





计算结果验证过程是信息论安全的。

本文第 1 节简要介绍博弈论概念、比特币交易以及 Micali-Rabin 随机向量表示技术.第 2 节对委托计算进行博弈分析.第 3 节提出一种理性委托计算协议.第 4 节对本协议进行安全性和其他性能分析.第 5 节给出结论.

## 1 预备知识

### 1.1 博弈论概念

**定义 1(博弈).** 博弈表达的基本式<sup>[20]</sup>由局中人集合  $P$ 、策略空间  $S$  和效用函数  $u$  这 3 个要素组成,即  $G=\{P,S,u\}$ , $S=\{S_1,\dots,S_n\}$ , $u=\{u_1,\dots,u_n\}$ .效用函数  $u_i:S\rightarrow R$ ( $R$  代表实数空间),它表示第  $i$  位局中人在不同策略组合下所得到的收益.

**定义 2(子博弈精炼纳什均衡).** 在一个具有完美信息的动态博弈中,各博弈方的策略构成一个策略组合满足:(1) 它为原博弈的纳什均衡;(2) 在整个动态博弈以及它的子博弈中都构成纳什均衡,那么这个策略组合称为该动态博弈的一个“子博弈精炼纳什均衡”。

### 1.2 比特币交易

在比特币协议<sup>[21]</sup>中,每个用户根据椭圆曲线算法 secp256k1 选取密钥对.一笔标准的比特币交易  $T$  是由金额  $amount$ 、签名  $sign$ 、接收方地址  $address$ 、未经花费的交易输出 UTXO(unspeut transaction output)组成,即  $T=(amount,sign,address,UTXO)$ ,其中  $sign=(amount,address,UTXO)$  是利用用户(交易发送方)私钥对  $amount$ 、 $address$ 、 $UTXO$  这 3 个部分的计算而获得的签名; $[T]=(amount,address,UTXO)$  称为交易主体  $body$ ;  $address$  是用用户公钥  $pk$  先经过 SHA160 取得 160 比特的摘要,再经 Base58check 编码后的地址.最简单的一笔交易是只有一个交易输入和一个交易输出(假设不存在交易费).其输入脚本为签名,输出脚本采用 P2PKH(pay-to-pub-key-hash)类型,即指定公钥的地址和签名验证脚本.图 1 示意有两个用户  $A$ 、 $B$ ,其密钥对为  $(A.sk,A.pk)$  和  $(B.sk,B.pk)$ ,用户  $A$  要向用户  $B$  发送一笔交易  $T_V$ . $T_V=(VB,\sigma_V,\pi_V,y)$ ,其中  $amount=VB,sign=\sigma_V,address=\pi_V,UTXO=y$ , $B$  为比特币符号. $\sigma_V$  为输入脚本  $in-script$ ——用于验证输出脚本  $\pi_V$ (即上一笔交易  $T_y$  的接收方  $A$  的地址和签名验证脚本指令)对  $T_V$  计算结果为真的证据;输出脚本  $out-script$  为  $\pi_V$ ,这是对这笔交易  $T_V$  的花费条件进行限制.

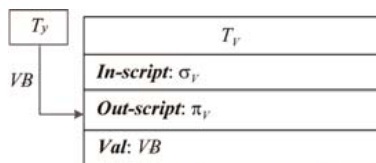


Fig.1 Bitcoin transaction

图 1 比特币交易

### 1.3 Micali-Rabin的随机向量表示技术

Micali-Rabin 利用随机向量表示值,采用 Pedersen 承诺<sup>[22]</sup>,通过零知识证明等式的正确性.设  $g,h$  是  $G_q$  的生成元, $q$  是大素数, $F_p$  为有限域, $p$  为 128 bit 的素数, $q>p$ .

**定义 1.** 设  $X$  的随机向量表示是  $X=(u,v)$ ,其中  $u,v\in F_p$ , $X$  的值是  $Val(X)=(u+v) \bmod p$ .

**定义 2.** 令  $u\in F_p$ ,存在一个辅助值  $r,r\in F_p$ ,对  $u$  进行承诺  $Com(u)=Com(u,r)=g^u h^r$ .

**定义 3.** 对  $X=(u,v)$  的分量进行承诺,即  $Com(X)=(Com(u),Com(v))$ .

**定义 4.** 假设有两行承诺值,其中  $1\leq i\leq n$ ,

$$Com(X_1),\dots,Com(X_n), Com(Y_1),\dots,Com(Y_n) \quad (1)$$

如果  $Val(X_i)=Val(Y_i),1\leq i\leq n$ ,那么这两行的承诺值是一致的.

**引理.** 如果有超过  $k$  个承诺值是假的,那么验证者接受的概率为  $(1/2)^k$ .

**证明:**对任意的  $Val(X_i)\neq Val(Y_i)$ ,这种情况不被发现的概率最多为  $1/2$ .因为独立、随机地选择一个  $c\leftarrow\{1,2\}$ ,

至少有  $k$  个等式不被发现的概率为  $(1/2)^k$ . □

## 2 委托计算的博弈分析

理性委托计算是结合博弈论和委托计算的思想,从参与者自利角度出发,通过效用函数来保障计算结果的可靠性.下面对理性委托计算进行分析.

