计算机研究与发展
Journal of Computer Research and Development

DOI:10.7544/issn1000-1239.2015.20150763
52(10):2247-2257,2015

实用安全多方计算协议关键技术研究进展

蒋 瀚　　徐秋亮
(山东大学计算机科学与技术学院　济南　250101)
(jianghan@sdu.edu.cn)

Advances in Key Techniques of Practical Secure Multi-Party Computation

Jiang Han and Xu Qiuliang
(*School of Computer Science and Technology, Shandong University, Jinan* 250101)

Abstract In the setting of secure multi-party computation, two or more parties with private inputs wish to compute some joint function of their inputs and achieve the security requirements of privacy, correctness, independence of inputs and more. Secure multi-party computation is not only the general basic research of secure protocol, but also applied in many applications such as coin-tossing, electronic voting and private information retrieval schemes. The research of secure multi-party computation provides a central tool in many area of cryptography. In recent years, secure multi-party computation has been advancing in leaps and bounds, especially in the practical techniques; the practical technology becomes a new attractive field in secure multi-party computation. In this paper, we introduce the main advances and results of practical secure multi-party computation, and focus on three major supporting techniques, which include garbled circuits optimization, cut-and-choose technique and oblivious transfer extension. These techniques significantly improve the efficiency of secure multi-party computation in different aspects.

Key words secure multi-party computation; garbled circuits optimization; cut-and-choose; oblivious transfer extensions; semi-honest adversaries; malicious adversaries

摘　要 在安全多方计算协议中,2个或多个持有秘密输入的参与方想要利用他们的输入来计算某个联合函数,并达到隐私性、正确性及输入无关性等安全要求.安全多方计算既是安全协议的一般性基础研究,也在许多应用领域(比如电子投票、网上合同签署、隐私信息检索等)有明确应用背景,其研究为密码学多个领域提供了核心工具.近年来,安全多方计算协议的研究,特别是在实用化技术方面取得了快速发展,协议实用化成为安全多方计算一个新的关注点.介绍了实用化安全多方计算协议研究的主要进展和成果,并重点介绍安全多方计算实用化的3个支撑性重要技术,包括混乱电路优化、剪切-选择技术及不经意传输扩展技术,这些技术在不同的方面显著提高了安全多方计算协议的效率.

关键词 安全多方计算;混乱电路优化;剪切-选择;不经意传输扩展;半诚实敌手;恶意敌手

中图法分类号 TP309

安全多方计算使得多个参与方能够以一种安全的方式正确执行分布式计算任务.具体来说,n个参与方$P_1,P_2,\cdots,P_n$希望利用各自的秘密输入共同计算n元功能函数(functionality),n元功能函数一般

收稿日期:2015-06-15;**修回日期**:2015-08-24
基金项目:国家自然科学基金项目(61173139,61572294);信息保障技术重点实验室开放基金项目(KJ-14-003);教育部高等学校博士学科点专项科研基金项目(20110131110027)

而言是一个随机函数，$f(x_1, x_2, \cdots, x_n) = (y_1, y_2, \cdots, y_n)$，每个参与方 P_i 持有秘密输入 x_i，计算完成后得到输出 y_i，且每个参与方除了自己的输入和输出以及由其可以推出的信息外得不到任何额外信息.

安全多方计算的研究从 20 世纪 80 年代 Yao 提出的百万富翁问题开始[1]，Yao 混乱电路[1]与 GMW 编译器[2]奠定了其初步理论基础，二、三十年来的研究积累了丰富理论成果. 特别地，对恶意敌手攻击下一般功能函数的安全多方计算协议的研究，大大促进了零知识证明、不经意传输、秘密共享等密码学基础原语的发展，对奠定安全协议可证明安全理论框架的基础起到重要作用，极大地推动了现代密码学基础理论的进展.

随着分布式计算、云计算等计算模式的发展，计算任务的参与方越来越多，计算任务执行的外部环境越来越复杂，为了保证某项计算任务的安全性，依靠传统的方法，不得不利用基础的密码算法部件构造越来越复杂的安全协议，而为了证明协议的安全性，又需要为协议定义多种安全特性，并一一去规约证明. 这种处理方法对保障复杂计算任务的安全性越来越力不从心.

一般化的安全多方计算协议，由于其计算任务无关性(可以计算任意的功能函数)，特别是其理想现实模拟的安全证明方法，不需要再考虑特定的安全属性及外部运行环境，所以对现阶段复杂应用的安全保障具有得天独厚的优势.

早期的安全多方计算成果仅仅停留在理论意义上，并不能真正的实用. 但是随着新形势下对于安全多方计算实用化的要求越来越迫切，近年来，安全多方计算实用化研究成为一个新的热点领域，并出现了加速发展的特征. 近年来，安全多方计算的论文成为顶级密码会的常备专题，甚至在 2014 年美密会上出现了 2 个安全多方计算专题：安全多方计算的基础和安全多方计算的实施. 在 2015 年国际密码协会与以色列巴伊兰大学联合举办的密码学第 5 届冬季学校中，聚集了安全多方计算领域内顶尖的研究者，而这届冬季学校的主题就是“实用安全多方计算进展”. 所有这些都体现出研究者们越来越关注安全多方计算协议实用化的研究.

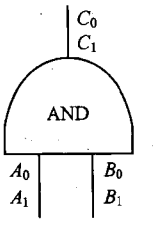
通过几年来的快速发展，安全多方计算已经逐步进入可以实用的阶段，研究者在实验室中已经完成了对 AES 等功能函数的安全多方计算实验，对于半诚实敌手下的协议，通信及计算负载已经不成为障碍，而对恶意敌手模型下的协议，虽然协议运行的时间仍然较长，但也不是完全不可以忍受的程度.

硬件计算速度的提高是协议效率提高的原因之一，包括一些研究者研究利用 GPU 来实现安全多方计算，但是，提高安全多方计算协议效率的更重要的因素在于一些新的设计思想及设计方法的发现，其中对效率提高影响最为显著的包括混乱电路的优化、cut-and-choose 技术的引入以及不经意传输扩展协议的出现. 我们将就这 3 种技术介绍目前最新的研究成果.

1 混乱电路优化研究进展

由于对每个功能函数 f，都存在一个与其等价的电路 C，因此最早的 Yao 协议[1]就是首先将功能函数转化为一个电路，然后针对电路的每个电路门进行混乱，最后逐次计算每个混乱门电路来实现对任意功能函数的安全多方计算.

