计 算 机 学 报
CHINESE JOURNAL OF COMPUTERS

恶意模型下的最大(小)值保密计算

李顺东 徐雯婷 王文丽 张萌雨

(陕西师范大学 计算机科学学院，西安 710062)

摘 要 安全多方计算是国际密码学界研究的热点，计算一组数据的最大(小)值问题是一个基本的计算问题，保密计算最大(小)值是安全多方计算的一个基础问题，在电子商务、保密招投标、保密数据挖掘等方面有广泛的应用,还可以作为基本模块用于构造更多的安全多方计算协议如各种保密优化协议、保密推荐协议、保密选优协议．目前这个问题的解决方案都只能抵抗被动攻击，尚没有见到能够抵抗主动攻击的解决方案．抵抗被动攻击的解决方案只能提供最基本的安全保障，在有可能遭受主动攻击的实际应用场景中无法保证安全．抵抗主动攻击的解决方案安全性更强，可以为大多实际应用场景提供安全保障，具有重要的理论与实际意义．本文针对保密数据所在范围已知而且范围不太大的应用场景，设计了一种保密数据编码方法，利用这种编码方法构造了抵抗被动攻击的最大(小)值的安全多方计算协议，方案非常简单、极易理解，并利用模拟范例证明了协议对于被动攻击是安全的；通过分析协议可能遭受的主动攻击，利用门限解密的密码系统、结合零知识证明和保密洗牌设计阻止或发现主动攻击的措施，把协议改造成能够抵抗主动攻击的安全协议，并用理想-实际范例证明了协议的安全性；分析了方案的效率并通过实验验证了协议的可行性．就我们所知，这是第一个能够抵抗主动攻击的最大(小)值问题解决方案.

关键词 安全多方计算，最大值，模拟范例，恶意模型，半诚实模型，理想-实际范例，零知识证明

中图法分类号 TP309

Secure Maximum (Minimum) Computation in Malicious Model

LI Shundong XU Wenting WANG Wenli ZHANG Mengyu

(School of Computer Science, Shaanxi Normal University, Xi'an 710062)

Abstract Secure multiparty computation is a key privacy-preserving technology which has extensive applications to preserve the privacy of private data in data sharing. It is also a focus of the international cryptographic community in recent years. It is a basic computing problem to compute the maximum (minimum) of a data set. Therefore, privately computing the maximum (minimum) is a fundamental problem of secure multiparty computation. Secure multiparty maximum (minimum) computation has extensive applications in E-commerce, secure bidding and auction, privacy preserving data-mining, private information retrieval, data sharing and artificial intelligence etc. The protocol for secure maximum (minimum) computation can also be used as a building block to construct other secure multiparty computation protocols such as secure optimization protocol, secure recommendation protocol, secure dynamic programming protocol and secure selection protocol. The existing protocols for this problem are only secure against passive attacks. As far as we know, there is no solution that is secure against active attacks. The protocols that are secure against passive attacks can only guarantee the most basic security. Because security in the semi-honest model is weak security, such protocols cannot be applied in practical scenarios that may suffer from active attacks. The protocols that are secure against active attacks can guarantee stricter security, can be used in more practical scenarios and have theoretical and

本课题得到中国国家自然科学基金(No.61272435)资助．李顺东(**通信作者**)，博士，教授，主要研究领域为公钥密码，密码协议，密码学与信息安全. E-mail: shundong@snnu.edu.cn．徐雯婷，硕士研究生，主要研究领域为信息安全协议设计. E-mail: xuwenting@snnu.edu.cn．王文丽，博士研究生，主要研究领域为密码学与信息安全. E-mail: wenliwang@snnu.edu.cn．张萌雨，硕士研究生，主要研究领域为密码学与信息安全. E-mail: zhangmengyu@snnu.edu.cn

practical importance. In this paper, we first design a new encoding method to encode a private data, and based on this new encoding method, we construct a protocol, denoted by Protocol 1, to privately compute the maximum (minimum) of private data set, which is secure in the semi-honest model. Protocol 1 is very simple and easily understood. We proved that Protocol 1 is secure against passive attacks by using the well-accepted simulation paradigm; we analyze the malicious attacks that the protocol may suffer from, and use threshold decryption cryptosystem, secure shuffling and zero-knowledge proof to prevent possible active attacks to convert the protocol into two protocols that are secure again active attacks. The first protocol that is secure against active attack, denoted by Protocol 2, is simpler and can be used in the scenarios where who owns the maximum (minimum) should be finally revealed while the privacy of other data should be preserved such as in a commercial auction or bidding scenarios. We prove that the protocol is secure against active attacks by using the ideal-vs-real paradigm; we analyze the efficiency of the protocol and test the feasibility of it. The experimental result shows that Protocol 2 is efficient and practical. The second protocol which is secure against active attack is more complicated and is of higher computational complexity. It uses a secure shuffling proof as a building block to resist some malicious behavior. It is arranged as Appendix A without security proof. This protocol can be used in the scenarios where the privacy of who owns the maximum (minimum) should also be preserved. To the best of our knowledge, these protocols for maximum (minimum) problem that can resist malicious attack are first proposed.

Key words Secure multiparty computation, maximum, simulation paradigm, malicious model, semi-honest model, ideal-vs.-real paradigm, zero-knowledge proof

1 引言

互联网、物联网、云计算与大数据的迅速普及，为人们采集数据、传送数据、利用数据带来了极大的便利，使得人们不但可以利用自己的数据，还可以方便地利用其它实体的数据进行计算、发现数据之间的规律，利用规律进行科学决策，促进科学技术、经济的发展，更好地管理社会．利用不同实体的数据进行联合计算能够取长补短、合作共赢，产生巨大的经济效益与社会效益．例如在流行病爆发但尚没有有效治疗药物的时期，医疗机构和医药研发机构之间的数据合作共享甚至能够拯救大量的生命．利用不同实体拥有的数据进行联合计算在未来的信息技术中将占有越来越重要的地位．

但不同实体所拥有的数据可能包含大量的隐私数据，如果进行联合计算时不进行适当的保护就极易泄露数据的隐私与机密，产生严重的后果．很多联合计算中都存在这个问题．因此保护联合计算中机密数据的机密、隐私数据的隐私是联合计算面临的一个严峻挑战．图灵奖获得者姚期智教授[1]以百万富翁问题引入的安全多方计算是联合计算中隐私保护的关键技术，是近年来国际密码学界研究的热点问题．在近几年的三大密码学顶级会议上，安全多方计算都是论文最多的研究方向之一[1]．图灵奖获得者Goldwasser[2]曾预言安全多方计算将成为计算科学一个必不可少的组成部分．著名密码学家Cramer[3]也曾语言：如果能够保密计算任何函数，计算科学就有了一个新的威力强大的工具．

姚期智教授在提出安全多方计算概念之后，基于不经意传输和电路计算提出了一种通用的安全两方计算工具[4]：混淆电路 (garbled circuit)方法．此后，Goldreich等提出了基于秘密共享和算术电路的通用安全多方计算协议[5,6]．但由于效率等多方面的限制，这些方法并不适合用于解决多数实际的安全多方计算问题．因此Goldreich[6]认为期望利用通用解决方案解决具体的安全多方计算问题是不切实际的，应该针对具体的问题设计具体的解决方案．

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出现的安全多方计算问题，这些问题包括保密的科学计算[1,7-10]、保密的数据挖掘[11,12]、保密的统计分析[13,14]、

[1] 参看近几年这些密码学顶级会议的论文集

保密的计算几何[15,16]、保密计算应用等许多问题[17-19]. 使一系列的保密计算问题获得解决、推动安全多方计算的发展.

通常人们在两种应用场景中研究安全多方计算, 一种是假设安全多方计算协议只可能遭受被动攻击, 即假设协议的所有参与者都是半诚实的, 又称半诚实模型下的安全多方计算; 另一种是假设安全多方计算协议可能遭受任意的主动攻击, 即假设协议的参与者是恶意的, 又称为恶意模型下的安全多方计算. 在半诚实模型中, 假设协议参与者在执行协议的过程中将会忠实地履行协议, 但是他们也可能会记录执行协议过程中自己收到的所有信息, 并试图(可能合谋)从这些信息中推导出其他参与者的隐私信息, 也就是说这种情况下参与者不会对协议发动主动攻击. 在这种应用场景中安全的协议只能提供最基本的安全保障, 防止因为疏忽大意而导致的隐私泄露. 但很多情况下参与者并不是半诚实的, 半诚实模型下安全的协议无法保障这些应用场景中的数据隐私.