假设参与者为委托方  $P_1$  和计算方  $P_2$ ,且都是理性参与者.委托方  $P_1$  拥有计算任务  $E$ ,计算任务  $E$  的价值为  $R(E)$ , $P(E)$  为计算方  $P_2$  计算任务  $E$  的成本, $W(E)$  为委托方  $P_1$  的计算任务  $E$  的支付成本,且  $R(E) > W(E) > P(E)$ .

委托方  $P_1$  将计算任务  $E$  和计算函数  $f$  发送给计算方  $P_2$ .计算方  $P_2$  接收计算任务  $E$ 、计算函数  $f$ ,运用自己的资源和计算函数  $f$  对计算任务  $E$  进行计算.计算方  $P_2$  计算完成后,将计算结果  $f(E)$  发送给委托方  $P_1$ .委托方  $P_1$  接收计算结果  $f(E)$ ,同时支付给计算方  $W(E)$ .由于参与者是理性的,因此计算方  $P_2$  可能发送错误的结果,委托方  $P_1$  可以惩罚计算方  $P_2$ .如图 2 所示.

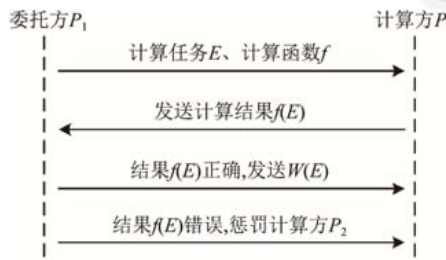


Fig.2 Rational delegation computation

图 2 理性委托计算

如图 3 所示,图中圆圈为委托方  $P_1$  或者计算方  $P_2$  的选择信息集(选择节点).首先计算方  $P_2$  有发送“正确结果”和“错误结果”两种行为选择, {正确,错误} 为计算方  $P_2$  的策略集.如果计算方选择发送正确的结果,则博弈结束,委托方  $P_1$  的效用  $u_1=R(E)-W(E)$ ,计算方  $P_2$  的效用  $u_2=W(E)-P(E)$ ;如果计算方  $P_2$  选择发送错误的结果,那么委托方  $P_1$  的效用  $u_1=-W(E)$ ,计算方  $P_2$  的效用  $u_2=W(E)$ ,因计算方  $P_2$  发送的结果是错误的,故不考虑计算成本.

由于参与者是理性的,为了自己的利益,计算方  $P_2$  可能会选择发送错误的结果,委托方  $P_1$  为防止计算方  $P_2$  欺骗自己,会惩罚计算方  $P_2$ .若计算方  $P_2$  发送错误的结果,委托方  $P_1$  会设置函数  $F$  惩罚计算方  $P_2$ ,此时委托方  $P_1$  的策略集为 {惩罚,不惩罚}.如果委托方  $P_1$  选择惩罚,委托方  $P_1$  的效用  $u_1=F(E)-W(E)$ ,计算方  $P_2$  的效用  $u_2=-(F(E)-W(E))$ ;若委托方  $P_1$  选择不惩罚,委托方  $P_1$  的效用  $u_1=-W(E)$ ,计算方  $P_2$  的效用  $u_2=W(E)$ ,且  $F(E) > W(E) > P(E)$ .如图 4 所示.

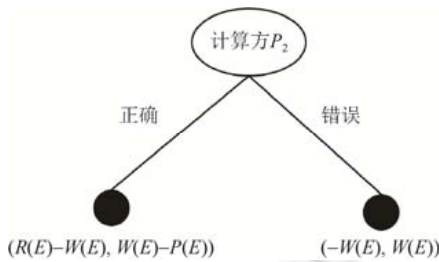


Fig.3 Game tree (I)

图 3 博弈树(I)

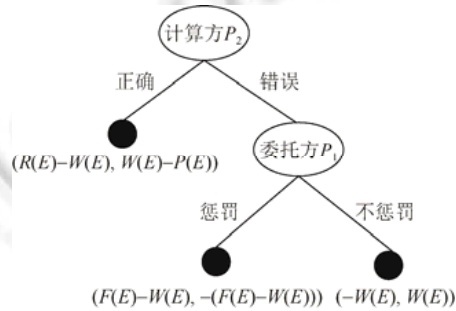


Fig.4 Game tree (II)

图 4 博弈树(II)

显然,理性委托计算可以用两个阶段来表示.

- (1) 计算方  $P_2$  从策略集 {正确,错误} 选择一个行动;

(2) 委托方  $P_1$  在观察到计算方  $P_2$  的行动后,从自己的策略集{惩罚,不惩罚}中选择一个行动.