图 1 以“与门”为例说明原始 Yao 混乱电路的构造，首先为电路中每一根线上的 2 位分别选择一个随机数与之对应，然后根据与门的真值表，利用输入线上的随机数完成对输出线上随机数的加密. 比如若第 1 根输入线的输入为 0，其对应的随机数为 A_0，第 2 根输入线的输入为 0，其对应的随机数为 B_0，根据与门的计算法则，输出线的输出应为 0，对应的随机值为 C_0，则计算密文 $E_{A_0}(E_{B_0}(C_0))$，其中 E 为一个对称加密算法. 按真值表计算完所有密文后，随机置换 4 个密文的顺序，形成混乱值表. 这样，Yao 混乱电路每个电路门需要 6 个随机数，其混乱值表共 4 行.



(a) Garbled AND Gate

τ	σ	τ^σ	Garbled Value
0 A_0	0 B_0	0 C_0	$E_{A_0}(E_{B_0}(C_0))$
0 A_0	1 B_1	0 C_0	$E_{A_0}(E_{B_1}(C_0))$
1 A_1	0 B_0	0 C_0	$E_{A_1}(E_{B_0}(C_0))$
1 A_1	1 B_1	1 C_1	$E_{A_1}(E_{B_1}(C_1))$

(b) Garbled computation table

Fig. 1 The garbled AND Gate and garbled computation table in Yaos' scheme.

图 1 Yao 方案与门混乱电路及混乱表

对于任意一个给定的计算任务，若将其转化为电路，其电路门的数量将是一个巨大的数字，而混乱电路将每一位通过一个随机数(属于某对称密码算法的密钥空间)及两重对称加解密运算进行混乱，计算负荷巨大，因此简化混乱电路的规模是提高安全多方计算协议效率直接有效的方法.

1.1　对混乱 XOR 门的优化

XOR 门是基本电路门之一,具有 $A\oplus R\oplus R=A$ 的计算特性.针对这一特殊性质,Kolesnikov 等人在 2008 年提出了一种称为 Free-XOR 的技术[3],对混乱 XOR 门进行了优化.Free-XOR 技术如图 2 所示,针对电路的 XOR 门,只对电路 2 根输入线的 0 比特选取 2 个随机值 A_0 和 B_0,然后再选择一个随机数 R,而混乱电路的其他的 4 个随机值可以通过如图 2 所示的计算获得.该技术被称为"Free",是因为对于 XOR 门,不再需要两重加解密来计算输出线上的混乱值.

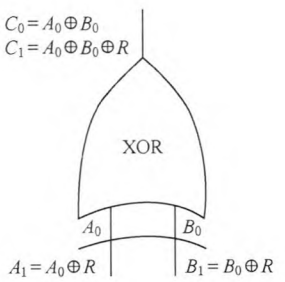


Fig. 2　The technic of Free-XOR.

图 2　Free-XOR 技术

Free-XOR 技术的效率很高,只需要 3 个随机值及简单的异或运算就可以构造混乱表.但遗憾的是,它只是在随机预言模型(random oracle, RO)下,对半诚实敌手是安全的.此外,由于电路一般是由 XOR 门与其他的门电路组合而成,而 Free-XOR 技术中每根输入输出线上 2 个输入对应随机数的"偏移"(即它们的异或)等于同一个随机数 R,这将影响到其他电路门优化技术的应用,在 1.2 节将会看到这一点.

在 2014 年美密会上,Kolesnikov 等人又将 Free-XOR 技术一般化为 FleXOR 技术[4].FleXOR 技术的基本原理如图 3 所示,其输入输出线上随机数的"偏移"不同,分别为 $\Delta_1,\Delta_2,\Delta_3$,而输出线上的随机数是通过 2 个过渡值 $\widetilde{A},\widetilde{B}$ 计算得来.当电路计算方

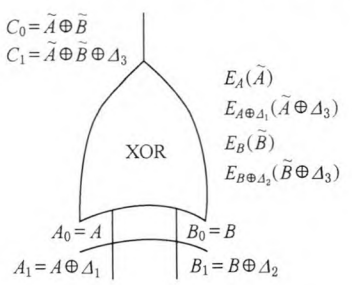


Fig. 3　The technic of FleXOR.

图 3　FleXOR 技术

拿到 2 个输入位所对应的随机值后,可以正确解密混乱值表中的 2 行,而 2 个解密值的异或,即为两个输入位进行或运算后计算结果所对应的随机值.

而在实际使用中,FleXOR 可以根据 XOR 门在整个电路中分布的具体情况,分别令 $\Delta_1=\Delta_3,\Delta_2=\Delta_3,\Delta_1=\Delta_2=\Delta_3$,再加上使用文献[5]中的行缩减技术(见 1.2 节),最终的混乱值表可以分别为 0 行、1 行或 2 行.

对于单纯的 XOR 门来说,Free-XOR 的效率比 FleXOR 效率更好,但是 FleXOR 可以与任意的其他门电路优化技术结合使用,所以对于整个混乱电路来说,FleXOR 技术可以比 Free-XOR 减少约 30%的电路规模.此外,FleXOR 还将 RO 假设减弱到 CR(correlation-robustness)假设,在 CR 模型下证明了该方法安全性,但仍然没有实现标准模型下的安全.

1.2　对混乱表的优化

Free-XOR 及 FleXOR 技术是针对于 XOR 门而设计的,对于其他的门电路,研究者提出一些方法来减少混乱表的行数.

Naor 等人在 1999 年提出一种方法[6],将混乱值表由 4 行减少到 3 行,他们的方法十分直接,如图 4 所示,还是以"与门"为例,将混乱值表的第 1 行直接规定为 0,然后修改随机数 $C_0=D_{A_0}(D_{B_0}(0))$.这种方法对任意的电路门都可以使用.特别的,令 $C_1=C_0\oplus R$,这个与门电路的输出线作为一个异或门的输入线时,对异或门就可以使用 Free-XOR 技术.

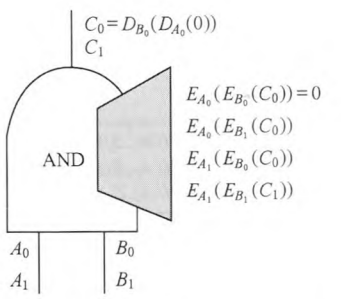


Fig. 4　Garbled computation with 3 rows.

