

Cut-and-Choose 双向不经意传输*

赵川, 蒋瀚, 魏晓超, 徐秋亮

(山东大学 计算机科学与技术学院, 山东 济南 250101)

通讯作者: 蒋瀚, E-mail: jianghan@sdu.edu.cn



摘要: 不经意传输作为现代密码学的一个基本工具,在安全协议的研究中起着重要作用.近年来,许多功能性更强的不经意传输变种被提出来,以适应不同的需求和环境.提出一个不经意传输变种,称为 cut-and-choose 双向不经意传输.基于同态加密给出该原语的一轮高效协议构造,且在半诚实模型下形式化证明了该协议的安全性.将 cut-and-choose 双向不经意传输运用到基于 cut-and-choose 技术的安全协议(尤其是安全两方计算)中,可以更具模块化地描述协议高层框架,降低协议交互轮数.此外,作为信息安全领域的一个底层基本工具,该原语本身也具有独立的研究意义.

关键词: 不经意传输;cut-and-choose;同态加密;半诚实模型;安全两方计算

中图法分类号: TP309

中文引用格式: 赵川,蒋瀚,魏晓超,徐秋亮.Cut-and-Choose 双向不经意传输.软件学报,2017,28(2):352-360. <http://www.jos.org.cn/1000-9825/5019.htm>

英文引用格式: Zhao C, Jiang H, Wei XC, Xu QL. Cut-and-Choose bilateral oblivious transfer. Ruan Jian Xue Bao/Journal of Software, 2017, 28(2): 352-360 (in Chinese). <http://www.jos.org.cn/1000-9825/5019.htm>

Cut-and-Choose Bilateral Oblivious Transfer

ZHAO Chuan, JIANG Han, WEI Xiao-Chao, XU Qiu-Liang

(School of Computer Science and Technology, Shandong University, Ji'nan 250101, China)

Abstract: Oblivious transfer is a fundamental tool in modern cryptography. It plays an important role in the research of security protocols. In recent years, many oblivious transfer variants with more powerful functionalities are proposed to fit in different kinds of requirements and scenarios. In this paper, a new oblivious transfer variant, called cut-and-choose bilateral oblivious transfer, is proposed. Based on homomorphic encryption, an efficient one-round protocol of this primitive is constructed along with rigorous security proof in semi-honest model. When applied in security protocols based on cut-and-choose technique (especially in secure two-party computation), cut-and-choose bilateral oblivious transfer enables a more modular high-level description of the protocol framework, and also reduces the round complexity of the protocols. Besides, as a basic tool in the information security area, this primitive itself is of independent research interest.

Key words: oblivious transfer; cut-and-choose; homomorphic encryption; semi-honest model; secure two-party computation

随着物联网以及移动互联网的快速发展,全球数据量呈指数级增长,大数据时代已经到来.在大数据产业蓬勃发展的同时,个人隐私数据泄露事件层出不穷,引起了人们的广泛关注.确保数据安全性、设计安全实用的多方协议成为亟待解决的问题.不经意传输(oblivious transfer)作为许多典型安全协议的核心基本工具,无论是在理论研究还是在应用研究中都担任了重要角色.

* 基金项目: 国家自然科学基金(61572294, 61173139)

Foundation item: National Natural Science Foundation of China (61572294, 61173139)

收稿时间: 2015-03-20; 采用时间: 2015-12-22; jos 在线出版时间: 2016-01-11

CNKI 网络优先出版: 2016-01-12 11:22:20, <http://www.cnki.net/kcms/detail/11.2560.TP.20160112.1122.002.html>