用子博弈精炼纳什均衡分析理性委托计算.第 1 阶段,计算方  $P_2$  选择发送正确的结果;第 2 阶段,委托方  $P_1$  选择惩罚计算方  $P_2$ .该策略组合是该博弈唯一的子博弈精炼纳什均衡,也是真正稳定的均衡.

### 3 理性委托计算协议

根据理性委托计算中纳什均衡,设计如下理性委托计算协议.在该协议中,理性参与者只有发送正确的结果,才能获得最大化利益,若发送错误的结果,会损害自身的利益;参与者若发现对方有欺骗行为,可以随时终止协议.也就是说,只有诚实执行协议的参与者的利益不会有任何损害.

假设在理性委托计算中,存在一个电子公告栏 SBB,该电子公告栏 SBB 用来张贴数据;数据被双方(委托方和计算方)所见,且双方看到的 SBB 上的内容是相同的;数据一旦被写入 SBB,就不能被删除修改.由于本协议参与方只有委托方和计算方,为确保诚实者的利益不受损害,引入一个可信第三方  $P_0$ .该可信第三方  $P_0$  可以帮助诚实者取回押金<sup>[17]</sup>.计算任务  $E$  的委托计算时间为  $t$ ,即在  $t$  时间内,一次委托计算任务完成(包括验证).同时将  $t$  时间分为 4 段, $0 < t_1 < t_2 < t_3 < t$ .

具体理性委托计算协议如下.

初始化阶段.

首先委托方  $P_1$  在 SBB 上公布计算任务  $E$  的描述信息以及押金  $De(E)$ .

接着委托方  $P_1$ 、计算方  $P_2$  以及可信第三方  $P_0$  根据椭圆曲线密码体制选取密钥对,用于交易的签名和验证.选取一条安全的椭圆曲线  $E$  和  $E$  的一个基点  $G$ ,  $G$  的阶为  $n$  ( $n$  为一个素数),秘密选择一个随机的整数  $1 < k_i < n$ ,计算  $d_i = k_i G$ ,然后公开  $(G, d_i)$ ,  $d_i$  为公钥,私钥为  $k_i$ ,  $i=0,1,2$ .

提交押金.假设委托方  $P_1$  委托的计算任务为  $E$ ,其支付成本为  $W(E)$ ,委托方  $P_1$  提交押金  $De(E)$  (其中,  $De(E) > W(E)$ ),提交押金将生成一笔交易  $T_{P_1}$ ,即  $T_{P_1} = (De(E), \sigma_{P_1}, \pi_{P_1}, \nu_{P_1})$ ,其中,  $\sigma_{P_1} = \text{sign}_{P_1}(De(E), \pi_{P_1}, \nu_{P_1})$ .比特币节点收到交易  $T_{P_1}$  后,先通过比特币脚本对交易有效性进行验证,若验证成功,则将交易广播至邻近节点,邻近节点再验证,通过验证后,再向邻近节点广播,直至全网大部分节点都接受到这笔交易.只要一个节点验证不成功,则丢弃这笔交易,不会再向邻近节点转发.最终,交易  $T_{P_1}$  被一个挖矿节点写进比特币账本中.在委托方  $P_1$  提交押金的脚本中,输入脚本  $in\_script$  为  $\sigma_{P_1}$ ,输出脚本  $out\_script$  为  $(\text{ver}_{P_1}(\text{body}, \sigma_{P_1}) \wedge \text{ver}_{P_2}(\text{body}, \sigma_{P_2})) \vee (\text{ver}_{P_1}(\text{body}, \sigma_{P_1}) \wedge \text{ver}_{P_0}(\text{body}, \sigma_{P_0})) \vee (\text{ver}_{P_2}(\text{body}, \sigma_{P_2}) \wedge \text{ver}_{P_0}(\text{body}, \sigma_{P_0}))$ ,押金为  $De(E)$ ,锁定脚本的时间  $tlock = t$ .对于计算方  $P_2$ ,也按照如上方式生成押金交易  $T_{P_2}$ .图 5 所示为委托方  $P_1$  生成的押金交易  $T_{P_1}$ .

$T_{P_1}$
<b>In-script:</b> $\sigma_{P_1}$
<b>Out-script:</b> $(\text{ver}_{P_1}(\text{body}, \sigma_{P_1}) \wedge \text{ver}_{P_2}(\text{body}, \sigma_{P_2})) \vee$ $(\text{ver}_{P_1}(\text{body}, \sigma_{P_1}) \wedge \text{ver}_{P_0}(\text{body}, \sigma_{P_0})) \vee$ $(\text{ver}_{P_2}(\text{body}, \sigma_{P_2}) \wedge \text{ver}_{P_0}(\text{body}, \sigma_{P_0})) \vee$
<b>Val:</b> $De(E)$
<b>tlock:</b> $t$

Fig.5 Deposit transaction  $T_{P_1}$

图 5 押金交易  $T_{P_1}$

计算阶段.