图 4　3 行混乱表

在 2009 年亚密会上,Pinkas 等人进一步将混乱值表减少到 2 行[5].他们的方法使用了多项式插值的性质,并且输出线上的随机数不再通过解密计算得到,而是直接通过插值函数计算得到.Pinkas 首先根据电路门真值表中 1 的个数的奇偶性将电路门划分为奇门(如与门和或门)及偶门(如异或门),

然后按奇门及偶门分别设计了混乱表行缩减的优化方法.具体做法如图5所示:

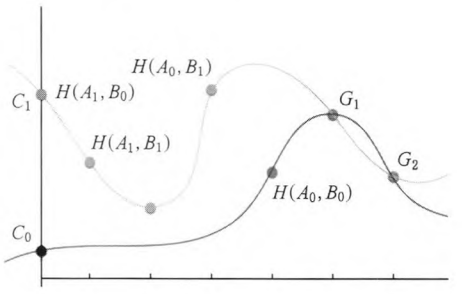


Fig. 5 Interpolation.

图5 插值计算

以或门(奇门)为例,首先为2根输入线选取4个随机数 A_0,A_1,B_0,B_1,然后选择一个Hash函数 H.取或门真值表中计算结果为1的3行所对应的3对随机密钥,计算 $H(A_1,B_1),H(A_0,B_1),H(A_1,B_0)$,然后利用这3点可以插值出一条曲线 $F(x)$,计算 $C_1=F(0)$;在 $F(x)$ 随机选取2个互异的非零点 G_1,G_2 作为公开参数,这样 $H(A_1,B_1),H(A_0,B_1),H(A_1,B_0)$ 中任意一点与 G_1,G_2 都可以形成3点插值以得到 $F(x)$,并计算出 $C_1=F(0)$.取或门真值表中计算结果为0的一行所对应的一对随机密钥,计算 $H(A_0,B_0)$,加上非零点 G_1,G_2,3点又可以插值出一条曲线 $G(x)$,计算 $C_0=G(0)$.通过这种方法,混乱表构造如图6所示,只有2行.在这种方法中,由于 C_0 和 C_1 的值由插值函数确定,也就是它们之间的"偏移"不能随意选择,因此该方法不能与Free-XOR技术一起使用.

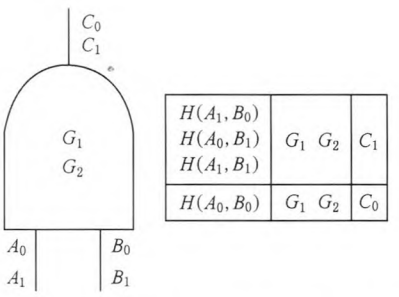


Fig. 6 Garbled computation with 2 rows.

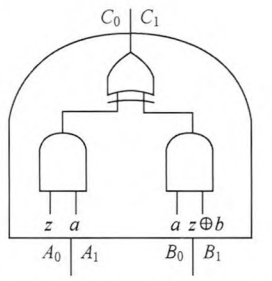
图6 2行混乱表

因为对门电路的真值表来说,其输出的真值有0或1两种情况,因此混乱电路中混乱值表最少也需要2行,Pinkas等人的方法已经将混乱表行数降低为最少,但该方法中多项式插值操作的主要开销为模幂运算,计算效率较低.

1.3 Half Gate优化技术

在2015年欧密会上,Zahur等人[7]提出了"半门"(Half-Gate)技术,进一步优化了混乱电路.

如图7所示,对一个与门来说,由于 $a\wedge b=(a\wedge z)\oplus(a\wedge(b\oplus z))$,其中 z 是一个随机比特,因此外部的 $a\wedge b$ 的电路门可以拆分成内部的 $(a\wedge z)$ 和 $(a\wedge(b\oplus z))$ 2个与门电路的异或.而内部的2个与门被称为"半门",它们具有一个特点:一个参与方知道一个输入,也就是,左边的与门,电路构造方知道 z,而右边的半门,电路计算方知道 $b\oplus z$.对于这类半门,首先利用Naor的行缩减技术[5],然后再使用Free-XOR技术[3],半门的混乱值表仅剩余一行(一个密文).从而外部的与门变为2个密文的异或,就可以使用Free-XOR技术.



$(a\wedge z)\oplus(a\wedge(b\oplus z))=a\wedge b$ z is a random bit

Fig. 7 The technic of Half-Gate.

图7 Half-Gate技术

利用Half-Gate技术,理论上可以减少33%的混乱电路规模,但是对电路计算者来说,每一个电路门都需要一步额外的Hash运算.

2 基于cut-and-choose技术的安全两方计算研究进展

最初Yao给出的安全两方计算协议[1]只能抵抗半诚实敌手的攻击,也就是说敌手是诚实但好奇的,他会诚实地执行协议,只是用观察到的消息来分析额外的结果.而对于恶意的敌手,他会主动改变协议的执行,从而获得期望的攻击结果.通用GMW编译器[2]使用零知识证明以及承诺等工具,迫使参与方必须依照协议来执行,从而可以将半诚实敌手下安全的协议编译为可抵抗恶意敌手攻击的协议.但也因为它大量地使用零知识证明以及承诺等协议,需要大量的公钥操作,非常低效.

为了避免大量使用零知识证明,同时又能规范参与方诚实执行协议,Pinkas等人[8]首次将cut-and-choose技术引入基于Yao混乱电路的安全两

方计算协议中.该方法要求混乱电路构造方构造 s 份(s 称为复制因子)混乱电路,混乱电路计算方随机选择其中一部分要求打开(称为检测电路)以检测电路的正确性;若全部检测通过,则双方像 Yao 协议一样计算剩余的每个电路(称为计算电路).基于 cut-and-choose 技术,当恶意的电路构造方有意地构造一个错误的电路时,有可能被选作检测电路而被发现,也有可能造成多个计算电路的输出不一致而被发现.通过适当的选取混乱电路的份数以及检测电路计算电路的划分方法,恶意敌手成功的可能行可以被降低到一个可忽略的概率,通常与 $-s$ 的指数相关.

尽管电路被扩展为 s 份,但由于混乱电路的构造方法,混乱电路的检测及计算都是高效的对称密码操作,因此 cut-and-choose 技术既可以抵抗恶意敌手攻击,又可以有效地提升效率.

鉴于 cut-and-choose 技术有效地减少了零知识证明等低效操作,这引起了研究者的广泛关注,并持续升温,近年来优秀的工作不断涌现.基于 cut-and-choose 技术应用的层次不同,这些工作大致可以分为 2 类:1)对整个混乱电路来实施 cut-and-choose 技术;2)针对混乱电路的电路门实施 cut-and-choose 技术.

