基于可验证混淆电路的合作式安全两方计算协议

张宗洋 刘翔宇 李威翰 陈 劳
(北京航空航天大学网络空间安全学院 北京 100191)

摘 要 本文针对基于混淆电路的安全两方计算协议通信复杂度过高的问题,研究如何进一步优化协议的性能.本文基于可验证混淆电路分享方案,在恶意敌手模型下实现了一种更高效的安全两方计算协议.主要创新性工作包括两个方面:(1)实现了一种新的可验证混淆电路分享方案.该方案延续了将混淆电路与秘密分享结合的思路,在 Three-Halves 混淆电路中结合可验证随机比特技术来生成混淆电路分享份额,保证了在恶意敌手模型下的安全性,与门运算的通信复杂度降低了约 25%,而或门运算仍是零通信开销的;(2)提出了一个安全两方计算协议.设计了一种合作式协议流程设计方案,通过划分布尔电路的方式,由两个参与方各承担一半电路的混淆或分析工作,合作完成计算任务,分摊了安全两方计算协议中的计算压力.利用提出的可验证混淆电路分享方案,基于哈希函数等密码学工具,本协议保证了电路计算的正确性以及在恶意敌手模型下的安全性.与 Emp-toolkit 两方协议相比,本协议通信时延优化了 1%~9%,计算时延优化了 5%~22%,通信量优化了 40%~60%.

关键词 混淆电路;秘密分享;掩码技术;布尔电路;安全两方计算
中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2022.02433

Efficient and Cooperative Secure Two-Party Computation Based on Authenticated Garbled Circuit

ZHANG Zong-Yang LIU Xiang-Yu LI Wei-Han CHEN Lao
(*School of Cyber Science and Technology, Beihang University, Beijing* 100191)

Abstract With the continuous improvement of informatization, new technologies, such as big data, artificial intelligence and blockchain, have been applied to all walks of life. These new technologies rely on data sharing across fields and enterprises. This way promotes the rapid development of industries such as finance, medical care, and commerce, but threatens the security of private data. How to achieve efficient data sharing and data interoperability while ensuring security is a problem to be solved in the current privacy protection field. Secure multi-party computation is one of high-level cryptography providing a new idea to ensure privacy. The secure two-party computation is the basis for constructing secure multi-party computation protocols. And it can also solve security problems of privacy data well in two-party applications, such as genome sequence alignment and pattern matching. Thus, secure two-party computation is one of the current research hotspots. The existing secure two-party computation protocols based on garbled circuit often have high communication complexity and poor performance in the malicious adversary model. This paper mainly studies how to optimize secure two-party computation protocols following the idea of Wang et al (CCS 2017). This paper implements a more efficient secure two-party computation protocol based on an authenticated garbling scheme under malicious adversary model.

收稿日期:2021-12-14;在线发布日期:2022-05-23.本课题得到国家重点研究计划项目(2021YFB3100400)、国家自然科学基金项目(61972017,72031001,61972310)、北京市自然科学基金(M22038,4202037)、中央高校基本科研业务费(YWF-22-L-1039)、云南省区块链应用技术重点实验室(培育)开放课题(YNB202101)资助.**张宗洋**,博士,副教授,中国计算机学会(CCF)高级会员,主要研究方向为区块链、密码学.E-mail: zongyangzhang@buaa.edu.cn. **刘翔宇**,硕士,中国计算机学会(CCF)会员,主要研究方向为密码学. **李威翰**,博士研究生,主要研究方向为零知识证明、区块链. **陈 劳**,博士研究生,助理研究员,主要研究方向为网络安全和区块链.

The main contributions are summarized as follows. (1) We implement a new authenticated garbling scheme. This scheme follows the idea of combining garbled circuit with secret sharing, which is proposed by Wang et al(CCS 2017). We combine the authenticated random bit sharing scheme with Three-Halves garbled circuit, and implement a new authenticated garbling scheme. This scheme reduces the communication complexity of AND gates operation by about 25%, while ensuring security against malicious adversaries. Meanwhile, XOR gates require no communication. (2) We propose a secure two-party computation protocol. We design a cooperative process scheme for the secure two-party computation protocol. This scheme aims at the problem of unbalanced computation pressure between garbler and evaluator. By dividing the Boolean circuit, each participant undertakes half of the circuit garbling or analyzing work and completes the computation task cooperatively. Then, based on the authenticated distributed garbling scheme proposed in this paper, hash function and other cryptographic technologies, this protocol guarantees correctness and security against malicious adversary model. Compared with the two-party computation protocol in Emp-toolkit, this protocol results in a 1%～9% improvement in the communication delay, a 5%～22% improvement in the computation delay and a 40%～60% improvement in the communication cost.

Keywords garbled circuit; secret sharing; Point-and-Permute; Boolean circuit; secure two-party computation

1 引 言

近些年来,跨领域、跨企业等方式的数据共享为商业营销、医疗科学等领域的发展带来了巨大契机,但是也增加了隐私泄露的风险.如何在保证数据安全的情况下共享数据并进行数据互操作,是隐私保护领域的研究热点.安全多方计算协议能够在不暴露个人输入数据的情况下,联合多参与方完成某计算任务,进而为解决数据共享中的数据安全问题提供了很好的解决方案.1880 年到 21 世纪初期,安全多方计算协议大多停留在理论性研究阶段,复杂度过高、计算能力有限.21 世纪初期至今,安全多方计算协议逐渐走向落地应用,用以满足现实应用场景的隐私保护需求.

通用安全多方计算协议实现针对布尔电路和算术电路的计算任务.根据核心技术的不同,通用安全多方计算协议大致可被划分为基于混淆电路[1]的安全多方计算协议和基于秘密分享[2-3]的安全多方计算协议.姚氏电路协议基于混淆电路实现,起源于姚期智院士提出的姚氏协议[4].为适应现实中数据规模大、计算复杂的应用场景,姚氏电路协议的高效性亟待进一步研究.本文重点研究两方场景下的姚氏电路协议,即基于混淆电路的安全两方计算协议.该类协议既可以用于构建多方场景下安全多方计算协议,也可以直接解决两方场景下的数据安全问题.

在基于混淆电路的安全两方计算协议中,两方交互次数为常数次,具有低时延的优势.假设布尔电路 C 的规模为 $|C|$,则该类协议的通信复杂度至少达 $O(|C|)$.当参与方输入的数据量级达到亿级时,布尔电路规模将呈爆炸式增长,混淆电路的规模也随之大幅增长.因此,该类协议的通信复杂度过高,吞吐量低,不适合在带宽有限的网络环境下使用.另外,该类协议仅具有半诚实敌手模型下的安全性,实现恶意敌手模型下的安全性需要结合秘密分享等其他技术.

1.1 本文贡献

本文设计了高效且安全的可验证混淆电路分享方案,主要研究了基于混淆电路的安全两方计算协议,具体贡献总结如下:

(1)提出了一个新的可验证混淆电路分享方案,基于将混淆电路与秘密分享结合的思路,在 Three-Halves 混淆电路中结合可验证随机比特技术,提出了混淆电路分享份额的新生成方案,保证了在恶意敌手模型下的安全性,或门运算仍是零通信开销的,与门运算的通信复杂度降低了约 25%.

(2)提出了一个安全两方计算协议,设计了合作式协议流程方案,针对参与方之间计算压力不平

衡的问题，将布尔电路划分为前后两半部分，由两个参与方各承担一半电路的混淆与分析工作，合作完成计算任务的，从而分摊计算压力. 基于掩码技术和承诺方案，保证了合作式协议流程方案中数据的隐私性与正确性. 本文提出的可验证混淆电路分享方案保证了该协议计算的正确性以及在恶意敌手模型下的安全性. Emp-toolkit[5] 是基于混淆电路的安全多方计算框架. 该框架最先实现了基于可验证混淆电路分享方案的安全两方计算协议. 与 Emp-toolkit 中恶意敌手模型下安全的安全两方计算协议相比，本文协议的通信时延优化了 1%～9%，计算时延优化了 5%～22%，通信量优化了 40%～60%.

1.2　相关工作

安全多方计算协议包含通用安全多方计算协议和解决特定应用问题的安全多方计算协议两大类. 其中，通用安全多方计算协议主要解决域上算术电路、布尔电路的计算问题，其可分为姚氏电路协议[4]、多方电路协议[6-7] 和混合模型协议[8-9]. 姚氏电路协议起源于姚期智院士的姚氏协议，该类协议只要将任意函数转换为布尔电路就可以完成计算，主要基于混淆电路技术实现，但通信量过高. 之后，学者们对混淆电路技术开展了大量优化工作. 其中，减小混淆表规模是主要的优化方向，现有的优化方案包括 P&P[10]、GRR3[11]、Free-XOR[12]、GRR2[13]、Half-Gates[14] 以及 Three-Halves[15]. 近年来，学者们还致力于研究减小完整混淆电路规模的优化方案，包括 SGC[16] 和 LogStack[17] 等.

基于混淆电路技术的安全计算协议仅能实现在半诚实敌手模型下的安全性. 如果需要实现在恶意敌手模型下的安全性，需要额外设计方案或结合其他密码学技术实现. 现有方案包括 Cut-and-Choose 方案[18-21]、非交互式方案[22]、基于 DDH 假设的可重随机化混淆电路[23]、单轮执行方案[24]、线上线下批处理方案[25]、DUPLO 方案[26]、可验证混淆电路分享方案[27-29] 等.

可验证混淆电路分享方案[27] 结合了混淆电路技术与秘密分享技术，可以在恶意敌手模型下以较少交互次数完成计算任务. 具体的，基于可验证随机比特分享技术，参与方分别生成一份混淆电路分享份额. 在这种情况下，恶意的混淆方不知道完整混淆电路的结构，无法通过终止计算信息推测出诚实计算方的输入. 文献[27]中协议使用的混淆电路技术为 Free-XOR[12]，两方协议的通信复杂度为 $4\kappa|C|+\kappa(|I|+|O|)$，其中 κ 表示安全参数，$|I|$ 表示电路输入集合的大小，$|O|$ 表示电路输出集合的大小. 文献[28]基于 Half-Gates[14] 提出了新的可验证的混淆电路分享方案，将安全两方计算协议的通信复杂度降低到 $2\kappa|C|+\kappa(|I|+|O|)$. 由于两方场景下的可验证混淆电路分享方案可以扩展到多方计算场景[29-30]，进一步优化两方场景下的可验证混淆电路分享方案具有重要的研究意义.

2　预备知识

2.1　安全模型

2.1.1　敌手模型

在安全两方计算协议中，敌手可能控制某一参与方参与计算，通过分析参与方之间的交互消息或不遵守协议执行流程，获取诚实参与方的输入信息，或者致使计算错误. 根据敌手的行为，敌手模型主要包括半诚实敌手模型和恶意敌手模型.

(1) 半诚实敌手模型. 在该模型下，敌手会遵循协议的流程完成计算，但是敌手试图分析交互消息得到额外的信息.

(2) 恶意敌手模型. 在该模型下，敌手能以任意的恶意行为破坏协议流程. 例如，敌手可能在协议执行的任何位置终止，导致诚实参与方无法获取输出结果；敌手可能随意伪造混淆电路，根据诚实参与方的终止情况推测其输入.

2.1.2　通用方案定义及安全属性

根据文献[15]，混淆电路技术的通用方案由算法组(Garble，Encode，Eval，Decode)组成.

定义 1.　混淆电路方案 G 由 4 个算法组成：

(1) $(F,e,d)\leftarrow \mathrm{Garble}(1^{\kappa},f)$；

(2) $X:=\mathrm{Encode}(e,x)$；

(3) $Y:=\mathrm{Eval}(F,X)$；

(4) $y:=\mathrm{Decode}(d,Y)$.

如果一个混淆电路方案是安全的，那么其一般具有 4 个安全属性，分别是正确性、隐私性、不经意性和可认证性.

(1) 正确性. 对于任意的电路 f 以及输入 x，在随机选取一个混淆电路副本之后，即 $(F,e,d)\leftarrow \mathrm{Garble}(1^{\kappa},f)$，等式

$$f(x)=\mathrm{Decode}(d,\mathrm{Eval}(\mathrm{Encode}(e,x)))$$

成立的概率接近于 1.

也就是说，若参与方均遵循协议，则能够计算得到正确的输出结果.

(2) 针对泄露函数 ϕ 的隐私性. 对于任意的电路

f 以及输入 x，参与方生成的实际混淆分布与任意模拟器 $\mathcal{S}(1^\kappa,\Phi(f),f(x))$ 仿真生成的随机混淆结构是不可区分的，即任何一个区分器可以区分如图 1 所示的现实实验与仿真实验的概率是可以忽略的. 这里所说的混淆结构包括混淆电路 F、输入线路对应标签 X 以及标签分析算法 d. $\Phi(f)$ 是有关电路 f 的提示信息.

现实实验	仿真实验
$(F,e,d)\leftarrow\text{Garble}(1^\kappa,f)$ $X:=\text{Encode}(e,x)$ return (F,X,d)	$(F,e,d)\leftarrow S(1^\kappa,\Phi(f),f(x))$ return (F,X,d)

图 1 隐私性装置图[15]

在协议的执行过程中，各参与方不会获得除了自己持有的秘密信息和计算结果以外的任何暴露隐私的额外信息. 此处强调的是从计算结果分析得到的信息不属于额外信息的范围.

(3) 针对泄露函数 ϕ 的不经意性. 对于任意的电路 f 以及输入 x，混淆方生成的实际混淆电路、输入对应的标签与任意模拟器 $\mathcal{S}(1^\kappa,\Phi(f))$ 仿真生成的随机混淆电路、随机标签是不可区分的，即任何一个区分器可以区分如图 2 所示的现实装置与仿真装置的概率是可以忽略的. $\Phi(f)$ 是有关电路 f 的提示信息.

现实装置	仿真装置
$(F,e,d)\leftarrow\text{Garble}(1^\kappa,f)$ $X:=\text{Encode}(e,x)$ return (F,X)	$(F,x)\leftarrow S(1^\kappa,\Phi(f))$ return (F,X)

图 2 不经意性实验图[15]

需要强调的是计算方通过混淆电路和输入线路单标签集合无法推测出原输入真值.

(4) 可认证性. 对于任意的电路 f 以及输入 x，任何一个具有多项式时间下可计算能力的敌手 $\mathcal{A}$，图 3 所示的分布输出 TRUE 的概率是不可忽略的.

仿真实验
$(F,e,d)\leftarrow\text{Garble}(1^\kappa,f)$ $X:=\text{Encode}(e,x)$ $Y\leftarrow\mathcal{A}(F,d,X)$ return $\text{Decode}(d,Y)\notin\{f(x),\perp\}$

图 3 可认证性实验图[15]

这里强调的是，只要计算得到 $\text{Decode}(d,Y)$ 的输出结果，就可以判断计算是否正确，因为敌手是无法伪造出正确结果的.

2.1.3 安全性定义

安全多方计算协议的安全性通过定义理想-现实模拟范式的方法实现. 理想世界范式 $\text{IDEAL}_{\mathcal{F},\mathcal{S}}$ 的定义中包含理想函数、模拟器以及敌手. 其中，诚实参与方的输入由模拟器生成. $\text{IDEAL}_{\mathcal{F},\mathcal{S}}$ 的定义包括三个要求：一是理想世界的敌手仅能提供输入信息，不能参与到计算当中，预防其可能的攻击；二是理想函数 $\mathcal{F}$ 永远不会被腐化，完成所有计算后，会将输出结果传递给模拟器；三是理想世界中的计算隐喻各类安全属性. 现实世界范式 REAL_π 的定义中没有理想函数，由所有参与方共同执行协议，其中一些参与方可能被敌手控制.

参考文献[31]，安全多方计算协议在半诚实敌手模型下的安全性描述如定义 2 所示.

定义 2. 半诚实模型的安全性. 给定一个协议 π，假设安全参数为 κ，如果存在一个模拟器 $\mathcal{S}$，使对于所有的输入 $x_1,\cdots,x_{|I|}$ 以及半诚实参与方集合 $\mathcal{C}$，满足：

$$\text{REAL}_\pi\{\kappa,\mathcal{C};x_1,\cdots,x_{|I|}\}\cong \text{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa,\mathcal{C};x_1,\cdots,x_{|I|}\},$$

则称协议 π 在半诚实敌手模型下安全地实现了理想函数 $\mathcal{F}$. 其中，$\cong$ 是指两个范式的概率分布是计算不可区分的.

参考文献[19]，安全计算协议在恶意敌手模型下的安全性描述如定义 3 所示.

定义 3. 恶意模型的安全性. 给定协议 π，假设现实世界中的敌手 $\mathcal{A}$ 控制的参与方集合为 $\text{corrput}(\mathcal{A})$，理想世界中的敌手 $\mathcal{S}$ 控制的参与方集合为 $\text{corrput}(\mathcal{S})$，如果对于现实世界中任意一个具有多项式时间内可计算能力的敌手 $\mathcal{A}$，存在一个满足 $\text{corrput}(\mathcal{A})=\text{corrput}(\mathcal{S})$ 的模拟器 $\mathcal{S}$，使对于所有诚实参与方的输入 $\{x_i\mid i\notin\text{corrput}(\mathcal{A})\}$，满足：

$$\text{REAL}_{\pi,\mathcal{A}}\{\kappa;\{x_i\mid i\notin\text{corrput}(\mathcal{A})\}\}\cong \text{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{x_i\mid i\notin\text{corrput}(\mathcal{S})\}\},$$

则称协议 π 在恶意敌手模型下安全地实现了理想函数 $\mathcal{F}$.

