中国科学: 信息科学 2021 年 第 51 卷 第 1 期: 75–88
SCIENTIA SINICA Informationis
论文

抗恶意敌手的百万富翁问题解决方案

李顺东*, 王文丽, 杜润萌

陕西师范大学计算机科学学院, 西安 710119
* 通信作者. E-mail: shundong@snnu.edu.cn

收稿日期: 2019–09–24; 修回日期: 2019–11–15; 接受日期: 2019–11–29; 网络出版日期: 2020–12–22

国家自然科学基金 (批准号: 61272435) 资助项目

摘要 安全多方计算是国际密码学界研究的热点, 百万富翁问题是安全多方计算最基础最重要的问题, 是构造其他安全多方计算协议的基本模块. 这个问题已经有许多解决方案, 但除了基于混淆电路的协议之外, 目前基于公钥加密算法的解决方案几乎都是半诚实模型下的解决方案, 抗恶意敌手的解决方案极少, 仅有的个别解决方案效率很低, 这制约着恶意模型下许多安全多方计算问题的解决. 抗恶意敌手的解决方案更符合安全多方计算的实际应用场景, 研究抗恶意敌手的百万富翁问题解决方案, 具有重要的理论与现实意义. 本文首先设计了一个半诚实模型下百万富翁问题的解决方案, 进一步分析了恶意敌手可能的恶意行为, 并用零知识证明和分割选择阻止或发现这些恶意行为, 将半诚实模型下安全的计算协议改造成恶意模型下安全的计算协议, 并用理想 – 实际范例证明了协议的安全性, 分析了恶意敌手攻击成功的概率和方案的效率. 理论分析表明与现有方案相比, 我们提出的方案效率至少提高 6 倍.

关键词 安全多方计算, 百万富翁问题, 恶意模型, 分割 – 选择, 零知识证明, 理想 – 实际范例

1 引言

随着互联网、物联网、云计算与大数据的迅速普及, 利用来自不同数据源的数据进行网络联合计算具有重大的现实意义, 成为计算的常态. 但如不加以防范, 在网络联合计算中数据的隐私与机密就极易泄露, 因此保护联合计算中数据的机密与隐私是网络联合计算面临的一个严峻挑战. 安全多方计算是实现任意函数合作保密计算的主要工具. 设计实现任意函数的安全计算协议是现代密码学的一项主要工作 [1], 也是密码学家一直在努力追求的目标.

安全多方计算是图灵奖获得者 Yao [2] 以百万富翁问题而引入的, 已经成为网络空间联合计算中机密与隐私保护的关键技术, 也成为国际密码学界研究的热点. 在近几年的三大密码学顶级会议上, 安

引用格式: 李顺东, 王文丽, 杜润萌. 抗恶意敌手的百万富翁问题解决方案. 中国科学: 信息科学, 2021, 51: 75–88, doi: 10.1360/SSI-2019-0226
Li S D, Wang W L, Du R M. Protocol for millionaires' problem in malicious models (in Chinese). Sci Sin Inform, 2021, 51: 75–88, doi: 10.1360/SSI-2019-0226

全多方计算都是论文最多的研究方向之一[3]. 图灵奖获得者 Goldwasser[4] 以及著名密码学家 Cramer 等[5] 都预言具有丰富理论基础和广阔应用背景的安全多方计算将成为计算科学一个不可分割的组成部分和新的威力强大的工具.

自 Yao 提出安全多方计算问题之后, Goldreich 等用心理游戏[6] 和电路计算[1] 方法进行了深入的研究. 他们通过将任意函数的计算转化为两种门电路 ("与" 门和 "非" 门电路) 的计算, 并利用不经意传输和秘密共享设计了这两种 "门" 电路的保密计算协议, 证明了在增强型限门置换存在的条件下, 任意函数的安全多方计算都是可能的. 但这样的解决方案只是从理论上证明安全计算任意函数的可能性, 并不能用来解决具体的安全多方计算问题. 因为除了一些简单的布尔运算之外, 将任意函数计算转化为电路计算并不是一件容易的事[7]; 即使能够转化, 这样的方案也因为计算复杂性太高而不实用. 因而 Goldreich 又指出, 用这样的方案来解决实际的安全多方计算问题是不可行的, 要高效地解决各种各样的安全多方计算问题, 应该针对具体应用场景的条件提出有针对性的解决方案.

在 Goldwasser 的预言与 Cramer 和 Goldreich 观点的激励下, 密码学家研究了各个应用领域中广泛出现的基本安全多方计算问题, 这些问题包括保密的科学计算[2,8~10]、保密的数据挖掘[11,12]、保密的统计分析[13,14]、保密的计算几何[15,16]、保密计算应用等[17,18]. 这些研究不仅使大量的安全多方计算问题得到切实可行的解决, 而且推动了安全多方计算理论与实际应用的发展. 目前安全多方计算不但是密码学的一个热点问题, 也已经投入实际商业应用[17].

安全多方计算有两个重要的应用场景, 一种是假设参与者都是半诚实的, 另一种是假设某些参与者是恶意的. 半诚实参与者会老老实实地执行协议, 但也会记录自己收到的所有信息以及自己在计算过程中所用的随机数, 在协议执行后对自己的数据和从协议中获得的数据进行分析, 试图推导出其他参与者的隐私信息, 这种行为称为被动攻击. 这种被动攻击行为只发生在协议执行之后[1]. 半诚实模型下的解决方案是设计恶意模型下解决方案的基础, 研究半诚实模型下的解决方案有重要的理论与实际意义. 但在很多实际应用场景中, 参与者并不都是半诚实的, 仅仅研究半诚实模型下的安全协议是不够的.

另一种情况是假设某些攻击者是恶意的 (恶意攻击者不超过参与者的一半), 在执行协议的过程中恶意攻击者可以主动采取任何可能的攻击行为. 对于恶意攻击者安全的协议, 一般来说对于半诚实参与者也是安全的, 因此这种模型更符合实际, 也更具有普适性, 研究这种模型下的安全多方计算协议具有更重要的理论意义和实际意义. 要解决实际网络联合计算中的隐私保护问题与机密保护问题, 迫切需要研究对恶意参与者安全的多方计算协议, 但由于设计恶意模型下安全协议的困难性, 目前这方面的研究还很少, 许多问题在恶意模型下仍然是公开问题.

设计对于恶意参与者安全的多方计算协议的一个普遍的方法是先设计出对于半诚实参与者安全的协议; 分析恶意参与者对于这样的协议可能采取什么攻击行为, 再针对这些恶意行为设计防范措施从而使半诚实模型下的安全协议成为恶意模型下安全的协议[1].

所谓百万富翁问题就是两个百万富翁 Alice 和 Bob 要保密比较他们财富多少的问题, 可以抽象为保密比较两个数的大小问题. 百万富翁问题是安全多方计算最基础、最重要的问题, 是构造其他安全多方计算协议的基础模块[19]. 因此构造恶意模型下百万富翁问题的解决方案具有重大的理论与现实意义, 本文旨在解决恶意模型下的百万富翁问题. 本文的贡献如下.

(1) 本文首先设计一种非常简单的半诚实模型下的百万富翁问题协议, 分析了该协议的安全性以及恶意攻击者可能实施的恶意行为.