在恶意模型中, 假设某些参与者是恶意的, 在执行协议的过程中可以采取任何可能的恶意行为对协议进行主动攻击. 显然对于恶意参与者安全的协议, 对于半诚实参与者也一定是安全的, 因此恶意模型更符合实际, 这种模型下安全的协议能够提供更强的安全保障, 研究恶意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与机密保护问题, 迫切需要研究恶意模型下的安全多方计算协议, 但是设计恶意模型下的安全多方计算协议要困难得多. Goldreich曾设计一个编译器[6], 该编译器可以将半诚实模型下安全的安全多方计算协议编译成恶意模型下安全的协议. 因为恶意模型协议都需要用到零知识证明且不同协议中需要的零知识证明不同, 编译器只好将任意零知识证明归约到一个NP完全(NPC)问题的零知识证明, 并用NPC问题的零知识证明解决问题, 而这样做的效率太低, 实际上用这样的方法得到恶意模型下安全的协议是不可行的. 恶意模型下的安全多方计算问题的研究还比较少, 相应的安全多方计算协议更少, 许多问题在恶意模型下的安全多方计算问题都还是公开问题.

计算一组数据的最大(小)值是一种基本的数据操作, 有极其广泛的应用, 因为人们做任何事情都希望收益最大或者成本最小, 这就需要在各种可能收益(成本)中确定最大(小)值. 拍卖就是要获得最大效益, 招标则是希望以最小的成本完成某项工作, 所有的优化[20]都是要选择收益最大或成本最小的方案. 如果这些工作的数据涉及到隐私数据就需要保密计算, 比如在保密拍卖或者招标中, 只有出价最高(低)的人才可以获得拍卖的商品(赢得招标), 而其他不能获得商品(落标)的人一般不愿意泄露自己的出价, 以免在今后的拍卖或者招标中处于不利地位[21]. 因此最大(小)值保密计算问题是安全多方计算的一个基础问题, 也是构造其他安全多方计算问题解决方案的基本模块, 在电子商务、保密招投标、保密数据挖掘等方面有广泛的应用, 还可以作为基本模块用于构造保密选拔、保密推荐、保密优化等协议. 通常要计算最大值, 不得不对数据进行两两比较, 但这样做会泄露很多信息. 而最大(小)值的安全多方计算要求只能泄露最大(小)值及其持有者的信息, 而不泄露能除此之外的任何信息. 不对数据进行两两比较而计算最大值是非常困难的, 是最大值计算的面临的挑战与难点, 因此保密计算一组数据的最大值也是一个比较困难的问题, 即使半诚实模型下的解决方案也非常有限[22-30].

1.1 有关工作

文献[22]利用分片混合技术和二分查找技术设计保密查询协议, 协议支持最大值的保密计算. 该文提出的协议需要一个不可信的数据聚合服务器负责为每个参与者分配一个基于身份信息的密钥, 并负责收集数据. 协议需要安全信道, 即使数据聚合服务器不参与合谋, 也只能抵抗30%的其他参与者的合谋攻击. 如果数据聚合服务器与任何一个参与合谋都可以得到很多信息. 协议的通信复杂性很高, 协议的安全性使其无法适用于典型的安全多方计算应用场景.

文献[23]基于加法同态加密算法和基于HMAC的密钥管理技术提出了一个时间序列数据的保密求和协议, 协议可以用于保密计算时间序列数据的最大值. 该文提出的协议也需要一个可信的权威服务器和一个不可信的数据聚合者, 方案只适用于时间序列数据的聚合. 文献也没有给出安全性证明.

文献[24]假设任何两个参与者之间从不通信, 并且有一个权威生成一些随机数 $s_1,\cdots,s_n\in\{0,1\}^\gamma$, 并将 $s_i,s_{i\bmod n+1}$ 发送给 P_i. 为计算最大值的每一比

特，权威要为所有参与者生成一个新鲜的随机数(nonce). 协议完全不考虑参与者之间合谋的可能，即使在典型的半诚实模型下也不能保证安全. 任何两个参与者合谋就可以得到处于他们之间的参与者数据的全部信息. 如果权威参与合谋，协议就没有任何安全性可言. 方案也会发生微小的错误.

文献[25]研究无线传感网络节点数据的最大(小)值查询协议，在这个协议中传感网络的拥有者拥有所有节点的密钥，即协议要借助一个权威的第三方. 协议逐比特确定最大(小)值，但是某些节点会获得其他一些节点的私密数据，安全性不高，无法在典型的安全多方计算场景中应用. 文献[26]设计了一个数据范围已知且范围不大的场景中的安全多方计算协议，协议利用两两比较的方法来确定最大值，协议也需要有一个生成密钥的权威.

上述的解决方案都需要一个可信或者不可信的第三者帮助实施保密计算，也都是为移动的智能设备如手机应用环境而设计的，不适用于标准的安全多方计算场景，在标准的安全多方计算环境使用是不安全的. 典型的安全多方计算场景中，各参与方地位完全相等，互不信任. 因此必须用参与者之间的交互代替可信或者不可信的数据聚合者和权威服务器. 在典型的安全多方计算场景中保密计算一组数据的最大值还是一个有待解决的问题.

文献[27]利用秘密共享方法构造了一个最大值保密计算协议，协议假设数据在一个范围之内，并逐比特确定最大值，协议也需要安全信道，通信复杂性较高. 文献[28]利用ElGamal加密算法的性质设计了保密数据范围已知，且范围较小时保密计算最大(小) 值的协议. 文献[29]利用保密替换技巧设计协议，降低了协议的计算复杂性. 这两个协议在半诚实模型下是安全的，在恶意模型下不安全. 此外，还有在密文数据上查询数据并返回Top-k数据的协议[30]可以在密文数据库中查询到密文数据的最大值，这和本文的应用场景不同. 目前还没有见到恶意模型下的最大(小)值保密计算方案. 因此构造恶意模型下最大(小)值的解决方案具有重要的理论与现实意义. 本文针对保密数据范围已知，而且范围不太大的情况(这种情况在实际中是很常见的，例如保密数据为年龄、血压、身高、体重、体温、学生分数、人数、速度、收入等都属于这种情况)首先设计了一种保密数据编码方法，利用这种方法设计了半诚实模型下安全的协议，在此基础上设计了恶意模型下安全的协议. 本文的主要贡献如下:

1. 提出了一种新的编码方法，利用此编码该方法设计了半诚实模型下安全的最大(小)值保密计算协议，协议不需要两两比较，因此是一种方法的创新，可用于解决许多新问题. 用广泛接受的模拟范例证明了协议在半诚实模型下是安全的. 协议可以作为基本的模块用于构造其他的安全多方计算解决方案.

2. 分析了恶意参与者对半诚实模型下安全的协议可能实施的攻击，利用门限解密的密码系统、结合零知识证明和保密洗牌证明设计阻止或发现恶意攻击的措施把该协议改造成恶意模型下安全的协议.

3. 用理想-实际范例证明了协议在恶意模型下是安全的. 分析了协议的效率，并通过实验验证了方案的可行性. 据我们所知，这是迄今为止第一个在恶意模型下安全的最大(小)值保密计算协议.

本文其余部分组织如下: 第二部分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; 第三部分介绍了编码方法和计算最大(小)值的原理，构造了半诚实模型下安全的最大(小)值计算协议，并用模拟范例证明了协议的安全性; 第四部分分析了恶意参与者可能实施的攻击和防范措施，将半诚实模型下安全的协议改造成了恶意模型下安全的协议，并用理想-实际范例证明了协议的安全性; 第五部分分析了协议的效率，并通过实验验证协议的可行性.

2 预备知识

构造本文的协议需要用到ElGamal门限密码系统，证明协议的安全性需要根据安全性的定义进行，阻止某些恶意行为需要零知识证明. 因此本部分介绍ElGamal门限密码系统，安全多方计算协议安全性的定义以及离散对数的零知识证明.

2.1 ElGamal公钥密码系统

ElGamal密码系统[31]的构造如下: 选择一个大素数 p，为保证安全 $p-1$ 至少要有一个大的素因子，g 是 Z_p^* 的生成元. 选择一个随机数 x 作为私钥，计算 $h=g^x \bmod p$ 作为公钥. 要加密消息 $m\in Z_p$，选择一个随机数 $r\in Z_p^*$，并计算

$$C=(c_1,c_2)=(g^r \bmod p, mh^r \bmod p). \tag{1}$$

要解密 $C=(c_1,c_2)$，只要把 $h^r \bmod p$ 消除即可.

因为 $h^r=(g^x)^r=g^{xr}\bmod p$，而 $c_1^x=(g^r)^x=g^{xr}\bmod p$，所以只要计算 $c_2(c_1^x)^{-1}\bmod p$ 即可以消除 $h^r\bmod p$ 恢复 m. 因此，解密可以表示成

$$m=c_2(c_1^x)^{-1}\bmod p.$$

ElGamal密码系统具有乘法同态性，即将 $C_1=(g^{r_1},m_1h^{r_1})$ 与 $C_2=(g^{r_2},m_2h^{r_2})$ 相乘可以得到一个新的密文 $C=(g^{r_1+r_2},m_1m_2h^{r_1+r_2})$，将这个结果解密得到的明文恰是 m_1m_2，即 $E(m_1)E(m_2)=E(m_1m_2)$.