在基于理想-现实模拟范式的安全性证明方法中，通过建立一系列混合模型实现从现实世界到理想世界的模拟. 在理想世界中，理想函数的计算是绝对安全且正确的，敌手仅能选择输入. 如果说明现实世界协议执行与理想世界协议执行的不可区分性以及正确性，就可以证明协议是安全的.

2.2 Three-Halves 混淆电路

Three-Halves 混淆电路[15]是基于 Free-XOR[12]技术实现的. 在 Three-Halves 混淆电路中，为了保证安全性，0/1 标签对之间的固定偏移量 Δ 是生成

混淆电路的混淆方私有的，这样可以防止输入线路的两个标签都被计算方获得，避免计算方伪造计算结果.

为了降低通信复杂度，Three-Halves 混淆电路采用了标签分片技术，将标签和偏移量分别切分为两个分片，然后基于分片构造混淆电路结构. 以与门 $g(\alpha,\beta,\gamma,\wedge)$ 为例，α 和 β 分别表示该与门的两条输入线路，γ 表示该与门的输出线路. 混淆方将输入线路的比特值 0 对应标签和偏移量划分为长度相等的两个子串，如式(1)所示：

$$\begin{bmatrix}\boldsymbol{L}_{\alpha,0}\\ \boldsymbol{L}_{\beta,0}\\ \Delta\end{bmatrix}=\begin{bmatrix}\boldsymbol{L}_{\alpha,0\boldsymbol{L}}\|\boldsymbol{L}_{\alpha,0\boldsymbol{R}}\\ \boldsymbol{L}_{\beta,0\boldsymbol{L}}\|\boldsymbol{L}_{\beta,0\boldsymbol{R}}\\ \Delta_{\boldsymbol{L}}\|\Delta_{\boldsymbol{R}}\end{bmatrix}\tag{1}$$

其中，“$\|$”表示串联连接两个分片. 根据 Free-XOR[12] 技术，假设各标签长度与安全参数 κ 相同，那么标签分片、偏移量分片的长度为 $\kappa/2$. 在这个基础上，可以得到各线路的比特值 1 对应标签的分片形式，如式(2)所示：

$$\begin{bmatrix}\boldsymbol{L}_{\alpha,1}\\ \boldsymbol{L}_{\beta,1}\end{bmatrix}=\begin{bmatrix}\boldsymbol{L}_{\alpha,0\boldsymbol{L}}\oplus\Delta_{\boldsymbol{L}}\|\boldsymbol{L}_{\alpha,0\boldsymbol{R}}\oplus\Delta_{\boldsymbol{R}}\\ \boldsymbol{L}_{\beta,0\boldsymbol{L}}\oplus\Delta_{\boldsymbol{L}}\|\boldsymbol{L}_{\beta,0\boldsymbol{R}}\oplus\Delta_{\boldsymbol{R}}\end{bmatrix}\tag{2}$$

假设存在哈希函数 $H(\cdot):\{0,1\}^{\kappa}\rightarrow\{0,1\}^{\kappa/2}$，那么基于标签对之间具有固定偏移量这一特性，可以推得式(3)和式(4)：

$$H(\boldsymbol{L}_{\alpha,0}\oplus\boldsymbol{L}_{\beta,0})=H(\boldsymbol{L}_{\alpha,1}\oplus\boldsymbol{L}_{\beta,1})\tag{3}$$

$$H(\boldsymbol{L}_{\alpha,0}\oplus\boldsymbol{L}_{\beta,1})=H(\boldsymbol{L}_{\alpha,1}\oplus\boldsymbol{L}_{\beta,0})\tag{4}$$

计算方得到与输入线路真值相关的标签对 $(\boldsymbol{L}_{\alpha,i},\boldsymbol{L}_{\beta,j})$ 之后，可以计算得的 3 个哈希值，分别是 $H(\boldsymbol{L}_{\alpha,i})$、$H(\boldsymbol{L}_{\beta,j})$ 和 $H(\boldsymbol{L}_{\alpha,i}\oplus\boldsymbol{L}_{\beta,j})$. 为了方便说明，将 $H(\boldsymbol{L}_{\alpha,i})$ 等标签称作单标签哈希值，将 $H(\boldsymbol{L}_{\alpha,i}\oplus\boldsymbol{L}_{\beta,j})$ 等标签称作组合标签哈希值. 如果混淆方基于各输入线路的单标签哈希值集合以及组合标签哈希值集合构造用于分析混淆表的混淆结构以及输出线路 0 值标签 $\boldsymbol{L}_{\gamma,0}$，那么计算方可以充分利用上述的 3 个哈希值分析混淆表，计算得到对应输出线路某值的一个标签. 根据文献[15]中的描述，对于与门 $g(\alpha,\beta,\gamma,\wedge)$，构造的混淆结构和输出线路 0 值标签满足式(5)：

$$\boldsymbol{V}\left(\boldsymbol{z}\left\|\begin{bmatrix}\boldsymbol{L}_{\gamma,0}\\ \boldsymbol{G}\end{bmatrix}\right.\right)\oplus\boldsymbol{M}\boldsymbol{H}=\boldsymbol{r}\left\|\left((\boldsymbol{R}\oplus[0\cdots0|\boldsymbol{t}])\cdot\begin{bmatrix}\boldsymbol{L}_{\alpha,0}\\ \boldsymbol{L}_{\beta,0}\\ \Delta\end{bmatrix}\right.\right)\tag{5}$$

式(5)的各部分具体描述如下：

(1) $\{\boldsymbol{L}_{\gamma,0},\boldsymbol{L}_{\alpha,0},\boldsymbol{L}_{\beta,0},\Delta\}$ 都是由式(1)所示的分片组成的向量形式. 其中，$\{\boldsymbol{L}_{\gamma,0},\boldsymbol{L}_{\alpha,0},\boldsymbol{L}_{\beta,0}\}$ 这三个向量形式均为 $[\boldsymbol{L}_{*,0\boldsymbol{L}},\boldsymbol{L}_{*,0\boldsymbol{R}}]^{\mathrm{T}}$，$\Delta$ 表示 $[\Delta_{\boldsymbol{L}},\Delta_{\boldsymbol{R}}]^{\mathrm{T}}$，$\boldsymbol{L}_{\gamma,0}$ 的分片 $\boldsymbol{L}_{\gamma,0\boldsymbol{L}}$ 和 $\boldsymbol{L}_{\gamma,0\boldsymbol{R}}$ 由式(5)生成；

(2) $\boldsymbol{G}=[\boldsymbol{G}_0,\boldsymbol{G}_1,\boldsymbol{G}_2]^{\mathrm{T}}$ 是混淆结构，包括 3 个分量，每个分量的长度均为 $\kappa/2$，用于分析混淆表，得到输出真值对应的输出标签 $\boldsymbol{L}_{\gamma}$；

(3) $\boldsymbol{H}$ 是基于输入线路的标签集合生成的哈希值向量，用于加密不同混淆行的输出对应标签，包括 $\boldsymbol{H}(\boldsymbol{L}_{\alpha,0})$，$\boldsymbol{H}(\boldsymbol{L}_{\beta,0})$，$\boldsymbol{H}(\boldsymbol{L}_{\alpha,1})$，$\boldsymbol{H}(\boldsymbol{L}_{\beta,1})$，$\boldsymbol{H}(\boldsymbol{L}_{\alpha,0}\oplus\boldsymbol{L}_{\beta,0})$ 和 $\boldsymbol{H}(\boldsymbol{L}_{\alpha,1}\oplus\boldsymbol{L}_{\beta,1})$ 这 6 个分量；

(4) $\boldsymbol{V}$ 和 $\boldsymbol{M}$ 是具有相同列空间的常量比特矩阵，维度分别是 8×5 和 8×6，秩都是 5；

(5) $\boldsymbol{t}$ 是标识了混淆行中输出值位置的位置信息矩阵，例如 $\boldsymbol{t}=[0,0,1,0]$，代表第三行混淆行对应的输出标签是 $\boldsymbol{L}_{\gamma,1}=\boldsymbol{L}_{\gamma,0}\oplus\Delta$，其它混淆行对应的输出标签是 $\boldsymbol{L}_{\gamma,0}$. 这里所说的混淆行是与真值表各行相对应的混淆表中的行信息. 因为式(5)等号两侧需要具有相同的空间维度，所以式(5)中的 $\boldsymbol{t}$ 实际上是维度为 8×2 的扩展形式，具体表示为

$$\boldsymbol{t}=\begin{bmatrix}0&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&0&0\end{bmatrix}^{\mathrm{T}};$$

(6) 矩阵 $\boldsymbol{R}$ 是根据 $\boldsymbol{t}$ 以及输入线路标签可能组合生成的控制比特矩阵，维度大小为 8×6. 为了不暴露输出值位置信息 $\boldsymbol{t}$ 给计算方，防止破坏隐私性，$\boldsymbol{R}$ 是与随机比特矩阵异或后的加密形式；

(7) $\boldsymbol{r}$ 是控制比特矩阵 $\boldsymbol{R}$ 的压缩表示形式，维度大小为 $8\times d$，小于矩阵 $\boldsymbol{R}$ 的维度. 发送压缩形式比特矩阵可以降低通信量.

其中，$d=2$ 适用于所有参与方均知道门类型的布尔门，$d=4$ 适用于仅混淆方知道门类型的布尔门. $\boldsymbol{z}$ 是 $\boldsymbol{r}$ 与 $lsb_{d/2}(\boldsymbol{H})$ 异或后得到的加密形式，满足式(6)：

$$\boldsymbol{V}\boldsymbol{z}\oplus\boldsymbol{M}\cdot lsb_{d/2}(\boldsymbol{H})=\boldsymbol{r}\tag{6}$$

混淆方将 $\boldsymbol{G}$ 和 $\boldsymbol{z}$ 发送给计算方，计算方基于 $\boldsymbol{G}$ 和一组输入线路单标签集合，通过式(6)计算 $\boldsymbol{r}$. 需要注意的是，$\boldsymbol{z}$ 有且仅有与输入线路单标签集合对应的那一行信息能够被计算方解密，进而计算得到 $\boldsymbol{r}$ 的局部信息.

基于 $\boldsymbol{r}$ 的局部信息可以解压缩分析得到矩阵 $\boldsymbol{R}$ 的局部视图. 通过式(7)，可以推得该组输入线路单标签集合对应的输出线路单标签集合. 其中，式(7)满足：

$$\begin{bmatrix}\boldsymbol{L}_{\gamma,0}\\ \boldsymbol{G}\end{bmatrix}\oplus\boldsymbol{M}\boldsymbol{H}=(\boldsymbol{R}\oplus[0\cdots0|\boldsymbol{t}])\cdot\begin{bmatrix}\boldsymbol{L}_{\alpha,0}\\ \boldsymbol{L}_{\beta,0}\\ \Delta\end{bmatrix}\tag{7}$$

2.3 可验证随机比特分享

可验证随机比特分享技术[32]能够实现比特信息的分享与验证. 假设 P_A 持有随机比特 a，参与方 P_A 和 P_B 为 a 生成可验证 MAC 值 $\boldsymbol{M}[a]$，满足：

$$\boldsymbol{M}[a]=\boldsymbol{K}[a]\oplus a\cdot\Delta_B,$$

其中，$\boldsymbol{K}[a]$ 是 P_B 选取的密钥，专门用于为随机比特 a 生成 MAC 值，Δ_B 是 P_B 私有的全局密钥. P_A 持有比特分享份额 $(a,\boldsymbol{M}[a])$，P_B 持有比特分享份额 $(\boldsymbol{K}[a],\Delta_B)$.

定义 4. 可验证随机比特分享函数[27]. 一个可验证随机比特分享函数 $\mathcal{F}_{\text{abit}}$ 定义为

$$\mathcal{F}_{\text{abit}}(a,(\boldsymbol{K}[a],\Delta_B))\rightarrow(\boldsymbol{M}[a]),$$

其中右箭头表示参与方交互后计算推出

假设参与方 P_A 持有随机比特 a 和随机比特 b，通过 $\mathcal{F}_{\text{abit}}$ 生成可验证的 MAC 值 $\boldsymbol{M}[a]=\boldsymbol{K}[a]\oplus b\cdot\Delta_B$ 和 $\boldsymbol{M}[b]=\boldsymbol{K}[b]\oplus b\cdot\Delta_B$，基于异或的特点，可验证随机比特分享技术满足异或同态的属性，即满足：

$$\boldsymbol{M}[a\oplus b]=\boldsymbol{K}[a]\oplus\boldsymbol{K}[b]\oplus(a\oplus b)\cdot\Delta_B.$$

通过验证函数 Open(a)，可以实现随机比特的公开以及正确性验证. 具体的，P_A 将 $(a,\boldsymbol{M}[a])$ 发送给 P_B，P_B 首先验证 $\boldsymbol{M}[a]=\boldsymbol{K}[a]\oplus a\cdot\Delta_B$ 是否成立. 如果成立，则接受比特 a，否则报错. 另外，Open(a,…) 表示验证多个随机比特的正确性，可以将所有随机比特的信息异或实现联合验证.

2.4 哈希函数

文献[15]说明了 Three-Halves 混淆电路技术中使用的哈希函数需要具有可调的循环相关健壮性(Tweakable Circular Correlation Robustness，TCCR)，具体如定义 5 所示.

定义 5. 哈希函数循环相关健壮性[15]. 给定一个哈希函数 H，为 H 定义一个相关的谕言机满足 $O_\Delta(X,\tau,b)=H(X\oplus\Delta,\tau)\oplus b\Delta$. 假设一个区分器不使用重复的一对 (X,τ) 调用谕言机 O_Δ，如果此时谕言机 O_Δ 的返回结果与调用随机谕言机的返回结果是不可区分的，则哈希函数 H 具有可调的循环相关健壮性(Tweakable Circular Correlation Robustness，TCCR).

为了适应 Three-Halves 混淆与分析方案的正确性与安全性需求，文献[15]对哈希函数的 TCCR 属性进行了一些重要修改，具体包括以下两点：

(1) 哈希函数的输入与输出长度不一致. 在 Three-Halves 混淆电路中采用了标签分片技术，每个标签分片的长度是标签的一半，基于各标签分片生成和分析混淆表. 因此，全局常量 Δ 被划分为 Δ_L 和 Δ_R，原来与哈希函数相关的谕言机需要修改为 $O_\Delta^*(X,\tau,b)=H(X\oplus\Delta,\tau)\oplus L(\Delta)$，其中 $L(\Delta)$ 是全局常量分片形式的线性组合函数，表示为 $L(\Delta_L\|\Delta_R)=a\Delta_L\oplus b\Delta_R$.

(2) 谕言机 O_Δ^* 与谕言机 O_Δ 相比，增强了随机性，因此不再需要使用某个特定满足强 TCCR 属性的理想哈希函数，哈希函数 H 可以来自某哈希族，在这种情况下完全可以满足 Three-Halves 混淆与分析方案的正确性与安全性需求.

基于上述两点修改，文献[15]提出了随机可调的循环相关健壮性(Randomized Tweakable Circular Correlation Robustness，RTCCR). 与强 TCCR 属性相比，哈希函数更容易满足 RTCCR 属性，足以支持混淆电路技术的安全性需求，可选择的哈希函数族规模更大. 具体证明过程可以参考文献[15]，这里不作过多赘述.

2.5 关键符号定义

本文后续章节中使用到的关键符号及其定义如表 1 所示.

表 1 关键符号定义

符号	定义说明
C	布尔电路
$\|C\|$	布尔电路 C 的大小
I	电路的输入线路集合
O	电路的输出线路集合
κ	安全参数
$P_{A/B}$	参与方的标识，分别用于标识两方
$\boldsymbol{G}_*$	混淆电路 $*$，用于电路混淆与分析
$H(\cdot)$	具有循环相关健壮性属性的哈希函数
$M(\cdot)$	可验证 MAC 值，用于验证比特的正确性
$K(\cdot)$	局部密钥，用于生成 MAC 值
Δ	全局密钥，用于生成 MAC 值
λ_*	布尔线路 $*$ 的掩码值
$:=$	赋值操作
$\leftarrow$	推导操作
$\xleftarrow{R}$	随机选取操作
$\cong$	概率分布在多项式时间计算能力下是不可区分的

本文研究的布尔电路由多个异或门和与门组成. 异或门可以在“零开销”的情况下完成计算，因此，本文布尔电路 C 的大小 $|C|$ 指布尔电路中包含的与门的数量.

3 基于 Three-Halves 的可验证混淆电路分享方案

3.1 方案设计

3.1.1 方案概述

Rosulek 和 Roy[15] 提出的 Three-Halves 混淆

电路实现了布尔电路中异或门的计算“零开销”，与此同时，与门的计算所需通信量低于之前最优的 Half-Gates 方案[14].

