父节点的 1/2，对二叉树除了“死”叶子节点外的每个节点引入两个值 S 与 T.

例 1 定义的三层二叉树(图 1)

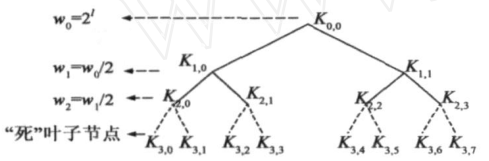

图1 三层二叉树

3 所需要的定义

定义 1[5] $y, g \in G_n$，称满足 $C = (y \| g \| g^s y^c \| m)$ 的二元组 $(c, s) \in \{0,1\}^k \times Z_n^*$ 为 y 关于 g 的离散对数知识对消息 $m \in \{0,1\}^*$ 的签名，即：$S_1\{x \mid y = g^x\}(m)$.

定义 2[8] $y_1, g, y_2, h \in G_n$，称满足 $C = H(y_1 \| g \| y_2 \| h \| g^s y_1^c \| h^s y_2^c \| m)$ 的二元组 $(c, s) \in \{0,1\}^k \times Z_n^*$ 为 y_1 关于 g 的离散对数知识 y_2 关于 h 的离散对数知识对消息 $m \in \{0,1\}^*$ 的签名，简记为：$S_2\{x \mid y_1 = g^x \wedge y_2 = h^*\}(m)$.

定义 3 强 RSA 问题[9] 设 $n = pq$ 为 RSA 模数，$(p = 2p' + 1, q = 2q' + 1$，且 p, p', q 都是素数$)$，元素 $u \in Z_n^*$，在 $e > 1$ 且 $q \nmid e$ 满足 $v^e \equiv u \bmod n$ 时，找一个二元组 (v, e) 是不可能的.

定义 4 Camenish-Lysyanskaya (CL 签名)[10] 协议是基于强 RSA 假设的，它是在一个用户与一个签名者之间进行，允许用户从签名者那里获得一个关于提问 C 的签名 σ(C 是具体关于 $(x_1, \cdots, x_l)$ 的提问)，但是签名者并不知道关于 C 的具体值. 签名者计算 $CLSign(C)$ 并发给用户，用户得到 $\sigma = sign(x_1, \cdots, x_l)$，且用户能通过 $Verif(\sigma, (x_1, \cdots, x_l)) = 1$ 验证签名是否正确. 在用户验证时，除了发给签名者必要的信息外，还要发给签名者知道关于 C 具体值 $(x_1, \cdots, x_l)$ 的知识签名：$PK\{\alpha_1, \cdots, \alpha_l, \beta: \beta = \text{sign}(\alpha_1, \cdots, \alpha_l)\}$.

4 节点可直接计算和节点可再分的方法

针对现有可分电子现金在用户花费时，需要从根节点一层一层计算的缺点，本文首次引入了节点可直接计算的方法和节点可再分的方法. 节点可直接计算的方法指用户在花费二叉树上的某一节点时，直接从根节点计算，这样使得用户在花费二叉树上的任一节点时所做的计算量相同，从而提高可分电子现金的效率；节点可再分的方法指在用户花费二叉树上分出的节点时，对现有的节点按原二叉树的方法进行分解，即二叉树上的节点既可以被直接花费，也可以在不满足花费要求时被用户分解后再花费，这样二叉树上的每个节点都可以被花费，从而提高可分电子现金的效率.

5 可分的电子现金的构造

5.1 开户协议

系统需要的参数：

G 是阶为 n 的群，在 G 中随机的选择生成元：$h_0, h_1, h_2 \in_R G$，$G_1 = \langle g_1 \rangle$ 是 Z_n^* 的阶为 o_1 的子群. 每个 $G_i = \langle g_i \rangle$ 必须为 $Z_{o_{i+1}}^*$ 的子群，由 o_1 计算出的 $2o_1 + 1, \cdots, 2o_{l+1} + 1$ 必须是素数，如果不是素数，则所有的数必须再由 o_1 重新计算，直到所有的数都是素数. 在各个群中随机的选择生成元如下：$g \in_R G$，$g_{1,0}, g_{1,1}, t_{1,0}, t_{1,1} \in_R G_1, \cdots, g_{l,0}, \cdots, g_{l,2^l-1}, t_{l,0}, \cdots, t_{l,2^l-1} \in_R G_l$，并且要保证他们各自相对于 $g_1, g_2, \cdots, g_l$ 的离散对数是不知道的.