ElGamal密码系统是概率密码系统，同一个明文可以加密成许多密文，这些密文都解密为同一个明文. 一个密文可以在不知道解密密钥的条件下转化为同一个明文的不同密文，只要利用算法的同态性，给密文乘以一个1的密文就可以实现这个功能，这个过程称为密文的重随机化或者密文刷新.

2.2 门限ElGamal公钥密码系统

ElGamal公钥密码系统的一个重要优势是可以在不需要分发者的情况下很方便地构造门限解密的ElGamal公钥密码系统. 门限解密的公钥密码系统可以由 n 参与者共同生成一个公钥，而私钥则由这 n 个参与者共享. 这样的密码系统中，至少需要 t 个人合作才能解密一个密文，少于 t 个人就不能解密，得不到关于明文的任何信息，这样的密码系统称为 (t,n) 门限密码系统. 门限密码系统是安全多方计算中对抗合谋攻击与欺骗的一种重要工具. (t,n) 门限密码系统可以抵抗 $t-1$ 个参与者的合谋攻击，我希望有 n 个参与者的安全多方计算协议要能够抵抗 $n-1$ 个参与者的合谋，所以安全多方计算需要平凡的门限密码系统，即 (n,n) 门限密码系统. (n,n) ElGamal门限密码系统可以这样构造:

首先，所有 n 个参与者选择一个大素数 p，一个生成元 $g\in Z_p^*$. 每一个参与者选择一个随机数 $sk_i\in Z_p^*$ 作为自己的私钥份额，计算并公布 $h_i=g^{sk_i}\bmod p$，则公钥为

$$h=\prod_{i=1}^{n}h_i=g^{sk_1+\cdots+sk_n}\bmod P. \tag{2}$$

加密计算与原始的ElGamal算法完全相同，即选择一个 $r\in Z_p^*$ 并计算 $c=(g^r\bmod p, mh^r\bmod p)$. 但是要解密一个密文 $c=(c_1,c_2)$ 需要每一个参与者 P_i 计算并公布 $c_1^{sk_i}\bmod p$，然后每个参与者都可以解密得到明文

$$m=c_1(\prod\nolimits_{i=1}^{n}c_1^{sk_i}\bmod p)^{-1}\bmod p$$

2.3 半诚实模型下的安全性

半诚实模型又称诚实但好奇模型或被动攻击模型，是一种重要的安全多方计算模型，在这种模型下每个参与者的行为与协议的要求保持一致，但会保留计算过程中收到的所有信息，并在协议执行后试图利用这些信息获得更多其他参与者的隐私信息，这种攻击称为被动攻击，它只发生在协议执行之后. 简单地说，如果参与者的任意合谋者集合在执行协议中得到的信息本质上都可以从他们的输入和输出直接得到，我们就说协议是安全的. 半诚实模型下协议安全性定义如下[6]:

定义 1. (半诚实行为下的安全性): 设 $\underline{f}:(\{0,1\}^*)^n\to(\{0,1\}^*)^n$ 是一个 n 元函数，记 $\underline{x}=(x_1,\cdots,x_n)$，则 $f(\underline{x})=(f_1(\underline{x}),\cdots,f_n(\underline{x}))$. 参与者 P_i 拥有保密数据 x_i，他们希望得到 $f(\underline{x})$ 的第 i 个元素 $f_i(\underline{x})$，而不愿意泄露 x_i. 设 Π 是一个计算 f 的 n 方协议. 对任意 $I=\{i_1,\cdots,i_t\}\subseteq\{1,\cdots,n\}$，令 $f_I(\underline{x}))=(f_{i_1}(\underline{x}),\cdots,f_{i_t}(\underline{x}))$. 执行协议 Π 的过程中第 i 个参与者得到的信息序列为 $view_i^{\Pi}(\underline{x})=(x_i,r_i,m_i^1,\cdots,m_i^t)$，其中 r_i 表示第 i 个参与者的随机数，m_i^j 表示第 i 个参与者收到的第 j 个消息. 令

$$view_I^{\Pi}(\underline{x})=(view_{i_1}^{\Pi}(\underline{x}),\cdots,view_{i_t}^{\Pi}(\underline{x})).$$

对于协议 Π，如果存在概率多项式时间算法 S 使得

$$\{S(I,(x_{i_1},\cdots,x_{i_t}),f_I(\underline{x}))\}_{\underline{x}\in(\{0,1\}^*)^n}\overset{c}{\equiv}\{view_i^{\Pi}(\underline{x})_{\underline{x}\in(\{0,1\}^*)^n}\} \tag{3}$$

成立，我们说 Π 保密计算 f，其中 $\overset{c}{\equiv}$ 表示计算不可区分.

要证明一个安全多方计算协议是安全的，就必须构造出使得(3)成立的模拟器 S.

2.4 证明离散对数相等

这里介绍证明离散对数相等的方法来自文献[32]，在恶意模型下的最大(小)值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,h 是其生成元，Alice知道 $\alpha=g^x,\beta=h^x$. 现在Alice要向Bob证明 $\log_g\alpha=\log_h\beta$ 而不泄露 x. 可以这么证明:

Alice在 $[0,A]$ 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 是一个散列函数，其值的范围为 $[0,B]$. 计算 $y=r+e\times x$. 只要证明 $(e,y)=[0,B]\times[0,A]$，且

$$e=H(g,h,\alpha,\beta,g^y/\alpha^e,h^y/\beta^e). \tag{4}$$

正确性: $g^y = g^{r+ex} = g^r(g^x)^e = g^r\alpha^e \Rightarrow g^y/\alpha^e = X$, $h^y = h^{r+ex} = h^r(h^x)^e = g^r\beta^e \Rightarrow h^y/\beta^e = Y$.

这个证明的原理在于即使不知道 x, 谁都可以算 e, 但如果不知道 x, 就无法计算这样一个 y, 使得 $g^y/\alpha^e = X, h^y/\beta^e = Y$ 成立.

2.5 恶意模型下的安全性

恶意模型的安全定义涉及到实际协议与理想协议的比较, 恶意模型下的理想协议与半诚实模型下的理想协议不同. 在恶意模型下, 在实际协议和理想协议中都有一些恶意行为是无法避免的, 恶意模型下的理想协议也必须考虑如何处理这些无法避免的行为, 因此要比半诚实模型下的理想协议复杂很多.

具体来说无论在理想协议还是实际协议中, 只要参与者不受胁迫, 下面三种恶意行为都无法阻止: (1) 拒绝参与协议; (2) 提供不真实的输入, 或者说替换自己的输入(因为实际数据是保密的, 所以没有办法判断一个参与者的输入是不是真实的保密数据); (3) 中途停止协议. 在实际协议与理想协议中都允许这三种恶意行为[6]. 除此之外的恶意行为都应当被阻止或者被发现.

恶意模型下安全的协议要能够阻止或者发现任何的恶意行为, 因此设计这样的协议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.

恶意模型下的理想协议 设 $P_1,\cdots,P_n$ 分别拥有数据 $x_1,\cdots,x_n$. 他们要借助于可信的第三者(Trusted Third Party-TTP)计算函数 $f(x_1,\cdots,x_n)$. 设恶意参与者集合为 $I=\{i_1,\cdots,i_t\}\subseteq[n]=\{1,\cdots,n\}$, 则 $\bar{I}=[n]\setminus I$ 为诚实参与者集合. 令 $\bar{x}=(x_1,\cdots,x_n), (\bar{x})_I=(x_{i_1},\cdots,x_{i_t})$, $f(\bar{x})=(f_1(\bar{x}),\cdots,f_n(\bar{x}))$, $f_I(\bar{x})=(f_{i_1}(\bar{x}),\cdots,f_{i_t}(\bar{x}))$. 恶意模型下的理想协议如下.

给TTP发送数据: 如果 $i\in\bar{I}, P_i$ 就会给TTP 发送真实的 x_i; 如果 $i\in I, P_i$ 就会根据 x_i 和 I 的策略, 决定不执行协议(不发送数据), 或者同意参与协议但在执行协议时发送虚假的 x_i'.

TTP给 I 的成员发送数据: TTP如果得到了 $\bar{x}$, 就独立计算 $f(\bar{x})$, 并将 $f_I(\bar{x})$ 发送给 I 的所有成员; 如果没有收到足够的数据, 就给 I 的成员发送一个表示协议无法执行的符号 $\perp$.