(2) 在半诚实模型下百万富翁问题协议的基础上, 利用零知识证明和分割选择 (cut-and-choose) 技术设计了恶意模型下的百万富翁问题协议. 该协议极其简单、极容易理解. 非常适合作为构造其他安

全多方计算协议的基本模块.

(3) 分析了协议中恶意参与者欺骗成功的概率, 在真实 – 理想框架下证明了方案的安全性, 这是迄今为止为数不多的恶意模型下安全的百万富翁问题协议. 分析了协议的效率, 理论分析表明本文提出的协议的效率至少比现有协议提高 6 倍.

本文其余部分组织如下: 第 2 节综述了百万富翁问题的研究现状; 第 3 节介绍了构造安全协议需要的一些基本知识与协议安全性的定义; 第 4 节介绍了一个半诚实模型下安全的百万富翁问题协议; 第 5 节提出了恶意模型下安全的百万富翁问题协议并证明了其安全性; 第 6 节是协议的性能分析; 第 7 节是本文的结论.

2 有关工作

百万富翁问题在用混淆电路 (garbled circuit) 解决的情况下有对恶意参与者安全的协议 [20~22], 这种协议采用分割 – 选择的方法, 需要 Alice 构造并发送许多电路; Bob 随机选择其中的一半电路, 要求 Alice 打开这些电路以检查其正确性; 最后利用另一半没有打开的电路进行保密计算, 并输出多数的结果. 但这种方法大大增加了计算复杂性、空间复杂性与通信复杂性 [20, 23], 而且构造电路本身也是一个计算复杂性很高的工作. 文献 [24,25] 致力于降低基于电路的协议的计算复杂性与通信复杂性, 使这种方法切实可行. 基于电路的协议也很难作为基本模块嵌入其他协议中, 因此除了一些简单的布尔函数之外, 要用混淆电路方法解决其他安全多方计算问题, 实际上还是不可行的.

文献 [26] 用门限解密的 lifted ElGamal 密码系统将 x, y 分比特加密, 然后利用零测试 (测试一组密文中是否有 0 的密文) 和批量相等测试 (一次测试多个数据是否相等) 通过逐位比较来判定 x, y 的大小. 提出的协议是可以验证的, 一定程度上能够防止恶意参与者的恶意行为, 但并没有给出恶意模型下的安全性定义, 也没有证明协议对于恶意参与者是安全的.

文献 [27] 研究的整数比较问题的输入是 x, y 的各个比特的密文 $E(x_i), E(y_i)$, 参与计算的双方都不知道 x, y, 输出也是一个密文即如果 $x > y$, 输出 $E(1)$, 如果 $x \leqslant y$, 则输出 $E(0)$. 文献证明了协议对于恶意参与者是安全的, 但这里的应用场景与百万富翁问题的应用场景完全不同.

文献 [28] 研究在一种特殊的应用场景中两个数的比较, 该方案需要两个半诚实的辅助计算者共享参与者的隐私数据, 并且提出辅助计算者是恶意的情况下如何改造协议使其对于主动攻击也是安全的. 这与我们的应用场景不同, 所提出的方案不适合本文的应用场景. 文献 [29] 改进了文献 [28] 的工作, 同样是用逐位比较的原理来比较两个数的大小, 该文提出了适合于明文输入和密文输入的协议, 适合于明文输入的协议与本文解决的问题相同. 最后说明可对文献 [28] 的方法进行改进使其成为对于恶意敌手也安全的协议, 但计算复杂性会大大提高, 文中并没有分析计算复杂性提高到什么程度. 本文拟设计高效的恶意模型下的百万富翁问题协议, 最后将本文设计的恶意模型下的安全计算协议与相关协议的效率进行了比较, 与现有协议的比较表明本文协议的效率至少提高 6 倍.

3 预备知识

构造本文的协议需要用到 Paillier 密码系统. 证明协议的安全性需要根据安全性的定义进行. 因此本节介绍 Paillier 门限密码系统以及安全多方计算协议安全性的定义.

3.1 Paillier 密码系统

Paillier 密码系统 [30] 的构造过程如下: 给定一个安全参数 k, 生成两个 k 比特的素数 p, q. 计算 $N = pq, \lambda = \mathrm{lcm}(p-1, q-1)$, 选择一个 g 使得 $\gcd(L(g^\lambda \bmod n^2), n) = 1$, 其中 $L(x) = \frac{x-1}{n}$, 公钥为 (g, N), 私钥为 λ.

要加密消息 $m \in Z_N$, 选择一个随机数计算 $c = g^m r^N \bmod N^2$.

要解密密文 $c \in Z_{N^2}$, 计算

$$m = \frac{L(c^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N.$$

Paillier 密码系统是概率密码系统, 也是语义安全的, 即任何一个明文 m 可以被加密成许多不同的密文, 没有多项式时间算法能够判定这些密文是不是能够解密成同一个明文.

Paillier 密码系统有一个重要的性质即加法同态性, 给定任意的两个密文:

$$c_1 = E(m_1) = g^{m_1} r_1^N \bmod N^2, \quad c_2 = E(m_2) = g^{m_2} r_2^N \bmod N^2,$$

则

$$c = c_1 c_2 = g^{m_1+m_2}(r_1 r_2)^N \bmod N^2 = E(m_1 + m_2),$$

$$c_1^k \bmod N^2 = (g^{m_1} r_1^N)^k \bmod N^2 = g^{km_1}(r_1^k)^N \bmod N^2 = E(km_1).$$

这个性质是基于 Paillier 加密算法的安全多方计算协议的基石.

3.2 证明离散对数相等

这里介绍证明离散对数相等的思想来自文献 [31] 并经过修改, 在恶意模型下的百万富翁问题协议中用于防止欺骗.

令 G 是一个阶数为 m 但 m 未知的循环群, g 是其生成元, h 是 G 中的一个元素. $\alpha = g^x, \beta = h^x$. 现在 Alice 要向 Bob 证明 $\log_g \alpha = \log_h \beta$ 但不泄露 x. 可如下证明.

(1) Bob 在 G 中随机选择一个 r, 计算 $X = g^r, Y = h^r, e = H(g, h, \alpha, \beta, X, Y)$, 其中 H 是一个单向散列函数, 把 r 发送给 Alice.

(2) Alice 计算 $y = r + e \times x, g^y, h^y$ 并把 g^y, h^y 发送给 Bob.

(3) Bob 验证 g^y, h^y 满足 $H(g, h, \alpha, \beta, g^y/\alpha^e, h^y/\beta^e) = e$ 即可.

正确性. 因为 $g^y = g^{r+ex} = g^r(g^x)^e = g^r\alpha^e = X\alpha^e, h^y = h^{r+ex} = h^r(h^x)^e = g^r\beta^e = Y\beta^e$, 所以 $g^y/\alpha^e = X, h^y/\beta^e = Y$, 进而推出 $H(g, h, \alpha, \beta, g^y/\alpha^e, h^y/\beta^e) = H(g, h, \alpha, \beta, X, Y) = e$.

这个证明的原理在于即使不知道 x, 给定 g^x, h^x, 可以计算 e, 也可以计算 g^{ex}, 但如果不知道 x, 就无法从 g^e 计算 g^{ex}, 也就无法计算这样一个 y, 使得 $g^y/\alpha^e = X, h^y/\beta^e = Y$. 反过来, 如果能够计算这样的 y 就表明一定知道 x.