然后银行、用户和商家计算各自的公私钥对 (sk_B, pk_B)、(sk_U, pk_U) 和 (sk_M, pk_M)，另外银行建立三个数据库：账户信息数据库 P、支付信息数据库 Q 和重复花费数据库 R.

5.2 取款协议

当用户从银行提取电子现金时，用户与银行要经过以下几步来完成取款：

(1) 用户→银行：(这一步用户可以提前计算) 用户随机的选 $s', r_U \in_R Z_n^*$，然后计算彼德森承诺[11]并发送给银行 $C' = h_0^{s'} h_1^u h_2^{r_U} (\bmod n)$ 和 pk_U，还有用户对自己知道的私钥和 C 对应秘密的零知识证明：

$$U = PK(\alpha, \beta, \gamma: pk_U = g^\alpha \wedge C' = h_0^\beta h_1^\alpha h_2^{r_U}) \quad (1)$$

(2) 银行→用户：银行验证零知识签名 U 的正确性，即验证等式(1)是否正确，若正确，银行选随机数 $r' \in_R Z_n^*$，并计算 $C = C' h_0^{r'} (\bmod n)$，并对电子现金进行 CL 签名[12]如下：$\sigma = C^{1/e} \bmod n = (h_0^{s'} h_1^u h_2^r h_0^{r'})^{1/e} \bmod n$，然后把 r', σ 传给用户. 并把得到的 C, pk_u, U, r', σ 存入数据库 P；

(3) 用户：用户计算 $s = s' + r' (\bmod n)$，并验证银行签名，即验证等式(2)是否正确：

$$\sigma^e \stackrel{?}{=} \left(h_0^s h_1^u h_2^r \right) \quad (2)$$

最后用户就能得到所提取的电子现金：

$C=(s,u,r,\sigma=\mathrm{Sign}(s,u,r))$.

5.3 花费协议

当用户想要从电子现金 C 中向商家花费价值 2^{l_1} 面额的电子现金时,就可以把 C 分割成一个或几个节点来花费,进行多次的花费协议就可以完成交易.本方案只解决花费某一节点情况,用户与商家要进行如下几步:

(1) 商家→用户:商家选随机数 r_M 并发给用户 $r_M\in_R Z_g^*$;

(2) 用户:用户计算 $R=H(pk_M\parallel r_M)$;

若用户花费的是由二叉树分出的节点,假设花费的是 K_{i_1,j_1} 分出的 $K_{i,j}$,随机的选择 $\tilde{g},\tilde{h}\in_R G_i,\tilde{g}_{i_1}\in_R G_{i_1},\tilde{g}_{i_1+1}\in_R G_{i+1},\cdots,\tilde{g}_i\in_R G_i$.用户经过算法 1:

输入:i,j

输出:$(\tilde{V}_0,\tilde{V}_{i_1},\cdots,\tilde{V}_i)$

$\tilde{r}\leftarrow Rand()$, $V\leftarrow g^s$, $\tilde{V}_0\leftarrow\tilde{g}^s\tilde{h}^r$

if $i=0$, return$(\tilde{V}_0,V)$

for $k=i_1$ to i

$\tilde{V}_k\leftarrow\tilde{g}_k^V$

$V\leftarrow g_{k,j}^V$

return $(\tilde{V}_0,\tilde{V}_{i_1},\cdots,V_i)$

然后用户计算:$LK=g_{i+1,2j-1}^V$, $RK=g_{i+1,2j}^V$,

$$S=LK\parallel RK,\ T=pk_U t_{i,j}^{V\cdot R}.$$

用户再利用 Fiat-Shamir 启发式算法[13]生成非交互的零知识证明,其中 $\sigma=\mathrm{Sign}(s,u,r)$ 向商家证明用户知道 $s,u,r,\tilde{V}_0=\tilde{g}^s\tilde{h}^r$ 向商家证明根节点是正确生成的,$\tilde{V}_{i+1}=\tilde{g}_{i+1}^{\alpha_{i+1,j2}}\wedge\tilde{V}_{i+1}=\tilde{g}_{i+1}^{\alpha_{i+1}}$向商家证明节点 $K_{i,j}$ 是正确生成的,$LK=g_{i+1,2j-1}^{\alpha_{i+1}}\wedge RK=g_{i+1,2j}^{\alpha_{i+1}}\wedge T=pk_U t_{i,j}^{R\cdot\alpha_i}$ 向商家证明所花费节点的 LK、RK 以及 T 是由用户正确计算得到的,则生成的零知识证明如下:

$$\begin{aligned}\Phi=PK\{&\sigma,s,u,r,\alpha_{i_1},\cdots,\alpha_i,\beta:\\ &\sigma=\mathrm{Sign}(s,u,r)\wedge\tilde{V}_0=\tilde{g}^s\tilde{h}^r\wedge\tilde{V}_{i_1}=\tilde{g}_{i_1}^s\wedge\tilde{V}_{i_1}\\ &=\tilde{g}_{i_1}^{\alpha_{i_1}}\wedge\tilde{V}_{i_1+1}=\tilde{g}_{i_1+1}^{\alpha_{i_1+1,j1}}\wedge\tilde{V}_{i_1+1}=\tilde{g}_{i_1+1}^{\alpha_{i_1+1}}\wedge\cdots\wedge\tilde{V}_{i+1}\\ &=\tilde{g}_{i+1}^{\alpha_{i+1,j2}}\wedge\tilde{V}_{i+1}=\tilde{g}_{i_1+1}^{\alpha_{i+1}}\wedge LK=\tilde{g}_{i+1,2j-1}^{\alpha_{i+1}}\wedge RK\\ &=\tilde{g}_{i+1,2j}^{\alpha_{i+1}}\wedge T=pk_U t_{i,j}^{R\cdot\alpha_i}\}\end{aligned}\tag{3}$$

若用户花费的是二叉树上的节点,假设花费的是 $K_{i,j}$,此时让算法 1 中的 $i_1=i$ 即可,同理,则生成的零知识证明为:

$$\begin{aligned}\Phi=PK\{&\sigma,s,u,r,\alpha_i,\beta:\sigma\\ &=\mathrm{Sign}(s,u,r)\wedge\tilde{V}_0=\tilde{g}^s\tilde{h}^r\wedge\tilde{V}_i=\tilde{g}_i^s\wedge\tilde{V}_i\\ &=\tilde{g}_i^{\alpha_i}\wedge\tilde{V}_{i+1}=\tilde{g}_{i+1}^{\alpha_{i+1,j2}}\wedge\tilde{V}_{i+1}=\tilde{g}_{i_1+1}^{\alpha_{i+1}}\wedge LK\\ &=\tilde{g}_{i+1}^{\alpha_{i+1}}\wedge LK=\tilde{g}_{i+1,2j-1}^{\alpha_{i+1}}\wedge RK\\ &=\tilde{g}_{i+1,2j}^{\alpha_{i+1}}\wedge T=pk_U t_{i,j}^{R\cdot\alpha_i}\}\end{aligned}\tag{4}$$

最后用户发给商家的电子现金是:

$C_1(2^{l_1},S,T,\Phi,R,\tilde{V}_0,\tilde{V}_i)$.

5.4 存款协议

当商家往银行存入电子现金时,商家只要发给银行 C_1,银行为了检测用户是否发生重复花费,需要经过以下几步检查:

(1) 银行检查零知识证明 Φ 有效性和它与 S、T 的一致性,若 C_1 是有效的电子现金,跳到下一步,否则银行拒绝存款,退出检查;

(2) 银行以所花费的节点为根节点,若所花费的是二叉树上的节点时,则与此节点对应的"死"叶子节点的值 $\tilde{S}_{k_j}$,即 $C_2(\tilde{S}_{k_j},S,T,\Phi,R,\tilde{V}_0,\tilde{V}_i)$,其中 $(1\le j\le 2^{l+1})$,若所花费的是二叉树上分出的节点时,则与此节点对应的"死"叶子节点的值 $\tilde{S}_{k_i}'$,即 $C_2'(\tilde{S}_{k_i}',S,T,\Phi,R,\tilde{V}_0,\tilde{V}_i)$.然后在数据库 Q 查找是否有相同的 $\tilde{S}_{k_j}$ 或 $\tilde{S}_{k_i}'$,若没有,跳到下一步,否则,跳到第四步.

(3) 银行检查两个电子现金中随机数 R 是否一样,若不一样,说明用户重复花费了,跳到下一步,否则说明商家对同一笔交易进行了重复存款,退出检查.