TTP给 $\bar{I}$ 的成员发送数据: 如果 $P_1\in I$, I 的成员收到 $f_I(\bar{x})$ 后可以决定是否让TTP给 $\bar{I}$ 的成员发送 $f_{\bar{I}}(\bar{x})$ (这个信息由P_1发送). 如果 I 的成员不让TTP给 $\bar{I}$ 发送 $f_{\bar{I}}(\bar{x})$ 或者不给TTP任何指示, TTP 就给 $\bar{I}$ 的成员发送符号 $\perp$; 如果 I 的成员让TTP给 $\bar{I}$ 的发送 $f_{\bar{I}}(\bar{x})$, TTP 就将 $f_{\bar{I}}(\bar{x})$ 发送给 $\bar{I}$ 的成员.

定义2[6]. 假设一个攻击者可以控制 I 中的所有成员, 执行协议时他按照策略 B 行事, 其中 B 是一个概率多项式时间算法, 我们用 (I,B) 表示这个攻击者. 在理想模型中, 在 (I,B) 控制下, 拥有辅助信息 z, 攻击者选择的随机数为 r, 输入为 $\bar{x}=(x_1,\cdots,x_n)$ 的条件下, f 的联合执行记作 $\mathbf{IDEAL}_{f,I,B(z)}(\bar{x})=\Upsilon(\bar{x},I,z,r)$, 其中 $\Upsilon(\bar{x},I,z,r)$ 定义如下:

如果 P_1 是诚实的, 即 $P_1\notin I$ 则

$$\Upsilon(\bar{x},I,z,r)=(f_{\bar{I}}(\bar{x}),B(\bar{x}_I,I,z,r,f_I(\bar{x}))),$$

其中 $\bar{x}=(x_1,\cdots,x_n')$ 且满足如果 $i\in I$ 则 $x_i=B(\bar{x}_I,I,z,r)_i$; 否则的话 $x_i=x_i$.

如果 P_1 不诚实,并且 $B(\bar{x}_I,I,z,r,f_I(\bar{x}))=\perp$, 则

$$\Upsilon(\bar{x},I,z,r)=(\perp^{|\bar{I}|},B(\bar{x}_I,I,z,r,f_I(\bar{x}),\perp)),$$

如果 P_1 不诚实但 $B(\bar{x}_I,I,z,r,f_I(\bar{x}))\neq\perp$ 则

$$\Upsilon(\bar{x},I,z,r)=(f_{\bar{I}}(\bar{x}),B(\bar{x}_I,I,z,r,f_I(\bar{x})))$$

定义3. 恶意模型下的安全性[6].

设 $f:(\{0,1\}^*)^n\to(\{0,1\}^*)^n$ 是一个 n 元函数, Π 是计算 f 的协议. 假设 $I,\bar{I},\bar{x},(\bar{x})_I,f(\bar{x}),f_I(\bar{x})$ 如前面定义. A 是描述实际协议中攻击者策略的多项式时间算法, 我们用 (I,A) 描述攻击者. 当输入为 $\bar{x}$, 辅助输入为 z 时, 在攻击者 (I,A) 的控制下, 协议 Π 的联合执行,记作 $\mathbf{REAL}_{\Pi,I,A(z)}(\bar{x})$, 定义为 n 个参与者之间的交互所产生的输出序列. I 中的参与者的消息由 $A(\bar{x}_I,I,z)$ 决定, $\bar{I}$ 中的参与者的消息则完全由 Π 决定. 具体地说, 恶意参与者的消息由 A 根据所有恶意参与者的初始输入、辅助输入 z 和包括诚实参与者的所有参与者到目前为止所发送的所有消息决定.

如果对于表示实际协议中恶意攻击者策略的任意概率多项式时间算法 A, 都存在一个表示理想协议中恶意攻击者策略的概率多项式时间算法 B, 使得对于任意的 $I\subseteq[n]$ 都有

$$\{\mathbf{IDEAL}_{f,I,B(z)}(\bar{x})\}_{\bar{x},z}\overset{c}{\equiv}\{\mathbf{REAL}_{\Pi,I,A(z)}\}_{\bar{x},z}, \qquad (5)$$

我们说 Π 保密计算函数 f.

3 半诚实模型下的最大值协议

问题描述: 参与者 $P_1,\cdots,P_n$ 分别拥有私密数据

$x_1,\cdots,x_n$，他们要计算 $\max\{x_1,\cdots,x_n\}$，$\min\{x_1,\cdots,x_n\}$ 而不泄露最大(小)值之外的任何其他信息.

计算原理： 假设 $x_1,\cdots,x_n\in\{s_1,\cdots,s_m\}$，其中 $s_1<\cdots<s_m$. 他们可以这么计算最大(小)值.

参与者 P_i 将自己的私密数据 x_i 编码为一个向量

$$V_i=(v_{i1},\cdots,v_{im}). \tag{6}$$

编码原则为: 如果 $x_i=s_j$，则 $v_{ij}=r\in Z_p^*$，其中 r 为不等于1的随机数; 否则，$v_{ij}=1$.

这样 $V_1,\cdots,V_n$ 构成一个矩阵

$$A=\begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1m}\\ v_{21} & v_{22} & \cdots & v_{2m}\\ \cdots & \cdots & \cdots & \cdots\\ v_{n1} & v_{n2} & \cdots & v_{nm}\end{pmatrix} \tag{7}$$

将矩阵的每一列的元素相乘得到一个向量

$$V=(\prod_{i=1}^{n}v_{i1},\prod_{i=1}^{n}v_{i2},\cdots,\prod_{i=1}^{n}v_{im})=(v_1,\cdots,v_m).$$

这个向量从右向左看第一个不等于1的 v_j 对应的 s_j 就是最大值，从左向右看不等于1的 v_i 对应的 s_i 就是最小值.

注1: $x_1,\cdots,x_n$ 可能只有一个值等于最大(小)值，也可能有多个值等于最大(小)值，也可能所有的数据都相等. 有时我们需要知道有多少个数据等于最大(小)值，有时候需要隐藏有多少个数据等于最大(小)值，这些都可以利用这里的原理设计出满足相应安全要求的协议.

例如 假设 $\{s_1,\cdots,s_m\}=\{1,2,\cdots,10\}$, $x_1=5$, $x_2=7$, $x_3=9, x_4=2$. 那么他们可以构造矩阵

$$A=\begin{pmatrix} 1 & 1 & 1 & 1 & 65 & 1 & 1 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1 & 1 & 781 & 1 & 1 & 1\\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 867 & 1\\ 1 & 758 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{pmatrix}.$$

将矩阵的每一列相乘得到

$$V=(1,758,1,1,65,1,781,1,867,1).$$

显然，从左到右第一个不等于1的位置 $i=2$ 对应的 s_2 就是最小值，从右到左第一个不等于1的位置 $j=9$ 对应的 s_9 就是最大值.

协议1 半诚实模型下的最大(小)值计算协议

输入: 参与者 $P_1,\cdots,P_n$ 分别拥有私密数据 $x_1,\cdots,x_n$.

输出: $f(x_1,\cdots,x_n)=(\min\{x_1,\cdots,x_n\},\max\{x_1,\cdots,x_n\})$.

准备: $P_1,\cdots,P_n$ 选择一个大素数 p 和一个 Z_p^* 的生成元 g，每个参与者 P_i 选择一个私钥份额 sk_i，联合生成门限解密的ElGamal密码系统的公钥

$$h=g^{\sum sk_i}\bmod p=\prod_{i=1}^{n}g^{sk_i}\bmod p.$$

1. $P_1,\cdots,P_n$ 分别生成自己数据的编码向量，用公钥加密这个向量并公布. 所有编码向量的密文构成一个 $n\times m$ 的密文矩阵

$$B=\begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1m}\\ c_{21} & c_{22} & \cdots & c_{2m}\\ \cdots & \cdots & \cdots & \cdots\\ c_{n1} & c_{n2} & \cdots & c_{nm}\end{pmatrix}, \tag{8}$$

其中 $c_{ij}=E(v_{ij})$.

2. 参与者计算矩阵每一列的乘积得到一个密文，所有列的乘积构成一个密文向量

$$C=(\prod_{i=1}^{n}c_{i1},\cdots,\prod_{i=1}^{n}c_{im})=(c_1,\cdots,c_m).$$

3. 所有参与者合作从左向右解密 $c_1,\cdots,c_m$，如果 $\mathbf{Dec}(c_i)\neq 1$，则 $\min\{x_1,\cdots,x_n\}=s_i$，停止解密.

4. 所有参与者再次合作从右向左解密 $c_m,\cdots,c_i$，如果 $\mathbf{Dec}(c_j)\neq 1$，则 $\max\{x_1,\cdots,x_n\}=s_j$，停止解密.

注2: 这个解决方案很容易推广到每个参与者有多个数，他们联合确定这些数据中的最大值与最小值的应用场合，只需要他们分别将自己数据的位置设置为随机数即可.

从计算原理可知这个协议的正确性是显然的，无需证明. 下面我们证明协议的安全性.

定理1. 协议1在半诚实模型下安全计算最大(小)值.