委托方  $P_1$  将计算任务  $E$ 、计算方式  $f$  发送给计算方  $P_2$ .计算方  $P_2$  利用自己的资源,结合计算方式  $f$  获得计算任务  $E$  的结果  $f(E)$ .接着计算方  $P_2$  在电子公告栏 SBB 上对  $f(E)$  进行  $3k$  行承诺  $(E, \text{Com}(f(E)_i^j))$ .电子公告栏 SBB 上的  $3k$  行承诺值采用 Micali-Rabin 的随机向量表示法,  $\text{Com}(f(E)_i^j) = (\text{Com}(u_i^j), \text{Com}(v_i^j))$ ,  $f(E)_i^j = (u_i^j, v_i^j)$ ,  $\text{Val}(f(E)_i^j) = (u_i^j + v_i^j) \bmod p, 1 \leq j \leq 3k$ .

验证和支付阶段.

计算方  $P_2$  将计算结果  $f(E)$  发送给委托方  $P_1$ , 由委托方  $P_1$  验证计算结果  $f(E)$  的正确性. 分以下 3 种常见情况进行讨论, 对于委托方  $P_1$  和计算方  $P_2$  没有按以下 3 种情况进行操作, 即超时操作的,  $t$  时间后, 委托方  $P_1$  和计算方  $P_2$  分别联系第三方  $P_0$ , 取回各自押金.

情况 1: 在  $t_1$  时间内, 计算方  $P_2$  将  $(E, f(E))$  发送给委托方  $P_1$ . 委托方  $P_1$  接收计算任务  $E$  的结果  $f(E)$ .

情况 2: 在  $t_1$  时间内, 计算方  $P_2$  将  $(E, f(E))$  发送给委托方  $P_1$ . 委托方  $P_1$  拒绝或者因为网络问题没有接收  $(E, f(E))$ , 即计算方  $P_2$  没有收到委托方  $P_1$  发送的接收  $(E, f(E))$  的证明.

情况 3: 在  $t_1$  时间内, 计算方  $P_2$  没有将  $(E, f(E))$  发送给委托方  $P_1$ .

对于情况 1, 在  $t_1$  时间内, 委托方  $P_1$  接收  $(E, f(E))$ . 在  $t_2$  时间内, 委托方  $P_1$  可以与计算方  $P_2$  进行交互式证明, 验证计算方  $P_2$  给出的计算结果  $f(E)$  的承诺值是否正确. 步骤如下.

a) 首先委托方  $P_1$  从电子公告栏 SBB 上任意选择一半的承诺值  $\langle E, Com(f(E)_i^j) \rangle$ , 验证计算任务结果  $f(E)$  的正确性. 委托方  $P_1$  秘密地与计算方  $P_2$  联系, 由计算方  $P_2$  秘密地向委托方  $P_1$  打开承诺值分量  $f(E)_i^j = (u_i^j, v_i^j)$ , 其中  $u_i^j, v_i^j \in F_p$ , 委托方  $P_1$  验证等式  $Val(f(E)_i^j) = (u_i^j + v_i^j) \bmod p$  是否成立.

b) 接着对于电子公告栏 SBB 上剩余的承诺值进行上下行一致性检验. 在电子公告栏 SBB 上, 对于每个值的承诺值, 每一列是一致的,  $Val(f(E)_i^j) = Val(f(E)_i^{j+1}), 1 \leq j \leq 3k$ . 委托方  $P_1$  对电子公告栏 SBB 剩余的一半承诺值  $\langle E, Com(f(E)_i^j) \rangle$  中的  $f(E)$  进行验证. 若委托方  $P_1$  要证明  $f(E)$  的第  $j$  行承诺值与第  $j+1$  行承诺值是一致的, 即证明等式  $Val(f(E)_i^j) = Val(f(E)_i^{j+1})$  成立. 假设  $f(E)_i^j$  的承诺为  $Com(f(E)_i^j) = (Com(u_i^j), Com(v_i^j)), f(E)_i^{j+1}$  的承诺为  $Com(f(E)_i^{j+1}) = (Com(u_i^{j+1}), Com(v_i^{j+1}))$ , 其中  $f(E)_i^j = (u_i^j, v_i^j), f(E)_i^{j+1} = (u_i^{j+1}, v_i^{j+1})$ . 计算方  $P_2$  根据  $f(E)_i^j = (u_i^j, v_i^j)$  以及  $f(E)_i^{j+1} = (u_i^{j+1}, v_i^{j+1})$  的分量值, 准备  $w$ .

$$w = (u_i^{j+1} - u_i^j) \bmod p, w = (v_i^j - v_i^{j+1}) \bmod p \tag{2}$$

委托方  $P_1$  可以通过抛硬币选择值  $c \leftarrow \{1, 2\}$ . 如果  $c=1$ , 计算方  $P_2$  将打开承诺值  $Com(u_i^j)$ 、 $Com(u_i^{j+1})$  以及  $-w$  的值验证等式  $u_i^j = (u_i^{j+1} + (-w)) \bmod p$ . 反之,  $c=2$ , 计算方  $P_2$  打开承诺值  $Com(v_i^j)$ 、 $Com(v_i^{j+1})$  以及  $w$  的值, 验证等式  $v_i^j = (v_i^{j+1} + w) \bmod p$ . 从这个等式中可以看出, 每次只打开承诺值的一半, 因此计算方  $P_2$  失败的概率至少为  $1/2$ . 若计算方  $P_2$  有超过  $k$  个承诺值是假的, 那么委托方  $P_1$  接受的概率为  $(1/2)^k$ .