(4) 若用户重复花费的是同一个节点时,把同一节点的两次花费 $(2^{l_1},\tilde{S}_{k_j},S,T,\Phi,R,\tilde{V}_0,\tilde{V}_i)$ 和 $(2^{l_1},\tilde{S}_{k_j},S,T',\Phi',R',\tilde{V}_0',\tilde{V}_i')$ 存到数据库 R 中,设 $T_1=T$, $T_2=T'$,然后利用下面计算揭示非法用户的身份:

$$pk_U=\left(T_1^{R_2}/T_2^{R_1}\right)^{1/(R_2-R_1)}\tag{5}$$

若用户重复花费的是有父子关系的节点时,把 $(2^{l_1},\tilde{S}_{k_i}',S,T,\Phi,R,\tilde{V}_0,\tilde{V}_i)$ 和 $(2^{l_1},\tilde{S}_{k_i}',S',T',\Phi',R',\tilde{V}_0',\tilde{V}_i')$ 存到数据库 R 中,假设 S 是 S 的祖先,并且使 $T_1=T$, $T_2=T'$, $R_1=R$, $R_2=R'$,其中 $g_{l_2'+1,2}^{V_{l_2'}}$ 是由 S 在算法 1 中计算出来的与 S 对应的 V 值,然后利用下面计算揭示非法用户的身份:

$$\left\{(T_2)^{1/R_2}/g_{l_2'+1,2}^{V_{l_2'}}\right\}^{R_2}=pk_U\tag{6}$$

在银行揭示非法用户身份后,就能追回用户重复花费的电子现金.

6 正确性与安全性分析

本系统在随机语言机模型下,若 CL 签名是不可伪造的,则可分电子现金是不可伪造的,而且银行能检测出双重花费用户的真实身份;由于 ElGamal 加解密问题是难解的,因此本系统具有匿名性,在 Diffie-Hellman 难解问题的基础上,可分电子现金是不可链接的.下面进行各种安全性分析时,假设用户花费的是原二叉树上节点.

(1) 正确性

要证明本协议是否是正确的,就需要证明上面的六个等式:(1)(2)(3)(4)(5)(6).

式(1)、(3)、(4)都是零知识证明,这些是正确的,而式(2)、(5)、(6)的证明如下:

式(2)的证明:

$$\sigma' = \left(\left(h_0^{s'} h_1^u h_0^r\right)^{1/e}\right)^e = h_0^{s'+r'} h_1^u h_2^r = h_0^s h_1^u h_2^r$$

式(5)的证明:

$$\left(T_1^{R_2} / T_2^{R_1}\right)^{1/(R_2-R_1)} = \left\{\left(pk_U t_{i,j}^{V\cdot R_1}\right)^{R_2} / \left(pk_U t_{i,j}^{V\cdot R_2}\right)^{R_1}\right\}^{1/(R_2-R_1)} = pk_U$$

式(6)的证明:

$$\left\{(T_2)^{1/R_2} / g_{I_2^2+1,2}^{V_2}\right\}^{R_2} = \left\{\left(pk_U g_{I_2^2+1,2}^{R_2\cdot V_2}\right)^{1/R_2} / g_{I_2^2+1,2}^{V_2}\right\}^{R_2} = pk_U$$

(2) 匿名性

用户与商家交易时,所花费的电子现金是 C_1,所有的零知识证明及 S, R, $\tilde{V}_0$, $\tilde{V}_1$ 不能给任何人关于用户的任何消息,但是 $T_1 = pk_U t_{i,j}^{V\cdot R}$ 里面含有用户的公钥 pk_U,如果攻击者 A 知道了用户公钥,用户就不具有匿名性了. 然而此时除了用户以外的任何人只知道关于用户公钥的信息 T_1, A 要想知道用户的公钥,就要从信息 T_1 中提取,而 T_1 是关于用户公钥的 ElGamal 密文,如果用户没有发生重复花费,那么除非破解了 ElGamal 加解密难题,否则不能得到用户公钥,然而这被公认为只能以可忽略的概率成功. 所以用户支付是匿名的.