证明. 因为在协议中所有参与者的地位是相同的，诚实的参与者受到的最严重的安全威胁是其他所有参与者合谋以获得该参与者的保密数据，这些攻击者构成的集合称为最大合谋攻击者集合. 如果协议对最大合谋攻击者集合是安全的，那么对于最大合谋攻击者集合的任何子集都是安全的. 不失一般性，假设 P_1 是诚实的. 最大攻击者集合为 $I=\{P_2,\cdots,P_n\}$, $\bar{x}=(x_1,\cdots,x_n)$. x_1 有两种情况:

(1) $\max\{\bar{x}\}=x_1$ 或者 $\min\{\bar{x}\}=x_1$ 且 $x_1\notin\{x_2,\cdots,x_n\}$. 此时，如果 $P_2,\cdots,P_n$ 合谋，x_1 将完全泄露. 这和理想协议完全相同，实际协议没有比理想协议泄露更多信息，协议是安全的.

(2) $\min\{\bar{x}\}<x_1<\max\{\bar{x}\}$. 在此情况下，$x_1$ 没有任何泄露，也和理想协议完全相同，协议是安全的，利用模拟范例证明如下.

在执行协议时，I 作为一个整体收到的消息只

有第一步 P_1 发出的 $(c_{11},\cdots,c_{1m})$ 和第三、第四步解密某个密文 $c_k=(c_{k1},c_{k2})$ 时收到的 $u_k=c_{k1}^{sk_1} \bmod p$. 假设 $\min\{\bar{x}\}=s_i,\max\{\bar{x}\}=s_j$，那么在协议执行过程中

$$view_I^{\Pi}(\bar{x})=\{c_{11},\cdots,c_{1m},u_1,\cdots,u_i,u_j,\cdots,u_m\},$$

其中 $u_k=c_{k1}^{sk_1} \bmod p\ (k=1,\cdots,i,j,\cdots,m)$. 现在我们通过构造满足(3)的 S 来证明协议的安全性. S 的工作过程如下:

给定输入 $(I,(x_2,\cdots,x_n),f_I(\bar{x}))$，$S$ 随机选择一个 x_1' 使得 $f(x_1',\cdots,x_n)=f(x_1,\cdots,x_n)$. 然后 S 在同一个ElGamal密码系统下模拟协议中的所有成员(模拟的 P_1 不参与合谋)执行协议得到密文 $c'_{11},\cdots,c'_{1m}$ 和 $u_1,\cdots,u_i,u_j,\cdots,u_m$. 令

$$S(I,(x_2,\cdots,x_n),f_I(\bar{x}))=$$

$$\{c'_{11},\cdots,c'_{1m},u_1,\cdots,u_i,u_j,\cdots,u_m\}.$$

在 P_1 不参与合谋的情况下 $c_{11},\cdots,c_{1m}$ 和 $c'_{11},\cdots,c'_{1m}$ 都是用概率加密算法加密的相同数量的密文，它们是计算不可区分的. u_k 和 $u_k'(k=1,\cdots,i,j,\cdots,m)$ 分别是 $(g^{r_k})^{sk_1} \bmod p$ 和 $(g^{r'_k})^{sk_1} \bmod p$ 形式的数. 因为

$$\{r_1,\cdots,r_i,r_j,\cdots r_m\}\overset{c}{\equiv}\{r'_1,\cdots,r'_i,r'_j,\cdots r'_m\},$$

所以

$$\{u_1,\cdots,u_i,u_j,\cdots,u_m\}\overset{c}{\equiv}\{u_1',\cdots,u_i',u_j',\cdots,u_m'\}.$$

因而得到

$$\{S(I,(x_2,\cdots,x_n),f_I(\bar{x}))\}\overset{c}{\equiv}\{view_I^{\Pi}(\bar{x})\}$$

因此，协议在半诚实模型下是安全的. 证毕

4 恶意模型下的协议

4.1 对协议1的主动攻击

我们已经证明协议1可以抵抗半诚实参与者的被动攻击，但是不能抵抗恶意敌手的主动攻击. 我们的目标是设计抵抗主动攻击的最大(小)值保密计算协议. 设计恶意模型下的安全多方计算协议，通常是先设计一个半诚实模型下的安全多方计算协议，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或者无法实施，或者被发现，从而迫使恶意敌手以半诚实的方式参与协议.

恶意敌手在协议执行过程中能做什么恶意行为呢？前述的三种无法阻止的恶意行为不予考虑，即使在理想模型下也不予考虑[6]. 除此之外，我们来分析针对协议1，参与者分别能做什么恶意行为.

因为在协议中每个参与者只是为其他参与者提供数据，其可能的攻击也只能隐藏在提供的数据中，即提供不符合形式(6)的向量的密文和不符合形式 $c_{k1}^{sk} \bmod p$ 的数据.

1. 提供不符合形式(6)的向量的密文有三种情况:

(1) 恶意参与者正确的编码应该将 v_s 设置为随机数，但他将 v_t 设置为随机数 $(1\le s,t\le m)$. 这种恶意行为等价于提供虚假输入值，因而可以不予考虑.

(2) 恶意参与者可能把自己编码的向量元素全设置为1，这样他可以不用自己的保密数据，知道其他参与者的最大(小)值，而不会被发现.

(3) 恶意参与者可以将多个元素设置成随机数干扰协议，比如某个参与者不希望最值出现在区间 $[s_3,s_6]$，他可以将自己的向量设置为 $(1,r,1,1,1,1,r,1,1,1)$.

2. 恶意参与者可能在第3，4步联合解密密文时提供错误的 u_k. 这样就使得诚实的参与者得到一个错误的结果，而他们则可以利用诚实参与者提供的正确的 u_k 计算出正确的结果.

上述恶意行为，在协议1中我们无法阻止也无法发现. 现在考虑阻止或发现这些恶意行为的措施.

对于1中第(2)种恶意行为，只需验证参与者提供的向量各分量的乘积是否为1即可. 如果各分量的乘积为1，则说明参与者在进行欺骗.

对于1中第(3)种恶意行为，从协议可以看出如果 r 出现的第1个位置大于真实的最小值，最后出现的位置小于真实的最大值，因为这些位置的密文不会被解密，所以不会对协议产生任何影响，可以不予考虑. 因此对于这种恶意行为只需考虑 r 出现的第1个(最后一个)位置小于(大于)真实的最小(大)值的情况，因为这两种情况下将导致协议产生错误输出.

如果在协议的最后让持有最大(小)值的参与者公开自己加密向量的每个分量时所选择的随机数，就能够发现第(2)，(3)两种恶意行为. 这样做是合理的，因为在任何招标、拍卖、电子商务活动、方案

优化以及选拔最佳人选等需要保密计算最大(小)值的场合，最后都要公布结果，而且公布的结果的合理性也应该经受检验(这样也可以防止多个参与者声称自己是最大值的拥有者).

对于2种恶意行为，可以采用零知识证明的方法阻止. 即要求参与者证明他提供的数据确实是 $c_{k1}^{sk} \bmod p$. 具体协议如下:

4.2 具体协议

协议2. 恶意模型下的最大(小)值保密计算

输入: $P_1,\cdots,P_n$ 分别输入 $x_1,\cdots,x_n \in \{s_1,\cdots,s_m\}$ 其中 $(s_1 <\cdots< s_n)$.

输出: $f(x_1,\cdots,x_n)=(\min\{x_1,\cdots,x_n\},\ \max\{x_1,\cdots,x_n\})$.

准备阶段: $P_1,\cdots,P_n$ 商定一个大素数 p， Z_p^* 的一个生成元 g. P_i 选择随机数 $sk_i \in Z_P^*$ 作为自己的私钥份额，并公布 $h_i = g^{sk_i} \bmod p$. $P_1,\cdots,P_n$ 计算

$$h=\prod_{i=1}^{n} h_i \bmod p$$

作为公钥.

1. 参与者 $P_i(i=1,\cdots,n)$ 分别构造符合形式(6)的向量 $E(V_i)=(E(v_{i1}),\cdots,E(v_{im}))$: 向量中有只有一个分量是随机数，其余分量均为1. 用公钥 h 加密自己的向量，并公布向量的密文.

2. 参与者合作验证向量 $E(V_i)=(E(v_{i1}),\cdots,E(v_{im}))$ 各分量的乘积 $\prod_{i=1}^{m} E(v_i)$，如果解密结果为随机数，则接受向量 $E(V_i)$，继续执行协议；否则说明 P_i 在进行欺骗，中止协议.