3.3 恶意模型下的安全性

恶意模型是一种更符合实际, 更具有普适性的模型, 一般来说恶意模型下安全的协议对于半诚实模型也是安全的. 恶意模型下安全的协议设计比半诚实模型下安全的协议设计难度大得多. 要证明设计的安全多方计算协议在恶意模型下是安全的, 必须证明它满足恶意模型下安全的定义. 普遍接受的恶意模型下的安全多方计算协议的安全性定义是 Goldreich 在文献 [1] 中给出的定义, 具体如下.

借助可信第三方的理想协议. Steven 和 Tom 分别拥有数据 x 与 y. 他们借助于可信的第三者 (trusted third party, TTP) 计算函数 $f(x,y)=(f_1(x,y),f_2(x,y))$. 执行协议后, 他们分别得到 $f_1(x,y)$ 与 $f_2(x,y)$ 而不泄漏 x 与 y. 协议如下所述.

(1) 输入. Steven 和 Tom 的隐私数据分别为 x 和 y.

(2) 提供给 TTP 的输入. 诚实的参与者总是给 TTP 提供 x, y. 恶意的参与者可能根据 x 或者 y 的情况, 决定不执行协议, 或者决定执行协议但在执行协议时给 TTP 提供一个虚假输入 x' 或 y'.

(3) TTP 给 Steven 发送数据. TTP 得到输入对 (x,y) 后, 独立计算 $f(x,y)$, 并将 $f_1(x,y)$ 发送给 Steven, 否则的话, 给 Steven 发送一个特殊的符号 $\perp$.

(4) TTP 给 Tom 发送数据. 如果 Steven 为恶意参与者, 它可能在收到 $f_1(x,y)$ 后不再理会 TTP. 在这种情况下, TTP 给 Tom 发送一个特殊的符号 $\perp$. 否则, TTP 将 $f_2(x,y)$ 发送给 Tom.

因为参与者除了从 TTP 获得自己相应的 $f_i(x,y)$ 之外, 得不到任何其他信息, 所以理想协议是最安全的协议. 如果一个实际协议能够达到与理想协议相同的安全性, 我们就说这个实际协议是安全的.

设 $F:\{0,1\}^*\times\{0,1\}^*\to\{0,1\}^*\times\{0,1\}^*$ 是一个概率多项式时间函数, $F_1(x,y)$ ($F_2(x,y)$) 表示 $F(x,y)$ 的第一个 (第二个) 元素. 设 $\overline{B}=(B_1,B_2)$ 是表示理想协议中参与者策略的一对概率多项式时间算法. 如果在执行协议过程中至少有一个 B_i ($i\in\{1,2\}$) 对于所有的 u,z,r,v 都有 $B_i(u,z,r)=u, B_i(u,z,r,v)=v$, 其中 u 是 B_i 的输入, z 是其辅助输入, r 是其选择的随机数, v 是从可信的第三者得到的局部输出 $F_i()$, 我们说这样的 $\overline{B}=(B_1,B_2)$ 是可接受的. 在理想模型中参与者拥有辅助信息 z 并以策略 $\overline{B}$ 联合计算 $F(x,y)$ 的过程记为 $\mathrm{IDEAL}_{F,\overline{B}(z)}(x,y)$, 定义为敌手均匀选择一个随机数 r, 令

$$\mathrm{IDEAL}_{F,\overline{B}(z)}(x,y)=\gamma(x,y,z,r),$$

其中 $\gamma(x,y,z,r)$ 定义如下:

- 如果 Steven 是诚实的, 那么有

$$\gamma(x,y,z,r)=(f_1(x,y'),B_2(y,z,r,f_2(x,y'))), \tag{1}$$

其中 $y'=B_2(y,z,r)$.

- 如果 Tom 是诚实的,

$$\gamma(x,y,z,r)=\begin{cases}(B_1(x,z,r,f_1(x',y),\perp),\perp), & \text{如果 } B_1(x,z,r,f_1(x',y))=\perp,\\ (B_1(x,z,r,f_1(x',y)),f_2(x',y)), & \text{否则.}\end{cases} \tag{2}$$

在两种情况下 $x'=B_1(x,z,r)$.

设 Π 是计算 F 的一个双方协议. $\overline{A}=(A_1,A_2)$ 是表示实际模型中参与者策略的两个概率多项式时间算法. 如果在执行协议过程中至少有一个 A_i ($i\in\{1,2\}$) 与 Π 所规定的策略一致, 我们说 $\overline{A}=(A_1,A_2)$ 是关于 Π 可接受的. 特别地, 这个 A_i 忽略其辅助输入. 当输入是 (x,y), 辅助输入为 z, 以策略 $\overline{A}$ 执行实际模型中的协议 Π 的过程记为 $\mathrm{REAL}_{\Pi,A(z)}(x,y)$, 定义为 $A_1(x,z)$ 和 $A_2(y,z)$ 之间交互所产生的输出对.

定义1 (恶意模型的安全性) 如果对于在真实协议中任何可接受的 $\overline{A}=(A_1,A_2)$, 都存在一个在理想协议中可接受的 $\overline{B}=(B_1,B_2)$ 使得

$$\{\mathrm{IDEAL}_{F,\overline{B}(z)}(x,y)\}_{x,y,z}\stackrel{c}{\equiv}\{\mathrm{REAL}_{\Pi,A(z)}(x,y)\}_{x,y,z}, \tag{3}$$

我们就说 Π 安全计算 F. 其中 $x,y,z\in\{0,1\}^*$ 使得 $|x|=|y|$ 并且 $|z|=\mathrm{poly}(|x|)$. 如果不会发生混淆的话, 我们也可以说 Π 是 F 的一个安全实现.

注1 恶意模型下的安全性定义, 蕴含着在两方计算的情况下, 至少有一方是诚实的才能保证协议是安全可行的. 如果两方都是恶意的, 是不可能设计出安全计算协议的 (参见文献 [1] 第 7 章 P606).

4 半诚实模型协议

半诚实模型是安全多方计算一个重要的模型, 目前绝大多数的安全多方计算协议都是在半诚实模型下设计的, 换句话说只有面对半诚实参与者的情况下才是安全的.

Paillier 密码系统的明文空间为 Z_N. $(Z_N,+)$ 构成一个加法群, 在这个群里是没有正负数之分的, 但如果我们限制实际处理的明文 m 都在 $[0,N/2)$ 之内, 即 $m\in\{0,\ldots,N/2-1\}$, 如果 $x+y=0 \bmod N$, 而 $0<x<N/2$, 那么一定有 $y>N/2$. 在这种情况下 y 是 x 的加法逆元, 如果认为 $x\geqslant 0$, 那么 y 就可以看作负数. 进一步假设 $x,y<N/2$, 如果 $(x-y) \bmod N<N/2$, 则 $x>y$; 如果 $(x-y) \bmod N>N/2$, 则 $x<y$. 利用这个原理可以设计出半诚实模型下百万富翁问题的安全多方计算协议.