(3) 不可伪造性

用户生成电子现金是 $C = (s, u, r, \sigma)$,如果攻击者 A 在取款的时候以诚实者的身份取款,然后在 A 诚实的花费后,银行提取出电子现金 $C = (s, u, r, \sigma)$,如果 A 对于同一个电子现金有多于一次的花费,则 A 有关于同一个消息 $m = (s, u, r)$ 的两个签名 σ 和 σ',而且银行认为这两个签名都是合法的. 这样在 CL 签名中关于同一个消息 m 就有多于一个的有效签名,然而由文献[10]可知 CL 签名是在基于复杂 RSA 假设上被证明是抵抗适应性选择消息攻击安全的,所以对于同一个消息 m 不可能有多于一个的有效签名. 因此该系统具有不可伪造性.

(4) 不可链接性

不可链接性分为两个:

(a) 银行不能链接用户的一次取款和与它相关的花费;

用户在花费时生成的电子现金是:C_1,由协议的匿名性可知银行只能以可忽略的概率从 C_1 中求出用户的公钥,所以银行不能把用户的这次花费和他的取款联系起来.

(b) 任何人都不能链接用户对同一电子现金的两次花费;

假如用户对同一电子现金的两次花费生成的电子现金分别是:C_1 与 $C_2(2^{l_2}, S', T', \Phi', R', \tilde{V}_0', \tilde{V}_i')$,要想链接这两次花费,那就要知道 S 与 S' 是否由同一个 s 生成;即判定这个等式是否相等:$\log_g\left(\log_{g_{i,j}} S\right)^2 = \log_g\left(\log_{g_{i,j}} S\right)'$,其中 $S = g_{i,j}^V$, $V = g^s$, $S' = g_{i,j}^{V'}$, $V' = g^s$,然而如果 A 从此等式中判断出 S 与 S' 是由同一个 s 生成,说明 A 解决了 Diffie-Hellman 难解问题,但 Diffie-Hellman 难解问题被认为在多项式时间内是不可能解决的. 所以 A 就不能确定 S 与 S' 是否由同一个 s 生成. 可见任何人是不能链接用户对同一电子现金的两次花费.

由(a),(b)可知此协议满足不可链接性.

(5) 可检测用户重复花费性

如果用户对同一电子现金重复花费了,那么用户两次生成的 S 是一样的,银行通过对数据库 Q 的检测就能知道,然后根据公式(4)和(5)确定用户公钥,再根据用户的公钥从数据库 P 中得到用户的真实身份,这样银行就能揭示非法用户的身份.

7 效率性分析

现对本文与文献[6]在效率上进行比较,在安全的前提下,定义各参数的长度为:l 为 10 层二叉树,$|n| = 1024$bits,$|l_H| = 160$bits,$|p| = |q| = 512$bits,同时我们认为形如 $g_1^{x_1} g_2^{x_2}$ 和 $g_1^{x_1} g_2^{x_2} g_3^{x_3}$ 的模指数运算,分别相当于 1.2 和 1.5 个单模指数运算.

本文与文献[6]所生成的电子现金都是:$C = (s, u, r, \sigma = \mathrm{Sign}(s, u, r))$,则电子现金的大小为:$4 \times 1024\text{bits} = 512\text{Bytes}$.

本文与文献[6]的取款是一样的,在取款时用户都需要计算:零知识证明 U、C 和验证等式(2),银行要计算 C 和生成签名 σ,因此用户和银行总共要做 10 次模指数运算.

下面用户和商家所做的计算量都是指花费二叉树上叶子节点时所需的计算量,本文在支付时,用户花费的是由根节点直接计算的叶子节点时,即:计算 V 需要 $1 + 1.2 + 2 \approx 4.2$ 次模指数运算,计算 S 与 T 需要 $2 + 1 = 3$ 次模指数运算,计算零知识证明 Φ 大约需要 22.4 次模指数运算,因此用户总共要做 30 次模指数运算;在用户花费的是由根节点一层一层计算而来的叶子节点时,即:计算 V 需要 $1 + 1.2 + 2 + 8 \approx 12.2$ 次模指数运算,计算 S 与 T 需要 $2 + 1 = 3$ 次模指数运算,计算零知识证明 Φ 大约需要 73.4 次模指数运算,因此用户总共要做 89 次模指数运算. 商家为了检测用户的 S 与 T 是不是正确生成的,最多需要做 73.4 次模指数运算. 而文献[6]在计算第 l 层的节点时,用户和商家分别要做 816

和 857 次模指数运算.

本协议和文献[6]对比如表 1.