可验证混淆电路分享方案. 在安全两方计算协议中，一般由一个参与方担任混淆方的角色来生成完整的混淆电路. 如果混淆方是恶意的，那么他可能构造错误的混淆表，导致计算方无法获得正确的输出而报错终止，而且混淆方能够通过该报错信息分析得知计算方的输入信息. 针对这一问题，Katz、Ranellucci 和 Rosulek[28] 提出了基于可验证随机比特分享[32]的可验证混淆电路分享方案. 首先，两个参与方共同生成各线路的掩码值，即两个参与方为线路各选择一个随机比特，两个随机比特的异或值作为该线路的掩码值，用于加密真值；然后，基于可验证随机比特分享，两个参与方分别生成混淆电路分享份额，在这种情况下，任何参与方都不知道完整的混淆电路，也不知道每一个混淆表中各混淆行对应的输出标签是哪一个. 即使其中一方作恶，也只能得到错误混淆电路的分析结果，而无法得知混淆表中错误混淆行的对应真值是什么.

针对混淆电路技术通信复杂度高的问题，基于 Three-Halves 混淆电路[15] 和可验证随机比特分享[32]，本节提出了一个恶意敌手模型下安全且通信开销更低的新可验证混淆电路分享方案.

3.1.2　具体构造

以一个与门 $g(\alpha,\beta,\gamma,\wedge)$ 为例. 假设 P_A 持有全局密钥 Δ_A，P_B 持有全局密钥 Δ_B. $\boldsymbol{M}[a]$ 表示随机比特值 a 的可验证 MAC 值，由拥有 a 的参与方持有. $\boldsymbol{K}[a]$ 是用于生成 $\boldsymbol{M}[a]$ 的局部密钥，由另一参与方持有. P_A 选取随机比特集合 $\{a_\alpha,a_\beta,a_\gamma\}$，$P_B$ 选取局部密钥集合 $\{\boldsymbol{K}[a_\alpha],\boldsymbol{K}[a_\beta],\boldsymbol{K}[a_\gamma]\}$，每个随机比特/局部密钥分别对应下标所标识的线路. 经过可验证随机比特分享函数 $\mathcal{F}_{\text{abit}}$ 计算后，参与方 P_A 拥有集合 $\{(a_\alpha,\boldsymbol{M}[a_\alpha]),(a_\beta,\boldsymbol{M}[a_\beta]),(a_\gamma,\boldsymbol{M}[a_\gamma])\}$. 同理，$P_A$ 选取随机字符串集合 $\{\boldsymbol{K}[b_\alpha],\boldsymbol{K}[b_\beta],\boldsymbol{K}[b_\gamma]\}$，$P_B$ 选取随机比特集合 $\{b_\alpha,b_\beta,b_\gamma\}$，经过 $\mathcal{F}_{\text{abit}}$ 计算之后，参与方 P_B 持有集合 $\{(b_\alpha,\boldsymbol{M}[b_\alpha]),(b_\beta,\boldsymbol{M}[b_\beta]),(b_\gamma,\boldsymbol{M}[b_\gamma])\}$.

基于 Beaver[33] 提出的乘法三元组方案计算得到输入线路掩码乘积值 $\lambda_\alpha\lambda_\beta$ 的分享份额 (a_γ^*,b_γ^*)，乘积值 $\lambda_\alpha\lambda_\beta$ 定义为与门输出乘积掩码值 λ_γ^*. 其中，份额 a_γ^* 由 P_A 持有，份额 b_γ^* 由 P_B 持有. 同理，经过 $\mathcal{F}_{\text{abit}}$ 计算之后，P_A 将拥有集合 $\{\boldsymbol{M}[a_\gamma^*],\boldsymbol{K}[b_\gamma^*]\}$，$P_B$ 将拥有集合 $\{\boldsymbol{K}[a_\gamma^*],\boldsymbol{M}[b_\gamma^*]\}$. 因此，与门 g 各线路的掩码值如式(8)定义：

$$\begin{bmatrix}\lambda_\alpha\\ \lambda_\beta\\ \lambda_\gamma\\ \lambda_\gamma^*\end{bmatrix}:=\begin{bmatrix}a_\alpha\oplus b_\alpha\\ a_\beta\oplus b_\beta\\ a_\gamma\oplus b_\gamma\\ a_\gamma^*\oplus b_\gamma^*\end{bmatrix}\tag{8}$$

由文献[27]可知，如式(9)所示，输出掩饰值可以使用线路掩码表示：

$$\begin{bmatrix}\hat{z}_{00}\\ \hat{z}_{01}\\ \hat{z}_{10}\\ \hat{z}_{11}\end{bmatrix}:=\begin{bmatrix}(\lambda_\alpha\wedge\lambda_\beta)\oplus\lambda_\gamma\\ (\lambda_\alpha\wedge\lambda_\beta)\oplus\lambda_\gamma\oplus\lambda_\alpha\\ (\lambda_\alpha\wedge\lambda_\beta)\oplus\lambda_\gamma\oplus\lambda_\beta\\ (\lambda_\alpha\wedge\lambda_\beta)\oplus\lambda_\gamma\oplus\lambda_\alpha\oplus\lambda_\beta\oplus 1\end{bmatrix}\tag{9}$$

假设 P_A 是混淆方，那么混淆表输出掩饰值的标签形式如式(10)表示：

$$\begin{bmatrix}\boldsymbol{L}_{\gamma,\hat{z}_{00}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{01}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{10}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{11}}\end{bmatrix}:=\begin{bmatrix}\boldsymbol{L}_{\gamma,0}\oplus\hat{z}_{00}\cdot\Delta_A\\ \boldsymbol{L}_{\gamma,0}\oplus\hat{z}_{01}\cdot\Delta_A\\ \boldsymbol{L}_{\gamma,0}\oplus\hat{z}_{10}\cdot\Delta_A\\ \boldsymbol{L}_{\gamma,0}\oplus\hat{z}_{11}\cdot\Delta_A\end{bmatrix}\tag{10}$$

输出值位置信息矩阵 $\boldsymbol{t}$ 标识混淆行中的输出值位置，因此可以使用式(9)作为式(5)中的输出值位置信息矩阵 $\boldsymbol{t}$，即

$$\boldsymbol{t}:=\begin{bmatrix}\hat{z}_{00}\\ \hat{z}_{01}\\ \hat{z}_{10}\\ \hat{z}_{11}\end{bmatrix}\tag{11}$$

根据式(8)、(9)，P_A 持有矩阵 $\boldsymbol{t}$ 的一个分享份额 $\boldsymbol{t_A}$ 表示如式(12)所示：

$$\boldsymbol{t_A}:=\begin{bmatrix}a_\gamma^*\oplus a_\gamma\\ a_\gamma^*\oplus a_\gamma\oplus a_\alpha\\ a_\gamma^*\oplus a_\gamma\oplus a_\beta\\ a_\gamma^*\oplus a_\gamma\oplus a_\alpha\oplus a_\beta\oplus 1\end{bmatrix}\tag{12}$$

P_B 持有矩阵 $\boldsymbol{t}$ 的另一个分享份额 $\boldsymbol{t}_B$ 表示如式(13)所示：

$$\boldsymbol{t}_B:=\begin{bmatrix}b_\gamma^*\oplus b_\gamma\\ b_\gamma^*\oplus b_\gamma\oplus b_\alpha\\ b_\gamma^*\oplus b_\gamma\oplus b_\beta\\ b_\gamma^*\oplus b_\gamma\oplus b_\alpha\oplus b_\beta\end{bmatrix}\tag{13}$$

这里需要注意的是，式(11)～(13)在实际计算时是维度为 8×2 的扩展形式，即公式中每一个分量是 $\begin{bmatrix}*&0\\0&*\end{bmatrix}$ 的列扩展形式. 根据参与方所持有的可验证随机比特分享份额信息，混淆行输出掩饰值的标签分享份额可以分别表示为式(14)和式(15)：

$$\begin{bmatrix}\boldsymbol{L}_{\gamma,\hat{z}_{00}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{01}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{10}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{11}}\end{bmatrix}_A = \boldsymbol{L}_{\gamma,0}\oplus \boldsymbol{t}_A\cdot\Delta_A\oplus \begin{bmatrix}\boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma]\\ \boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma]\oplus\boldsymbol{K}[b_\alpha]\\ \boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma]\oplus\boldsymbol{K}[b_\beta]\\ \boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma]\oplus\boldsymbol{K}[b_\alpha]\oplus\boldsymbol{K}[b_\beta]\end{bmatrix} \tag{14}$$

$$\begin{bmatrix}\boldsymbol{L}_{\gamma,\hat{z}_{00}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{01}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{10}}\\ \boldsymbol{L}_{\gamma,\hat{z}_{11}}\end{bmatrix}_B = \begin{bmatrix}\boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma]\\ \boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma]\oplus\boldsymbol{M}[b_\alpha]\\ \boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma]\oplus\boldsymbol{M}[b_\beta]\\ \boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma]\oplus\boldsymbol{M}[b_\alpha]\oplus\boldsymbol{M}[b_\beta]\end{bmatrix} \tag{15}$$

根据式(14)和式(15),式(5)的混淆结构可以划分为两个分享份额,即 $\boldsymbol{G}=[\boldsymbol{G}]_A\oplus[\boldsymbol{G}]_A$. 其中,$P_A$ 生成的混淆分享份额 $[\boldsymbol{G}]_A$ 满足式(16):

$$\boldsymbol{V}\left(\boldsymbol{z}\left\|\begin{bmatrix}\boldsymbol{L}_{\gamma,0}\\ [\boldsymbol{G}]_A\end{bmatrix}\right.\right)\oplus\boldsymbol{M}\boldsymbol{H}= \boldsymbol{r}\left\|\left((\boldsymbol{R}\oplus[0\cdots0\,|\,\boldsymbol{t}_A])\cdot\begin{bmatrix}\boldsymbol{L}_{\alpha,0}\\ \boldsymbol{L}_{\beta,0}\\ \Delta_A\end{bmatrix}\right)\oplus\boldsymbol{K}\right. \tag{16}$$

其中,$\boldsymbol{K}:=\begin{bmatrix}(\boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma])\\ (\boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma]\oplus\boldsymbol{K}[b_\alpha])\\ (\boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma]\oplus\boldsymbol{K}[b_\beta])\\ (\boldsymbol{K}[b_\gamma^*]\oplus\boldsymbol{K}[b_\gamma]\oplus\boldsymbol{K}[b_\alpha]\oplus\boldsymbol{K}[b_\beta])\end{bmatrix}$.

P_B 生成的混淆分享份额 $[\boldsymbol{G}]_B$ 满足式(17):

$$[\boldsymbol{G}]_B:=\begin{bmatrix}(\boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma])\\ (\boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma]\oplus\boldsymbol{M}[b_\alpha])\\ (\boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma]\oplus\boldsymbol{M}[b_\beta])\\ (\boldsymbol{M}[b_\gamma^*]\oplus\boldsymbol{M}[b_\gamma]\oplus\boldsymbol{M}[b_\alpha]\oplus\boldsymbol{M}[b_\beta])\end{bmatrix} \tag{17}$$

需要注意的是,因为 $\boldsymbol{V}$ 和 $\boldsymbol{M}$ 的行维度都是 8,而且基于标签分片计算,所以式(16)中的 $\boldsymbol{K}$ 和式(17)各分量都是 $\begin{bmatrix}*_{\boldsymbol{L}} & 0\\ 0 & *_{\boldsymbol{R}}\end{bmatrix}$ 的分片扩展形式.

综上,基于式(16)和式(17)分别获得一份混淆电路份额. 基于可验证随机比特分享技术[32],生成各线路掩码值所使用的随机比特是可验证的,因此两个混淆分享份额也是可验证的.

参照文献[15],可验证混淆电路分享方案由 4 个算法组成,分别是混淆算法、标签映射算法、电路分析算法和真值映射算法,可实现布尔电路混淆与混淆电路分析这两个核心任务. 另外,为了获得和分析混淆结构中的控制比特矩阵 $\boldsymbol{R}$,本文使用了文献[15]中的控制比特矩阵生成算法 SampleR($\boldsymbol{t}$,leak)以及控制比特矩阵解析算法 DecodeR($\boldsymbol{r}$,leak,i,j)实现,算法过程完全一致,这里不再赘述.

混淆算法 $(F,e,d)\leftarrow\text{Garble}(1^\kappa,f)$. F 是混淆电路,e 是混淆算法,d 是分析算法. 具体过程如下:

1. 分析电路 f 得到布尔电路结构信息可以表示为 $\Phi(f):=(\text{inputs},\text{outputs},\text{in},\text{leak},\text{eval})$,由 P_A 和 P_B 共享,各部分信息分别表示:

① inputs 是电路输入线路的数量;

② outputs 是电路输出线路的数量;

③ in 是布尔电路中所有线路以及布尔门的编号集合. 编号工作是按照电路的拓扑结构顺序完成. 对于编号为 g 的布尔门,两条输入线路编号分别是 $\text{in}_1(g)$ 和 $\text{in}_2(g)$,满足 $\text{in}_i(g)<g$;

④ leak 是布尔门类型信息,$\text{leak}(g)$ 表示编号为 g 的布尔门的类型;

⑤ eval 是布尔门对应的运算.

2. 从满足 RTCCR 属性的哈希函数族 $\mathcal{H}$ 里选取一个哈希函数:$H\leftarrow\mathcal{H}$.

3. 混淆方 P_A 随机选取全局密钥:

$$\Delta_A\xleftarrow{R}\begin{bmatrix}1\|GF(2^{\kappa/2-1})\\ GF(2^{\kappa/2})\end{bmatrix}.$$

4. P_A 为各输入线路生成 0 标签、掩码随机比特份额以及掩码随机比特份额密钥,具体描述为:

① 对于所有输入线路 $k\in[1,\text{inputs}]$,随机选取 0 标签:

$$\boldsymbol{L}_{k,0}\xleftarrow{R}\begin{bmatrix}0\|GF(2^{\kappa/2-1})\\ GF(2^{\kappa/2})\end{bmatrix};$$

② 随机选取 $a_k\leftarrow\{0,1\}$ 和随机比特 b_k 的密钥:

$$\boldsymbol{K}[b_k]\xleftarrow{R}\begin{bmatrix}GF(2^{\kappa/2})\\ GF(2^{\kappa/2})\end{bmatrix}.$$

5. P_B 为各输入线路生成掩码随机比特份额以及掩码随机比特份额密钥.

对于所有输入线路 $k\in[1,\text{inputs}]$,随机选取比特 $b_k\xleftarrow{R}\{0,1\}$,同时,选取随机比特 a_k 的密钥:

$$\boldsymbol{K}[a_k]\xleftarrow{R}\begin{bmatrix}GF(2^{\kappa/2})\\ GF(2^{\kappa/2})\end{bmatrix}.$$

6. 根据电路拓扑结构,P_A 生成混淆电路份额 $[\boldsymbol{G}]_A$,具体过程描述如下:

① 对于所有布尔门 $k\in[\text{inputs},|f|]$,输入线路的 0 标签分别是 $\boldsymbol{L}_{\alpha,0},\boldsymbol{L}_{\beta,0}:=\boldsymbol{L}_{\text{in}_1(k),0},\boldsymbol{L}_{\text{in}_2(k),0}$,掩码随机比特份额分别是 $a_\alpha,a_\beta:=a_{\text{in}_1(k)},a_{\text{in}_2(k)}$;

② 如果布尔门是异或门,那么输出线路的 0 标签是 $\boldsymbol{L}_{k,0}:=\boldsymbol{L}_{\alpha,0}\oplus\boldsymbol{L}_{\beta,0}$,掩码随机比特份额是 $a_k:=a_\alpha\oplus a_\beta$;

③ 布尔门不是异或门,则处理步骤如下:

步骤 1. 随机选取输出线路的掩码比特份额:$a_k\xleftarrow{R}\{0,1\}$;

步骤 2. 通过乘法三元组算法获得输出线路的乘积掩码比特份额:$a_k^*\leftarrow\text{Triple}$;

步骤 3. 随机选取乘积掩码随机比特 b_k^* 的密钥:

$$\boldsymbol{K}[b_k^*] \xleftarrow{R} \begin{bmatrix} GF(2^{\kappa/2}) \\ GF(2^{\kappa/2}) \end{bmatrix};$$

步骤 4. 计算输出线路位置比特信息矩阵份额：

$$\boldsymbol{t}_A := \begin{bmatrix} a_k^* \oplus a_k \\ a_k^* \oplus a_k \oplus a_\alpha \\ a_k^* \oplus a_k \oplus a_\beta \\ a_k^* \oplus a_k \oplus a_\alpha \oplus a_\beta \oplus 1 \end{bmatrix},$$

并将 $\boldsymbol{t}_A$ 扩展为维度为 8×2 的形式.