c) 委托方  $P_1$  与计算方  $P_2$  交互式证明后, 若验证结果正确, 在  $t_3$  时间内, 委托方  $P_1$  将准备一笔支付交易  $T_{pay}$ , 计算方  $P_2$  收到委托方  $P_1$  的支付金额  $W(E)$ , 图 6 所示为支付交易  $T_{pay}$ . 接着委托方  $P_1$  和计算方  $P_2$  创造交易  $T_{De-pi}$ , 在  $t$  时间后, 领回各自押金  $De(E), i=1, 2$ , 图 7 所示为委托方  $P_1$  创建的取回押金的交易  $T_{De-p1}$ ; 若验证结果不正确, 委托方  $P_1$  将自己与计算方  $P_2$  交互式证明的结果发送给第三方  $P_0$ . 在  $t$  时间后, 第三方  $P_0$  与委托方  $P_1$  联合签名, 创建交易  $T_{De-p2|(p1 \wedge p0)}$ , 取走计算方  $P_2$  的押金. 当然, 同时创建交易  $T_{De-p1|(p1 \wedge p0)}$ , 取回自己的押金. 图 8 所示为取走计算方  $P_2$  的押金交易  $T_{De-p2|(p1 \wedge p0)}$ . 这里, 计算方  $P_2$  押金归委托方  $P_1$  所有, 不考虑第三方  $P_0$ .

$T_{pay}$
<b>In-script:</b> $\sigma_{pay}$
<b>Out-script:</b> $ver_{p_2}(body, \sigma_{p_2})$
<b>Val:</b> $W(E)$
<b>lock:</b> 0

Fig.6 Payment transaction  $T_{pay}$

图 6 支付交易  $T_{pay}$



$T_{De-p1}$
<b>In-script:</b> $\sigma_{De-p1}, \sigma_{De-p2}$
<b>Out-script:</b> $ver_{p1}(body, \sigma_{De-p1})$
<b>Val:</b> $De(E)$
<b>lock:</b> 0

Fig.7 Withdrawal deposit transaction  $T_{De-p1}$ 图 7 取回押金交易  $T_{De-p1}$ 

$T_{De-p2 (p1 \wedge p0)}$
<b>In-script:</b> $\sigma_{De-p1}, \sigma_{De-p0}$
<b>Out-script:</b> $ver_{p1}(body, \sigma_{De-p2 (p1 \wedge p0)})$
<b>Val:</b> $De(E)$
<b>lock:</b> 0

Fig.8 Withdrawal deposit transaction  $T_{De-p2|(p1 \wedge p0)}$ 图 8 取回押金交易  $T_{De-p2|(p1 \wedge p0)}$ 

对于情况 2:委托方  $P_1$  拒绝或者因为网络问题没有接收( $E, f(E)$ ),即计算方  $P_2$  没有收到接收证明。

a) 在  $t_2$  时间内,计算方  $P_2$  将联系第三方  $P_0$ ,计算方  $P_2$  将对( $E, f(E)$ )进行签名,得到  $\sigma_{P_1}(E, f(E))$ ,计算方  $P_2$  将( $\sigma_{P_1}(E, f(E)), E, f(E)$ )发送给第三方  $P_0$ ;第三方  $P_0$  验证签名后(因计算方  $P_2$  主动联系第三方,不考虑签名错误问题),加入自己的签名  $\sigma_{P_0}$ ,发送( $\sigma_{P_0}(\sigma_{P_1}(E, f(E)), E, f(E)), \sigma_{P_1}(E, f(E)), E, f(E)$ )给委托方  $P_1$ 。

b) 若时间  $t_2$  内,委托方  $P_1$  接收( $\sigma_{P_0}(\sigma_{P_1}(E, f(E)), E, f(E)), \sigma_{P_1}(E, f(E)), E, f(E)$ ),那么在  $t_3$  时间内委托方  $P_1$  与计算方  $P_2$  进行交互式验证, $t$  时间内进行支付,其情况与情况 1 相同;若时间  $t_2$  内,委托方  $P_1$  依然拒绝接收( $E, f(E)$ ).那么  $t_2$  时间后,计算方  $P_2$  联合第三方  $P_0$ ,在  $t$  时间后,第三方  $P_0$  与计算方  $P_2$  联合签名,创建交易  $T_{De-p1|(p2 \wedge p0)}$ ,取走委托方  $P_1$  的押金,同时创建交易  $T_{De-p2|(p2 \wedge p0)}$ ,取回自己的押金。