3. 所有参与者生成的编码向量的密文构成一个形如(7)的密文矩阵. 将密文矩阵的第 i 列元素相乘作为向量 C 的第 i 个分量，这样形成一个新的密文向量

$$C=(c_1,\cdots,c_m)=((c_{11},c_{12}),\cdots,(c_{m1},c_{m2})).$$

4. 所有参与者联合从左到右解密向量 C 的每一个分量 $c_1,\cdots,c_m$，解密时每个参与者要用零知识证明的方式向所有参与者证明自己提供的部分解密数据是正确的，即假设解密 $c_k=(c_{k1},c_{k2})$ 时参与者 P_i 提供的数据为 u_k，他必须证明 $\log_g h_i = \log_{c_{k1}} u_k$. 如果某个参与者不能证明这一点，则说明他在欺骗，中止协议. 如果 $\mathbf{Dec}(c_i)\neq 1$，则 $\min\{x_1,\cdots,x_n\}=s_i$. 停止从左到右解密向量.

5. 所有参与者联合从右向左解密向量 C 的每一个分量 $c_m,\cdots,c_i$，解密时每个参与者要用零知识证明的方式向所有参与者证明自己提供的解密数据是正确的. 如果发现欺骗就中止协议. 如果 $\mathbf{Dec}(c_j)\neq 1$，则 $\max\{x_1,\cdots,x_n\}=s_j$.

6. 持有最小值和最大值的参与者公布自己向量的每个分量和加密每个分量时所选择的随机数，其他参与者验证该参与者没有欺骗.

注3: 我们强调协议并不能阻止参与者进行欺骗，但如果参与者进行欺骗，就能够被发现. 任何恶意模型下安全的协议都只能做到这一点.

注4: 在很多应用中最后都需要公开最大(小)值持有者的身份并经受检验，因此协议中要求持有最大(小)值得持有者公布自己的向量并经受检验是合理的. 如果某些应用中要求不能泄露最大(小)值的参与者，可以采用附录中的协议.

4.3 协议的正确性

如果协议在执行过程中没有中止，说明所有参与者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(也可能有第(3)种的恶意行为，但其设置的随机数位置值介于最大值和最小值位置之间，这些恶意行为既不会导致协议产生错误输出，也不会泄露更多的信息)，协议的正确性如同协议1,计算原理保证了协议的正确性.

4.4 协议的安全性

前面已经分析了协议执行过程中可能受到的恶意攻击以及所采取的预防措施，下面证明协议的安全性定理:

定理2. 协议2在恶意模型下是安全的.

证明概要. 与证明协议1的安全性相同，因为各个参与者地位平等，我们也仅考虑最大合谋攻击者集合. 如果协议对最大合谋攻击者集合是安全的，那么对于最大合谋攻击者集合的任何子集的合谋攻击者集合都是安全的. 不失一般性，假设 P_n 是诚实的，最大合谋攻击者集合为 $I=\{P_1,\cdots,P_{n-1}\}$. 我们将证明对于在实际协议中控制 I 中所有参与者的攻击者所采取的任何任意概率多项式时间算法策略 A，在理想协议中都存在一个概率多项式时间算法策略 B 使得(5)成立.

如果协议在第2步中止，因为没有输出任何信息，所有参与者的密文都没有进行任何操作，不会泄露任何信息，所以协议是安全的.

我们需要将真实协议中任何 A 转化为理想协议中相应的 B. 首先考虑在执行实际协议时攻击者所获得的信息. 在执行协议2时，I 作为一个整体收到的消息如下:

(1)第一步 P_n 发出的由一个分量为随机数，$m-1$ 个分量为1的向量的密文，因为这些密文是用

门限解密的概率密码系统加密的，没有 P_n 的合作 I 的成员无法解密，不能从这个密文向量得到 x_n 的任何信息.

(2) 设 $\min\{x_1,\cdots,x_n\}=s_i,\max\{x_1,\cdots,x_n\}=s_j$，在解密 $c_k=(c_{k1},c_{k2})(k=1,\cdots,i,j\cdots,m)$ 时 P_n 发送的 $u_k=c_{k1}^{sk_n} \bmod p$ 和零知识证明 u_k 正确的信息 t_k.

执行实际协议时，协议的输入为 $\bar{x}=(A(x_1,\cdots x_{n-1}),x_n)$，即合谋者的输入取决于合谋攻击者的策略 A. 如果在协议第4,5步解密过程中任何一个 $P_i\in I$ 不能零知识证明自己提供的解密数据是正确的，协议中止. 此时 P_n 将无法得到 $f(\bar{x})$，只能输出 $\perp$，但 I 则可能得到正确的 $f(\bar{x})$. 攻击者输出什么取决于攻击者的策略，即攻击者将输出 $A(\bar{x}_I,I,r,z,C,u_k,t_k,f(\bar{x}))$，因此，

$$\{\mathbf{REAL}_{\Pi,I,A(z)}\}_{\bar{x},z}=\{A(\bar{x}_I,I,r,z,C,u_k,t_k,f(\bar{x})),\perp)\}.$$

如果协议没有中止，P_n 将收到并输出 $f(\bar{x})$. I 也收到 $f(\bar{x})$，但 I 输出什么取决于 I 的策略 A 和协议执行过程收到的消息，即 $A(\bar{x}_I,I,r,z,C,u_k,t_k,f(\bar{x}))$，因此

$$\{\mathbf{REAL}_{\Pi,I,A(z)}\}_{\bar{x},z}=\{A(\bar{x}_I,I,r,z,C,u_k,t_k,f(\bar{x})),f(\bar{x})\}.$$

现在转向理想协议，在理想协议中 P_n 给TTP提供自己的保密数据 x_n. B 把 $\bar{x}_I$ 提供给 A，从 A 得到 $A(\bar{x}_I)$ 并发送给TTP. TTP得到 $\bar{x}=(A(\bar{x}_I),x_n)=(A(x_1,\cdots,x_{n-1}),x_n)$，自己计算 $f(\bar{x})$，并把 $f(\bar{x})$ 发送给 B.

B 随机选择一个 x_n' 使得 $f(A(x_1,\cdots,x_{n-1}),x_n')=f(A(x_1,\cdots,x_{n-1}),x_n)$. B 利用 x_n 与 I 执行协议，并给 I 提供 x_n 的编码的密文向量 C_n' 和零知识证明所需要的 u_k', t_k'.

如果实际协议执行中因为任何一个合谋者没有通过零知识证明而中止，B 就不让TTP给 P_n 发送计算结果，P_n 只能输出 $\perp$. 如果实际协议没有中止而是执行完所有步骤结束，B 就让TTP给 P_n 发送计算结果.

两种情况下，B 都用 $(\bar{x}_I,I,r,z,C_n',u_k',t_k',f(\bar{x}))$ 调用 A. A 输出什么，B 就输出什么，即输出 $A(\bar{x}_I,I,r,z,C_n',u_k',t_k',f(\bar{x}))$. 因此，如果实际协议中止时

$$\{\mathbf{IDEAL}_{f,I,B(z)}(\bar{x})\}_{\bar{x},z}=\{A(\bar{x}_I,I,r,z,C'_n,u'_k,t'_k,f(\bar{x})),\perp\}.$$

如果实际协议执行完毕而结束，

$$\{\mathbf{IDEAL}_{f,I,B(z)}(\bar{x})\}_{\bar{x},z}=\{A(\bar{x}_I,I,r,z,C'_n,u'_k,t'_k,f(\bar{x})),f(\bar{x})\}.$$

比较两种情况下的 $\{\mathbf{IDEAL}_{f,I,B(z)}(\bar{x})\}_{\bar{x},z}$ 和 $\{\mathbf{REAL}_{\Pi,I,A(z)}\}_{\bar{x},z}$ 可以发现 P_n 的输出是相同的. 因此只要证明 $A(\bar{x}_I,I,r,z,C_n',u_k',t_k',f(\bar{x}))$ 与 $A(\bar{x}_I,I,r,z,C_n,u_k,t_k,f(\bar{x}))$ 计算不可区分即可.

C_n',C_n 是用概率加密算法加密的向量的密文，因此，$C_n\equiv_c C'_n$. u_k 和 u_k' 分别是 $(g^{r_k})^{sk_n} \bmod p$ 和 $(g^{r'_k})^{sk_n} \bmod p$ 形式的数 ($k=1,\cdots,i,j\cdots,m$)，所以 $u_k\equiv_c u_k'$. 零知识证明理论保证零知识证明的 t_k,t_k' 计算不可区分. 所以

$$A(\bar{x}_I,I,r,z,C_n,u'_k,t'_k,f(\bar{x}))\equiv_c A(\bar{x}_I,I,r,z,C_n,u_k,t_k,f(\bar{x})).$$

进而得到

$$\{\mathbf{IDEAL}_{f,I,B(z)}(\bar{x})\}_{\bar{x},z}\equiv_c\{\mathbf{REAL}_{\Pi,I,A(z)}\}_{\bar{x},z}.$$

因此，协议在恶意模型下是安全的. 证毕

5 协议效率分析与测试

因为尚没有见到恶意模型下的最大值保密计算协议，所以无法与现有协议的效率进行比较，我们只分析测试本文协议的效率. 半诚实模型下的协议1主要是作为设计恶意模型下安全多方计算协议2的基石为协议2服务的，这里我们不分析协议1的计算复杂性(效率)，只分析协议2的计算复杂性. 我们用最费时的模指数运算的次数来度量计算复杂性，忽略其他费时很少的运算.