在文献 [32] 中作者提出了利用复杂的密文比特分解协议, 根据 x,y 的密文 $E(x),E(y)$ 得到一个比特 b 的共享份额 b_1,b_2, 根据 b_1,b_2 可以计算出真正的 b. 协议需要借助一个计算服务提供者 (computation service provider, CSP)、云平台 (cloud platform, CP) 和一个密钥产生中心 (key generation center, KGC). 根据上述 Paillier 加密算法的性质与处理负数的原理, 在文献 [32] 的思想基础上, 我们总结出可以解决半诚实模型下安全的百万富翁问题的协议. 为便于描述, 我们定义函数:

$$F(x,y)=\begin{cases}1, & \text{if } x>y,\\ 0, & \text{if } x=y,\\ -1, & \text{if } x<y.\end{cases}\tag{4}$$

协议 1 对于半诚实参与者 Steven 和 Tom 是安全的, 但如果 Steven 或者 Tom 是恶意的, 协议就不再安全. 第 5 节我们将改进该协议使其成为对于恶意参与者安全的协议.

协议 1 半诚实模型下的百万富翁问题协议

输入: Steven 和 Tom 的输入分别为 x,y.

输出: $F(x,y)$.

准备: Steven 运行 Paillier 加密系统的密钥生成算法, 生成系统的公钥 (g,N) 使得 $x,y<N/2$ (同时保证本协议中选定的随机数 s 满足 $s(x-y)$ 或者 $s(y-x)$ 小于 $N/2$, 这可以通过选择充分大的 N 来保证) 和私钥 λ, 并把公钥发送给 Tom.

1. Steven 用公钥加密 x 得到 $c_1=g^x r_1^N \bmod N^2$, 并把 c_1 发送给 Tom.

2. Tom 选择新的随机数 s,r_2 计算

$$c_2=c_1^s\cdot(g^{sy})^{-1}r_2^N \bmod N^2=g^{s(x-y)}(r_1^s r_2)^N \bmod N^2,$$

把 c_2 发给 Steven.

3. Steven 解密 c_2 得到 $s(x-y)$, 如果 $s(x-y)<N/2$, 则 $F(x,y)=1$; 如果 $N/2<s(x-y)<N$ 则 $F(x,y)=-1$; 如果 $s(x-y)=0$, 则 $F(x,y)=0$.

4. Steven 将 $s(x-y)$ 告诉 Tom.

5 恶意模型下的协议

解决问题的思路. 设计恶意模型下的安全多方计算协议, 通常是首先设计一个半诚实模型下安全的协议, 然后分析恶意的敌手将会如何攻击这个协议, 再设计针对这些攻击的防范措施, 使得敌手的恶意行为无法实施, 或者一旦实施就会被发现, 从而迫使恶意敌手以半诚实的方式参与协议.

首先要清楚有些恶意行为在理想协议中都无法阻止, 恶意模型下安全的协议同样也无法阻止这样的恶意行为. 但恶意模型下安全的协议必须与理想模型协议同样安全, 要保证理想模型下无法实施的恶意行为在设计的协议中也无法实施. 理想协议无法阻止的恶意行为有 3 种 [1]: (1) 拒绝参与协议; (2) 提供虚假的输入, 或者说替换自己的输入; (3) 中途停止协议 (在得到自己需要的信息之后, 阻止其他参与者得到他们需要的信息). 实际协议也不考虑这些恶意行为. 除此之外, 在执行协议时 Steven 和 Tom 可以实施下述恶意行为, 我们无法阻止但必须能够发现.

(1) Steven 能够实施的恶意行为 (此时假设 Tom 是半诚实的) 包括两种情况: (i) 加密 x 需要选择随机数 r_1 的时候, 不选择真正的随机数以便于后面分析出 Tom 的数据. 这种情况可不考虑, 因为不管 Steven 选择什么数, 只要 Tom 选择的是随机数, Steven 就无法从 r_1 得到期望的信息. (ii) 在解密后告诉 Tom 一个错误的 $s(x-y)$ 值的范围, 使 Tom 得到了错误的结论. 协议必须能够发现这种恶意行为.

(2) Tom 可能实施的是恶意行为 (假设 Steven 是半诚实的): Tom 计算 $c_2 = g^{s(x-y)}(r_1^s r_2)^N \bmod N^2$ 的时候选择的 s, r_2 不是随机数. Tom 不可能从 r_2 得到任何信息, 因为解密消除了 r_2 的影响. 如果选择 $s > N/2$, 那么当 Steven 公布 $s(x-y)$ 的范围时, 他就得到正确的结论而 Steven 却得不到正确的结论.

对于 Steven 告诉一个错误的 $s(x-y)$ 的恶意行为, 可以用这样的方法发现这种恶意行为: 要求在协议执行过程中, Steven 用零知识证明的方法向 Tom 证明 $s(x-y)$ 确实是正确的计算结果. 但这样又导致一个新的问题: 如果 Tom 知道了正确的 $s(x-y)$, 就可以根据这个值确定 x 的值.

为了防止这些恶意行为, 我们的思想是: 协议中的随机数 s 由 Steven 和 Tom 共同生成, 使得 s 符合协议的要求, 且 Steven 和 Tom 都不知道 s 的值, 但又能够利用 s 来完成上面的协议, 必要的时候还能够证明计算过程是正确的. 5.1 小节给出具体协议.

5.1 具体协议

具体协议见协议 2.

注2 (1) 协议适合 x, y 的范围不知道的情况; (2) 协议并不能阻止参与者进行欺骗, 但能够发现欺骗. 在执行这个协议时, 如果 Steven 诚实, 他就能够发现 Tom 的欺骗; 如果 Tom 诚实也能发现 Steven 的欺骗. 但如果 Steven 和 Tom 都是恶意的, 理论上已经证明在这种情况下不可能设计出安全的协议 (参见文献 [1] 第 7 章 P606).

5.2 协议的正确性

参数满足 $t_i|x-y| < N_t/2$ 与 $s_i|x-y| < N_s/2$ 时才能保正确性, 理论上这一点很容易做到, 假设 x, y, s_i, t_i 都不超过 τ 比特, 那么只要使得 N_s, N_t 的比特数超过 $2\tau+1$ 比特即可. 协议的第 2 步, 第 3 步就是为了保证满足 $s_i|x-y| < N_s/2$, $t_i|x-y| < N_t/2$. 在满足此条件的前提下, 如果协议在执行过程中没有中止, 说明 Steven 和 Tom 都在以半诚实的方式执行协议. 只要参与者是以半诚实的方式执行协议, 协议的正确性就可以保证.

协议 2 恶意模型下的百万富翁问题协议

输入: Steven 输入 x, Tom 输入 y.

输出: $F(x,y)$.

准备阶段: Steven 和 Tom 分别生成自己 Paillier 密码系统的公钥 $(g_s,N_s),(g_t,N_t)$ 并计算 $u=g_s^{\lambda_s} \bmod N_s^2, v=g_t^{\lambda_t} \bmod N_t^2$. Steven 和 Tom 交换 (g_s,N_s,u) 和 (g_t,N_t,v).

1. Steven, Tom 分别选择 m 个随机数 s_i,t_i $(i=1,\ldots,m)$ (随机数应该有许多小的素因子) 并计算

$$(c_{1s}^i,c_{2s}^i)=(g_s^{s_ix} \bmod N_s^2, g_s^{s_i} \bmod N_s^2),\quad (c_{1t}^i,c_{2t}^i)=(g_t^{t_iy} \bmod N_t^2, g_t^{t_i} \bmod N_t^2),$$

分别公布 $(c_{1s}^i,c_{2s}^i),(c_{1t}^i,c_{2t}^i)$.