对于情况 3:在  $t_1$  时间内,计算方  $P_2$  没有将计算任务  $E$  的结果  $f(E)$  发送给委托方  $P_1$ 。

a) 在  $t_2$  时间内,委托方  $P_1$  将联系第三方  $P_0$ ,委托方  $P_1$  将( $\sigma_{P_1}(proof), \sigma_{P_1}(E), proof, E$ )发送给第三方  $P_0$ , $proof$  是指委托方  $P_1$  没有收到计算结果的证据;第三方  $P_0$  将( $\sigma_{P_0}(\sigma_{P_1}(E)), \sigma_{P_1}(E), E$ )发送给计算方  $P_2$ ,要求计算方  $P_2$  将任务  $E$  的计算结果发送给委托方  $P_1$ 。

b) 若  $t_2$  时间内,委托方  $P_1$  接收计算结果  $f(E)$ ,并与计算方  $P_2$  进行交互式验证,其情况与情况 1 相同;若在时间  $t_2$  内计算方  $P_2$  没有将计算结果  $f(E)$  发送给委托方  $P_1$ ,那么  $t_3$  时间内,委托方  $P_1$  联系第三方  $P_0$ ,在  $t$  时间后,第三方  $P_0$  与委托方  $P_1$  联合签名,创建交易  $T_{De-p2|(p1 \wedge p0)}$ ,取走计算方  $P_2$  的押金,同时创建交易  $T_{De-p1|(p1 \wedge p0)}$ ,取回自己的押金。

## 4 协议分析

### 4.1 安全性分析

我们从公平性、正确性、隐藏性这 3 个方面分析理性委托计算协议。

**定理 1(该理性委托计算协议具有公平性).**

证明:在理性委托计算协议中,参与人都是理性的,为保障自己的利益,参与人可能会选择欺骗对方.为保障协议的公平性,在协议开始阶段,委托方  $P_1$  和计算方  $P_2$  都提交了一笔特殊构造的比特币押金.只要双方诚实执行协议, $t$  时间后就可以各自收回押金;若任意一方不诚实,那么就会受到失去押金的惩罚.当情况 1 中验证结果错误时,委托方  $P_1$  将自己与计算方  $P_2$  交互式证明的结果发送给第三方  $P_0$ .在  $t$  时间后,第三方  $P_0$  与委托方  $P_1$  联合签名,创建交易  $T_{De-p2|(p1 \wedge p0)}$ ,取走计算方  $P_2$  的押金,作为对计算方  $P_2$  欺骗自己的惩罚;当出现情况 2 和情况 3 时,诚实的一方在规定时间内可以联系第三方  $P_0$ ,让第三方  $P_0$  联系另一方,若另一方收到第三方通知后,可选择遵守协议,那么委托计算过程继续;若另一方收到第三方通知后,仍然不遵守协议,那么诚实方联系第三方  $P_0$ ,在  $t$  时间后,第三方  $P_0$  与诚实方联合签名,取走不遵守协议的参与方押金。□

**定理 2(该理性委托计算协议具有正确性).**

证明:在情况 1 中,如果委托方  $P_1$  和计算方  $P_2$  都是诚实的,即他们遵守协议规则.首先,在  $t_1$  时间内,计算方  $P_2$  将计算任务的结果  $f(E)$  按时发送给委托方  $P_1$ ;其次,在  $t_2$  时间内,双方进行交互式证明,验证计算方  $P_2$  给出的

计算结果  $f(E)$  的承诺值是否正确;若验证结果正确,那么在  $t_3$  时间内,委托方  $P_1$  将支付一笔钱作为酬劳给计算方  $P_2$ ;最后,在  $t$  时间后,双方领回各自押金  $De(E)$ .  $\square$

**定理 3(该理性委托计算协议具有隐藏性).**

证明:一方面,在验证支付阶段,第三方  $P_0$  不会获得关于计算结果  $f(E)$  的任何信息.在本协议中,可信第三方  $P_0$  仅仅作为一个联系方,协助委托方  $P_1$  和计算方  $P_2$  完成委托计算的过程,同时帮助诚实的一方惩罚欺骗者,取走欺骗者的押金.

另一方面,在计算阶段,计算方  $P_2$  采用 Micali-Rabin 的随机向量表示技术,对计算结果  $f(E)$  在 SBB 上进行  $3k$  行承诺  $(E, Com(f(E)_i^j))$ ,  $1 \leq j \leq 3k$ . 在验证支付阶段,委托方  $P_1$  和计算方  $P_2$  进行交互式证明,验证计算方  $P_2$  给出的计算结果  $f(E)$  的承诺值是否正确.首先验证计算任务结果  $f(E)$  的正确性.委托方  $P_1$  秘密地与计算方  $P_2$  联系,由计算方  $P_2$  秘密地向委托方  $P_1$  打开承诺值分量,因此第三方不知道具体的计算结果  $f(E)$  的信息;接着验证计算任务结果  $f(E)$  上下的一致性,验证时由委托方  $P_1$  抛硬币来随机选择打开承诺值的一个分量,根据这一个分量,是无法推断出  $f(E)$  的,因为 Pedersen 承诺方案是信息论隐藏的.因此在验证时不会泄漏计算结果  $f(E)$ .  $\square$

## 4.2 性能分析

下面给出本协议与其他协议的性能比较,见表 1.