步骤 5. 通过文献[15]中的控制比特矩阵生成算法 SampleR($\boldsymbol{t}$,leak),计算控制矩阵以及其压缩表示形式：$\boldsymbol{R},\boldsymbol{r} \leftarrow$ SampleR($\boldsymbol{t}_A$,leak(k))；

步骤 6. 计算输出值的密钥矩阵：

$$\boldsymbol{K} := \begin{bmatrix} (\boldsymbol{K}[b_\gamma^*] \oplus \boldsymbol{K}[b_\gamma]) \\ (\boldsymbol{K}[b_\gamma^*] \oplus \boldsymbol{K}[b_\gamma] \oplus \boldsymbol{K}[b_\alpha]) \\ (\boldsymbol{K}[b_\gamma^*] \oplus \boldsymbol{K}[b_\gamma] \oplus \boldsymbol{K}[b_\beta]) \\ (\boldsymbol{K}[b_\gamma^*] \oplus \boldsymbol{K}[b_\gamma] \oplus \boldsymbol{K}[b_\alpha] \oplus \boldsymbol{K}[b_\beta]) \end{bmatrix},$$

并将 $\boldsymbol{K}$ 表示为维度为 8×2 的分片扩展形式.

步骤 7. 生成份额 $[\boldsymbol{G}]_A$：

$$\left(\boldsymbol{z} \middle\| \begin{bmatrix} \boldsymbol{L}_{k,0} \\ [\boldsymbol{G}]_A \end{bmatrix}\right) =$$

$$\boldsymbol{V}^{-1}\left(\boldsymbol{r} \middle\| \left((\boldsymbol{R} \oplus [0\cdots 0 \mid \boldsymbol{t}_A]) \cdot \begin{bmatrix} \boldsymbol{L}_{\alpha,0} \\ \boldsymbol{L}_{\beta,0} \\ \Delta_A \end{bmatrix}\right) \oplus \boldsymbol{K} \oplus \boldsymbol{MH}\right) \tag{18}$$

7. P_B 生成混淆电路份额 $[\boldsymbol{G}]_B$，过程描述如下：

① 对于所有布尔门 $k\in$[inputs,$|f|$]，门输入线路的掩码随机比特份额分别表示为 $b_\alpha := b_{\mathrm{in}_1(k)}$，$b_\beta := b_{\mathrm{in}_2(k)}$，掩码随机比特份额的 MAC 值分别由函数 $\mathcal{F}_{\mathrm{abit}}$ 得到：$\boldsymbol{M}[b_\alpha],\boldsymbol{M}[b_\beta] \leftarrow \mathcal{F}_{\mathrm{abit}}$；

② 如果布尔门是异或门，那么输出线路的掩码随机比特份额是 $b_k := b_\alpha \oplus b_\beta$；

③ 如果布尔门不是异或门，则处理步骤如下：

步骤 1. 随机选取输出线路掩码的随机比特份额：$b_k \xleftarrow{R} \{0,1\}$；

步骤 2. 通过 Beaver 乘法三元组算法[33]获得输出线路的乘积掩码比特份额：$b_k^* \leftarrow$Triple；

步骤 3. 获得乘积掩码随机比特 b_k^* 的 MAC 值：$\boldsymbol{M}[b_k^*] \leftarrow \mathcal{F}_{\mathrm{abit}}$；

步骤 4. 生成混淆电路份额 $[\boldsymbol{G}]_B$：

$$[\boldsymbol{G}]_B := \begin{bmatrix} (\boldsymbol{M}[b_\gamma^*] \oplus \boldsymbol{M}[b_\gamma]) \\ (\boldsymbol{M}[b_\gamma^*] \oplus \boldsymbol{M}[b_\gamma] \oplus \boldsymbol{M}[b_\alpha]) \\ (\boldsymbol{M}[b_\gamma^*] \oplus \boldsymbol{K}[b_\gamma] \oplus \boldsymbol{M}[b_\beta]) \\ (\boldsymbol{M}[b_\gamma^*] \oplus \boldsymbol{K}[b_\gamma] \oplus \boldsymbol{M}[b_\alpha] \oplus \boldsymbol{M}[b_\beta]) \end{bmatrix} \tag{19}$$

式(18)的分量均是 $\begin{bmatrix} *_{\boldsymbol{L}} & 0 \\ 0 & *_{\boldsymbol{R}} \end{bmatrix}$ 形式.

8. 根据电路输出线路标签生成标签分析算法 D. 具体的，对于所有输出线路 $k\in$outputs 和比特值 $j\in\{0,1\}$，电路输出线路对应标签值的加密方式，即标签分析算法为 $D_k^j := H'(\boldsymbol{L}_{k,0} \oplus j\Delta_A, k)$.

经过第 1～8 步之后，得到混淆结构三元组 $\{F=(\Phi(f),\boldsymbol{H},([\boldsymbol{G}]_A,[\boldsymbol{G}]_B),\boldsymbol{z}),e=(\Delta_A,L,(a,b)),d=(\Phi(f),D)\}$.

标签映射算法 $X \leftarrow$Encode(e,x). X 是输入线路值对应标签集合.

通过 Open($\cdot,\cdots$)验证输入线路随机比特值的正确性之后，参与方恢复自己的输入线路的掩码值，将掩码值与真值异或得到输入线路的掩饰值，并生成输入标签集合. 具体的，首先，对于所有输入线路 $k\in$[1,inputs]，参与方验证并恢复输入线路掩码 $\lambda_k \leftarrow$Open(a_k,b_k)；然后，参与方 P_A 生成输入掩饰值的标签 $X_k^j := \boldsymbol{L}_{k,0} \oplus (\lambda_k \oplus x_k)\Delta_A$. 其中，$P_B$ 计算得到自己的输入线路的掩饰值之后发送 P_A，由 P_A 生成相应标签并发送给 P_B.

电路分析算法 $Y \leftarrow$Eval(F,X). Y 是输出线路值对应标签集合.

P_B 本地恢复完整的混淆电路，根据电路的拓扑结构信息，利用输入线路掩饰值对应标签集合分析混淆电路，具体过程描述如下：

1. 解析电路相关信息：

(inputs,outputs,in,leak)$\leftarrow\Phi$；

2. 对于所有布尔门 $k\in$[inputs,$|f|$]，输入线路的标签分别是 $\boldsymbol{L}_\alpha := \boldsymbol{L}_{\mathrm{in}_1(k)}$，$\boldsymbol{L}_\beta := \boldsymbol{L}_{\mathrm{in}_2(k)}$，掩饰值分别是 $i := lsb(\boldsymbol{L}_\alpha)$，$j := lsb(\boldsymbol{L}_\beta)$，$lsb(*)$表示取 $*$ 的最低比特位；

3. 如果布尔门是异或门，那么输出线路的标签是 $\boldsymbol{Y}_k := \boldsymbol{L}_{\alpha,0} \oplus \boldsymbol{L}_{\beta,0}$，掩码随机比特份额是 $a_k := a_\alpha \oplus a_\beta$；

4. 如果布尔门不是异或门，则处理步骤如下：

① 恢复完整的混淆电路：$\boldsymbol{G}_k := [\boldsymbol{G}]_A \oplus [\boldsymbol{G}]_B$；

② 计算分析输出线路标签 $\boldsymbol{X}_{ij}$；

$$(\boldsymbol{r} \parallel \boldsymbol{X}_{ij}) := \boldsymbol{V}_{ij}\left(\boldsymbol{z}_k \middle\| \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{G}_k \end{bmatrix}\right) \oplus \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} H(\boldsymbol{L}_\alpha, 3k-3) \\ H(\boldsymbol{L}_\beta, 3k-2) \\ H(\boldsymbol{L}_{\alpha,0} \oplus \boldsymbol{L}_{\beta,0}, 3k-1) \end{bmatrix} \tag{20}$$

③ 通过文献[15]中控制比特矩阵解析算法 DecodeR($\boldsymbol{r}$,leak,i,j)获取控制比特矩阵的局部视图：$\boldsymbol{R}_{ij}$:= DecodeR($\boldsymbol{r}$,leak,i,j)；

④ 计算输出线路标签：$\boldsymbol{Y}_k := \boldsymbol{X}_{ij} \oplus \boldsymbol{R}_{ij} \begin{bmatrix} \boldsymbol{L}_\alpha \\ \boldsymbol{L}_\beta \end{bmatrix}$.

真值映射算法 $y \leftarrow$Decode(d,Y).

使用标签分析算法 d 获得输出线路对应掩饰值，然后再通过 Open($\cdot,\cdots$)验证输出线路随机比特值的正确性之后，恢复完整掩码后计算输出值 y，具体过程描述如下：

1. 参与方分别解析布尔电路的相关信息：(inputs,outputs,in,leak)$\leftarrow\Phi$；

2. 初始化输出比特集合 $y:=$empty list;

3. 对于所有的输出线路 $k\in[\text{outputs}]$,如果存在 j,使 $D_k^j=H'(X_k,k)$成立,则添加 j 到 y 集合;

4. 参与方公开验证并恢复电路输出线路掩码 $\lambda_k\leftarrow \text{Open}(a_k,b_k)$;

5. 恢复真值 $y:=y\oplus\lambda_k$.

3.2 安全性分析

定理 1. 恶意模型的安全性. 如果 Three-Halves 混淆电路和可验证随机比特分享是安全的,那么实现的可验证混淆电路分享方案在(Triple,$\mathcal{F}_{\text{abit}}$)-混合模型下安全地计算电路 f,并可以抵抗概率多项式时间恶意敌手$\mathcal{A}$的攻击.

证明. 假设参与方 P_A 担任混淆方,P_B 担任计算方,则依次应用可验证混淆电路分享方案的混淆算法 $\text{Garble}(1^\kappa,f)$,标签映射算法 $\text{Encode}(e,x)$,电路分析算法 $\text{Eval}(F,X)$以及真值映射算法 $\text{Decode}(d,Y)$可以完成计算并获得结果.

文献[15]中证明了 Three-Halves 混淆电路技术具有正确性、隐私性、不经意性以及可认证性. 因此,如果本节提出的可验证混淆电路分享方案是安全的,则根据 Three-Halves 混淆电路技术推得该方案具有正确性、隐私性、不经意性以及可认证性. 另外,假设可验证随机比特技术的实现是通过$\mathcal{F}_{\text{abit}}$实现,安全性证明过程可参照文献[27],这里不再详述.

正确性. 根据算法 $\text{Garble}(1^\kappa,f)$的具体描述,可以推得每个布尔门混淆行对应输出线路的标签集合 $\boldsymbol{X}$ 如式(21)所示:

$$\boldsymbol{X}=\boldsymbol{L}_{k,0}\oplus(\boldsymbol{R}\oplus[0\cdots0\,|\,\boldsymbol{t}])\begin{bmatrix}\boldsymbol{L}_{\alpha,0}\\ \boldsymbol{L}_{\beta,0}\\ \Delta_A\end{bmatrix}\tag{21}$$

其中,根据式(8)到式(17),可以得知输出值位置信息矩阵 $\boldsymbol{t}=\boldsymbol{t}_A\oplus\boldsymbol{t}_B$,因此计算方恢复得到完整混淆电路结构 $\boldsymbol{G}$,就可以推得式(21).

根据算法 $\text{Eval}(F,X)$的具体描述,计算方通过算法 $\text{DecodeR}(\boldsymbol{r},\text{leak},i,j)$获取控制比特矩阵 $\boldsymbol{R}$ 的局部视图,即根据 i 和 j 定位到其中一行 $R_{i,j}$. 根据控制比特矩阵 $\boldsymbol{R}$ 的性质,可以推得 $\boldsymbol{R}\begin{bmatrix}\boldsymbol{L}_{\alpha,0}\\ \boldsymbol{L}_{\beta,0}\\ \Delta_A\end{bmatrix}$ 的其中一行为 $R_{i,j}\begin{bmatrix}\boldsymbol{L}_\alpha\\ \boldsymbol{L}_\beta\end{bmatrix}$. 因此,计算方得到各布尔门的输出线路标签如式(22)所示:

$$\begin{aligned}Y_k&=\boldsymbol{L}_{k,0}\oplus\boldsymbol{R}\begin{bmatrix}\boldsymbol{L}_\alpha\\ \boldsymbol{L}_\beta\end{bmatrix}\\&=\boldsymbol{L}_{k,0}\oplus t_{i,j}\Delta_A\\&=\boldsymbol{L}_{k,0}\oplus(\text{eval}(\lambda_i\oplus i,\lambda_j\oplus j)\oplus\lambda_k)\Delta_A\end{aligned}\tag{22}$$

因为 Δ_A 的末位为 1,i 和 j 分别是输入掩饰值,所以取 $lsb(Y_k)$可以作为输出掩饰值形式,通过算法 $\text{Decode}(d,Y)$可以得到 $D_k^{y_k}=H'(X_k,k)$. 假设 $D_k^{y_k}\neq D_k^{1-y_k}$ 不成立的概率是可以忽略的,那么通过 $\text{Open}(\cdot,\cdots)$验证并恢复输出线路掩码值 λ_k 之后,就可以得到正确的输出真值 $y_k:=lsb(Y_k)\oplus\lambda_k$.

对于可验证混淆电路分享方案 G,假设现实世界中的恶意敌手$\mathcal{A}$控制的参与方集合为 $\text{corrput}(\mathcal{A})$,理想世界中的敌手$\mathcal{S}$控制的参与方集合为 $\text{corrput}(\mathcal{S})$,理想函数$\mathcal{F}$是电路 f 的理想执行. 如果对于敌手$\mathcal{A}$,在理想世界存在一个满足 $\text{corrput}(\mathcal{A})=\text{corrput}(\mathcal{S})$的模拟器$\mathcal{S}$,使诚实参与方的输入$\{x_i\,|\,i\notin\text{corrput}(\mathcal{A})\}$,满足:

$$\begin{aligned}&\text{REAL}_{G,\mathcal{A}}\{\kappa;\{x_i\,|\,i\notin\text{corrput}(\mathcal{A})\}\}\cong\\&\qquad\text{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{x_i\,|\,i\notin\text{corrput}(\mathcal{S})\}\},\end{aligned}$$

则称可验证混淆电路分享方案 G 在恶意敌手模型下安全地实现了理想函数$\mathcal{F}$.

文献[15]中证明了 Three-Halves 混淆电路技术在半诚实敌手模型下是安全的,具有正确性、隐私性、不经意性以及可认证性. 因此,可以根据 Three-Halves 混淆电路技术推得本节提出的可验证混淆电路分享方案的安全性. 这里所说的安全性是半诚实敌手下安全的,也就是说假设敌手会遵循协议完成混淆电路生成和分析工作. 另外,假设可验证随机比特技术的实现是通过$\mathcal{F}_{\text{abit}}$实现,安全性证明过程可参照文献[27],这里不再详述.

接下来分别考虑恶意 P_A 和恶意 P_B 这两种情况,通过理想-现实模拟的方法证明定理 1.

情况 1(P_A恶意). 假设概率多项式时间敌手$\mathcal{A}$腐化了参与方 P_A,可以构造一个模拟器$\mathcal{S}$,既内部执行一个敌手$\mathcal{A}$装置来模拟 P_A,也模拟诚实参与方 P_B 的行为,模拟器$\mathcal{S}$通过访问一个理想函数$\mathcal{F}$来分析 f.

根据可验证混淆电路方案 G 的四个算法,模拟器$\mathcal{S}$的模拟行为描述如下.

混淆算法. $\mathcal{S}$模拟生成并记录 P_A 和 P_B 持有的所有信息,包括布尔电路 f 的结构信息,具有 RTC-CR 属性的哈希函数 H,全局密钥 Δ_A,各输入线路的 0 标签集合、各掩码随机比特份额以及掩码随机比特份额密钥,通过理想乘法三元组算法 Triple 获得的与门输出线路的掩码比特分享份额,以及通过$\mathcal{F}_{\text{abit}}$获得的验证 MAC 值集合;

然后,$\mathcal{S}$接受敌手$\mathcal{A}$构造的混淆电路份额$[\boldsymbol{G}]_A$,$\mathcal{S}$模拟 P_B 生成混淆电路份额$[\boldsymbol{G}]_B$.

标签映射算法. $\mathcal{S}$接受敌手$\mathcal{A}$发送来的所有信息，使用输入 $y:=0$ 来模拟 P_B参与输入线路标签映射过程，生成并发送 P_B输入线路的掩饰值给敌手$\mathcal{A}$. 另外，$\mathcal{S}$接受敌手$\mathcal{A}$发送的 P_A输入线路掩饰值，并通过混淆算法获得的线路掩码值恢复得到 P_A输入线路真值，然后发送给理想函数$\mathcal{F}$得到计算结果 $f(x,y)$.

电路分析算法. $\mathcal{S}$接受敌手$\mathcal{A}$发送的所有信息，使用相同输入 $y:=0$ 来模拟 P_B参与混淆电路分析过程，恢复完整混淆电路结构并计算输出线路值的标签集合.

真值映射算法. $\mathcal{S}$模拟 P_B使用标签分析算法 d 获得输出线路对应掩饰值，如果获取掩饰值成功，说明电路分析算法得到的输出线路标签集合是有效标签，$\mathcal{S}$计算 $f(x,0)$，与对应线路掩码异或获得掩饰值形式，作为输出掩饰值发送给敌手$\mathcal{A}$；否则，说明电路分析过程错误，发送报错终止信息给敌手$\mathcal{A}$.

根据模拟证明方法原理，通过建立一系列混合模型来证明本方案在现实世界与理想世界执行的不可区分性. 在混合模型中，第一个模型是现实执行情形，最后一个模型是理想世界的执行情形.