不经意传输是一个两方传输任务,其中一个参与方称为发送方,提供多个字符串;另一个参与方称为接收方,提供一个索引以选取其中 1 个或多个.任务执行完成后,接收方获得其索引所对应的字符串,对其他字符串一无所知;发送方没有输出,且不知道接收方获得了哪些字符串.自 Rabin 在 1981 年提出不经意传输以来^[1],针对该领域的研究就受到了密码学研究者的广泛关注.在后续的研究中,有很多工作重点关注设计更高效或更安全的不经意传输协议^[2-6].另外,也有很多工作提出了功能性更强的不经意传输变种,用来增强调用不经意传输的外层协议的安全性或提高协议效率,如承诺不经意传输^[7,8]、不经意传输扩展^[9-11]等.值得注意的是,Lindell 等人 TCC 2011 会议上提出了 cut-and-choose 不经意传输^[12],将其应用在基于混乱电路和 cut-and-choose 技术的安全两方计算协议中,高效地解决了 Kiraz 等人指出的选择性失败攻击问题^[13].然而,该变种仅能用来传输单个参与方的混乱密钥.为了传输另一个参与方的混乱密钥以完成混乱电路检测和计算,协议还需要参与方进行多轮交互.显然,基于该原语的协议交互复杂度过高.

受 cut-and-choose 不经意传输的启发,本文提出 cut-and-choose 双向不经意传输这一新原语.与 cut-and-choose 不经意传输相比,新原语具有更强大的功能,可以在安全两方计算中一次性传输两个参与方的所有相关密钥.除了能够解决选择性失败攻击问题以外,该原语还具有以下 3 个优势:

- (1) 降低外层协议的交互轮数,这在协议双方交互受限的场景中是非常关键的;
- (2) 更具模块化、更清晰地描述协议过程,使协议更易于理解和分析;
- (3) 可用于其他 cut-and-choose 场景中,其本身具有独立的研究意义.

基于同态加密,本文构造出仅需 1 轮交互的 cut-and-choose 双向不经意传输协议,并基于标准的安全性定义,在半诚实模型下形式化证明该协议的安全性.

本文第 1 节回顾一些预备知识.第 2 节给出新原语 cut-and-choose 双向不经意传输的功能函数定义,并基于同态加密构造具体协议,在半诚实模型下给出严格的安全性证明.第 3 节阐述该原语在基于 cut-and-choose 技术的安全两方计算中的应用.第 4 节总结全文,并展望后续的研究工作.

1 预备知识

本节介绍相关预备知识,包括不经意传输、cut-and-choose 不经意传输、同态加密方案以及半诚实模型下的安全性定义.

1.1 不经意传输

一个标准的 2 取 1 不经意传输是一个两方功能函数,其中一个参与方称为发送方 S ,输入为两个 n 比特字符串 y_0, y_1 ,另一个参与方称为接收方 R ,输入为一个选择比特 $\tau \in \{0, 1\}$.该任务执行完成后, R 获得 y_τ ,而对 $y_{1-\tau}$ 一无所知; S 没有输出,且不知道 R 获得了 y_0, y_1 中的哪一个.这些性质可以由下面的功能函数 \mathcal{F}_{ot} 给出.

功能函数 \mathcal{F}_{ot} .

输入:

-- S 输入 $y_0, y_1 \in \{0, 1\}^n$.

-- R 输入 $\tau \in \{0, 1\}$.

输出:

-- S 输出 \perp (表示为空).

-- R 输出 y_τ .

1.2 Cut-and-Choose 不经意传输

Lindell 等人提出的 cut-and-choose 不经意传输是在 2 取 1 不经意传输的基础上,为 R 引入一个 cut-and-choose 指示比特 j ,以便让 R 选择是要获得 S 的所有两个输入值,还是只需要获得其中一个.我们将 Lindell 等人提出的该原语做了适当简化,将其写成下面的功能函数 \mathcal{F}_{cot} 以便表述.

功能函数 \mathcal{F}_{cot} .

输入:

--S 输入 $y_0, y_1 \in \{0, 1\}^n$.

--R 输入 (j, τ) , 其中 $j \in \{0, 1\}$ 为 cut-and-choose 指示比特, $\tau \in \{0, 1\}$ 为 R 的选择比特.

输出:

--S 输出 \perp .

--R 输出 z :

当 $j=1$ 时, z 为 (y_0, y_1) ;

当 $j=0$ 时, z 为 y_τ .