**Table 1 Protocol comparison**

**表 1 协议比较**

	计算复杂度	通信复杂度
文献[23]	$O(n)$	$\geq 3$
文献[24]	$O(1)$	1
文献[17]	$O(1)$	1
本文协议	$O(1)$	1

文献[23]基于 cut-and-choose 协议和秘密共享,提出了一个公平支付方案.计算复杂度为  $O(n)$ ,通信复杂度至少为 3,其中,  $n$  为 ringer 的数目.但该方案需要一个银行来生成支付代币,若交易成本太高,那么选择银行作为支付中介就不是理想选择.此外,该方案性能较差.

文献[24]基于传统的电子现金系统,提出了适合于分布式外包计算的新的公平有条件支付方案,运用第三方解决了外包者和工人的信任问题.计算复杂度为  $O(1)$ ,通信复杂度为 1.但该方案提出的有条件支付仍然需要一个银行产生以及验证代币.

文献[17]基于比特币和承诺抽样技术构造了一个公平支付协议,取消了银行作为支付中介.在方案中,委托者需要付一笔比特币押金.只要工人是诚实的,无论委托者行为是诚实的或恶意的,工人均可以得到支付.文献[17]中引入了一个半可信第三方,帮助委托方取回押金.计算复杂度为  $O(1)$ ,通信复杂度为 1.

本文协议基于比特币和 Micali-Rabin 的随机向量表示技术,设计一种新的理性委托计算协议.计算复杂度为  $O(n)$ ,通信复杂度为 1.协议中引入了一个可信第三方,帮助委托方和计算方取回押金.与文献[23,24]相比,本文协议不需要一个银行,因此不用担心交易成本较高问题;与文献[17]相比,在本文协议中委托方和计算方都要提交押金,一旦计算方出现不遵守协议的行为,委托方可以联合第三方取走计算方的押金.只要委托方和计算方遵守协议,最终他们都能公平、正确地得到计算结果和收益.

接下来对协议花费的时间进行评估.为了简单起见,本文仅考虑时间和任务量的关系.本文协议的时间开销主要是委托方和计算方提交押金交易时间、计算和验证任务两部分时间组成.在初始化阶段,委托方和计算方提交押金交易.按照比特币协议规范,一般认为整个比特币网络平均每 10 分钟产出一个块.因此,本文将提交押金时间设为恒定 10 分钟(本文不考虑比特币分叉问题).在计算阶段,计算方对委托方的任务进行计算,理论上来说,任务量大,计算时间开销大(这里,为了方便,表示为线性关系);在验证阶段,委托方采用 Micali-Rabin 随机向量来表示技术验证计算结果,与任务量大小无关,因而将验证时间设为常数.总的来说,计算和验证任务的时间开销随着任务量的增加而增加.图 9 给出了所提协议时间消耗和任务量的关系.



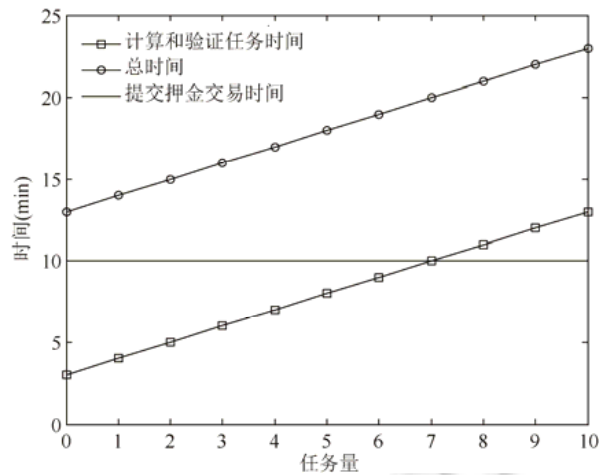


Fig. 9 Time overhead of the protocol

图 9 协议时间开销

## 5 结论

本文基于博弈论研究了委托计算问题,详细分析博弈论环境下委托计算中参与者的策略和效用,并提出了基于比特币和 Micali-Rabin 的随机向量表示技术的公平理性委托计算协议.在理性委托计算中,委托方和计算方仅关心自身的收益,做决策的依据都取决于其效用.文中仅给出了一对一型理性委托计算协议.在边缘计算环境中往往存在一对多甚至多对多的理性委托计算协议,这将是下一步工作的方向.