2.1 对整个电路实施的 cut-and-choose

对整个电路实施 cut-and-choose 技术一般先由电路构造方先产生 s 份混乱电路,而电路计算方产生一个 cut-and-choose 挑战串,用于划分检测电路与计算电路.对于检测电路来说,每个门电路的输入输出线上所有的随机值以及门电路对应的混乱值表都要发送给电路计算方以用于检测.而对于计算电路来说,对每个门电路来说,其对应的混乱值表是要发送给电路计算方的,但是对于门电路输入线上的随机数,其发送分为 2 种情况.对于与电路构造方所对应的输入线,因为电路构造方知道自己真实输入,所以可以直接将与自己真实输入对应的随机数发送给电路构造方;而对于与电路计算方所对应的输入线,由于电路构造方不知道电路计算方的真实输入,所以需要与电路计算方执行一个 1-out-of-2 的 OT 协议,将与电路计算方真实输入对应的随机数茫然发送给电路计算方.

在基于 cut-and-choose 技术的安全两方计算协议中,电路的复制因子是一个关键的参数.首先协议的安全性与之有关,混乱电路复制的份数越多,敌手作弊被发现的概率越大,协议得到错误输出的可能(也就是错误概率)越小,但同时通信复杂度与计算复杂度也越大.因此在设计错误概率关于复制因子尽可能低的方案成为设计高效协议的关键.

文献[9]在理想/现实模拟范例下给出了可证安全的高效安全两方计算协议,其错误概率为 $2^{-s/17}$.之后,文献[10-11]分别给出了错误概率为 $2^{-0.311s}$ 和 $2^{-0.32s}$ 的结果.2013 年有 3 个工作[12-14]分别使用不同的方法设计出错误概率为 2^{-s} 的协议,其共同特点是只要所有计算电路中有一个是正确的,就能保证协议计算结果的正确性.

将 cut-and-choose 技术应用在安全多方计算协议中,需要克服 2 个比较关键的问题,分别是输入一致性问题以及选择失败攻击问题.

2.1.1 输入一致性问题

使用 cut-and-choose 技术需要将混乱电路复制多份,这就可能引发参与方在各混乱电路中输入不一致的问题,从而给协议的安全性带来影响.输入一致性问题包括电路计算方的输入一致性问题及电路构造方的输入一致性问题.

对于电路计算方的输入一致性问题,可以通过构造一个特殊的 OT 协议得到解决.早在欧密会 2007 上,Lindell 等人[9]构造了一种称为批处理 cut-and-choose OT 的协议,发送方输入 ns 对随机数,对应于 s 个混乱电路中,与电路计算方输入相对应的随机数.接收方输入 2 个挑战值,一个 n 位的值,对应于接收方的真实输入;一个 s 位的挑战值对应于 cut-and-choose 挑战串.该 OT 的输出分 2 种情况:当 cut-and-choose 挑战串的某比特为 1 时,该比特对应检测电路,接收方获得该检测电路上对应的 $2n$ 个随机数;当 cut-and-choose 挑战串的某比特为 0 时,该比特对应计算电路,接收方获得该计算电路中与与自己真实输入对应的 n 个随机数.这样,由于接收方的真实输入只被输入一次,因此不存在输入一致性问题.这种思想一直被后续工作采用来解决电路计算方的输入一致性问题.

对电路构造方的输入一致性问题,情况则比较复杂.早期的工作使用承诺方案来解决这一问题.文献[9,15]使用承诺方案给出了需要 $O(ns^2)$ 次对称密码操作的方法,其中 n 为参与方输入长度,s 为混乱电路的数目.这 2 个工作都避免了使用低效的非对称密码操作,但其通信代价过高,需要传输 $O(ns^2)$ 个承诺.

在 TCC2011 上,Lindell 等人[10]提出一种新的方法,利用计算代价代替通信代价,解决电路构造方

的输入一致性问题.具体的做法是,对于电路的 n 条与电路构造方对应的输入线,每条选择 2 个随机数,以这 $2n$ 个随机数为基础(底数);而对于 s 份电路,每份电路选择一个随机数作为指数,作用于 $2n$ 个基础底数上,产生混乱电路输入线上的随机值,此时数据的随机性是基于离散对数问题的.这样,对于 s 份混乱电路中的随机数,不再是完全随机选择的,而是让对应于同一输入线的 s 个随机值具备内在联系(有共同的底数),从而保证电路构造方输入一致性问题.这种方法将通信代价降为 $O(n+s)$,但需要 $O(ns)$ 次低效的模幂操作.基于文献[10]的思想,Mohassel 在美密会 2013 上[16],Shelat 在 CCS2013 上[17]将解决该问题的计算复杂度降为 $O(ns)$ 次对称密码操作.

在 TrustCom2015 上,赵川等人[18]进一步延展了 cut-and-choose OT 的思想,提出了一种 cut-and-choose 双向 OT 的密码学原语,对于一个混乱门电路来说,发送方 S 输入 $(x_0, x_1, y_0, y_1, \sigma)$,其中 x_0, x_1, y_0, y_1 对应于输入线上的 4 个随机数,σ 是电路构造方的真实输入,接收方输入 (j, τ),其中 j 是 cut-and-choose 挑战串,τ 是电路计算方的真实输入.协议执行完毕后,接收方的输出分 2 种情况:当 $j=1$ 时,接收方的输出为 (x_0, x_1, y_0, y_1);当 $j=0$ 时,接收方的输出为为 (x_σ, y_τ).该协议进行批处理后,发送方与接收方的真实输入都只被输入一次,因此发送方与接收方输入一致性问题都可以被解决.该 cut-and-choose 双向 OT 协议是基于加同态密码体制设计的,并在恶意敌手模型下利用理想现实模拟的方法证明了安全性.

2.1.2 选择失败攻击问题

在 2006 年,Kiraz 等人[19]指出文献[8]的 cut-and-choose 安全两方计算协议中存在一种选择失败攻击的漏洞,这是由 OT 协议本身的缺陷所带来的.当一个恶意的电路构造方在发送计算电路中与电路计算方输入相关的随机值时,在所执行的 OT 协议中刻意地构造一个正确的值、一个错误的值,随后观察电路计算方的反应,可以获得电路计算方 1 b 的输入.这是因为,电路计算方在后续的计算中,如果产生错误并终止协议,即诸计算电路产生了不一致的输出值,意味着在 OT 中获得了错误的值,反之如果电路计算方不终止协议,意味着在 OT 中获得了正确的值.为了抵抗这种攻击,文献[19]从 OT 协议本身入手,指出使用 committed OT 或 committing OT 技术,在 OT 协议执行过程中,就避免了数据发送方使用错误的数据,但是 committed OT 或 committing OT 本身的效率不高.