1.3 同态加密方案

记某公钥加密方案为 $M=(Gen, Enc, Dec)$, 其中, Gen 表示密钥生成算法, Enc 表示加密算法, Dec 表示解密算法. 将使用公钥 pk 对明文 m 的加密记为 $Enc_{pk}(m)$. 直观上, 给定两个密文 $c_1=Enc_{pk}(m_1)$ 和 $c_2=Enc_{pk}(m_2)$, 若在不知道对应私钥和明文的前提下可以高效计算 $c_1 \cdot c_2$ 满足 $c_1 \cdot c_2=Enc_{pk}(m_1+m_2)$, 则称该公钥加密方案具有加法同态性. 这里要求计算 $c_1 \cdot c_2$ 的结果是对 m_1+m_2 的一次随机加密. 正式地, 有如下定义:

定义 1(加法同态加密). 对于一个公钥加密方案 $M=(Gen, Enc, Dec)$, 如果对于任意的 n , 任意 $Gen(1^n)$ 的输出 (pk, sk) , 满足:

$$\{pk, c_1=Enc_{pk}(m_1), c_1 \cdot Enc_{pk}(m_2)\} \cong \{pk, Enc_{pk}(m_1), Enc_{pk}(m_1+m_2)\},$$

其中, 等号 \cong 表示计算不可区分, 则 M 是一个加法同态加密方案.

1.4 安全性定义

不经意传输作为一类具体的安全两方计算问题, 其安全性可由安全两方计算的标准安全模型来定义. 本文使用半诚实模型下的安全性定义^[14]来刻画所构造的 cut-and-choose 双向不经意传输协议的安全性.

定义 2(半诚实模型下的安全性). 令 $f(\cdot, \cdot)=(f_1, f_2)$ 为任意 $\{0, 1\}^* \times \{0, 1\}^* \rightarrow \{0, 1\}^* \times \{0, 1\}^*$ 多项式时间功能函数, π 为一个计算 $f(\cdot, \cdot)$ 的两方协议. 给定输入 (x, y) (参与方 P_1 的输入为 x , P_2 的输入为 y) 和安全参数 n , $P_i (i \in \{1, 2\})$ 在协议 π 中的视图记为 $view_i^\pi(x, y, n)=(w, r^i, m_1^i, \dots, m_t^i)$, 其中, $w \in \{x, y\}$, r^i 为 P_i 的内部随机带, m_j^i 为 P_i 接收到的第 j 条消息; P_i 的输出记为 $output_i^\pi(x, y, n)$. 另外, 记双方的联合输出为 $output^\pi(x, y, n)=(output_1^\pi(x, y, n), output_2^\pi(x, y, n))$.

如果下列条件满足, 则协议 π 在半诚实模型下安全计算功能函数 $f(\cdot, \cdot)$.

首先, 要求满足正确性, 即满足:

$$\{output^\pi(x, y, n)\}_{x, y, n} \cong \{f(x, y)\}_{x, y}.$$

其次, 要求存在概率多项式时间算法 \mathcal{S}_1 和 \mathcal{S}_2 , 满足:

$$\{\mathcal{S}_1(1^n, x, f_1(x, y))\}_{x, y, n} \cong \{view_1^\pi(x, y, n)\}_{x, y, n},$$

$$\{\mathcal{S}_2(1^n, y, f_2(x, y))\}_{x, y, n} \cong \{view_2^\pi(x, y, n)\}_{x, y, n},$$

其中, $x, y \in \{0, 1\}^*$, 满足 $|x|=|y|$, $n \in \mathbf{N}$.

2 Cut-and-Choose 双向不经意传输

这一节首先介绍并形式化定义 cut-and-choose 双向不经意传输功能函数, 然后基于同态加密给出其协议构造, 并形式化证明该协议满足半诚实模型下的标准安全性定义(定义 2).