## References:

- [1] Xue R, Wu Y, Liu MH, Zhang LF, Zhang R. Progress in verifiable computation. SCIENTIA SINICA Informationis, 2015,45(11): 1370–1388 (in Chinese with English abstract).
- [2] Goldwasser S, Micali S, Rackoff C. The knowledge complexity of interactive proof systems. SIAM Journal on Computing, 1989, 18(1):186–208.
- [3] Babai L. Trading group theory for randomness. In: Sedgewick R, ed. Proc. of the 17th Annual ACM Symp. on Theory of Computing. New York: ACM, 1985. 421–429.
- [4] Arora S, Safra S. Probabilistic checking of proofs: A new characterization of NP. Journal of the ACM (JACM), 1998,45(1):70–122.
- [5] Gennaro R, Gentry C, Parno B, Raykova M. Quadratic span programs and succinct NIZKS without PCPs. In: Johansson T, Nguyen PQ, eds. Proc. of the EUROCRYPT. Berlin: Springer-Verlag, 2013. 626–645.
- [6] Gentry C. A fully homomorphic encryption scheme [Ph.D. Thesis]. Stanford: Stanford University, 2009.
- [7] Gentry C. Fully homomorphic encryption using ideal lattices. In: Mitzenmacher M, ed. Proc. of the ACM Symp. on the Theory of Computing (STOC). New York, 2009. 169–178.
- [8] Gennaro R, Wichs D. Fully homomorphic message authenticators. In: Sako K, Sarkar P, eds. Proc. of the ASIACRYPT. Berlin: Springer-Verlag, 2013. 301–320.
- [9] Azar PD, Micali S. Rational proofs. In: Karloff H, Pitassi T, eds. Proc. of the Annual ACM Symp. on Theory of Computing. New York: ACM, 2012. 1017–1028.
- [10] Azar PD, Micali S. Super-Efficient rational proofs. In: Kauffman RJ, Bichler M, Lau HC, Yang Y, Yang C, eds. Proc. of the 14th Annual ACM Conf. on Electronic Commerce (EC). New York: ACM, 2013. 29–30.
- [11] Guo S, Hubacek P, Rosen A, Vald M. Rational arguments: Single round delegation with sublinear verification. In: Naor M, ed. Proc. of the 5th Annual Conf. on Innovations in Theoretical Computer Science (ITCS). New York: ACM, 2014. 523–540.

- [12] Campanelli M, Gennaro R. Sequentially composable rational proofs. In: Decision and Game Theory for Security. Switzerland: Springer Int'l Publishing, 2015. 270–288.
- [13] Chen J, Mccauley S, Singh S. Rational proofs with multiple provers. In: Sudan M, ed. Proc. of the 2016 ACM Conf. on Innovations in Theoretical Computer Science. New York: ACM, 2016. 237–248.
- [14] Inasawa K, Yasunaga K. Rational proofs against rational verifiers. IACR Cryptology ePrint Archive, 2017,2017:270.
- [15] Bentov I, Kumaresan R. How to use bitcoin to design fair protocols. In: Garay JA, Gennaro R, eds. Proc. of the Int'l Cryptology Conf. Berlin, Heidelberg: Springer-Verlag, 2014. 421–439.
- [16] Andrychowicz M, Dziembowski S, Malinowski D, Mazurek L. Secure multiparty computations on bitcoin. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. San Jose: IEEE, 2014. 443–458.
- [17] Huang H, Chen XF, Wu QH, Huang XY, Shen J. Bitcoin-Based fair payments for outsourcing computations of fog devices. Future Generation Computer Systems, 2016.
- [18] Bistarelli S, Mantilacci M, Santancini P, Francesco S. An end-to-end voting-system based on bitcoin. In: Proc. of the Symp. on Applied Computing. New York: ACM, 2017. 1836–1841.
- [19] Micali S, Rabin MO. Cryptography miracles, secure auctions, matching problem verification. Communications of the ACM, 2014, 57(2):85–93.
- [20] Osborne M. An Introduction to Game Theory. New York: Oxford University Press, 2004.
- [21] Narayanan A, Bonneau J, Felten E, Miller A, Goldfeder S. Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction. Princeton: Princeton University Press, 2016.
- [22] Pedersen TP. Non-Interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum J, ed. Proc. of the Int'l Cryptology Conf. Berlin, Heidelberg: Springer-Verlag, 1991. 129–140.
- [23] Carbunar B, Tripunitara M. Fair payments for outsourced computations. IEEE Trans. on Parallel & Distributed Systems, 2010,23(2): 1–9.
- [24] Chen XF, Li J, Susilo W. Efficient fair conditional payments for outsourcing computations. IEEE Trans. on Information Forensics & Security, 2012,7(6):1687–1694.

#### 附中文参考文献:

- [1] 薛锐,吴迎,刘牧华,张良峰,章睿.可验证计算研究进展.中国科学:信息科学,2015,45(11):1370–1388.



尹鑫(1992—),女,山东莱芜人,硕士,CCF 学生会员,主要研究领域为密码学,安全协议.



王海龙(1993—),男,硕士,CCF 学生会员,主要研究领域为密码学,安全协议.



田有亮(1982—),男,博士,教授,主要研究领域为算法博弈论,密码学与安全协议,大数据隐私保护.