混合模型 1. $\mathcal{S}$模拟生成并记录 P_A和 P_B持有的信息. $\mathcal{S}$模拟诚实参与方 P_B使用在现实执行中的真实输入 y 来参与现实协议执行情形.

混合模型 2. $\mathcal{S}$模拟执行混淆算法，其他行为与混合模型 1 一致.

Triple 和$\mathcal{F}_{\text{abit}}$是理想算法和函数，因此，敌手$\mathcal{A}$在混合模型 2 和混合模型 1 的视图中，都是随机比特值或随机字符串，因此混合模型 1 和混合模型 2 是不可区分的.

混合模型 3. $\mathcal{S}$模拟执行标签映射算法，其他行为与混合模型 2 一致.

可验证随机比特分享技术是安全的，敌手$\mathcal{A}$不能分析得到 P_B输入线路的完整掩码信息，无法从 P_B输入线路的掩饰值恢复其输入真值，敌手$\mathcal{A}$在混合模型 3 和混合模型 2 的视图中仍只有随机比特值，因此这两个模型是不可区分的.

混合模型 4. $\mathcal{S}$模拟执行电路分析算法，其他行为与混合模型 3 一致.

在混合模型 4 中，敌手$\mathcal{A}$未接收任何信息. 因此，混合模型 4 和混合模型 3 是不可区分的.

混合模型 5. $\mathcal{S}$模拟执行真值映射算法，其他行为与混合模型 4 一致.

Three-Halves 混淆电路技术是安全的，具有隐私性和可认证性. 如果敌手$\mathcal{A}$在混合模型 2 中发送的是根据已有信息构造的正确混淆电路份额$[\boldsymbol{G}]_A$，那么敌手$\mathcal{A}$可以收到输出线路掩饰值，因为不知道输出线路完整掩码信息，所以在敌手$\mathcal{A}$的视图中输出线路掩饰值是随机比特值，此时与混合模型 3 中的视图一致. 如果敌手$\mathcal{A}$在混合模型 2 中发送的是部分伪造的混淆电路份额$[\boldsymbol{G}]_A$，那么将收到报错信息，但是敌手仅持有混淆电路份额$[\boldsymbol{G}]_A$，无法推测$\mathcal{S}$在分析完整混淆电路的哪部分导致最后结果的分析错误，此时敌手没有获得额外的信息用于推测$\mathcal{S}$的模拟输入，视图仍与混合模型 3 一致. 因此，混合模型 5 和混合模型 4 是不可区分的.

根据混合模型的不可区分可传递性，混合模型 1 和混合模型 5 是不可区分的，而混合模型 5 是理想世界的执行情形，满足：

$$\text{REAL}_{G,\mathcal{A}}\{\kappa;\{y\mid y\text{ 是诚实参与方 }P_B\text{ 的输入}\}\}\cong \text{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{y\mid y\text{ 是模拟器}\mathcal{S}\text{ 的模拟输入}\}\}.$$

情况 2(P_B恶意). 假设概率多项式时间的敌手$\mathcal{A}$腐化了参与方 P_B，与情况 1 类似，可以构造一个模拟器$\mathcal{S}$. 此时，根据可验证混淆电路方案 G 的四个算法，模拟器$\mathcal{S}$的模拟行为描述如下：

混淆算法. 与情况 1 一致，$\mathcal{S}$模拟生成并记录 P_A和 P_B持有的所有信息. 然后，$\mathcal{S}$模拟 P_A生成混淆电路份额$[\boldsymbol{G}]_A$，发送给敌手$\mathcal{A}$.

标签映射算法. $\mathcal{S}$接受敌手$\mathcal{A}$发送来的所有信息，使用输入 $x:=0$ 来模拟 P_A参与输入线路标签映射过程，生成并发送布尔电路所有输入线路的掩饰值给敌手$\mathcal{A}$. 另外，$\mathcal{S}$接受敌手$\mathcal{A}$发送的 P_B输入线路掩饰值，并通过混淆算法获得的线路掩码值恢复得到 P_A输入线路真值，然后发送给理想函数$\mathcal{F}$得到计算结果 $f(x,y)$.

电路分析算法. 敌手$\mathcal{A}$恢复完整混淆电路结构并计算输出线路值的标签集合.

真值映射算法. 如果 $\mathcal{S}$接收到来自敌手$\mathcal{A}$的输出线路掩饰值，那么根据持有的两个参与方信息恢复输出线路掩码，与输出线路掩饰值异或得到输出真值. 否则，直接终止.

类似地，针对情况 2，建立混合模型如下：

混合模型 1. $\mathcal{S}$模拟生成并记录 P_A和 P_B持有的信息. $\mathcal{S}$模拟诚实参与方 P_A使用在现实执行中的真实输入 x 来参与现实协议执行情形.

混合模型 2. $\mathcal{S}$模拟执行混淆算法，其他行为与混合模型 1 一致.

与情况 1 同理，混合模型 1 和混合模型 2 是不

可区分的.

混合模型 3. $\mathcal{S}$模拟执行标签映射算法,其他行为与混合模型 2 一致.

可验证随机比特分享技术是安全的,敌手$\mathcal{A}$不能分析得到 P_A输入线路的完整掩码信息,接收的标签集合都是随机字符串形式,敌手$\mathcal{A}$在混合模型 3 的视图中仍为随机比特值或随机字符串,因此混合模型 3 和混合模型 2 是不可区分的.

混合模型 4. $\mathcal{S}$模拟执行电路分析算法,其他行为与混合模型 3 一致.

在混合模型 4 中,敌手$\mathcal{A}$恢复完整的混淆电路结构并分析. Three-Halves 混淆电路技术是安全的,具有隐私性、不经意性,敌手$\mathcal{A}$无法根据完整的混淆电路结构以及输入线路掩饰值的标签集合推测 P_A的输入真值. 因此,混合模型 4 和混合模型 3 是不可区分的.

混合模型 5. $\mathcal{S}$模拟执行真值映射算法,其他行为与混合模型 4 一致.

可验证随机比特分享技术是安全的,敌手$\mathcal{A}$可以验证并恢复得到输出线路掩码,计算得到输出真值,这与现实情形一致. 因此,混合模型 5 和混合模型 4 是不可区分的. 同理,根据混合模型的不可区分可传递性,混合模型 1 和混合模型 5 是不可区分的,而混合模型 5 是理想世界的执行情形,满足:

$$\mathrm{REAL}_{\mathrm{G},\mathcal{A}}\{\kappa;\{x|x\text{ 是诚实参与方}P_A\text{的输入}\}\}\cong \mathrm{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{x|x\text{ 是模拟器}\mathcal{S}\text{的模拟输入}\}\}.$$

证毕.

3.3 复杂度分析

Three-Halves 混淆电路技术[15]中标签之间仍具有固定异或偏移量,因此在本节提出的可验证混淆电路分享方案中,XOR 门仍是“零开销”的. 与门的总通信量主要由混淆电路中的与门数量决定,每个与门的混淆结构大小为 $1.5\kappa+2d$. 其中,d 是控制比特矩阵 $\boldsymbol{R}$ 的压缩加密形式的行维度. 在本文中默认两个参与方均知晓布尔电路中所有布尔门的类型,因此 $d=2$.

表 2 总结了已有的可验证混淆电路分享方案的计算复杂度与通信复杂度,并与所提出的方案进行了对比. 经分析表明,本方案计算完整电路的通信复杂度为$(1.5\kappa+2d)|C|+\kappa(|I|+|O|)$,其中$|C|$表示电路中与门的数量,$\kappa$ 表示安全参数,$|I|$表示电路输入线路数量,$|O|$表示电路输出线路数量. 本方案的计算复杂度主要由哈希计算的次数决定,为一个与门构造混淆结构需计算 6 次哈希函数,分析其混淆结构需计算 3 次哈希函数,因此总计算复杂度为 $9\kappa|C|$. 与文献[27]相比,本方案优化了约 $2.5\kappa|C|$;与文献[28]相比,本方案优化了约 $0.5\kappa|C|$. 在网络条件受限的情况下,本文提出的方案是有意义的,可以通过文献[15]的实验分析总结得到.

表 2 可验证混淆电路分享方案复杂度对比表

协议	混淆电路技术	计算复杂度	通信复杂度
WRK17[27]	Free-XOR[12]	$6\kappa\|C\|$	$4\kappa\|C\|+\kappa(\|I\|+\|O\|)$
KRRW18[28]	Half-Gates[14]	$6\kappa\|C\|$	$2\kappa\|C\|+\kappa(\|I\|+\|O\|)$
本方案	Three-Halves[15]	$9\kappa\|C\|$	$(1.5\kappa+2d)\|C\|+\kappa(\|I\|+\|O\|)$

4 高效安全两方计算协议

4.1 合作式流程设计方案

4.1.1 方案概述

安全两方计算协议的传统流程划分为五个阶段,分别是函数无关预处理阶段、函数相关预处理阶段、输入预处理阶段、电路分析阶段以及输出阶段. 两个参与方一般分别担任混淆方和计算方的角色. 混淆方为布尔电路的每个布尔门真值表生成对应的混淆表. 计算方根据电路拓扑结构逐门分析混淆表,直至获得输出结果.

在函数无关预处理阶段,两方获得生成混淆电路所需要的前置信息,例如各线路掩码份额等. 在函数相关预处理阶段,担任混淆方角色的参与方根据布尔电路的拓扑结构生成混淆电路. 具体的,混淆方为布尔电路中每个布尔门对应的真值表生成一张混淆表,混淆表的每一行与两条门输入线路的 0/1 比特值可能组合相关,在混淆表中布尔值都以标签的形式表示(0 标签和 1 标签),标签一般是一个随机字符串. 在输入预处理阶段,两方通信获得电路输入线路标签集合中与输入线路真值对应的标签子集合. 在电路分析阶段,计算方使用输入线路所取真值对应的标签,对混淆电路进行逐门分析. 具体的,计算方使用布尔门两条输入线路分别对应的标签,对混淆表进行逐行分析,有且仅有一个门输出线路的标签(0 标签或 1 标签)能够被计算方得到. 直到分析得出电路输出线路的标签. 在输出阶段,混淆方揭示电路输出标签与输出真值的对应关系,获得输出结果.

在掩码技术[10]中,计算方可以通过掩饰值定位到混淆表中的某一行,不再需要对混淆表进行逐行分析. 因此,在基于混淆电路技术的两方安全计算协议中,混淆方生成混淆电路的计算复杂度一般要比计算方

分析混淆电路的计算复杂度高. 以 Three-Halves[15]混淆电路技术为例，混淆方为 1 个与门生成混淆表需要调用 6 次哈希函数，计算方分析与门的混淆表只需要调用 3 次哈希函数. 在这一示例中，因为计算压力主要集中在哈希计算上，所以忽略了有关异或等计算操作的描述.

综上，基于混淆电路技术的安全两方计算协议存在计算复杂度不平衡的问题. 针对这一问题，本节设计了一个合作式协议流程设计方案，在两个参与方之间分摊计算压力，提升实例化协议的效率. 在该方案中，两个参与方同时担任混淆方和计算方这两个角色，共同完成混淆电路的生成和分析过程，从而实现将函数相关预处理阶段和电路分析阶段的总计算压力分摊到两个参与方上，提升实例化协议的实现效率.

首先，根据电路的大小，将完整的电路划分成前半电路和后半电路这两个子电路，这里主要根据与门的数量划分(异或门“零开销”). 然后，在协议相关预处理阶段，两个参与方分别为前半电路和后半电路生成混淆子电路；在电路分析阶段，由后半电路的混淆方完成前半电路的混淆子电路分析任务，由前半电路的混淆方完成后半电路的混淆子电路分析任务.

仍以 Three-Halves[15]混淆电路技术为例，假设电路规模为 $|C|$，安全参数为 κ，混淆方计算复杂度达 $6\kappa|C|$，计算方的计算复杂度达 $3\kappa|C|$. 结合合作式流程设计方案之后，两方的计算复杂度分别达 $4.5\kappa|C|$，分摊了计算压力. 类似的，本协议流程方案可以适配到基于 Free-XOR[12]、Half-Gates[14]等混淆电路技术实现的安全两方计算协议当中.

在该方案中，两个参与方分别生成和分析前半电路和后半电路的混淆子电路，如何将前半电路的输出线路值映射到后半电路的混淆子电路输入线路的标签是一个关键问题. 基于可验证随机比特分享技术实现的掩码技术可以解决这一问题. 在该技术中，门的各线路标签的下角标仅作为掩饰值来定位混淆行，与真实的比特值不是对应的. 掩饰值是真实比特值的加密形式，因此可以直接暴露给计算方. 计算方完成前半电路的混淆电路分析工作以及后半电路的混淆电路生成工作. 计算方可以在本地实现前半电路输出线路掩饰值与后半电路输入线路标签的对应.

4.1.2　具体构造

假设前半电路的输出线路集合表示为 O_{Front}，后半电路的输入线路集合表示为 I_{behind}. 其中，O_{Front} 和 I_{behind} 集合的大小是一样的，即对应相同的电路中间线路. 另外，G 表示 C 的一个混淆电路，协议中的两个参与方分别表示为 P_A 和 P_B，安全两方计算协议的通用五阶段表示为五元组 $(\text{Func}_{\text{Inde}}, \text{Func}_{\text{De}}, \text{Input}_{\text{Pre}}, \text{Eval}, \text{Output})$.

另外，在协议中使用的承诺方案表示为一个两元组 $(\text{Com}(\cdot), \text{OpenCom}(\cdot))$，其中 $\text{Com}(\cdot)$ 表示承诺算法方对括号中的内容生成一份承诺，$\text{OpenCom}(\text{Com}(\cdot))$ 表示承诺方将承诺 $\text{Com}(\cdot)$ 的内容公开，由接收方验证承诺 $\text{Com}(\cdot)$ 的正确性，如果相等，则接收方接受承诺；否则，接收方拒绝承诺.

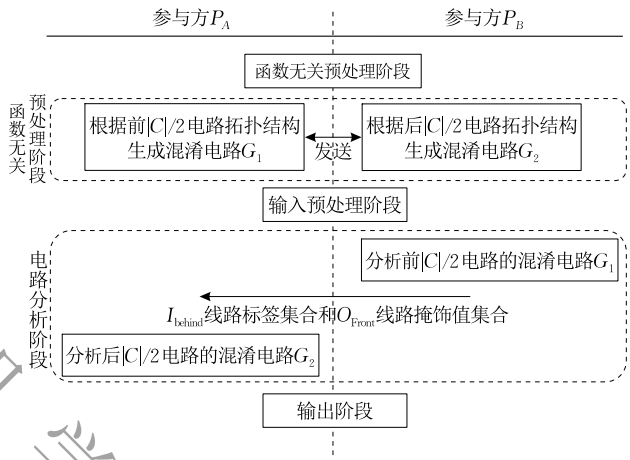


图 4　合作式协议流程图

合作式协议流程如图 4 所示，具体描述如下：

1. 在函数无关预处理阶段 $\text{Func}_{\text{Inde}}$，$P_A$ 和 P_B 生成混淆电路需要的信息，如基于 OT 技术为布尔电路各线路生成随机掩码份额及其相应的验证值、为输入线路生成 0/1 比特值对应的标签对等；

2. 在函数相关预处理阶段 Func_{De}，P_A 为前 $|C|/2$ 电路构造一个混淆电路 $\boldsymbol{G}_1$，并为输出线路集合 O_{Front} 中各输出线路可取的比特值与相应标签生成一对承诺 $(\text{Com}(0,\boldsymbol{L}_0), \text{Com}(1,\boldsymbol{L}_1))$，将混淆电路 $\boldsymbol{G}_1$ 和相应承诺对发送给 P_B；P_B 为后 $|C|/2$ 电路构造一个混淆电路 $\boldsymbol{G}_2$，并为输出线路集合 O 中各输出线路可取的比特值与相应标签生成一对承诺 $(\text{Com}(0,L'_0), \text{Com}(0,L'_1))$，将混淆电路 $\boldsymbol{G}_2$ 和相应承诺对发送给 P_A；

3. 在输入相关预处理阶段 $\text{Input}_{\text{Pre}}$，$P_A$ 将输入线路掩饰值集合对应的单标签集合发送给 P_B. 电路输入线路集合 I 就是前 $|C|/2$ 电路的输入线路集合. 其中，P_A 的各输入线路真值掩饰形式对应的标签由 P_A 直接发送给 P_B，P_B 的各输入线路真值掩饰形式对应的标签由与 P_A 通信交互获得；

4. 在电路分析阶段 Evaluate，P_B 首先基于在步骤 3 获取的电路输入线路集合 I 对应的标签，分析在步骤 2 获取的混淆电路 $\boldsymbol{G}_1$，得到一组前半电路的输出线路集合 O_{Front} 对应

的标签.其中,输出线路集合 O_{Front} 中各输出线路 O_i 对应掩饰值 v_i 通过验证 OpenCom(Com(v_i,$\boldsymbol{L}_{v_i}$))后获得.集合 O_{Front} 和 I_{behind} 是一致的.P_B 将 O_{Front} 中各线路的真值的掩饰形式对应的混淆电路 $\boldsymbol{G}_2$ 输入线路标签集合发送给 P_A;P_A 使用这一组标签分析在步骤 2 获取的混淆电路 $\boldsymbol{G}_2$,得到一组后半电路输出线路集合,即电路输出线路集合 O 中各输出线路的标签;

5.在输出阶段,输出线路集合 O 中各输出线路 O_i 对应掩饰值 v_i 通过 OpenCom(Com(v_i,$\boldsymbol{L}_{v_i}$))获得,然后 P_B 公开集合 O 中各输出线路的掩码值,P_A 和 P_B 分别基于掩码值恢复得到电路的输出真值.