选择失败攻击的发现一个直接的结果就是当计算电路产生不一致的输出时,电路计算方不终止协议,而采用大多数的输出值作为最终的计算结果.这种方式虽然可以避免选择失败攻击,但是也增加了敌手欺骗成功的概率,因此要使敌手欺骗成功的概率不变,必需增大复制因子,从而造成协议效率变低.

文献[9]采用将计算方的每个输入位扩展为 s 位的异或方式,当敌手进行选择失败攻击后,只能获得扩展后的一位,而用户的真实输入是 s 位的异或,因此用户的真实输入泄漏的概率降至可忽略.该方法的计算代价为 $O(s\times\max(4n, 8s))$ 次非对称密码操作和 $O(s\times\max(4n, 8s))$ 次对称密码操作.此后,文献[10-11]分别使用 cut-and-choose OT 和改进的 committing OT 给出了需 $O(ns)$ 次非对称密码操作的解决方法,本质上没有改变输出大多数一致结果的思想.

在 2013 年美密会上,Lindell[12] 提出了一种创造性的方法,利用一个额外的"惩罚电路"实现了对选择失败攻击的抵抗.在他的方法中,先完成计算电路的计算,如果计算电路的输出不一致,电路计算方就获得 2 个不同的输出,这 2 个不同的输出可以构成一个有效的证据;如果计算电路的输出一致,电路计算方以计算电路的输出及一个随机数,构成一个无效的证据.下一步电路构造方与电路计算方执行一个附加的安全多方计算协议,该协议电路构造方输入其在原本两方计算协议中的输入,而电路计算方输入证据,当证据有效时,电路计算方可以获得电路计算方的真实输入,从而可以本地计算出功能函数的真实输出;当证据无效时,电路计算方得不到任何输出,这个两方计算函数的电路可以称为"惩罚电路".随后双方再完成检测电路的检测,根据检测电路的结果来完成整个安全两方计算协议.通过惩罚电路,恶意敌手唯一作弊而不被发现的情况就是:作弊电路未被选作检测电路;同时计算电路的输出为一致的并且是错误的.这就将协议得到错误输出的概率降到最小.

2.2 对电路门实施的 cut-and-choose

与上述将 cut-and-choose 技术应用到整个电路级别不同,Nielsen 和 Orlandi[20] 在 TCC2009 上首次提出将 cut-and-choose 技术应用到电路的门上,这种方法被其称之为"LEGO",寓意着混乱电路门可以像乐高积木一样被拼接为整个混乱电路.LEGO

方法要求混乱门构造方构造很多混乱的"与非"门发送给电路的计算方,计算方随机选择其中一部分要求打开以检测其正确性,检测通过后,剩余部分被随机置换,并在构造方的帮助下正确拼接成一个容错的 Yao 混乱电路.该容错混乱电路中的每个"门"都是一个冗余门,称为桶(bucket),每个桶中有若干个混乱门.在计算过程中,每个桶的输出取该桶中所有门的输出中占大多数的值,经过一层层计算,最终获得正确的电路输出值.该方法同样具有错误概率,但由于其复制因子相对于基于整个电路的 cut-and-choose 技术来说要低,所以协议的渐近效率更高.不过,由于该方法需要用到 Pedersen 同态承诺,因此每个门需要进行常数次模乘幂运算,效率较低,并且该构造中的密钥值是$\mathbb{Z}_p$中的群元素,而非随机比特串,因此许多对 Yao 混乱电路的优化方法(如 free-XOR)并不适用.此外,该方案的安全性是建立在一个特殊的数论假设之上的,不具备一般性.

Frederiksen 等人在欧密会 2013 上对文献[20]中的 LEGO 方案进行了改进[21],称为 MiniLEGO.他们首先摒弃了 Pedersen 同态承诺,而采用了一种基于 OT 协议的 XOR-同态承诺协议.由于 OT 协议可以被高效的扩展(见本文第 3 节),仅少量的基础 OT 需要公钥操作,大量的扩展 OT 仅需要对称密码操作就可完成.鉴于基础 OT 的数量远远小于扩展 OT 的数量,因此平均每个 OT 仅需要少量的对称密码操作就可以完成,进一步,平均到对每个门的 XOR-同态承诺,也仅需要少量的对称密码操作就可以完成.同时,鉴于 XOR-同态承诺不再要求随机数为群元素,因此 MiniLEGO 可以使用标准的电路门进行构造,这样对 Yao 混乱门的已知优化方法都适用,所有这些,使得 MiniLEGO 的整体效率较之前的 LEGO 方法大大提高.

下面的表 1 给出了对基于 cut-and-choose 技术的安全两方计算协议的各项对比.

表 1 基于 cut-and-choose 技术的安全两方计算协议对比

Table1 Comparison of Secure Two-Party Computation Protocols Based on cut-and-choose

Level of cut-and-choose	Reference	Model/Assumption	Computational Complexity		Communication Complexity	Error Rate
			Symmetric	Non-Symmetric		
Circuit Level	Ref[9]	SM	$O(s\|C\|+s^2n)$	$O(s^2n)$	$O(ts\|C\|+ts^2n)$	$2^{-s/17}$
	Ref[10]	SM,DDH	$O(s\|C\|)$	$O(sn)$	$O(ts\|C\|)$	$2^{-0.311s}$
	Ref[18]	SM,DDH	$O(s\|C\|)$	$O(sn)$	$O(ts\|C\|)$	$2^{-0.32s}$
	Ref[11]	SM,DDH	$O(s\|C\|)$	$O(sn)$	$O(ts\|C\|)$	2^{-s}
	Ref[13]	ROM,CDH	$O(s\|C\|)$	$O(sn)$	$O(ts\|C\|)$	$2^{-s+O(\log s)}$
	Ref[12]	SM,DL,DDH	$O(s\|C\|)$	$O(n)$	$O(ts\|C\|)$	2^{-s}
Gate Level	Ref[19]	UC,DL,CoRH	$O(s\|C\|/\log(\|C\|))$	$O(s\|C\|/\log(\|C\|))$	$O(ts\|C\|/\log(\|C\|))$	2^{-s}
	Ref[20]	ROM	$O(s\|C\|/\log(\|C\|))$	$O(s)$	$O(ts\|C\|/\log(\|C\|))$	2^{-s}

Note: n is the length of the input/output, s is a secure parameter in statistical, t is a secure parameter in computation, $|C|$ is the circuit size. SM denotes standard model, ROM denotes random oracle model.