表 1　离线可分电子现金的对比

方　案	电子现金大小	取款计算量	支付时用户计算量	支付时商家计算量	所能花费电子现金总额
文献[6]	512	10	816	857	2^{t-1}
本　文			89	73.4	$t2^{t-1}$

注:(1) t 为二叉树的层数.

(2) 电子现金的单位是:Bytes,取款计算量、支付时用户计算量和支付时商家计算量的单位是:次模指数运算.

(3) 表 1 中支付时用户和商家所需要的计算量是指在花费叶子节点时所需要的.

8　结论

本文针对现有的可分电子现金协议存在用户花费时要从根节点一层一层计算的缺点,首次将节点直接计算和可再分的方法引入到可分电子现金协议中,在无可信第三方的情况下,构建了一个新型的离线可分电子现金协议. 在该协议中,引入节点可再分的方法,二叉树的每个节点都可以被花费,同一电子现金分出的节点是不可链接的,而且引入节点可直接计算的方法,使得二叉树上每个节点的计算量都一样,从而大大的提高了可分电子现金的整体效率.

参考文献:

[1] T Okamato, K Ohata. Universal electronic cash[A]. Proceedings of Cypto '91[C]. Berlin: Springer-Verlag, 1992. 324 - 337.

[2] T Okamoto. An efficient divisible electronic cash scheme[A]. Advances in Cryptology-CRYPTO '95[C]. German: Springer-Verlag, 1995, 438 - 451.

[3] A Chan, Y frankel, Y Tsiounis. Easy come-easy go divisible cash[A]. Advances in Cryptology- EUROCRYPT '98[C]. Finland: Springer-Verlag, 1998. 561 - 575.

[4] T Nakanishi, M Sugiyama. Unlinkable divisible electronic cash [A]. ISW '00[C]. Australia: Springer-Verlag, 2000. 121 - 134.

[5] 彭冰,洪帆和崔国华. 基于零知识证明签名和强 RSA 问题的可分电子现金[J]. 通信学报, 2006, 27(7): 12 - 19.
Peng Bing, Hong Fan, Cui Guohua. Divisible e-cash based on signatures of zero-knowledge proof and strong RSA problem [J]. Journal on Communication. 2006, 27(7): 12 - 19. (in Chinese)

[6] S Canard, A Gouget. Divisible e-cash systems can be truly anonymous[A]. Advances in Cryptology-EUROCRYPT '07 [C]. Barcelona, Spain: Springer-Verlag, 2007. 482 - 497.

[7] 李梦东,杨义先. 无可信第三方的离线电子现金的匿名性控制[J]. 电子学报. 2005, 3(33): 456 - 458.
Li Mengdong, Yang Yixian. Revocable anonymous off-line e-cash scheme without TTP[J]. Acta Electronica Sinica, 2005, 3 (33): 456 - 458. (in Chinese)

[8] J Camenisch, M Michels. Proving in zero-knowledge that a number n is the product of two safe primes[A]. Advances in Cryptology-EUROCRYPT '99[C]. Czech Republic: Springer-Verlag, 1999. 107 - 122.

[9] E Fujisaki, T Okamoto. Statistical zero knowledge protocols to prove modular polynomial relations[A]. Advances in Cryptology - CRYPTO '97[C]. Berlin: Springer-Verlag. 1997. 16 - 30.

[10] J Camenisch, A Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps[A]. Advances in Cryptology-Crypto '04[C]. California: Springer-Verlag, 2004. 56 - 72.

[11] T Pedersen. Non-interactive and information theoretic secure verifiable secret sharing[A]. Advances in Cryptology-CRYPTO '91[C]. Berlin: Springer-Verlag, 1991. 129 - 140.

[12] J Camenisch, A Lysyanskaya. A signature scheme with efficient protocols[A]. In Giuseppe Persiano, editor, Security in communication networks [C]. Italy: Springer-Verlag, 2002. 268 - 289.

[13] A Fiat, A Shamir. How to prove yourself: practical solutions to identification and signature problems[A]. Advances in Cryptology-Crypto '86[C]. California: Springer-Verlag, 1986. 186 - 194.

作者简介:

刘文远　男,1968 年生于黑龙江密山,哈尔滨工业大学计算机应用技术博士,研究方向:电子商务、信息安全.
E-mail: wyliu@vip.163.com

张江霄　男,1983 生于河北邢台,硕士研究生,研究方向:电子商务、信息安全、密码学.