2.1 功能函数

我们说标准的不经意传输和 cut-and-choose 不经意传输实现的功能都是单向的. 也就是说, S 输入两个值等待 R 进行选择, 而其本身并没有主动选择权. 如果在 cut-and-choose 不经意传输的基础上, 为 S 增加两个字符串 x_0 和 x_1 , 使其能够通过一个新的选择比特 σ 对 R 接收到这两个字符串中的哪一个拥有决定权, 就构成了 cut-and-

choose 双向不经意传输.当然,增加的这两个字符串被 R 接收到 1 个还是 2 个,依然由 R 的 cut-and-choose 指示比特 j 来决定.需要注意的是,引入选择比特 σ 会带来以下问题:当 j 为 0 时, R 必须知道 x_0 和 x_1 这两个值应该获得哪一个.如果按照正常顺序 x_0, x_1 发送这两个值,则 R 可以获得 σ 的值,这与 σ 需要保密相矛盾.因此,需要引入一个置换比特 b 来对 x_0 和 x_1 的位置进行随机置换,从而达到隐藏 σ 的目的.这样, R 就可以在不知道 σ 的情况下获得 x_σ . 置换比特的引入会对接收方的输出造成影响:当 cut-and-choose 指示比特 j 为 1 时,接收方除了获得被置换位置后的两个字符串 x_b, x_{1-b} 之外,还需要拿到 b 的值,以获得正常顺序的 x_0, x_1 ; 当 cut-and-choose 指示比特 j 为 0 时,接收方除了获得由发送方指定应该获得的 x_σ 之外,实际上也获得了 x_σ 的位置信息 $\sigma \oplus b$, 即当获得 x_b, x_{1-b} 中的第 1 个时,说明 $\sigma \oplus b$ 为 0, 反之则说明 $\sigma \oplus b$ 为 1. 该功能可以由下面的功能函数 \mathcal{F}_{ccbot} 给出.

功能函数 \mathcal{F}_{ccbot} .

输入:

--S 输入 $(x_0, x_1, y_0, y_1, b, \sigma)$, 其中 $x_0, x_1, y_0, y_1 \in \{0, 1\}^n$, $b \in \{0, 1\}$ 为置换比特, $\sigma \in \{0, 1\}$ 为 S 的选择比特.

--R 输入 (j, τ) , 其中 $j \in \{0, 1\}$ 为 cut-and-choose 指示比特, $\tau \in \{0, 1\}$ 为 R 的选择比特.

输出:

--S 输出 \perp .

--R 输出 z :

当 $j=1$ 时, z 为 $(x_b, x_{1-b}, 1-b, y_0, y_1)$;

当 $j=0$ 时, z 为 $(x_\sigma, y_\tau, \sigma \oplus b)$, 其中 $\sigma \oplus b$ 指示 x_σ 的位置信息.

2.2 协议构造

Cut-and-choose 双向不经意传输可以基于同态加密构造.主要思想是利用加密方案的同态性,将接收方 cut-and-choose 指示比特的密文与发送方的输入进行特定运算.具体构造请见协议 1.

协议 1. Cut-and-Choose 双向不经意传输协议.

输入:发送方 S 输入 $(x_0, x_1, y_0, y_1, b, \sigma)$;接收方 R 输入 (j, τ) .

辅助输入:安全参数 1^n ;满足定义 1 的选择明文攻击(CPA)安全的加法同态加密方案 $M=(Gen, Enc, Dec)$.

协议过程:

步骤 1. R 将 τ 编码为两个比特 $\tau_0 \tau_1$, 其中 $\tau_\tau=1, \tau_{1-\tau}=0$. 具体来说,如果 $\tau=1$, 则编码为 $\tau_0 \tau_1=01$; 如果 $\tau=0$, 则编码为 $\tau_0 \tau_1=10$. 另外, R 将 j 编码为两个比特 $j_0 j_1=j_0$. 然后, R 生成一组密钥 $(pk, sk) \leftarrow Gen(1^n)$, 公开公钥 pk , 并用 pk 对 $(j_0 j_{\tau_0} + \tau_0 j_{\tau_1} + \tau_1)$ 进行加密, 将加密后得到的密文三元组 $(Enc_{pk}(j_0), Enc_{pk}(j_{\tau_0} + \tau_0), Enc_{pk}(j_{\tau_1} + \tau_1))$ 发送给 S .