(1) 生成公钥需要 n 次模指数运算; (2) 加密一个明文需要2次模指数运算，n 个参与者每个人加密1个 m 维向量需要 $2mn$ 次模指数运算; (3) 解密一个密文需要 n 次模指数运算，如果 $\min\{\bar{x}\}=s_i$ 和 $\max\{\bar{x}\}=s_j$，得到 $\min\{\bar{x}\}$ 需要 ni 次模指数运算，得到 $\max\{\bar{x}\}$ 需要 $n(m-j)$ 次模指数运算. (4) 证明提供了正确的解密数据需要4次模指数运算; 所有参与者的证明需要 $4ni+4n(m-j)$ 次模指数运算. 验证最大值与最小值持有者没有欺骗需要 $2m$ 次模指数运算，协议共需要 $2(m+1)n+5n(i+m-j)+4m$ 次模指数运算.

协议效率测试. 一般的恶意模型下的安全多方计算协议基本上都只进行了计算复杂性分析，没有给出实际的测试数据[9,33-35]，实际的效率是比较低的. 因为没有恶意模型下的最大值保密计算协议可以比较，我们只验证协议的实际效率. 与半诚实模

型下的协议相比，恶意模型下的协议只增加了 $4ni+4(m-j)+4m$ 次模指数运算，效率是非常高的．我们在下面的实验环境下测试了协议的效率：Intel(R) Core(TM)i5-9400 CPU 2.90GHz, 2.90GHz, 内存8GB 64位Windows10操作系统，编程语言为Pycharm+Python3.9．ElGamal密码系统的参数 p 为512, 1024, 1536, 2048, 2560比特素数，加密的随机数为64-80比特的随机数时，协议的运行时间如图1所示．

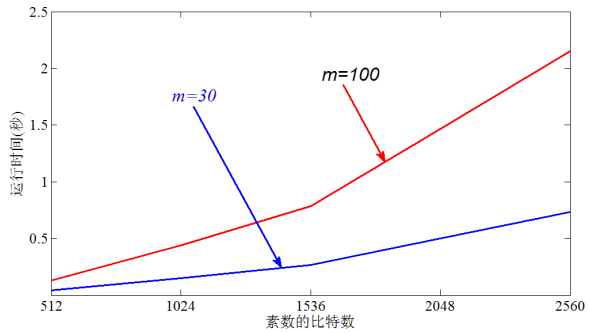

图1. 协议的运行时间随素数大小的变化趋势

图中的红色曲线是 $n=10, m=30, i=5, j=27$ 时，协议运行时间随素数大小变化的趋势；蓝色曲线是 $n=10, m=100, i=10, j=92$ 时，协议运行时间随素数大小变化的趋势．

注5． 测试数据忽略了置换的零知识证明．根据文献[36]的分析，1个参与者生成自己的置换并进行零知识证明需要16m次模指数运算，因此置换的零知识证明需要增加16nm次模指数运算．考虑到置换的零知识证明，协议的执行时间将增加到目前的4倍左右．

从实验结果可以看出恶意模型下保密计算最大(小)值的协议2在 $n=10, m=100, i=10, j=92, |p|=2560$ 时，运行时间只有2.3秒，协议的效率是比较高的，也是完全可以接受的．

6 结论

保密计算一组数据的最大(小)值问题是安全多方计算领域一个最重要、最基础的问题，可以作为基本模块用来构造其他安全多方计算协议，在实践中有广泛的应用前景．现有的解决方案有限而且只是半诚实模型下安全的协议．本文构造了一个恶意模型下安全的计算协议，这是第一个恶意模型下安全的最大(小)值计算协议，理论分析与试验数据都表明协议的效率是比较高的，但协议适用范围有一定的限制．该协议也需要最大(小)值的持有者公布自己加密数据时所选择的随机数进行验证．如果最大(小)值的持有者也需要保密，可以用附录A中的协议．今后我们将研究保密数据范围未知和数据范围没有限制条件下的最大(小)值计算协议以及其他问题的解决方案．

参 考 文 献

[1] Yao A C. Protocols for secure computations//Proceedings of 23rd Annual Symposium on Foundations of Computer Science. Chicago, USA, 1982: 160-164

[2] Goldwasser S. Multi party computations: past and present// Proceedings of the sixteenth annual ACM symposium on principles of distributed computing. Santa Barbara, USA, 1997: 1-6

[3] Cramer R, Damgard I B. Secure multiparty computation. London: Cambridge University Press, 2015

[4] Yao A C. How to generate and exchange secrets (extended abstract)// Proceedings of the Twenty-Seventh Annual Symposium on Foundations of Computer Science. Toronto, Canada, 1986: 162-167

[5] Goldreich O, Micali S, Wigderson A. How to play any mental game//Proceedings of the nineteenth annual ACM symposium on Theory of computing. New York, USA, 1987: 218-229

[6] Goldreich O. Foundations of cryptography: volume 2, basic applications. London,UK: Cambridge university press, 2004

[7] Tang C M, Shi G H, Yao Z A. Secure multi-party computation protocol for sequencing problem. Scientia Sinica Informationis, 2011, 41(07): 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(07): 789-797)

[8] Fagin R, Naor M, Winkler P. Comparing information without leaking it. Communications of the ACM, 1996, 39(5): 77-85

[9] Freedman M J, Hazay C, Nissim K, et al. Efficient set intersection with simulation-based security. Journal of Cryptology, 2016, 29(1): 115-155

[10] Kim E Y, Lee H S, Park J. Towards round-optimal secure multiparty computations: multikey FHE without a CRS. International Journal of Foundations of Computer Science, 2020, 31(2): 157-174

[11] Sin G T, Cao J N, Lee C S. DAG: A General Model for Privacy-Preserving Data Mining. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(1): 40-53

[12] Liu J, Tian Y, Zhou Y, et al. Privacy preserving distributed data mining based on secure multi-party computation. Computer Communications, 2020, 153: 208-216

[13] Du W L, Atallah M J. Privacy-preserving cooperative statistical

analysis//Proceedings of the Annual Computer Security Applications Conference, New Orleans, USA, 2001. 102-110

[14] Vaidya J. Privacy-preserving statistics. IEEE Computer, 2018, 51(9): 8-9

[15] Atallah M J, Du W L. Secure multi-party computational geometry// Proceedings of the Workshop on Algorithms and Data Structures, Providence, USA, 2001: 165-179

[16] Chen Z H, Li S D, Chen L C, et al. Fully privacy-preserving determination of point-range relationship. Scientia Sinica Informationis, 2018, 48(02): 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(02): 187-204)

[17] Xu C, Xie X, Zhu L H, et al. PPLS: a privacy-preserving location-sharing scheme in mobile online social networks. Science China: Information Sciences. 2020, 63(3): 132105:1–132105:11

[18] Zhao C, Zhao S N, Zhao M H, et al. Secure multi-party computation: theory, practice and applications. Information Sciences, 2019, 476(5): 357-372

[19]Xu J, Wang A D, Wu J, et al. SPCSS: social network based privacy-preserving criminal suspects sensing. IEEE Transactions on Computational Social Systems, 2020, 7(1): 261-274

[20] Gupta N, Gade S, Chopra N, et al. Preserving Statistical Privacy in Distributed Optimization. IEEE Control System Letter, 2021, 5(3): 779-784

[21] Lindell Y. Secure Multiparty Computation (MPC). IACR Cryptology ePrint Arch, 2020: 300

[22] Shi J, Zhang R, Liu Y, et al. PriSense: privacy-preserving data aggregation in people-centric urban sensing system//Proceedings of the IEEE INFOCOM, San Diego, USA, 2010. 758-766

[23] Li Q H, Cao G H, Porta T F. Efficient and privacy-preserving data aggregation in mobile sensing. IEEE Transactions on Dependable Secure Computing, 2014, 11(2): 115-129

[24] Zhang Y, Chen Q J, Zhong S. Efficient and privacy-preserving min and *k*th min computations in mobile sensing systems. IEEE Transactions on Dependable and Secure Computing, 2017, 14(1): 9-21

[25] Dai H, Ji Y, Xiao F, et al. Privacy-Preserving MAX/MIN Query Processing for WSN-as-a −Service//Proceedings of International Federation of Information Processing Networking conference, Warsaw, Poland, 2019: 1-9

[26]Guan Y G, Lu R X, Zheng Y D, et al. Achieving Efficient and Privacy-Preserving Max Aggregation Query for Time-Series Data//Proceedings of IEEE International Conference on Communications, Dublin, Ireland, 2020: 1-6