合作式协议流程设计方案的前提假设是两个参与方已知完整电路的拓扑结构.合作式协议流程设计方案可以实现两个参与方之间计算压力的分摊,在计算复杂电路的过程中,不会有其中一方因为计算压力大而成为瓶颈.在恶意场景下,因为前半混淆电路的分析要早于后半混淆电路,关于前半电路分析的正确性检查可以与后半混淆电路的分析过程并行完成,如果检查到错误,协议可以提前终止,不必等后半混淆电路分析的完成.

本流程设计方案根据电路的大小进行了子电路的划分.基于 Free-XOR[12] 技术实现的安全两方计算协议能够实现异或门计算和通信“零开销”,即无需交互,通过本地进行简单的逻辑计算就可以完成计算.如果完整电路中与门的数量要在电路中分布集中在后半电路,那么合作式协议流程设计方案无法发挥作用.在这种情况下,本方案可以稍加改进,选择根据电路中与门的数量来划分前半电路和后半电路,让两个子电路中的与门数量大致相等,从而实现计算压力的分摊.因此,两方需要在计算前已知电路的拓扑结构.

4.2 协议设计

本节设计了一个基于混淆技术的高效安全两方计算协议.通过合作式协议流程设计方案,本协议中将计算压力分摊到两个参与方;通过第三节提出的可验证混淆电路分享方案,本协议牺牲了部分参与方的计算复杂度,而且能够抵抗恶意敌手的选择失败攻击.因此,协议实例可以提升效率.

本协议包括两个参与者,分别表示为 P_A 和 P_B.协议的具体步骤描述如下:

步骤 1. 函数无关预处理阶段 $\text{Func}_{\text{Inde}}$,为了后续的混淆电路生成和分析任务,$P_A$ 和 P_B 需要完成如下工作.

1. 生成全局密钥 Δ_A/Δ_B 以及线路密钥集合;

2. 通过可验证随机比特方案 $\mathcal{F}_{\text{abit}}$,为电路输入线路、与门输出线路生成掩码比特分享份额以及验证 MAC 值集合;

3. 通过 Beaver 三元组方案 Triple,为与门的输出线路生成掩码运算比特分享份额;

4. P_A 为前$|C|/2$电路的输入线路集合随机选取 0 标签集合,P_B 为后$|C|/2$电路的输入线路集合随机选取 0 标签集合.

步骤 2. 函数相关预处理阶段 Func_{De}.

1. P_A 和 P_B 通过第 3.1.2 节的算法 Garble,为前$|C|/2$电路构造一个混淆电路 $\boldsymbol{G}_1$,其中,P_A 生成混淆份额$[\boldsymbol{G}_1]_A$并发送给 P_B,P_B 本地持有混淆份额$[\boldsymbol{G}_1]_A$;

2. P_A 和 P_B 通过第 3.1.2 节的算法 Garble,为后$|C|/2$电路构造一个混淆电路 $\boldsymbol{G}_2$,其中,P_B 生成混淆份额$[\boldsymbol{G}_2]_B$并发送给 P_A,P_A 本地持有混淆份额$[\boldsymbol{G}_2]_A$,这里需要注意的是,P_B 需基于自己的全局密钥 Δ_B 来生成$[\boldsymbol{G}_2]_B$,也就是说 P_A 和 P_B 在 Garble 流程中角色互换.

步骤 3. 输入相关预处理阶段 $\text{Input}_{\text{Pre}}$.

1. P_A 和 P_B 通过 Open(•,…)验证输入线路掩码随机比特份额的正确性,分别本地生成自己对应的输入线路掩饰值,发送给另一方;

2. P_A 将与电路输入线路集合 I 中各输入线路真值的掩饰形式对应的标签发送给 P_B,电路输入线路集合 I 就是前$|C|/2$电路的输入线路集合.

步骤 4. 电路分析阶段 Evaluate.

1. P_B 恢复得到完整的混淆电路结构 $\boldsymbol{G}_1$,使用在步骤 3 获取的输入线路集合 I 对应的标签集合,通过第 3.1.2 节的算法 Eval,分析 $\boldsymbol{G}_1$,得到一组前$|C|/2$电路的输出线路集合 O_{Front} 中各输出线路的标签以及掩饰值;

2. P_B 将 O_{Front} 集合中各线路的掩饰值以及各掩饰值对应到后$|C|/2$电路输入线路标签集合发送给 P_A,另外前$|C|/2$电路的与门输出掩饰值也发送给 P_A;

3. P_A 恢复得到完整的混淆电路结构 $\boldsymbol{G}_2$,使用来自 P_B 的后$|C|/2$电路输入线路对应标签集合,通过第 3.1.2 节的算法 Eval,分析 $\boldsymbol{G}_2$,得到一组后$|C|/2$电路的输出线路集合 O,即完整电路输出集合中各线路的标签以及相应掩饰值.同时,P_A 基于可验证随机比特技术的异或同态属性,对接收到的前$|C|/2$电路掩饰值计算的正确性进行验证,具体验证过程与文献[27]一致,如果前$|C|/2$电路正确性检验过程发现错误,则直接终止协议.

步骤 5. 输出阶段 Output.

1. P_A 和 P_B 通过第 3.1.2 节的算法 Decode 获得集合 O 中各输出线路真值,P_B 公开集合 O 中各输出线路的掩码值;

2. P_B 基于可验证随机比特技术的异或同态属性,对接收到的后$|C|/2$电路掩饰值计算的正确性进行验证,同样地,具体验证过程也与文献[27]一致,基于可验证随机比特分享技术和桶联合验证技术[32]实现.如果后$|C|/2$电路的正确性检验过程出现了错误,则直接终止协议.如果正确性验证通过,则通过算法 Decode 获得的输出线路真值.

4.3　安全性分析

定理 2.　恶意模型的安全性. 如果 H 是一个具有 RTCCR 属性的哈希函数，那么第 4.2 节提出的协议 π 能以(Triple，$\mathcal{F}_{\text{abit}}$，Com)-混合模型安全地计算电路 f，并抵抗具有多项式时间内可计算能力的恶意敌手 $\mathcal{A}$ 攻击.

证明.　附录 1 给出了第 4.1 节合作式协议流程设计方案的安全性分析. 对于协议 π，假设现实世界中的恶意敌手 $\mathcal{A}$ 控制的参与方集合为 corrput($\mathcal{A}$)，理想世界中的敌手 $\mathcal{S}$ 控制的参与方集合为 corrput($\mathcal{S}$)，理想函数 $\mathcal{F}$ 是电路 f 的理想执行. 如果对于现实世界中任意一个敌手 $\mathcal{A}$，存在一个满足 corrput($\mathcal{A}$)=corrput($\mathcal{S}$)的模拟器 $\mathcal{S}$，使对于诚实参与方的输入$\{x_i \mid i \notin \text{corrput}(\mathcal{A})\}$，满足：

$$\text{REAL}_{\pi,\mathcal{A}}\{\kappa;\{x_i \mid i \notin \text{corrput}(\mathcal{A})\}\} \cong \text{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{x_i \mid i \notin \text{corrput}(\mathcal{S})\}\},$$

则称协议 π 在恶意敌手模型下安全地实现了理想函数 $\mathcal{F}$.

接下来分别考虑恶意 P_A 和恶意 P_B 两种情形，通过理想-现实模拟的方法证明定理 3.

情况 1(P_A 恶意). 假设拥有多项式时间计算能力的敌手 $\mathcal{A}$ 腐化了参与方 P_A，然后构造一个模拟器 $\mathcal{S}$，既内部执行一个敌手 $\mathcal{A}$ 装置来模拟 P_A，也模拟诚实参与方 P_B 的行为，模拟器 $\mathcal{S}$ 通过访问一个理想函数 $\mathcal{F}$ 来分析(f，$f_{\text{前}1/2}$).

根据协议流程，模拟器 $\mathcal{S}$ 行为描述如下：

步骤 1. $\mathcal{S}$ 模拟生成并记录 P_A 和 P_B 持有的信息，具体包括如下内容：

1. 全局密钥(Δ_A，Δ_B)以及线路密钥集合；

2. $\mathcal{F}_{\text{abit}}$ 发给两个参与方的电路输入线路、与门输出线路的掩码比特分享份额以及验证 MAC 值集合；

3. Beaver 三元组方案 Triple 发给两个参与方的与门输出线路生成掩码运算比特分享份额；

4. 前$|C|/2$ 电路的输入线路的 0 标签集合和后$|C|/2$ 电路的输入线路集合比特值 0 的标签集合.

步骤 2. $\mathcal{S}$ 接受所有内部敌手装置发送的信息，使用输入 0 来模拟 P_B 参与混淆电路生成过程，发送生成的后半电路混淆分享份额给 P_A.

步骤 3. $\mathcal{S}$ 接受敌手 $\mathcal{A}$ 发送的 P_A 输入线路掩饰值，并通过预处理阶段获得的线路掩码值恢复得到 P_A 输入线路真值，然后发送给 $\mathcal{F}$ 得到：

$$z=f(x,y) \text{和} \ z_{1/2}=f_{\text{前}1/2}(x,y).$$

步骤 4. $\mathcal{S}$ 接受所有内部敌手装置发送的信息，也使用输入 $y:=0$ 来模拟 P_B 参与混淆电路分析过程. 关于前半电路的输出线路集合，$\mathcal{S}$ 计算 $f_{\text{前}1/2}(x,0)$，与对应线路掩码异或获得掩饰值形式，作为输出掩饰值发送给敌手 $\mathcal{A}$.

步骤 5. $\mathcal{S}$ 接受所有内部敌手装置发送的信息，模拟 P_B 的行为. 如果一个诚实参与方发现错误终止，那么 $\mathcal{S}$ 也模拟终止. 如果没有终止，$\mathcal{S}$ 通过预处理阶段生成的信息以及敌手 $\mathcal{A}$ 发来的信息，计算得到敌手 $\mathcal{A}$ 的输入值 x，发送给 $\mathcal{F}$.

根据模拟证明方法原理，如果在现实执行中，半诚实敌手 $\mathcal{A}$ 的输出与诚实参与方 P_B 输出联合视图与在理想世界协议的执行过程中 $\mathcal{S}$ 与参与方 P_B 的输出联合视图是不可区分的，即满足：

$$\text{REAL}_{\pi,\mathcal{A}}\{\kappa;\{x_i \mid x_i \notin \{P_A\}\}\} \cong \text{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{x_i \mid i \notin \{P_A\}\}\},$$

则称此协议在恶意敌手模型下安全地实现了理想函数 $\mathcal{F}$，可以说明 P_A 是恶意的情况下 π 协议是安全的. 其中，诚实输入 $x_i=y$.

具体的，需要通过建立一系列混合模型来证明该结论. 在混合模型中，第一个模型是协议的现实执行情形，最后一个模型是理想世界中协议的执行情形. 如果各相邻的混合模型是不可区分的，那么通过混合模型之间的传递性可以递推得到混合模型 1 与混合模型 3 之间的不可区分性.

混合模型 1. $\mathcal{S}$ 模拟诚实参与方 P_B 使用在现实执行中的真实输入 y 来参与现实协议执行情形. 同时，函数无关预处理阶段的信息都由 $\mathcal{S}$ 模拟生成.

混合模型 2. $\mathcal{S}$ 接受敌手 $\mathcal{A}$ 发送的 P_A 输入掩饰值，并通过预处理阶段获得的线路掩码值恢复得到 P_A 输入线路真值，其他行为与混合模型 1 一致. 如果参与方 P_B 在协议的任意步骤发生终止，那么 $\mathcal{S}$ 发送终止信息给 $\mathcal{F}$；否则，发送 P_A 输入真值给 $\mathcal{F}$.

根据上述描述，可以发现敌手 $\mathcal{A}$ 在混合模型 1 和混合模型 2 的视图中，所有线路真值都是掩饰值形式或随机字符串形式，因此两个视图是一致的. 根据定理 1，可以推得 P_B 在混合模型 1 和混合模型 2 中的输出信息是不可区分的.

混合模型 3. $\mathcal{S}$ 模拟诚实参与方 P_B 使用 $y:=0$ 来参与协议，关于前半电路的输出线路集合，$\mathcal{S}$ 计算 $f_{\text{前}1/2}(x,0)$，与对应线路掩码异或获得掩饰值形式，作为输出掩饰值发送给敌手 $\mathcal{A}$，并忽略敌手 $\mathcal{A}$ 的信息，其他行为与混合模型 2 中一致.

敌手 $\mathcal{A}$ 本地关于诚实参与方 P_B 输入的信息是随机字符串形式的标签以及掩饰形式，敌手 $\mathcal{A}$ 在混合模型 2 和混合模型 3 中的视图应是一致的. 在混合模型 2 中，敌手 $\mathcal{A}$ 可能随意构造混淆结构$[\boldsymbol{G}_1]_A$和

$[\boldsymbol{G}_2]_A$来影响后续计算的正确性,或者执行可选择失败攻击,即构造错误的$[\boldsymbol{G}_1]_A$和$[\boldsymbol{G}_2]_A$,然后通过参与方P_B在分析过程中因出现分析错误而终止的情况,分析得到P_B的输入.

因为在第3.1节的可验证混淆电路分享方案中,线路掩码是分享形式的,在混淆电路的生成和分析过程中,各布尔门输出线路掩码不被任何参与方知晓.如果$\mathcal{A}$发送的混淆结构中包含错误的掩码份额,参与方P_B可以通过验证函数Open检验到该错误.即使错误没有被检验到,由于协议引入了正确性检查机制,因此参与方P_B可以在协议的分析阶段终止.但$\mathcal{A}$不知道完整的掩码信息,也就不知道有关混淆行的位置信息,无法分析得到终止错误信息是对应于哪一混淆行,也就无法得知P_B的输入信息.因此,混合模型2中的终止情况不会让敌手$\mathcal{A}$的视图包含掩饰值集合与随机字符串以外的信息.根据混合模型的不可区分可传递性,混合模型1和混合模型3是不可区分的,而混合模型3是理想世界的执行情形,满足:

$$\mathrm{REAL}_{\pi,\mathcal{A}}\{\kappa;\{x_i|x_i\notin\{P_A\}\}\}\cong$$
$$\mathrm{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{x_i|i\notin\{P_A\}\}\}.$$

情况2(P_B恶意).假设拥有多项式时间计算能力的敌手$\mathcal{A}$腐化了参与方P_B,然后构造一个模拟器$\mathcal{S}$,既内部执行一个敌手$\mathcal{A}$装置来模拟P_B,也模拟诚实参与方P_A的行为,模拟器$\mathcal{S}$通过访问一个理想函数$\mathcal{F}$来分析$(f,f_{\text{前}1/2},f_{\text{后}1/2})$.根据协议流程,模拟器$\mathcal{S}$的具体行为描述如下:

步骤1. $\mathcal{S}$模拟生成并记录P_A和P_B持有的信息,具体包括如下内容:

1. 全局密钥(Δ_A,Δ_B)以及线路密钥集合;

2. $\mathcal{F}_{\text{abit}}$发给两个参与方的电路输入线路、与门输出线路的掩码比特分享份额以及验证MAC值集合;

3. Beaver三元组方案Triple发给两个参与方的与门输出线路生成掩码运算比特分享份额;

4. 前$|C|/2$电路的输入线路0标签集合和后$|C|/2$电路的输入线路0标签集合.

步骤2. $\mathcal{S}$接受所有内部敌手装置发送的信息,也使用输入为0来模拟P_A参与混淆电路生成过程,发送生成的前半电路混淆分享份额给P_B.

步骤3. $\mathcal{S}$接受敌手$\mathcal{A}$发送的P_B输入线路掩饰值,并通过预处理阶段获得的线路掩码值恢复得到P_B的输入线路真值,然后发送给$\mathcal{F}$,得到:

$$z=f(x,y),f_{\text{前}1/2}(x,y),f_{\text{后}1/2}(f_{\text{前}1/2}(x,y)).$$

步骤4. $\mathcal{S}$接受所有内部敌手装置发送的信息,也使用输入$x:=0$来模拟P_A参与混淆电路分析过程.关于后半电路的输出线路集合,$\mathcal{S}$计算$f_{\text{后}1/2}(f_{\text{前}1/2}(0,y))$,与对应线路掩码异或获得掩饰值形式,作为输出掩饰值发送给敌手$\mathcal{A}$.

步骤5. $\mathcal{S}$接受所有内部敌手装置发送的信息,模拟P_A的行为.如果一个诚实参与方P_A发现错误终止,那么$\mathcal{S}$也模拟终止.如果没有终止,$\mathcal{S}$通过预处理阶段生成的信息以及敌手$\mathcal{A}$发来的信息,计算得到敌手$\mathcal{A}$的输入值y以及前半电路理想函数输出值$f_{\text{前}1/2}(x,y)$,发送给$\mathcal{F}$.