2. 利用分割 – 选择的思想, Steven 从 m 组 (c_{1t}^i,c_{2t}^i) 中任意选取 $m/2$ 组 (c_{1t}^i,c_{2t}^i), 要求 Tom 公布对应的 t_iy. Steven 验证 $(t_iy<N_t/2)\wedge(g_t^{t_iy} \bmod N_t^2=c_{1t}^i)$. 如果验证通过则执行下一步, 否则停止协议.

3. Tom 从 m 组 (c_{1s}^i,c_{2s}^i) 中任意选取 $m/2$ 组 (c_{1s}^i,c_{2s}^i), 要求 Steven 公布对应的 s_ix. Tom 验证这些明文都满足 $(s_ix<N_s/2)\wedge(g_s^{s_ix} \bmod N_s^2=c_{1s}^i)$. 如果验证通过则执行下一步, 否则停止协议.

4. Steven 和 Tom 分别从剩下的 (c_{1t}^i,c_{2t}^i) 和 (c_{1s}^i,c_{2s}^i) 随机选择一个 (c_{1t}^j,c_{2t}^j) 和 (c_{1s}^i,c_{2s}^i), 并分别选取 $s\in Z_t^*$ 和 $t\in Z_s^*$. Steven 计算

$$c_t=E_t(st_j(x-y))=(c_{2t}^j)^{sx}(c_{1t}^j)^{-s}r_1^{N_t} \bmod N_t^2=g_t^{st_j(x-y)}r_1^{N_t} \bmod N_t^2,$$

Tom 计算

$$c_s=E_s(s_it(x-y))=(c_{1s}^i)^t(c_{2s}^i)^{-ty}r_2^{N_s} \bmod N_s^2=g_s^{s_it(x-y)}r_2^{N_s} \bmod N_s^2,$$

并将结果发给对方.

5. Steven 计算 $m_s=c_s^{\lambda_s} \bmod N_s^2$, Tom 计算 $m_t=c_t^{\lambda_t} \bmod N_t^2$. 他们将 m_s,m_t 发送给对方.

6. 双方都用零知识证明的方法证明计算是正确的, 即证明 $\log_{c_s} m_s=\log_{g_s} u$ 和 $\log_{c_t} m_t=\log_{g_t} v$. 如果任何一方通不过证明, 则通不过证明的一方是恶意的.

7. 如果都通过证明, Tom 可以计算 $L(m_s)/L(u)$ 得到 $s_it(x-y)$, 进而计算出 $s_i(x-y)$ 并判断出 $F(x,y)$, 但仍然可能被 Steven 欺骗, 后面将分析 Steven 欺骗成功的概率. Steven 可以计算 $L(m_t)/L(v)$ 得到 $st_j(x-y)$, 进而计算 $t_j(x-y)$ 并判断出 $F(x,y)$, 但也可能被 Tom 欺骗, Tom 欺骗成功的概率与 Steven 欺骗成功的概率完全相等).

5.3 协议的安全性

安全性分析. 在这个协议中双方的地位完全相同, 执行的操作也完全相同, 双方的安全地位也完全相同, 因此我们只分析 Steven 的可能恶意行为及其对 Tom 数据的隐私和协议正确性的影响.

(1) 协议第 1 步要求 Steven 在计算 (c_{1s}^i,c_{2s}^i) 时必须都使用相同的 x, 但他可能使用不同的 x. 第 2 步验证也发现不了, 这将导致在第 4 步 Tom 选择的 (c_{1s}^i,c_{2s}^i) 中的 x 不是真实的 x, 这种情况等同于 Steven 更改自己的输入, 因为理想协议也无法避免这种情况, 根据约定不予考虑.

(2) 在第 5 步中 Steven 要零知识证明解密的结果 m_s 是正确的, 这一步无法进行欺骗, 如果剩下的 $m/2$ 组 (c_{1s}^i,c_{2s}^i) 中的 s_ix 也满足 $s_ix<N_s/2$, Tom 选择 $t<N_t/2$, 那么公布 m_s 后, Tom 就可以计算 $F(x,y)$.

(3) 在这个过程中 Steven 唯一可能实施成功的恶意行为就是他选择的某个 s_i 不满足要求, 在第 2 步验证时没有发现, 恰恰在第 4 步又被 Tom 选中, 这样 Tom 将得不到正确的结论. 但 Steven 并不能从中得到 y 的信息, 因为对于他来说 $s_it(x-y)$ 是一个不定方程 (只有一个方程, 但未知数是两个).

经过分析计算, 如果 Steven 要用上述方法进行欺骗, 他的最优选择是在 m 组 (c_{1s}^i,c_{2s}^i) 中只有一组不符合要求, 其他 $m-1$ 组都符合要求, 这样欺骗成功概率最大, 等于 $1/m$. 其他情况都不是最优, 成功的概率更小. 以 $m=10$ 为例, 如果只有一组不符合要求, 欺骗成功的概率为

$$\text{Steven 欺骗成功的概率}=\frac{C_9^5}{C_{10}^5}\times\frac{1}{5}=\frac{1}{10},$$

但如果有 5 组不符合要求, 欺骗成功的概率降为

$$\text{Steven 欺骗成功的概率} = \frac{C_5^5}{C_{10}^5} = \frac{1}{252}.$$

如果选择 $m = 50$ 这两个概率分别降到 1/50 和 7.9×10^{-15}. 如果选择多于一半的组不满足要求, 欺骗成功的概率降为 0 (在验证阶段总会被发现). Tom 可能的恶意行为与欺骗成功的概率完全相同 (这个概率可以自己调整), 因此协议是安全的. 下面我们用理想 – 实际范例证明协议是安全的.

定理1 协议 2, 记作 Π, 对于恶意参与者是安全的.

证明概要 假设在执行 Π 时, 协议双方采取的一个可接受的策略对为 $\overline{A} = (A_1, A_2)$. 要证明协议 Π 对于恶意参与者是安全的, 必须能够将执行 Π 时可接受的策略对 $\overline{A} = (A_1, A_2)$ 转化成理想协议中相应的策略对 $\overline{B} = (B_1, B_2)$, 使得 Π 中 A_1, A_2 的输出与理想模型中 B_1, B_2 的输出计算不可区分. 因为不允许 A_1 和 A_2 同时都不诚实, 所以我们分别处理 A_1 诚实与 A_2 诚实的两种情况.

我们首先分析 A_1 诚实的情况. 在这种情况下, B_1 是协议确定的, 它将根据协议的规定执行协议 (不会中止协议 Π), 并输出协议 Π 规定的输出. 只需要将真实模型敌手 A_2 转化成理想模型敌手 B_2 (因为假设的理想模型敌手 B_2 并不知道 A_2 在面对某个问题时如何决策, 所以只有调用 A_2 才能知道如何决策).

如果在执行协议 Π 时 A_1 是诚实的, 那么

$$\text{REAL}_{\Pi,\overline{A}}(x, y) = \{F(x, A_2(y)), A_2((c_{1s}^i, c_{2s}^i), m_s, S)\},$$

其中 S 是在零知识证明过程中 A_2 收到的消息序列, $i = 1, \ldots, m$. 在理想模型中, 只要我们找到一个可接受的策略对 $\overline{B} = (B_1, B_2)$, 他们的输出与 $\text{REAL}_{\Pi,\overline{A}}(x, y)$ 计算不可区分即可.