步骤 2. S 将 σ 编码为两个比特 $\sigma_0 \sigma_1$, 其中 $\sigma_\sigma=1, \sigma_{1-\sigma}=0$. 具体来说,如果 $\sigma=1$, 则编码为 $\sigma_0 \sigma_1=01$; 如果 $\sigma=0$, 则编码为 $\sigma_0 \sigma_1=10$. 然后, S 利用 R 的公钥 pk 计算 $(Enc_{pk}(j_1), Enc_{pk}(\sigma_0), Enc_{pk}(\sigma_1), Enc_{pk}(b))$, 并计算密文五元组:

$$\begin{aligned} w_b &= (Enc_{pk}(j_{\sigma_b}) \cdot Enc_{pk}(\sigma_b))^{x_b}, \\ w_{1-b} &= (Enc_{pk}(j_{\sigma_{1-b}}) \cdot Enc_{pk}(\sigma_{1-b}))^{x_{1-b}}, \\ w_2 &= (Enc_{pk}(j_b) \cdot Enc_{pk}(b))^{1-b}, \\ w_3 &= (Enc_{pk}(j_{\tau_0} + \tau_0))^{y_0}, \\ w_4 &= (Enc_{pk}(j_{\tau_1} + \tau_1))^{y_1}. \end{aligned}$$

计算完成后,将密文五元组 $(w_b, w_{1-b}, w_2, w_3, w_4)$ 发送给 R .

步骤 3. R 用私钥 sk 对接收到的密文五元组进行解密, 得到明文五元组 $(u_b, u_{1-b}, u_2, u_3, u_4)$:

- 当 $j=1$ 时, 令 $(u_b, u_{1-b}, u_2, u_3, u_4) = (x_b, x_{1-b}, 1-b, y_0, y_1)$;
- 当 $j=0$ 时, 忽略 u_2 的值, 令 u_b, u_{1-b} 中不为 0 的值为 x_σ , 即 u_b, u_{1-b} 中的第 $\sigma \oplus b + 1$ 个; 令 u_3, u_4 中的第 $\tau + 1$ 个为 y_τ , 得到输出 $(x_\sigma, y_\tau, \sigma \oplus b)$.

2.3 安全性证明

定理 1. 假定 $M=(Gen,Enc,Dec)$ 是一个满足定义 1 的 CPA 安全的加法同态加密方案,则协议 1 在定义 2 下安全计算功能函数 \mathcal{F}_{ccbot} .

证明:令 \mathcal{F}_{ccbot} 为 cut-and-choose 双向不经意传输功能函数, π 为计算 \mathcal{F}_{ccbot} 的协议 1,根据定义 2,如果下列条件满足,我们就说协议 π 在半诚实模型下安全计算 \mathcal{F}_{ccbot} .

首先满足正确性,即

$$\{output^\pi((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau),n)\}_{x_0,x_1,y_0,y_1,b,\sigma,j,\tau,n} \cong \{\mathcal{F}_{ccbot}((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau))\}_{x_0,x_1,y_0,y_1,b,\sigma,j,\tau,n}$$

其次,存在概率多项式时间算法 \mathcal{S}_1 和 \mathcal{S}_2 满足:

$$\{\mathcal{S}_1(1^n,(x_0,x_1,y_0,y_1,b,\sigma))\}_{x_0,x_1,y_0,y_1,b,\sigma,n} \cong \{view_1^\pi((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau),n)\}_{x_0,x_1,y_0,y_1,b,\sigma,j,\tau,n} \quad (1)$$

$$\{\mathcal{S}_2(1^n,(j,\tau,z))\}_{j,\tau,z,n} \cong \{view_2^\pi((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau),n)\}_{x_0,x_1,y_0,y_1,b,\sigma,j,\tau,n} \quad (2)$$

其中 $x_0,x_1,y_0,y_1 \in \{0,1\}^n, z \in \{0,1\}^*, b,\sigma,j,\tau \in \{0,1\}, n \in \mathbf{N}$.

下面我们对协议 1 的安全性给出形式化证明.