如果在现实协议执行过程中,敌手$\mathcal{A}$的输出与诚实参与方P_A输出组成的联合视图与在理想世界协议的执行过程中$\mathcal{S}$与诚实参与方P_A的输出联合视图的概率分布是不可区分的,则称此协议在恶意敌手模型下安全地实现了理想函数$\mathcal{F}$,可以说明P_B是恶意的情况下π协议是安全的.其中,输入$x_i=x$.与情况1一样,通过建立一系列混合模型来证明该结论.

混合模型1. $\mathcal{S}$模拟诚实P_A使用真实输入x来参与现实协议执行情形.同时,函数无关预处理阶段的信息都由$\mathcal{S}$模拟生成.

混合模型2. $\mathcal{S}$接受敌手$\mathcal{A}$发送的P_B输入线路掩饰值,并通过预处理阶段获得的线路掩码值恢复得到P_B输入线路真值,其他行为与混合模型1一致.如果参与方P_A在协议的任意步骤发生终止,那么$\mathcal{S}$发送终止信息给$\mathcal{F}$;否则,发送P_B输入线路真值给$\mathcal{F}$.

根据上述描述,可以发现敌手$\mathcal{A}$在混合模型1和混合模型2的视图中,所有线路真值仍是掩饰值形式或随机字符串形式,因此两个视图是一致的.根据定理1,可以推得P_B在混合模型1和混合模型2中的输出信息是不可区分的.

混合模型3. $\mathcal{S}$模拟诚实参与方P_A使用$x:=0$来参与协议,关于后半电路的输出线路集合,$\mathcal{S}$计算$f_{\text{后}1/2}(f_{\text{前}1/2}(0,y))$,与对应线路掩码异或获得掩饰值形式,作为输出掩饰值发送给敌手$\mathcal{A}$,并忽略其返回信息,其他行为与混合模型2一致.

敌手$\mathcal{A}$本地关于诚实参与方P_B输入的信息是随机字符串形式的标签以及掩饰形式,敌手$\mathcal{A}$在混合模型2和混合模型3中的视图应是一致的.在混合模型2中,敌手$\mathcal{A}$可能随意构造混淆结构$[\boldsymbol{G}_1]_B$和$[\boldsymbol{G}_2]_B$来影响后续计算的正确性和隐私性.

与情况1同理,P_A可以通过验证函数Open检验到该错误,正确性验证过程中发生终止也不会让敌手$\mathcal{A}$得知输入信息.

因此,混合模型2中的终止情况不会让敌手$\mathcal{A}$的视图包含掩饰值集合与随机字符串以外的信息,混合模型2和混合模型3是不可区分的.根据混合

模型的不可区分可传递性，混合模型 1 和混合模型 3 是不可区分的，满足：

$$\mathrm{REAL}_{\pi,\mathcal{A}}\{\kappa;\{x_i|x_i\notin\{P_B\}\}\}\cong$$
$$\mathrm{IDEAL}_{\mathcal{F},\mathcal{S}}\{\kappa;\{x_i|i\notin\{P_B\}\}\}.$$

根据情况 1 和情况 2 的结论，协议 π 可以在恶意敌手模型下安全地计算电路 f.　　证毕.

4.4　复杂度分析

在第 4.2 节的步骤 4 的 2～3 中，为了让 P_A 能够基于前半电路的输出结果分析后半电路的混淆电路，P_B 需要在本地完成从前半电路的输出线路集合 O_{Front} 到后半电路的输入线路集合 I_{behind} 的标签映射工作，然后将对应集合 I_{behind} 的标签集合发送给 P_A. 因为本协议的合作式协议流程设计方案是基于掩码技术实现的，P_B 能够获知输出线路的真值的掩饰值形式，所以标签映射工作是简单的比特值-标签映射，这一部分的计算复杂度可忽略. 因为要发送额外的标签集合，通信复杂度有额外的 $O(I_{\mathrm{behind}})$ 开销，总通信复杂度达 $O(|C|+|I|+|O|+|I_{\mathrm{behind}}|)$. 因为部分线路通信量的增加与完整电路大小相比是少量的，所以本协议通信复杂度仍属于 $O(|C|)$ 级别.

由第 3.3 节中关于本文提出的可验证混淆电路分享方案可知，由于 $d=2$，与安全参数 κ 相比是可以忽略的. 因此，本协议与文献[27]的协议相比，通信量降低了约 $2.5\kappa|C|$，计算复杂度增加了 $3\kappa|C|$；与文献[28]的协议相比，通信量降低了约 $0.5\kappa|C|$，计算复杂度增加了 $3\kappa|C|$.

在同等安全级别下，相比于文献[27]和文献[28]，本协议的通信复杂度得到了优化，实现了计算压力的分摊. 另外，根据文献[15]，本协议使用的哈希函数仅需要满足 RTCCR 属性[15]. 与文献[27]和文献[28]相比，本协议可以选择更加轻量化的哈希函数，使得协议实例效率得到了进一步提升；并可以根据不同的布尔电路组成，进一步优化哈希函数调用次数. 因此，总计算复杂度不大于 $9\kappa|C|$，基于合作式协议流程方案，两个参与方的计算复杂度均不大于 $4.5\kappa|C|$. 本协议与基于可验证混淆电路分享方案实现的协议对比总结如表 3 所示.

表 3　可验证混淆电路分享方案复杂度对比表

协议	计算复杂度		通信复杂度
	P_A	P_B	
WRK17[27]	$4\kappa\|C\|$	$2\kappa\|C\|$	$4\kappa\|C\|+\kappa(\|I\|+\|O\|)$
KRRW18[28]	$4\kappa\|C\|$	$2\kappa\|C\|$	$2\kappa\|C\|+\kappa(\|I\|+\|O\|)$
本协议	$\leq 4.5\kappa\|C\|$	$\leq 4.5\kappa\|C\|$	$(1.5\kappa+2d)\|C\|+\kappa(\|I\|+\|O\|)+\|I\|_{\mathrm{behind}}$

5　实验与分析

5.1　仿真系统设计

基于本文第 4 节提出的安全两方计算协议，本节实现了仿真系统，该仿真系统的系统架构如图 5 所示，自上而下包括应用服务层、数据编译层、计算协议层以及密码组件层.

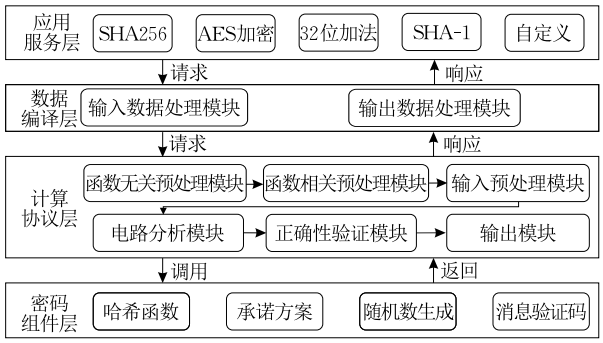


图 5　系统架构图

应用服务层提供了 SHA-1、SHA256、AES 加密以及 32 位加法这四类算法服务，支持自定义布尔电路的计算，具有服务可扩展性；数据编译层实现了其他数据类型与布尔类型之间的转换处理；计算协议层实现了第 4 节提出的协议，用于完成任务的安全计算；密码组件层提供了哈希函数等密码技术，支持计算协议层的计算.

5.1.1　测试方案

本文对仿真系统的性能进行了测试，并与 Emp-toolkit 进行对比分析. Emp-toolkit[5] 中实现了在恶意敌手模型下安全且基于混淆电路的安全两方计算协议，其可验证混淆电路方案来自文献[27].

测试方案包括计算时延测试、通信量测试以及通信时延测试这三个部分，分别测试 AES、32 位加法、SHA-1 以及 SHA256 这四种算法，具体的测试方法如下：

(1) 计算时延测试. 在每个算法计算任务的执行过程中，将从导入数据开始到输出计算结果所用的时间作为 1 次计算时延结果，基于 10 种不同的输入输出集测试 10 次，取平均耗时作为最终测试结果，单位为 ms；

(2) 通信量测试. 记录在每个算法计算任务执行过程中产生的通信报文规模，单位为字节；

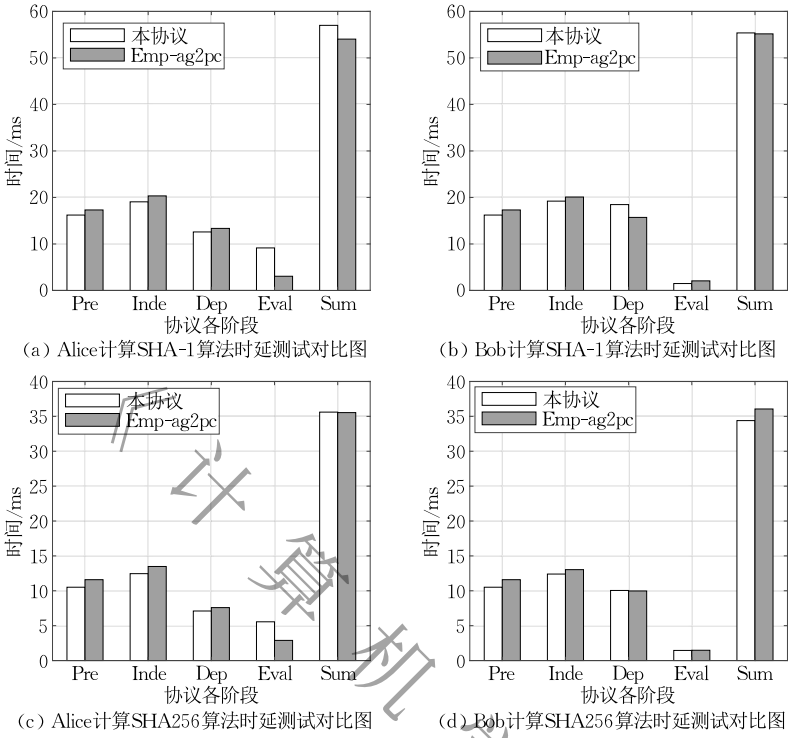
(3) 通信时延测试. 在每个计算任务执行过程中，将参与方每次通信交互的时间作为 1 次通信时延结果，基于 10 种不同的输入输出集测试 10 次，取平均耗时作为最终测试结果，单位为 ms.

5.1.2 测试环境

本文实现的仿真系统在操作系统 MacOS 11.5.1 上进行测试，处理器为 Apple M1，内存为 16 GB，编程软件为 CLion C/C++2021.2，使用的编程语言版本为 C++11，编译器为 Clang 12.0.5.

5.2 实验分析

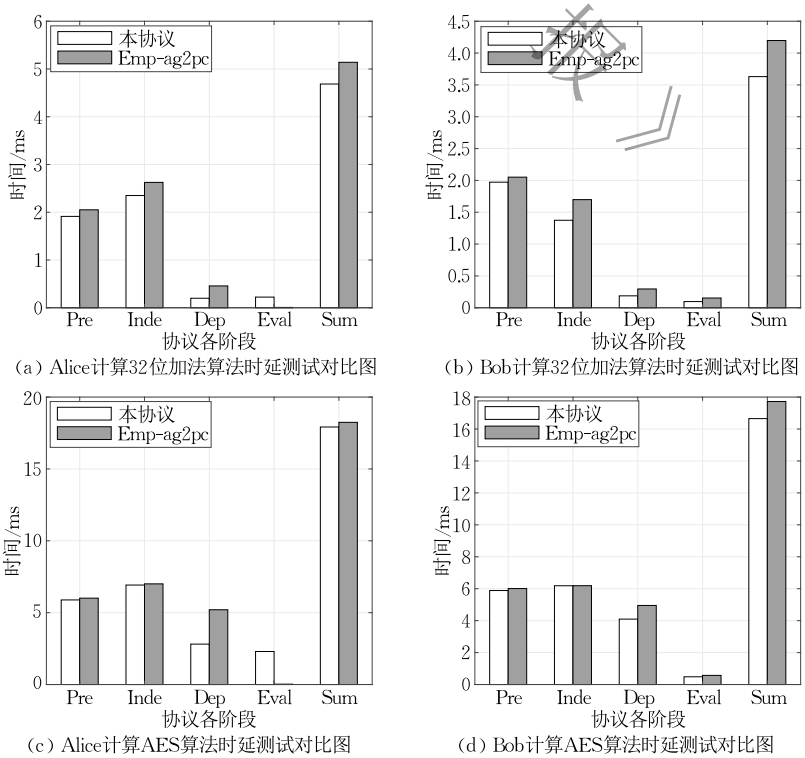
在性能测试方面，将本仿真系统的安全两方计算协议和 Emp-toolkit 两方协议[5] 围绕计算时延、通信量以及通信时延三个指标进行了对比，测试结果对比如图 6～图 9 所示.



(a) Alice计算SHA-1算法时延测试对比图　(b) Bob计算SHA-1算法时延测试对比图

(c) Alice计算SHA256算法时延测试对比图　(d) Bob计算SHA256算法时延测试对比图

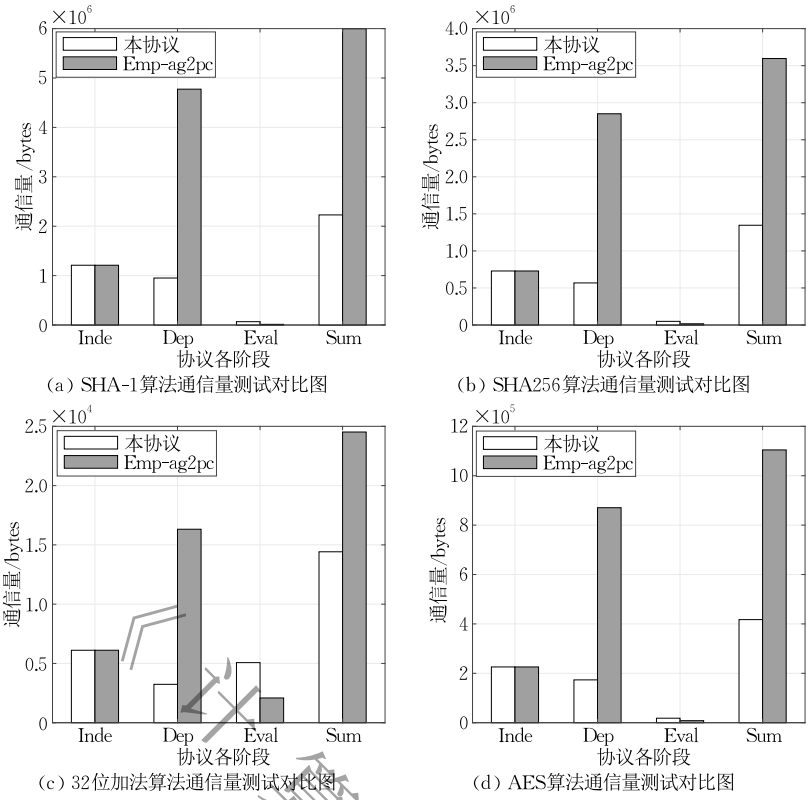
图 6 SHA-1/SHA256 算法计算时延测试结果对比图



(a) Alice计算32位加法算法时延测试对比图　(b) Bob计算32位加法算法时延测试对比图

(c) Alice计算AES算法时延测试对比图　(d) Bob计算AES算法时延测试对比图

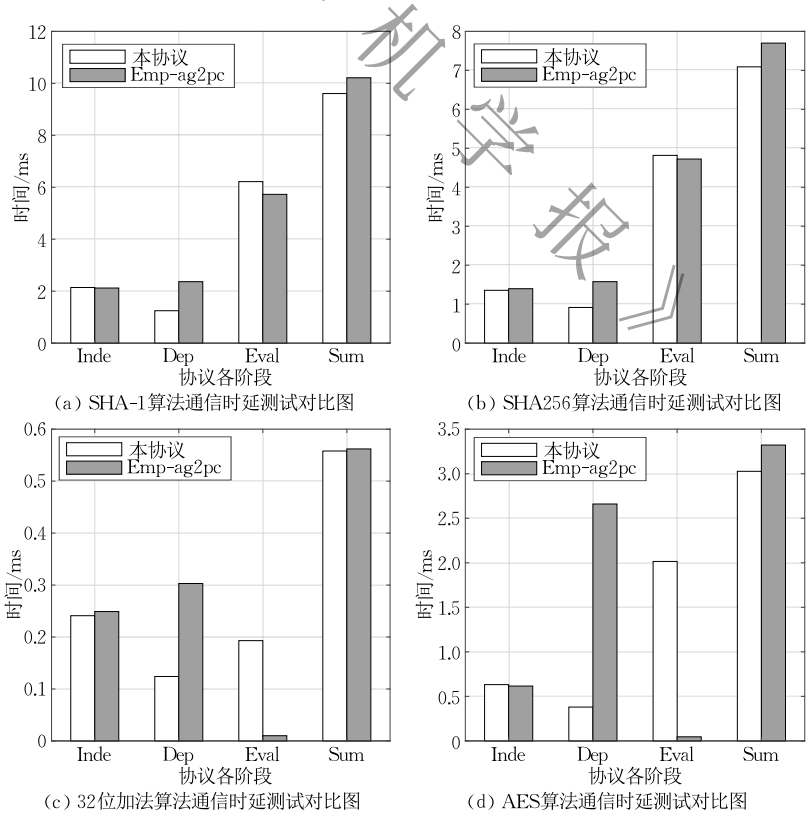
图 7 32 位加法/AES 算法计算时延测试结果对比图



(a) SHA-1算法通信量测试对比图　(b) SHA256算法通信量测试对比图

(c) 32位加法算法通信量测试对比图　(d) AES算法通信量测试对比图

图 8　通信量测试结果对比图(10 组平均值)



(a) SHA-1算法通信时延测试对比图　(b) SHA256算法通信时延测试对比图

(c) 32位加法算法通信时延测试对比图　(d) AES算法通信时延测试对比图

图 9　通信时延测试结果对比图(10 组平均值)

在测试过程中，记两个参与方分别为 Alice 和 Bob. 在本协议层中，Alice 负责前半布尔电路的混淆，以及后半混淆电路的恢复与分析工作，Bob 负责前半混淆电路的恢复与分析，以及后半布尔电路的混淆工作. 在 Emp-toolkit 两方协议中，Alice 负责布尔电路的混淆生成工作，Bob 负责完整混淆电路

的恢复与分析工作.测试结果分别记录了分阶段的时延(通信量)及总时延(总通信量),包括预生成阶段,以及协议执行的函数无关预处理阶段、函数相关预处理阶段、输入预处理阶段、电路分析阶段以及输出阶段这 5 个阶段.