(1) 在理想模型中, 因为 B_1 是模仿诚实的 A_1 的行为, 所以它会给 TTP 发送真实的 x (并且自己收到消息后允许给 B_2 发送消息, 所以最后 B_2 一定能够得到消息). 不诚实的 B_2 会给 TTP 发送什么, 则取决于 B_2 的策略, 而 B_2 的策略应该与 A_2 的策略一致, 因此需要调用 A_2 来决定给 TTP 发送什么信息. B_2 把 y 发送给 A_2, 从 A_2 得到 $A_2(y)$, 即实际执行协议时 A_2 使用的隐私数据. B_2 给 TTP 发送 $A_2(y)$, 从 TTP 得到 $F(x, A_2(y))$ (这个结果也给 B_1, 因为只有在 B_1 得到结果而且没有中止协议的前提下, B_2 才能得到结果).

(2) 现在 B_2 要利用从 TTP 得到的 $F(x, A_2(y))$, 设法得到一个与实际执行协议时 A_2 得到的 $\text{view}_{A_2}^{\Pi}(x, A_2(y))$ 计算不可区分的 $\text{view}_{B_2}^{F}(x, A_2(y))$, 并把 $\text{view}_{B_2}^{F}(x, A_2(y))$ 交给 A_2, 输出 A_2 的输出.

- B_2 随机选择 x' 使得 $F(x', A_2(y)) = F(x, A_2(y))$. B_2 以 x' 模拟协议, 即 B_2 装扮成 A_1 与 A_2 执行协议, 给 A_2 发送协议的第 1 步所需要的所有消息.
- B_2 公布协议第 2 步 A_2 要求 A_1 公布的信息.
- B_2 和 A_2 执行协议的剩余部分, 得到相应的 m_s' 并用零知识证明的方法向 A_2 证明 m_s' 计算用的 λ 是正确的, 记录下证明过程中发出的所有消息序列 S'.

(3) B_2 用 $((c_{1s}^{i'}, c_{2s}^{i'}), m_s', S')$ 调用 A_2. 输出 $A_2((c_{1s}^{i'}, c_{2s}^{i'}), m_s', S')$. 这样我们得到

$$\text{IDEAL}_{F,\overline{B}}(x, y) = \{F(x, A_2(y)), A_2((c_{1s}^{i'}, c_{2s}^{i'}), m_s', S')\}.$$

对于 A_2 来说, 因为 $\text{REAL}_{\Pi,\overline{A}}(x, y)$ 和 $\text{IDEAL}_{F,\overline{B}}(x, y)$ 右边的 $c_{1s}^i \overset{c}{\equiv} c_{1s}^{i'}, c_{2s}^i \overset{c}{\equiv} c_{2s}^{i'}, m_s' \overset{c}{\equiv} m_s$ (前面两个 $\overset{c}{\equiv}$ 是因为它们都用同样的概率加密算法加密的密文, 后一个 $\overset{c}{\equiv}$ 是因为它们是由随机数

$s_i(x-A_2(y))$ 和 $s'_i(x-A_2(y))$ 计算出的结果, 而 s_i, s'_i 是计算不可区分的), 零知识证明的系统保证 $S \overset{c}{\equiv} S'$, 所以有

$$\{\mathrm{IDEAL}_{F,\overline{B}}(x,y)\} \overset{c}{\equiv} \{\mathrm{REAL}_{\Pi,\overline{A}}(x,y)\}.$$

现在我们转向 A_2 诚实的情况. 这种情况下, B_2 是协议确定的, 它将根据协议的规定执行协议, 并输出协议 Π 规定的输出. 只需要将真实模型敌手 A_1 转化成理想模型敌手 B_1 即可. 执行协议 Π 的时候 A_1 输出什么完全取决于 A_1 的策略和它获得的 $\mathrm{view}^{\Pi}_{A_1}$, 即根据 A_1 在最后一步是否愿意公布解密的结果并完成零知识证明, 可以分为两种情况. 如果不公布结果或不进行零知识证明 (视为 A_1 中止协议) 时

$$\mathrm{REAL}_{\Pi,\overline{A}}(x,y) = \{A_1((c^i_{1t}, c^i_{2t}), m_t, S), \perp\},$$

其中 S 是在零知识证明过程中 A_1 收到的消息序列, $i = 1, \ldots, m$. 如果公布结果且通过零知识证明, A_2 将收到 $F(A_1(x), y)$, 此时

$$\mathrm{REAL}_{\Pi,\overline{A}}(x,y) = \{A_1((c^i_{1t}, c^i_{2t}), m_t, S), F(A_1(x), y)\}.$$

在理想模型中, B_2 也遵守协议 Π 并输出协议规定的输出结果. B_1 接受输入 x, 局部运行 A_1, 得到 A_1 在实际执行协议时将会发送的信息 $A_1(x)$. B_1 将 $A_1(x)$ 发送给 TTP, 从 TTP 得到 $F(A_1(x), y)$. 如果实际协议中 A_1 公布自己解密的结果并通过最后的零知识证明, 那么在理想模型中 B_1 通知 TTP 也给 B_2 发送结果, B_2 将得到 $F(A_1(x), y)$; 如果实际协议中 A_1 不公布结果或者通不过相应的零知识证明, 在理想模型中 B_1 通知 TTP 不给 B_2 发送结果, B_2 将得到 $\perp$.

B_1 利用 $F(A_1(x), y)$ 调用 A_1 并给 A_1 提供它期望得到的所有消息, 可得到一个 $\mathrm{view}^{F}_{B_1}(A_1(x), y)$, 该 $\mathrm{view}^{F}_{B_1}(A_1(x), y)$ 与 A_1 在执行实际协议时得到的 $\mathrm{view}^{\Pi}_{A_1}(A_1(x), y)$ 计算不可区分.

(1) B_1 随机选择一个 y' 使得 $F(A_1(x), y') = F(A_1(x), y)$, B_1 以 y' 模拟协议, 即 B_1 扮演成 A_2 与 A_1 执行协议. 给 A_1 提供执行协议 Π 所需要的信息, 接收 A_1 发送给它的消息. 在此过程中 B_1 需要生成自己的 Paillier 密码系统的公钥 (这个系统要与 A_2 的密码系统基本相同), 公布对应的 (g'_t, N'_t, v'). 根据 y' 生成对应的 $(c^{i'}_{t_1}, c^{i'}_{t_2})$.

(2) B_1 公布协议第 2 步 A_1 要求 A_2 公布的信息 (B_1 装扮成 A_2) 让 A_1 验证. 接收 A_1 选择的 c'_t.

(3) B_1 计算 m'_t 发送给 A_1, 并用零知识证明计算 m'_t 的过程是诚实的, 即证明 $m'_t = (c'_t)^{\lambda'_t} \bmod (N'_t)^2$.

(4) 到此为止, B_1 得到了 $((c^{i'}_{1t}, c^{i'}_{2t}), m'_t, S')$ (可以不考虑 A_1 后续的恶意行为, 因为此时已经得到了实际执行协议时 A_1 需要的信息, 结合 A_1 的策略, 已经能够决定其输出, 所不同的只是 A_2 是否能得到 $F(A_1(x), y)$). 他可能根据自己的策略不再给 B_2 提供需要的信息 (不解密 c'_s, 或者不进行零知识证明. 这两种行为均视为中止协议).