3 OT 扩展研究进展

不经意传输(oblivious transfer, OT)是最基础的密码学原语之一,最早由 Rabin 在 1981 年提出[22],在 OT 的最常用的 1-out-of-2 版本[23]中,发送方持有一对消息 m_0, m_1,而接收者持有一个选择比特 b,当 OT 协议执行完成之后,接收者可以获得消息 m_b,但对消息 m_{1-b}一无所知,并且发送者不知道关于 b 的任何信息.

OT 协议是所有安全多方计算协议中的最基本工具,在基于 Yao 混乱电路[1]构造的 SMPC 协议中,参与方每 bit 输入需要一个 OT 协议,而在基于 GMW 范式[7]构造的 SMPC 协议中,布尔电路的每一个 And 门需要至少一个 OT 协议.因此在恶意敌手模型下的实际的安全多方计算协议所需要执行的 OT 次数需要数百万次.举例来说,利用 TinyOT[24]技术,当计算 AES 电路时,需要使用 2^{19} 次 OT 协议,而计算隐私集合求交(private set intersection, PSI)电路时,需要使用 2^{30} 次 OT 协议.因此 OT 协议执行的效率成为了影响安全多方计算协议效率的最重要的因素.

早在1988年,Impagliazzo和Rudich就已经证明了不能使用黑盒构造的方法从一个单向函数来实现OT协议[25].因此,尽管现在OT协议的设计技术已经非常成熟高效,但它任然需要模指数运算,比如Peikert等人提出的高效OT协议[26],每秒钟可以计算350次OT协议,也就是说执行2^{20}次OT协议(安全计算AES所需的OT协议量级),需要超过45 min,执行2^{30}次OT协议(安全计算PSI所需的OT协议量级),需要超过45 000 min(超过1个月),这在实际应用中是远远不能接受的.

在密码学的发展过程中,公钥密码学的出现解决的传统对称密码学中密钥管理的困难,但是公钥密码学需要低效的模指数运算,不适合用于加/解密大数据,于是研究者提出使用公钥密码来加密对称密码算法的密钥,然后使用对称密码算法来加/解密实际数据的方法,被称为混合加密.借鉴这种思想,在1996年Beaver首先提出一种称为OT扩展的技术[27],一个OT扩展协议通过运行少量的OT协议(例如几百个)作为基础,通过使用廉价的伪随机置换操作,来获得大量(例如几百万个)OT协议执行的效果.文献[27]构造的OT扩展协议使用了随机OT的工具,效率不高,仅具备理论上的意义.而在美密会2003上,Ishai等人提出了一个半诚实敌手模型下安全的OT扩展协议[28],是第1个高效地将n个基础OT协议扩展为$m(\gg n)$个OT协议的工作.

协议输入:发送方的输入$(x_1^0,x_1^1),(x_2^0,x_2^1),\cdots,(x_m^0,x_m^1)$,为$m$对随机密钥值;接收方的输入$\sigma=\sigma_1,\sigma_2,\cdots,\sigma_m$为$m$位二进制串.

协议输出:发送方输出为空;接收方输出为$x_1^{\sigma_1},x_2^{\sigma_2},\cdots,x_m^{\sigma_m}$.

1) 第1阶段:基础OT阶段(如图8所示).

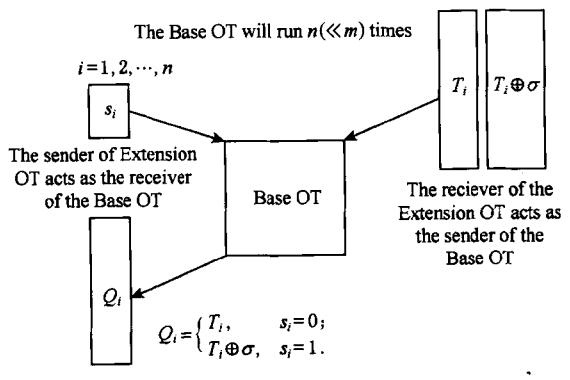


Fig. 8 The map of first stage.

图8 第1阶段示意图

① 接收方随机选择n个长度为m的随机串$T_1,T_2,\cdots,T_n$,对$i=1,2,\cdots,n$,计算n个数据对$(T_i,T_i\oplus\sigma)$;

② 发送方随机选择n位二进制串$s=s_1,s_2,\cdots,s_n$.

③ 对$i=1,2,\cdots,n$,OT扩展协议的接收方作为一个基础OT协议的发送方,输入$(T_i,T_i\oplus\sigma)$,而OT扩展协议的发送方作为一个基础OT协议的接收方,输入s_i,双方执行一个基础的OT协议,OT扩展协议的发送方获得Q_i.

第1阶段完成之后,发送者获得n个m位的二进制串$Q_1,Q_2,\cdots,Q_n$,将其分别看作一个m维列向量,$Q_1,Q_2,\cdots,Q_n$可以拼合成一个$m\times n$的矩阵,相应的接收者的输入$T_1,T_2,\cdots,T_n$也可以看成一个矩阵,分别记2个矩阵的第i行为$Q(i)$和$T(i)$,如图9所示:

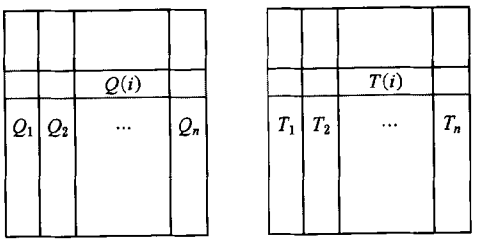


Fig. 9 Result of Base OT.

图9 基础OT执行结果

可见,由基础OT的执行过程,对接收者输入$\sigma=\sigma_1,\cdots,\sigma_m$的第$i$位$\sigma_i$,对$j=1,2,\cdots,n$,如果$\sigma_i=0$,则无论$s_i$等于什么,$Q(i)=T(i)$;如果$\sigma_i=1$,则对$Q(i)$和$T(i)$的第$j$位$Q_j(i)$和$T_j(i)$:

若$s_j=0$,则$Q_j(i)=T_j(i)=T(i)\oplus 0=T(i)\oplus s_j$;

若$s_j=1$,则$Q_j(i)=T_j(i)\oplus 0=T_j(i)\oplus 1=T_j(i)\oplus s_j$.

总之,无论s_j等于什么,$Q_j(i)=T_j(i)\oplus s_j$,从而$Q(i)=T(i)\oplus s$.

2) 第2阶段:扩展阶段.

对$i=1,2,\cdots,m$,

① 发送方发送:

$$y_i^0=H(i,Q(i))\oplus x_i^0,$$

$$y_i^1=H(i,Q(i)\oplus s)\oplus x_i^1;$$

② 接收方计算$x_i^{\sigma_i}=H(i,T(i))\oplus y_i^{\sigma_i}$.