[27]Huang Y, Zeng P, Choo K K. An Efficient Privacy-Preserving Protocol for Computing *k*th Minimum Value in P2P Networks. Journal of Circuits System Computation, 2020, 29(9): 2050138:1-2050138:20

[28]Dou J W, Ma L, Li S D. Secure Multi-Party Computation for Minimum and Its Applications. ACTA ELECTRONICA SINICA, 2017, 45(07): 1715-1721(窦家维, 马丽, 李顺东. 最小值问题的安全多方计算及其应用. 电子学报, 2017, 45(07): 1715-1721)

[29] Yang X Y, Li S D, Kang J. Private substitution and Its Applications in Private Scientific computation. Chinese Journal of Computers, 2018, 41(5): 1132-1142(杨晓艺, 李顺东, 亢佳. 保密替换及其在保密科学计算中的应用. 计算机学报, 2018, 41(5): 1132-1142)

[30] Yang Y, Liu X M, Deng R H. Multi-user multi-keyword rank search over encrypted data in arbitrary languages. IEEE transactions on dependable and secure computing, 2020, 17(2): 320-334

[31] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE transactions on information theory, 1985, 31(4): 469-472

[32] Fouque P A, Poupard G, Stern J. Sharing decryption in the context of voting or lotteries//Proceedings of International Conference on Financial Cryptography, Anguilla, UK, 2000: 90-104

[33] Lindell Y. Fast cut-and-choose-based protocols for malicious and covert adversaries. Journal of Cryptology, 2016, 29(2): 456-490

[34] Frederiksen T K, Pinkas B, Yanai A. Committed MPC-Maliciously Secure Multiparty Computation from Homomorphic Commitments. IACR Cryptology. ePrint Arch. 2017: 550

[35] Hazay C, Yanai A. Constant-Round Maliciously Secure Two-Party Computation in the RAM Model. Journal of Cryptology, 2019, 32(4): 1144-1199

[36] Bayer S, Groth J. Efficient Zero-Knowledge Argument for Correctness of a Shuffle//Proceedings of 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, 2012: 263-280

附录A 不泄露最大值拥有者的协议

输入: $P_1,\cdots,P_n$ 分别输入 $x_1,\cdots,x_n\in\{s_1,\cdots,s_m\}$，其中 $(s_1<\cdots<s_n)$.

输出: $f(x_1,\cdots,x_n)=(\min\{x_1,\cdots,x_n\},\ \max\{x_1,\cdots,x_n\})$.

准备阶段: $P_1,\cdots,P_n$ 商定一个大素数 p，Z_p^* 的一个生成元 g. P_i 选择随机数 $sk_i\in Z_P^*$ 作为自己的私钥份额, 并公布 $h_i=g^{sk_i}\bmod p$. $P_1,\cdots,P_n$ 计算

$$h=\prod_{i=1}^{n}h_i \bmod p$$

作为公钥.

1. 参与者 P_1 构造一个符合形式(6)的向量 $V=(v_1,\cdots,v_m)$:

向量中有只有一个分量是随机数, 其余分量均为1. P_1 用公钥 h 加密向量 V 得到 $E(V)=(E(v_1),\cdots,E(v_m))$, 并公布向量的密文和加密时选择的随机数, 其他参与者验证确实如此. 如果验证通过则接受 $E(V)$, 继续执行协议; 否则说明 P_1 在欺骗, 中止协议.

2. 每个参与者 P_i 通过对 $E(V)$ 进行置换和重随机化操作生成符合自己的保密数据编码的向量的密文 $E(V_i)$, 并用文献[36]的方法证明自己的 $E(V_i)$ 确实是 $E(V)$ 的置换. 如果证明通过, 则接受 $E(V_i)$ 并继续执行协议; 否则说明 P_i 在欺骗, 中止协议.
3. 所有参与者的 $E(V_i)$ 合成一个形如(7)的密文矩阵. 将密文矩阵的第 i 列元素相乘作为向量 C 的第 i 个分量, 这样形成一个新的密文向量

$$C=(c_1,\cdots,c_m)=((c_{11},c_{12}),\cdots,(c_{m1},c_{m2})).$$

4. 所有参与者联合从左到右解密向量 C 的每一个分量 $c_1,\cdots,c_m$, 解密时每个参与者要用零知识证明的方式向所有参与者证明自己提供的解密数据是正确的, 即假设解密 $c_k=(c_{k1},c_{k2})$ 时参与者 P_i 提供的数据为 u_k, 他必须证明 $\log_g h_i=\log_{c_{k1}} u_k$. 如果某个参与者不能证明这一点, 则说明他在欺骗, 中止协议. 如果 $\mathbf{Dec}(c_i)\neq 1$, 则 $\min\{x_1,\cdots,x_n\}=s_i$. 停止从左到右解密向量.
5. 所有参与者联合从右向左解密向量 C 的每一个分量 $c_m,\cdots,c_i$, 解密时每个参与者要用零知识证明的方式向所有参与者证明自己提供的解密数据是正确的. 如果发现欺骗就中止协议. 如果 $\mathbf{De}(\mathbf{c}_j)\neq$, 则 $\max\{x_1,\cdots,x_n\}=s_j$.

LI Shundong, Ph. D., professor. His main research interests include public key cryptography, cryptographic protocol design and information security.

XU Wenting, M. S. candidate. Her research interest includes information secure protocol design.

WANG Wenli, Ph. D candidate. Her main research interests include modern cryptography and information security.

ZHANG Mengyu, M. S. candidate. Her main research interests include modern cryptography and information security.

Background

In the information age, data has become the most important strategic resource of a country, of an enterprise and even of a person. Due to various constrictions, no entity (an organization, an enterprise or a person) can obtain all data it needs and every entity usually needs the data owned by other entities to help their decision-making, that is, different entities perform cooperative computation on their private data to share the data, to benefit each other to finally achieve win-win. Data sharing can create value. The more data are shared, the more value will be created. But data owned by different entities often contain much private information. If the rntities shre their data and perform computation on the data without proper protection, the privacy will be easily disclosed carelessly, and it will result in some serious consequence or economic or fame loss. The risk of privacy disclosing seriously hinder such cooperative computation.

Secure multiparty computation was first introduced by Yao as millionaires' problem. Goldreich et al thoroughly studied secure multiparty computation and established the theoretical basis of it. It is now a key privacy-preserving technology for cooperative computation in information era. Using secure multiparty computation, distrusted parties can cooperatively perform computation on their private data to explore the relationship between data, to mine data value while preserving the data privacy to make full use of private data to improve economic and social management and to benefit the human society. The cooperative computation will benefit the distrusted parties to securely share their private data without worrying privacy disclosing. Secure multiparty computation is general cryptographic computation primitive which needs homomorphic cryptosystems, secret sharing, bit commitment, zero-knowledge proof, oblivious transfer, one-way hash function, signature and so on as building blocks. Secure multiparty computation research can promote the development of such branches of the cryptography. Therefore, secure multiparty computation becomes a focus of the international cryptographic community in recent years.

Computing the maximum of a data set is a basic data operation. Secure computing the maximum of a data set, is one of the most important problem of secure multiparty computation. The protocols for this problem are building blocks of many other secure multiparty computation protocols in many secure applications such as secure voting, secure

suction and bidding, secure optimization, secure recommendation. This problem has not been thoroughly studied. Even the protocols for this problem that are secure in the semi-honest model are scarcely. The existing protocols need a trusted authority to aggregate data and only work in some smart phone application scenarios. They cannot resist any collusion attacks and are even not secure in the semi-honest model, to say nothing of any active attacks in the malicious model. Such protocols cannot even guarantee the weakest security. Therefore, they cannot be used in many practical scenarios. Protocols that are secure in the malicious model can guarantee enough security in practical scenarios and of more important theoretical significance. To the best of our knowledge, there is no protocol for computing maximum that are secure in the malicious model. Privately compute the maximum of private data in the malicious model is still an open problem. It is urgent and of theoretical importance to study protocols that are secure in the malicious model. This is the reason why we study this problem.

In this paper, we first introduce a protocol to securely compute the maximum (minimum) in the semi-honest model. Then we use zero-knowledge proof of discrete logarithm, of secure shuffling and result verifiability to modify it such that it can resist active attack or can find atcive attack, and finally convert the protocol to one that is secure in the malicious model. The protocol is the first which is secure in the malicious model. We have studied secure multiparty computation for nearly 20 years. We established a secure multiparty computation laboratory equipped with enough computers and a vigor research team. Our study is fruitful. We have published more than 100 papers to address various secure multiparty computation problems. We have innovated some new techniques and some new methods such as secure substitution, encryption-and-choose, encoding, secure permutation and compute-encrypt-choose-compute to solve many secure multiparty computation problems such as secure comparing, secure maximum computation, secure sorting, secure Boolean computation, secure vector computation, secure set computation, secure graph computation, secure computational geometry.

Our study has been supported by three projects of the national Natural Science Foundation of China.