(5) B_1 用 $((c^{i'}_{1t}, c^{i'}_{2t}), m'_t, S')$ 调用 A_1. A_1 输出什么, B_1 就输出什么, 即输出 $A_1((c^{i'}_{1t}, c^{i'}_{2t}), m'_t, S')$. 如果 (B_1, B_2) 在理想模型中 B_1 通知 TTP 不给 B_2 发送结果时我们定义

$$\mathrm{IDEAL}_{F,\overline{B}}(x,y) = \{A_1((c^{i'}_{1t}, c^{i'}_{2t}), m'_t, S'), \perp\}.$$

如果 (B_1, B_2) 在理想模型中 B_1 通知 TTP 给 B_2 发送结果时我们定义

$$\mathrm{IDEAL}_{F,\overline{B}}(x,y) = \{A_1((c^{i'}_{1t}, c^{i'}_{2t}), m'_t, S'), F(A_1(x) - y')\}.$$

无论哪种情况在实际协议和理想模型中 A_2 和 B_2 的输出是相同的. 只要证明 $((c_{1t}^{i'}, c_{2t}^{i'}), m_t', S')$ 与 $((c_{1t}^{i}, c_{2t}^{i}), m_t, S)$ 是计算不可区分的即可. 而这个不可区分是显然的, 因为 $c_{1t}^{i'}, c_{2t}^{i'}$ 和 c_{1t}^{i}, c_{2t}^{i} 都是用语义安全的 Paillier 加密系统加密得到的密文, 它们是计算不可区分的. m_t 和 m_t' 都是随机数乘以一个常数, 再进行加密与模指数运算得到的数, 仍然是计算不可区分的; 零知识证明保证 $S \stackrel{c}{\equiv} S'$. 所以

$$\{\mathrm{REAL}_{\Pi,\overline{A}}(x,y)\} \stackrel{c}{\equiv} \{\mathrm{IDEAL}_{F,\overline{B}}(x,y)\}.$$

综上所述, 在真实协议中任何可接受的概率多项式时间的算法对 $\overline{A} = (A_1, A_2)$, 都存在一个在理想模型中可接受的概率多项式时间的算法对 $\overline{B} = (B_1, B_2)$ 使得

$$\{\mathrm{IDEAL}_{F,\overline{B}}(x,y)\} \stackrel{c}{\equiv} \{\mathrm{REAL}_{\Pi,A}(x,y)\},$$

因此协议在恶意模型下是安全的.

6 协议效率分析与比较

协议 2 的性能比较. 文献 [26] 通过逐位比较来判定 x, y 的大小. 假设 x, y 均为 L 比特, 则方案需要 $(L+2)(L-1)+2(2L+k(4k-2))+7L = L^2+12L+8k^2-4k-2$ 模指数运算, 其中 k 是一个与 L 无关的小参数. 如果令 $L = 10, k = 2$ 则需要 242 次模指数运算.

文献 [27] 研究的整数比较问题的输入是 x, y 的各个比特的密文, 参与计算的双方都不知道 x, y, 输出也是一个密文, 双方都不知道对应的明文, 即如果 $x > y$, 输出 1 的密文, 如果 $x \leqslant y$, 则输出 0 的密文. 对于 L 比特的 x, y, 协议需要 $124L$ 模指数运算, 其通信复杂性为 6 轮. 协议对于恶意敌手是安全的.

文献 [28] 研究的问题与本文研究的问题应用场景不同, 所提出的方案不适合本文的应用场景. 文献 [29] 改进了文献 [28] 的工作, 同样是用逐位比较的原理来比较两个数的大小, 该文提出了适合于明文输入和密文输入的协议, 适合于明文输入的协议与本文问题的应用场景相同. 半诚实模型下的协议需要 $\frac{1}{2}(L-1)L$ 次模指数运算, 当 $L = 20$ 时, 需要 180 次模指数运算. 经过优化可以减少约 50% 的运算量, 但运算量还是很大的. 最后说明可以用文献 [28] 的方法改造成对于恶意敌手安全的协议, 但计算复杂性会大大提高.

协议 2 中 Steven 和 Tom 需要各自生成 m 组模指数, 各自需要 $2m$ 次模指数运算, 共需要 $4m$ 次模指数运算; 各需要验证 $m/2$ 个模指数, 分别需要 $m/2$ 次模指数运算, 共需要 m 次模指数运算; 一次离散对数的零知识证明需要 6 次模指数运算, 两个参与者各做 1 次离散对数的零知识证明, 共需要 12 次模指数运算; 共需要 $5m+12$ 模指数运算. 模指数运算的次数与比较的数据大小没有关系, 通过简单的分析, 一般 $m = 20$ 就足够了, 在这样的条件下, 协议 2 的效率显著高于现有协议的效率 (这些协议与我们的协议并不具有可比性, 因为应用场景和安全性不同, 但为了说明方案的效率, 可以粗略进行比较), 具体比较如表 1. 表中的第 2 行数据是当 $m = 20, L = 20, k = 2$ 时的模指数运算次数, 文献 [29] 的数据是半诚实模型协议的数据, 恶意模型协议的模指数运算数据将大幅度增加, 其他都是恶意模型下的协议数据.

注3 因为对恶意参与者安全的协议一般都需要用比特承诺、分割选择和零知识证明迫使恶意参与者只能像半诚实参与者一样行事, 否则将会被发现. 增加的比特承诺、分割选择和零知识证明将导致计算复杂性大大提高、协议效率明显降低, 使得抗恶意参与者协议与抗半诚实参与者协议的效率根本无法相比. 我们可以采用预处理的方式或者计算外包的方式提高效率, 且这两种方法在我们的协议

表 1 协议效率比较 (以模指数运算为比较基准)

Table 1 Efficiency comparison of different protocols (the benchmark is modular exponentation)

Protocol 2	Protocol of [26]	Protocol of [27]	Protocol of [29]
$5m+12$	$L^2+12L+8k^2-4k-2$	$124L$	$\frac{1}{2}(L-1)L$(semi-honest)
112	662	2480	190(semi-honest)

中都是可行的, 因为主要的计算就是在分割选择阶段计算 $g_s^{s_i} \bmod p, g_t^{t_i} \bmod p$, 这部分计算和保密数据没有关系, 既可以预计算也可以外包计算. 这么做计算效率至少可以提高一倍.

7 结论

百万富翁问题是安全多方计算领域一个最重要、最基础的问题, 它的性能严重影响以其为基础构建的许多安全多方计算协议的性能. 现有的百万富翁问题协议都是半诚实模型下安全的, 对于恶意敌手是不安全的. 这样我们就无法利用现有协议来构造其他恶意模型下安全的协议, 从而解决其他的安全多方计算问题. 本文利用分割 – 选择方法和零知识证明构造了一个恶意模型下安全的百万富翁问题协议, 解决了安全多方计算领域一个基础性的问题. 未来我们将以此为基础构造更多恶意模型下安全的协议, 解决一些具体的安全多方计算问题, 推动安全多方计算研究的发展. 进一步提高协议的效率, 或者设计更高效的协议也是我们今后工作的方向.