首先,根据 cut-and-choose 指示比特 j 的取值分两种情况对协议 1 的正确性进行证明.

情况 1) 当 $j=1$ 时, $\mathcal{F}_{ccbot}((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau))$ 的输出为 $(x_b,x_{1-b},1-b,y_0,y_1)$,我们证明在此情况下,协议 1 的输出也是如此.

在协议 1 中,当 $j=1$ 时 $j_0=1,j_1=0$,有 $j_b+b=1, b \in \{0,1\}$,所以,

$$\begin{aligned} w_b &= (Enc_{pk}(j_{\sigma_b}) \cdot Enc_{pk}(\sigma_b))^{x_b} = Enc_{pk}(x_b \cdot (j_{\sigma_b} + \sigma_b)) = Enc_{pk}(x_b), \\ w_{1-b} &= (Enc_{pk}(j_{\sigma_{1-b}}) \cdot Enc_{pk}(\sigma_{1-b}))^{x_{1-b}} = Enc_{pk}(x_{1-b} \cdot (j_{\sigma_{1-b}} + \sigma_{1-b})) = Enc_{pk}(x_{1-b}), \\ w_2 &= (Enc_{pk}(j_b) \cdot Enc_{pk}(b))^{1-b} = Enc_{pk}((1-b) \cdot (j_b + b)) = Enc_{pk}(1-b), \\ w_3 &= (Enc_{pk}(j_{\tau_0} + \tau_0))^{y_0} = Enc_{pk}(y_0 \cdot (j_{\tau_0} + \tau_0)) = Enc_{pk}(y_0), \\ w_4 &= (Enc_{pk}(j_{\tau_1} + \tau_1))^{y_1} = Enc_{pk}(y_1 \cdot (j_{\tau_1} + \tau_1)) = Enc_{pk}(y_1). \end{aligned}$$

这样,对 $(w_b, w_{1-b}, w_2, w_3, w_4)$ 解密得到的明文为 $(x_b, x_{1-b}, 1-b, y_0, y_1)$.

因此,当 $j=1$ 时,协议 1 的输出为 $(x_b, x_{1-b}, 1-b, y_0, y_1)$,满足正确性要求.

情况 2) 当 $j=0$ 时, $\mathcal{F}_{ccbot}((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau))$ 的输出为 $(x_{\sigma}, y_{\tau}, \sigma \oplus b)$,我们证明在此情况下,协议 1 的输出也是如此.

在协议 1 中,当 $j=0$ 时 $j_0=0,j_1=0$,有 $j_b+b=b, b \in \{0,1\}$,所以,

$$\begin{aligned} w_b &= (Enc_{pk}(j_{\sigma_b}) \cdot Enc_{pk}(\sigma_b))^{x_b} = Enc_{pk}(x_b \cdot (j_{\sigma_b} + \sigma_b)) = Enc_{pk}(x_b \cdot \sigma_b), \\ w_{1-b} &= (Enc_{pk}(j_{\sigma_{1-b}}) \cdot Enc_{pk}(\sigma_{1-b}))^{x_{1-b}} = Enc_{pk}(x_{1-b} \cdot (j_{\sigma_{1-b}} + \sigma_{1-b})) = Enc_{pk}(x_{1-b} \cdot \sigma_{1-b}), \\ w_2 &= (Enc_{pk}(j_b) \cdot Enc_{pk}(b))^{1-b} = Enc_{pk}((1-b) \cdot (j_b + b)) = Enc_{pk}((1-b) \cdot b) = Enc_{pk}(0), \\ w_3 &= (Enc_{pk}(j_{\tau_0} + \tau_0))^{y_0} = Enc_{pk}(y_0 \cdot (j_{\tau_0} + \tau_0)) = Enc_{pk}(y_0 \cdot \tau_0), \\ w_4 &= (Enc_{pk}(j_{\tau_1} + \tau_1))^{y_1} = Enc_{pk}(y_1 \cdot (j_{\tau_1} + \tau_1)) = Enc_{pk}(y_1 \cdot \tau_1). \end{aligned}$$