针对OT扩展的研究,主要集中在2个方面:1)在半诚实敌手模型下进一步提高协议的效率;2)力争在恶意敌手模型下设计OT扩展协议.

3.1 半诚实敌手模型下高效的OT扩展协议

尽管Ishai等人的OT扩展方案[28]已经惊人的高效,但是2013年,有2篇工作在半诚实敌手模型下对其作了进一步的改进.

在2013年美密会上,Kolesnikov等人[29]将文献[28]中的1-out-of-2的OT扩展方案发展为1-out-of-n的OT扩展方案.由于一个传输$\log n$长比特串的1-out-of-n的OT方案等于$\log n$个传输1 b的1-out-of-2的OT方案,因此将1-out-of-n的OT扩展应用到安全多方计算协议中,相当于对安全多方计算中用到的传输1 b的1-out-of-2的OT方案进行批处理,再加上OT扩展技术,总体上提高安全多方计算协议的效率.Kolesnikov的方案可以将计算夫再度和通讯复杂度降低为原来的$1/O(\log n)$.

利用1-out-of-k的OT方案,将文献[28]方案的通信及计算复杂性降为原来的$1/O(\log n)$.他们观察到,在文献[28]原始方案的第1阶段,如图9所示,双方传输的是一个$m\times n$矩阵,而传输的方式是按列来执行OT协议,双方执行的是n个基础的1-out-of-2的OT方案,传输的是n个m位长的字符串.

在2013年CCS上,Asharov等人[30]首先在标准模型下构造了一个新的OT方案,并将其用于OT扩展,降低了计算和通信复杂性,随后,又针对基于Yao混乱电路和GMW范例的安全多方计算协议中OT协议的不同使用特点,分别对OT扩展协议进行了优化.随后他们进行了实验验证,将他们优化后的OT扩展应用到已有的安全多方计算框架,对一个具有1.29×10^9个与门的求解编辑距离(Levenshtein distance)问题的电路,他们的实验结果是每秒钟可以计算1.2×10^6个与门.这表明了半诚实敌手模型下安全的多方计算协议,其效率瓶颈已经不再是计算量,而变成了通讯量.

3.2 恶意敌手模型下的OT扩展协议

虽然我们说上述协议[28]是在半诚实敌手模型下安全的,但事实上,它可以抵抗恶意的发送者,但是不能抵抗恶意的接收者的攻击.如果恶意的接收者在第1阶段执行n个基础OT协议过程中,使用了不同的σ,则可以抽取发送方选择的随机数s,从而在第2阶段扩展阶段同时获得x_i^0,x_i^1.因此为了防止恶意接收者攻击,主要手段就是对接收者的输入进行一致性检测.为了获得恶意敌手模型下安全的OT扩展协议,现有下述2类方法:

1) 文献[28]中提出的利用cut-and-choose技术的方法,文献[31-32]进一步发展了这一方法,这类方法利用cut-and-choose技术来检测接收方的输入一致性,其检测步骤作用在每一个OT扩展阶段,增加了扩展OT协议的计算量,相对于半诚实的协议来说,计算负载增加比例较大,并且这类方法都是在Random Oracle模型下证明安全的.

2) 首先由Nielsen等人[33]在2012年美密会提出的方案利用Hash函数进行一致性检测,并且其检测步骤作用于基础OT阶段,仅增加了基础OT协议的计算量,由于基础OT协议的数量远远少于扩展OT的数量,因此该方法效率较高,但该方案仅在Random Oracle模型中证明了安全性.

在2015年欧密会上,Asharov等人在文献[33]的基础上作了进一步的改进[34],增加了少量的基础OT,减少了大量的输入一致性检测,从而提高了效率.在第1阶段的n次基础OT中,文献[33]的输入一致性检测要保证扩展OT的发送方在任意两次的OT都使用相同的,以n次OT中的输入为顶点,如果两次OT中的输入进行一致性检测,则在这两点间连接一条边,所以文献[33]的一致性检测的次数是一个n个顶点的完全图的边数.Asharov方法是增大n的值,然后在n个顶点的图中选择一个随机d次正则图来进行一致性检测,这种方法将大大减少一致性检测的次数,他们进一步讨论了参数n,d的选择方法.除了效率提升之外,该方案不但在Random Oracle模型中证明了安全性,进一步,如果要求进行一致性检测Hash函数满足一种称为最小熵鲁棒性(min-entropy correlation)的特性,该协议可以在标准模型下证明安全性.

4 结束语

从本文的介绍来看,对于半诚实敌手下的安全多方计算协议,其效率已经得到了充分的提升,影响其效率的瓶颈已经成为通信带宽,而不再是计算量.为了得到恶意敌手下安全的协议,现在的思想就是避免或减少使用低效的公钥操作(如零知识证明及OT等),而尽量采用高效的对称密钥操作.现有的途径,如cut-and-choose及OT扩展都需要用户多次使用自己的输入,这就带来了输入一致性检测的问题,为了降低敌手欺骗成功的概率,又引入了额外承诺及零知识证明操作,并且敌手欺骗成功的概率

越低,所需的计算量越大.对于恶意敌手模型下的协议,还有待更好的解决方案.

从最初仅仅停留在理论上的成果,发展到20世纪初的可用阶段,再到目前的可实用阶段,安全多方计算协议在实用化的道路上一直前进,并且近年来有加速发展的趋势,其发展历程同公钥密码学的发展类似.我们期待在不久的将来,安全多方计算协议能够真正用于实际问题的解决,在信息安全应用领域,可以像目前的公钥密码学一样无所不在.