参考文献

1 Goldreich O. Foundations of Cryptography-Volume 2: Basic Applications. London: Cambridge University Press, 2009

2 Yao A C. Protocols for secure computations. In: Proceedings of the 23rd Annual Symposium on Foundation of Computer Science, Chicago, 1982. 160–164

3 Ishai Y, Rijmen V. Advances in Cryptology–EUROCRYPT 2019. Proceedings of the 38th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2019. 97–185, 351–406, 473–561

4 Goldwasser S. Multi party computations: past and present. In: Proceedings of the 16th Annual ACM Symposium on Principles of Distributed Computing, Santa Barbara. 1997. 1–6

5 Cramer R, Damgard I B. Secure Multiparty Computation. London: Cambridge University Press, 2015

6 Goldreich O, Micali S, Wigderson A. How to play any mental game. In: Proceedings of the 19th Annual ACM Symposium on Theory of Computing, New York, 1987. 218–229

7 Kreuter B, Shelat A, Shen C H. Billion-gate secure computation with malicious adversaries. In: Proceedings of USENIX Security Symposium 2012, Bellevue, 2012. 285–300

8 Marszalek Z. Parallel fast sort algorithm for secure multiparty computation. J Univ Comput Sci, 2018, 24: 488–514

9 Freedman M J, Hazay C, Nissim K, et al. Efficient set intersection with simulation-based security. J Cryptol, 2016, 29: 115–155

10 Kissner L, Song D. Privacy-preserving set operations. In: Proceedings of Annual International Cryptology Conference, Santa Barbara, 2005. 241–257

11 Agrawal R, Srikant R. Privacy-preserving data mining. SIGMOD Rec, 2000, 29: 439–450

12 Li S X, Mu N K, Le J Q, et al. Privacy preserving frequent itemset mining: maximizing data utility based on database reconstruction. Comput Secur, 2019, 84: 17–34

13 Du W, Atallah M J. Privacy-preserving cooperative statistical analysis. In: Proceedings of the 17th Annual Computer Security Applications Conference, New Orleans, 2001. 102–110

14 Wang Z B, Pang X Y, Chen Y H, et al. Privacy-preserving crowd-sourced statistical data publishing with an untrusted server. IEEE Trans Mobile Comput, 2019, 18: 1356–1367

15 Atallah M J, Du W. Secure multi-party computational geometry. In: Proceedings of Workshop on Algorithms and Data Structures, Providence, 2001. 165–179
16 Chen Z H, Li S D, Chen L C, et al. Fully privacy-preserving determination of point-range relationship. Sci Sin Inform, 2018, 48: 187–204 [陈振华, 李顺东, 陈立朝, 等. 点和区间关系的全隐私保密判定. 中国科学: 信息科学, 2018, 48: 187–204]
17 Zhao C, Zhao S, Zhao M, et al. Secure multi-party computation: theory, practice and applications. Inf Sci, 2019, 476: 357–372
18 Choudhuri A R, Goyal V, Jain A. Founding secure computation on blockchains. In: Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques, Darmstadt, 2019. 351–380
19 Tang C M, Shi G H, Yao Z A. Secure multi-party computation protocol for sequencing problem. Sci Sin inf Sci, 2011, 41: 789–797 [唐春明, 石桂花, 姚正安. 排序问题的安全多方计算协议. 中国科学: 信息科学, 2011, 41: 789–797]
20 Shelat A, Shen C H. Two-output secure computation with malicious adversaries. In: Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tallinn, 2011. 386–405
21 Canetti R, Poburinnaya O, Venkitasubramaniam M. Equivocating yao: constant-round adaptively secure multiparty computation in the plain model. In: Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, Montreal, 2017. 497–509
22 Lindell Y, Pinkas B. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In: Proceedings of the 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, 2007. 52–78
23 Lindell Y. Fast cut-and-choose-based protocols for malicious and covert adversaries. J Cryptol, 2016, 29: 456–490
24 Ben D A, Nisan B, Pinkas B. FairplayMP: a system for secure multi-party computation. In: Proceedings of ACM Conference on Computer and Communications Security, Virginia, 2008. 257–266
25 Pinkas B, Schneider T, Smart N, et al. Secure two-party computation is practical. In: Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, 2009. 250–267
26 Peng K, Boyd C, Dawson E, et al. An efficient and verifiable solution to the millionaire problem. In: Proceedings of International Conferecnec of Information Security and Cryptology, Seoul, 2004. 51–66
27 Garay J A, Schoenmakers B, Villegas J. Practical and secure solutions for integer comparison. In: Proceedings of International Conference on Theory and Practice of Public Key Cryptography, Harbin, 2007. 330–342
28 Damgård I, Geisler M, Krøigard M. Homomorphic encryption and secure comparison. J Appl Cryptol, 2008, 1: 22–31
29 Veugen T. Improving the DGK comparison protocol. In: Proceedings of IEEE International Workshop on Information Forensics and Security, Costa Adeje, 2012. 49–54
30 Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, Prague, 1999. 223–238
31 Fouque P A, Poupard G, Stern J. Sharing decryption in the context of voting or lotteries. In: Proceedings of International Conference on Financial Cryptography, Anguilla, 2000. 90–104
32 Liu X, Choo K K R, Deng R H, et al. Efficient and privacy-preserving outsourced calculation of rational numbers. IEEE Trans Depend Secure Comput, 2018, 15: 27–39

Protocol for millionaires' problem in malicious models

Shundong LI*, Wenli WANG & Runmeng DU

School of Computer Science, Shaanxi Normal University, Xi'an 710119*, China*
* Corresponding author. E-mail: shundong@snnu.edu.cn

Abstract Secure multiparty computation is a focus of the international cryptographic community. The millionaires' problem is the most important problem in secure multiparty computation and is a building block for constructing other secure multiparty computation protocols. Several solutions are available to solve this problem, but except for protocols based on garbled circuits, the existing solutions based on public key cryptosystems are only secure in semihonest models. No solution based on a public key cryptosystem is secure against malicious adversaries. This state restricts the resolution of many secure multiparty computation problems in malicious scenarios. A solution that is secure in malicious models is highly applicable in practical application scenarios and is generally appealing. Therefore, the study of the solution to the millionaires' problem in a malicious model is of great theoretical and practical significance. In this work, we propose a multiparty computation protocol for the millionaires' problem that is secure in a semihonest model. The proposed protocol is simple and easily understandable. We analyze the possible malicious behaviors in this protocol and use zero-knowledge proof and cut-and-choose techniques to resist possible malicious behaviors and thereby convert the protocol into one that is secure in the malicious model. We prove that the proposed protocol is secure in the malicious model by using the well-accepted ideal-real paradigm. Theoretical efficiency analysis shows that the efficiency of our protocol is at least six times that of existing protocols.

Keywords secure multiparty computation, millionaires' problem, malicious model, cut-and-choose, zero-knowledge proof, ideal-real paradigm

Shundong LI was born in 1963. He received his Ph.D. degree in Computer Science and Technology from Xi'an Jiaotong University, Xi'an, China, in 2003. Currently, he is a professor and Ph.D. supervisor in the School of Computer Science at Shaanxi Normal University. His current research interests include cryptography and information security.

Wenli WANG was born in 1991. She received her master's degree in Mathematics from Shaanxi Normal University, Xi'an, China, in 2019. Currently, she is a Ph.D. candidate in Computer Science at Shaanxi Normal University. Her main research interests include applied mathematics and applied cryptography.

Runmeng DU was born in 1994. Currently, she is a master's degree candidate in Software Engineering at Shaanxi Normal University. Her main research interests include cryptography and information security.