在对比图中,预生成阶段 Pre 包括乘法三元组的生成阶段、信息验证码的生成与验证阶段. Inde 指函数无关预处理阶段,Dep 指函数相关预处理阶段以及输入预处理阶段,Eval 指电路分析阶段以及输出阶段.

为了实现合作式流程设计方案,在预生成阶段以前,本文提出的协议还包括额外的电路划分阶段,在参与方已知算法的布尔电路结构以后,可以在本地完成前半电路和后半电路的划分,为了分摊两方的计算压力,两个子电路中与门的数量大致相同.电路划分阶段可以在协议执行之前任意时刻完成,两方无需额外交互,不对协议执行造成影响,因此,实验测试未包含电路划分阶段的计算时延.

(1)计算时延测试

在计算时延测试中,测试了协议五阶段的时延以及总时延.对比图中的 Sum 包含协议五阶段总计算时延.如图 6、图 7 所示,与 Emp-toolkit 相比,四种算法的 Pre 阶段、Inde 阶段和 Dep 阶段的平均计算时延分别得到了不同程度的优化.由于计算复杂度的增长,Eval 阶段的平均计算时延有较明显增长.从算法总计算时延来看,除了在 SHA-1 算法中,两方的平均计算时延增长了约 5%,在 SHA256、32 位加法和 AES 算法中,两方的平均计算时延分别优化了约 5%、22%和 8%.

(2)通信量测试

在通信量测试中,由于乘法三元组的生成个数与布尔电路结构中的与门数量相关,Pre 阶段与文献[27]一致.因此,仅测试了 Inde 阶段、Dep 阶段和 Eval 阶段.Sum 阶段指这三个阶段的总通信量.

如图 8 所示,在计算 SHA-1 等四种算法的时候,与 Emp-toolkit 相比,Inde 阶段和 Eval 阶段的通信量略有增长;由于可验证混淆电路分享方案通信复杂度的降低,Dep 阶段的通信量有大幅度降低.三阶段的通信量优化了约 40%～60%.

(3)通信时延测试

在通信时延测试中,与通信量测试同理,也仅测试了 Inde 阶段、Dep 阶段和 Eval 阶段这三个阶段的通信时延.Sum 指这三个阶段的总通信时延.

如图 9 所示,在计算 SHA-1 等四种算法的时候,与 Emp-toolkit 相比,Inde 阶段平均通信时延略有增长;由于可验证混淆电路分享方案通信复杂度的降低,Dep 阶段的平均通信时延有明显降低;由于合作式流程设计方案中前半电路计算完成后需要两方交互后再继续后半电路的计算,Eval 阶段的平均通信时延明显增长.从三阶段总通信时延来看,本文提出协议的平均通信时延优化了 1%～9%.

6 结束语

本文针对基于混淆电路的安全两方计算协议通信复杂度过高的问题,基于 Three-Halves 混淆电路技术[15]和可验证随机比特分享技术[27]设计了一个新的可验证混淆电路分享方案,改进了协议流程设计,提出了一个新的安全两方计算协议.本文证明了该协议在恶意敌手模型下是安全的,优化了协议实例的性能.

致 谢 感谢所有作者为本文所付出的努力,本文四个作者贡献一致.

参 考 文 献

[1] Zhang Zheng, Zhang Fang-Guo. Garbled circuits and indistinguishability obfuscation. Journal of Cryptologic Research, 2019, 6(5): 541-560(in Chinese)
(张正,张方国. 混淆电路与不可区分混淆. 密码学报, 2019, 6(5): 541-560)

[2] Shamir A. How to share a secret. Communications of the Association for Computing Machinery, 1979, 22(11): 612-613

[3] Blakley G R. Safeguarding cryptographic keys//Proceedings of the AFIPS 1979 National Computer Conference (AFIPS). New York, USA, 1979: 313-317

[4] Yao C. How to generate and exchange secrets//Proceedings of the 27th Annual Symposium on Foundations of Computer Science (FOCS). Toronto, Canada, 1986: 162-167

[5] Emp-toolkit. https://github.com/emp-toolkit

[6] Goldreich O, Micali S, Wigderson A. How to play any mental game or a completeness theorem for protocols with honest majority//Proceedings of 19th Annual Symposium on Theory of Computing (STOC). New York, USA, 1987: 218-229

[7] Ben-Or M, Goldwasser S, Wigderson A. Completeness theorems for non-cryptographic fault-tolerant distributed//Proceedings of the 20th Annual Symposium on Theory of Computing (STOC). Chicago, USA, 1988: 1-10

[8] Demmler D, Schneider T, Zohner M. ABY: A framework for efficient mixed-protocol secure two-party computation//Proceedings of the 22nd Annual Network and Distributed System Security Symposium (NDSS). San Diego, USA, 2015: 1-15

[9] Patra A, Schneider T, Suresh A, et al. ABY2.0: Improved mixed-protocol secure two-party computation//Proceedings of the 30th Annual USENIX Security Symposium (USENIX). Boston, USA, 2021: 2165-2182

[10] Beaver D, Micali S, Phillip R. The round complexity of secure protocols//Proceedings of the 22nd ACM Symposium on Theory of Computing (STOC). Baltimore, USA, 1990: 503-513

[11] Naor M, Pinkas B, Sumner R. Privacy preserving auctions and mechanism design//Proceedings of the 1st ACM Conference on Electronic Commerce. Denver, USA, 1999: 129-139

[12] Kolesnikov V, Schneider T. Improved garbled circuit: Free-XOR gates and applications//Proceedings of the 35th International Colloquium on Automata, Languages, and Programming (ICALP). Reykjavik, Iceland, 2008: 486-498

[13] Pinkas B, Schneider T, Smart N P, et al. Secure two-party computation is practical//Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT). Tokyo, Japan, 2009: 250-267

[14] Zahur S, Rosulek M, Evans D. Two halves make a whole-reducing data transfer in garbled circuits using half gates//Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Sofia, Bulgaria, 2015: 220-250

[15] Rosulek M, Roy L. Three Halves make a whole? Beating the half-gates lower bound for garbled circuits//Proceedings of the 41st Annual International Cryptology Conference (CRYPTO). Virtual Event, USA, 2021: 94-124

[16] Heath D, Kolesnikov V. Stacked garbling-garbled circuit proportional to longest execution path//Proceedings of the 40th Annual International Cryptology Conference (CRYPTO). Santa Barbara, USA, 2020: 763-792

[17] Heath D, Kolesnikov V. LogStack: Stacked garbling with $O(b\log b)$ computation//Proceedings of the 40th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Zagreb, Croatia, 2021: 3-32

[18] Lindell Y, Pinkas B. An efficient protocol for secure two-party computation in the presence of malicious adversaries//Proceedings of the 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Barcelona, Spain, 2007: 52-78

[19] Lindell Y, Pinkas B. Secure two-party computation via cut-and-choose oblivious transfer//Proceedings of the 8th Theory of Cryptography Conference (TCC). Providence, USA, 2011: 329-346

[20] Lindell Y. Fast cut-and-choose based protocols for malicious and covert adversaries//Proceedings of the 33rd Annual International Cryptology Conference (CRYPTO). Santa Barbara, USA, 2013: 1-17

[21] Araki T, Barak A, Furukawa J, et al. Optimized honest-majority MPC for malicious adversaries-breaking the 1 billion-gate per second barrier//Proceedings of the 38th IEEE Symposium on Security and Privacy (S&P). San Jose, USA, 2017: 843-862

[22] Afshar A, Mohassel P, Pinkas B, et al. Non-interactive secure computation based on cut-and-choose//Proceedings of the 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Copenhagen, Denmark, 2014: 387-404

[23] Zhao Qing-Song, Zeng Qing-Kai, Liu Xi-Meng, et al. Verifiable computation using re-randomizable garbled circuits. Journal of Software, 2019, 30(2): 399-415(in Chinese)
(赵青松，曾庆凯，刘西蒙等. 基于可重随机化混淆电路的可验证计算. 软件学报，2019，30(2)：399-415)

[24] Wang X, Malozemoff A J, Katz J. Faster two-party computation secure against malicious adversaries in the single-execution setting//Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Paris, France, 2017: 399-424

[25] Mohassel P, Rosulek M. Non-interactive secure 2PC in the offline/online and batch settings//Proceedings of the 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT). Paris, France, 2017: 425-455

[26] Kolesnikov V, Nielsen J B, Rosulek M, et al. DUPLO: Unifying cut-and-choose for garbled circuits//Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS). Dallas, USA, 2017: 3-20

[27] Wang X, Ranellucci S, Katz J. Authenticated garbling and efficient maliciously secure two-party computation//Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS). Dallas, USA, 2017: 21-37

[28] Katz J, Ranellucci S, Rosulek M, et al. Optimizing authenticated garbling for faster secure two-party computation//Proceedings of the 38th Annual International Cryptology Conference (CRYPTO). Santa Barbara, USA, 2018: 365-391

[29] Yang K, Wang X, Zhang J. More efficient MPC from improved triple generation and authenticated garbling//Proceedings of the 27th ACM Conference on Computer and Communications Security (CCS). Virtual Event, USA, 2020: 1627-1646

[30] Wang X, Ranellucci S, Katz J. Global-scale secure multiparty computation//Proceedings of the 24th ACM Conference on Computer and Communications Security (CCS). New York, USA, 2017: 39-56

[31] Lindell Y, Pinkas B. A proof of security of Yao's protocol for two-party computation. Journal of Cryptology, 2009, 22(2): 161-188

[32] Nielsen J B, Nordholt P S, Orlandi C, et al. A new approach to practical active-secure two-party computation//Proceedings of the 32nd Annual International Cryptology Conference (CRYPTO). Santa Barbara, USA, 2012: 681-700

[33] Beaver D. Efficient multiparty protocols using circuit randomization//Proceedings of the 11th Annual International Cryptology Conference (CRYPTO). Santa Barbara, USA, 1991: 420-432

附录 1.

合作式流程设计方案的安全性. 如果混淆电路方案 G, 不经意传输方案 OT 以及基于可验证随机比特分享的掩码技术是安全的,承诺方案 Com 具有隐藏性和绑定性,那么合作式协议流程设计方案具有正确性、隐私性、不经意性和可认证性.

(1) 正确性. 在第 4.1.2 节描述流程的步骤 2 中,两个参与方根据混淆方案 G 分别生成前半电路的混淆电路 $\boldsymbol{G}_1$ 和后半电路的混淆电路 $\boldsymbol{G}_2$. 根据定义 1 的描述,如果 G 是安全的,那么混淆电路 $\boldsymbol{G}_1$ 和混淆电路 $\boldsymbol{G}_2$ 具有正确性. 也就是说,在流程步骤 4 中,P_B 分析 $\boldsymbol{G}_1$ 得到的前半电路输出是正确的. P_B 在分析完 $\boldsymbol{G}_1$ 之后,本地完成前半电路输出值与后半电路输入线路标签的对应工作,将后半电路输入值对应标签发送给 P_A. 因为承诺方案 Com 具有隐藏性和绑定性,所以 P_A 可以收到正确的后半电路输入线路值对应的标签集合. 在经过第 4.1.2 节的步骤 5 之后,两个参与方可以得到完整电路的正确输出结果.

(2) 隐私性. 安全的混淆电路方案 G 具有隐私性,因此前半电路的混淆电路 $\boldsymbol{G}_1$ 和后半电路混淆电路 $\boldsymbol{G}_1$ 的生成与分析过程分别是安全的. 本方案中的掩码技术是基于可验证随机比特分享份额技术实现的,任意一方不知道线路的完整掩码,仅持有掩码份额. 因此,基于定义 4 的安全性[27],如果不公开中间线路的掩码值,P_B 在完成混淆电路 $\boldsymbol{G}_1$ 分析工作后,是无法通过前半电路输出掩饰值恢复真值的.

(3) 不经意性. 安全的混淆电路方案 G 具有不经意性是通过调用不经意传输方案 OT 实现的. 在第 4.1.2 节流程步骤 2 中,P_A 仅发送各输入线路的一个标签组成的标签集合给 P_B. 在步骤 2 之后,P_B 持有混淆电路 $\boldsymbol{G}_1$ 和与各输入线路对应的 1 个标签. 因为不经意传输方案 OT 是安全的,具有隐私性和不经意性,P_B 不会知道有关输入线路的另一个标签的任何信息,P_A 也不会知道有关 P_B 输入的任何信息. 在步骤 3 中,P_B 能够分析得到前半混淆电路 $\boldsymbol{G}_1$ 的输出. 但是,输出是以掩饰值形式存在的,P_B 不会获得完整电路 C 的中间计算信息. P_B 将对应后半电路 $\boldsymbol{G}_2$ 各输入线路的 1 个标签组成的标签集合发送给 P_A,P_A 不会知道有关完整电路的中间输入线路另一个标签的任何信息.

(4) 可认证性. 因为安全的混淆方案 G 具有可认证性,所以两个参与方在第 4.1.2 节的步骤 2 生成的混淆电路 $\boldsymbol{G}_1$ 和 $\boldsymbol{G}_2$ 具有可认证性. 参与方在步骤 4～5 分别分析得到的输出要么是正确的子电路输出值,要么是无效输出. 因此,在经过步骤 5 之后,就可以判断参与方是否是通过分析 $\boldsymbol{G}_1$ 和 $\boldsymbol{G}_2$ 获得的完整电路输出结果.

ZHANG Zong-Yang, Ph. D., associate professor. His research interests include blockchain and cryptography.

LIU Xiang-Yu, M.S. Her current research interest is cryptography.

LI Wei-Han, Ph.D. candidate. His current research interests include zero-knowledge proofs and blockchain.

CHEN Lao, Ph. D. candidate, assistant researcher. His current research interests include cybersecurity and blockchain.

Background

With the continuous improvement of informatization, new technologies, such as big data, artificial intelligence and blockchain, have been applied to all walks of life. These new technologies rely on data sharing across fields and enterprises. This way promotes the rapid development of industries such as finance, medical care, and commerce, but threatens the security of private data. How to achieve efficient data sharing and data interoperability while ensuring security is a problem to be solved in the current privacy protection field.

Secure multi-party computation which is one of high-level

cryptography provides a new idea to solve this problem. The secure two-party computation is the basis for constructing secure multi-party computation protocols. And it can also solve security problems of privacy data well in two-party applications, such as genome sequence alignment and pattern matching. Thus, secure two-party computation is one of the current research hotspots.

Secure two-party computation based on Yao's garbled circuit is one of the main-steam research for secure two-party computation, resulting in constant-round. Compared with other protocols, secure two-party computation protocol in Yao's world has high communication complexity.

This paper proposed an authenticated garbling scheme based on Three-Halves garbled circuit. This scheme continues the idea of combining garbled circuit with secret sharing. This scheme is secure in malicious adversary model while implementing a 25% improvement in the communication complexity. Then, proposed a cooperative process for secure two-party protocol. This scheme distributes the computation pressure to the two parties based on Point-and-Permute technology. In addition, this scheme helps the honest party to detect the malicious adversary in advance. Finally, proposed a secure two-party computation protocol by using the authenticated garbling scheme, the cooperative process scheme and other cryptographic technologies. This protocol is secure under malicious adversary model. Compared with secure two-party computation protocol proposed in Emp-toolkit, this protocol results in a 1%～9% improvement in the communication delay, a 5%～22% improvement in the computation delay and a 40%～60% improvement in the communication cost.

The research in this paper is partially supported by the National Key Research and Development Program of China (No. 2021YFB3100400), the National Natural Science Foundation of China (Nos. 61972017, 72031001, 61972310), the Beijing Natural Science Foundation (No. M22038, 4202037), the Fundamental Research Funds for the Central Universities (No. YWF-22-L-1039), and the Yunnan Key Laboratory of Blockchain Application Technology (YNB202101).