参 考 文 献

[1] Yao A. How to generate and exchange secrets [C] //Proc of the 27th IEEE Symp on Foundations of Computer Science (FOCS1986). Los Alamitos, CA: IEEE Computer Society, 1986: 162-167

[2] Goldreich O, Micali S, Wigderson A. How to play any mental game—A completeness theorem for protocols with honest majority [C] //Proc of the 19th Annual ACM Symp on Theory of Computing. New York: ACM, 1987: 218-229

[3] Kolesnikov V, Schneider T. Improved garbled circuit: Free XOR gates and applications [G] //LNSC 5126: Automata, Languages and Programming. Berlin: Springer, 2008: 486-498

[4] Kolesnikov V, Mohassel P, Rosulek M. FleXOR: Flexible garbling for XOR gates that beats free-XOR [G] //LNCS 8617: Advances in Cryptology (CRYPTO 2014). Berlin: Springer, 2014: 440-457

[5] Pinkas B, Schneider T, Smart N P, et al. Secure two-party computation is practical [G] //LNCS 5912: Advances in Cryptology (ASIACRYPT 2009). Berlin: Springer, 2009: 250-267

[6] Naor M, Pinkas B, Sumner R. Privacy preserving auctions and mechanism design [C] //Proc of the 1st ACM Conf on Electronic Commerce. New York: ACM, 1999: 129-139

[7] Zahur S, Rosulek M, Evans D. Two halves make a whole [G] //LNCS 9057: Advances in Cryptology (EUROCRYPT 2015). Berlin: Springer, 2015: 220-250

[8] Pinkas B. Fair secure two-party computation [G] //LNCS 2656: Advances in Cryptology (Eurocrypt 2003). Berlin: Springer, 2003: 87-105

[9] Lindell Y, Pinkas B. An efficient protocol for secure two-party computation in the presence of malicious adversaries [G] //LNCS 4515: Advances in Cryptology (EUROCRYPT 2007). Berlin: Springer, 2007: 52-78

[10] Lindell Y, Pinkas B. Secure two-party computation via cut-and-choose oblivious transfer [G] //LNCS 6597: Advances in TCC 2011. Berlin: Springer, 2011: 329-346

[11] Shelat A, Shen C H. Two-output secure computation with malicious adversaries [G] //LNCS 6632: Advances in cryptology (EUROCRYPT 2011). Berlin: Springer, 2011: 386-405

[12] Lindell Y. Fast cut-and-choose based protocols for malicious and covert adversaries [G] //LNCS 8043: Advances in Cryptology (CRYPTO 2013). Berlin: Springer, 2013: 1-17

[13] Brandão L T A N. Secure two-party computation with reusable bit-commitments, via a cut-and-choose with forge-and-lose technique [G] //LNCS 8270: Advances in Cryptology (ASIACRYPT 2013). Berlin: Springer, 2013: 441-463

[14] Huang Y, Katz J, Evans D. Efficient secure two-party computation using symmetric cut-and-choose [G] //LNCS 8043: Advances in Cryptology (CRYPTO 2013). Berlin: Springer, 2013: 18-35

[15] Mohassel P, Franklin M. Efficiency tradeoffs for malicious two-party computation [M] //LNCS 3958: Public Key Cryptography (PKC 2006). Berlin: Springer, 2006: 458-473

[16] Mohassel P, Riva B. Garbled circuits checking garbled circuits: More efficient and secure two-party computation [G] //LNCS 8043: Advances in Cryptology (CRYPTO 2013). Berlin: Springer, 2013: 36-53

[17] Shelat A, Shen C. Fast two-party secure computation with minimal assumptions [C] //Proc of the 2013 ACM SIGSAC Conf on Computer & Communications Security. New York: ACM, 2013: 523-534

[18] Zhao Chuan, Jiang Han, Wei Xiaochao, et al. Cut-and-choose bilateral oblivious transfer and its application [C] // Proc of the 14th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Los Alamitos, CA: IEEE Computer Society, 2015: 384-391

[19] Kiraz M, Schoenmakers B. A protocol issue for the malicious case of Yao's garbled circuit construction [C] //Proc of the 27th Symp on Information Theory in the Benelux. Eindhoven: Werkgemeenschap voor Informatie-en Communicatietheorie, 2006: 283-290

[20] Nielsen J B, Orlandi C. LEGO for two-party secure computation [G] //LNCS 5444: Theory of Cryptography. Berlin: Springer, 2009: 368-386

[21] Frederiksen T K, Jakobsen T P, Nielsen J B, et al. Minilego: Efficient secure two-party computation from general assumptions [G] //LNCS 4515: Advances in Cryptology (EUROCRYPT 2013). Berlin: Springer, 2013: 537-556

[22] Rabin M O. How to exchange secrets by oblivious transfer, TR-81 [R]. Boston: Harvard University, 1981

[23] Even S, Goldreich O, Lempel A. A randomized protocol for signing contracts [J]. Communications of the ACM, 1985, 28(6): 637-647

[24] Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation [G] //LNCS 7417: Advances in Cryptology (CRYPTO 2012). Berlin: Springer, 2012: 681-700

[25] Impagliazzo R, Rudich S. Limits on the provable consequences of one-way permutations [G] //LNCS 403: Advances in Cryptology (CRYPTO'88). Berlin: Springer, 1990: 8-26

[26] Peikert C, Vaikuntanathan V, Waters B. A framework for efficient and composable oblivious transfer [G] //LNCS 5157: Advances in Cryptology (CRYPTO 2008). Berlin: Springer, 2008: 554-571

[27] Beaver D. Correlated pseudorandomness and the complexity of private computations [C] //Proc of the 28th Annual ACM Symp on Theory of Computing (STOC'96). New York: ACM, 1996: 479-488

[28] Ishai Y, Kilian J, Nissim K, et al. Extending oblivious transfers efficiently [G] //LNCS 2729: Advances in Cryptology (CRYPTO 2003). Berlin: Springer, 2003: 145-161

[29] Kolesnikov V, Kumaresan R. Improved OT extension for transferring short secrets [G] //LNCS 8043: Advances in Cryptology (CRYPTO 2013). Berlin: Springer, 2013: 54-70

[30] Asharov G, Lindell Y, Schneider T, et al. More efficient oblivious transfer and extensions for faster secure computation [C] //Proc of the 20th ACM SIGSAC Conf on Computer & Communications Security (CCS13). New York: ACM, 2013: 535-548

[31] Nielsen J B. Extending oblivious transfers efficiently-how to get robustness almost for free [DB/OL]. IACR Cryptology ePrint Archive, 2007 [2015-06-07]. http://eprint.iacr.org/2007/215

[32] Harnik D, Ishai Y, Kushilevitz E, et al. OT-combiners via secure computation [G] //LNCS 4948: Theory of Cryptography. Berlin: Springer, 2008: 393-411

[33] Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation [G] //LNCS 7417: Advances in Cryptology (CRYPTO 2012). Berlin: Springer, 2012: 681-700

[34] Asharov G, Lindell Y, Schneider T, et al. More efficient oblivious transfer extensions with security for malicious adversaries [G] //LNCS 9056: Advances in Cryptology (EUROCRYPT 2015). Berlin: Springer, 2015: 673-701

Jiang Han, born in 1974. Lecturer of Shandong University since 2009. His main research interests include cryptography and information security, especially secure multi-party computation.

Xu Qiuliang, born in 1960. Professor and PhD supervisor in Shandong University. His main research interests include public key cryptography and multi-party secure computation.