可以看到,当 $j=0$ 时,无论 b 取何值, w_2 都是对 0 的加密.根据协议,我们忽略这一项,分别对 w_b, w_{1-b} 和 w_3, w_4 进行处理.对 w_b, w_{1-b} ,由于 $\sigma_\sigma=1, \sigma_{1-\sigma}=0$,所以在对 w_b, w_{1-b} 解密得到的明文中,一个为 0,一个为 x_σ ,其中 x_σ 的位置信息为 $\sigma \oplus b$.具体来说,若 $\sigma=1$,则 $\sigma_1=1, \sigma_0=0$,对 w_b, w_{1-b} 解密可得到 0 和 x_1 (顺序不定, x_1 的位置信息为 $\sigma \oplus b=1 \oplus b$);若 $\sigma=0$,则 $\sigma_0=1, \sigma_1=0$,对 w_b, w_{1-b} 解密可得到 0 和 x_0 (顺序不定, x_0 的位置信息为 $\sigma \oplus b=0 \oplus b$).对 w_3, w_4 ,由于 $\tau_\tau=1, \tau_{1-\tau}=0$,所以在对 w_3, w_4 解密得到的明文中,一个为 y_τ ,一个为 0,其中 y_τ 的位置由 τ 直接确定.

因此,当 $j=0$ 时,协议 1 的输出为 $(x_{\sigma}, y_{\tau}, \sigma \oplus b)$,满足正确性要求.

综上所述,有 $\{output^\pi((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau),n)\}_{x_0,x_1,y_0,y_1,b,\sigma,j,\tau,n} \cong \{\mathcal{F}_{ccbot}((x_0,x_1,y_0,y_1,b,\sigma),(j,\tau))\}_{x_0,x_1,y_0,y_1,b,\sigma,j,\tau,n}$

下面我们分别证明等式(1)和等式(2)成立.

情况 1) S 被腐化,证明等式(1)成立.

为了证明 S 被腐化时等式(1)成立,我们需要构造一个模拟器 \mathcal{S}_1 ,给定输入 $(x_0, x_1, y_0, y_1, b, \sigma)$ 和安全参数 1^n ,输出 S 在协议 1 中的视图,并证明该视图和真实协议执行中 S 的视图 $view_1^\pi((x_0, x_1, y_0, y_1, b, \sigma), (j, \tau), n)$ 计算不可区分。

根据协议 1, S 所能看到的视图消息仅为密文 $(Enc_{pk}(j_0), Enc_{pk}(j_{\tau_0+\tau_0}), Enc_{pk}(j_{\tau_1+\tau_1}))$ 。为了模拟该消息,我们令模拟器 \mathcal{S}_1 在明文空间随机选取明文 m_1, m_2, m_3 , 使用 R 的公钥 pk 进行加密,得到密文 $(Enc_{pk}(m_1), Enc_{pk}(m_2), Enc_{pk}(m_3))$ 。这样,给定输入 $(1^n, (x_0, x_1, y_0, y_1, b, \sigma))$, 模拟器 \mathcal{S}_1 的输出为

$$(x_0, x_1, y_0, y_1, b, \sigma, r, Enc_{pk}(m_1), Enc_{pk}(m_2), Enc_{pk}(m_3)),$$

其中, r 为协议过程中使用的随机数。

以上即是模拟器 \mathcal{S}_1 的构造。我们接下来证明等式(1)成立,即

$$\{\mathcal{S}_1(1^n, (x_0, x_1, y_0, y_1, b, \sigma))\}_{x_0, x_1, y_0, y_1, b, \sigma, n} \cong \{view_1^\pi((x_0, x_1, y_0, y_1, b, \sigma), (j, \tau), n)\}_{x_0, x_1, y_0, y_1, b, \sigma, j, \tau, n}.$$

由模拟器 \mathcal{S}_1 的构造可知, $\{\mathcal{S}_1(1^n, (x_0, x_1, y_0, y_1, b, \sigma))\} = \{(x_0, x_1, y_0, y_1, b, \sigma, r, Enc_{pk}(m_1), Enc_{pk}(m_2), Enc_{pk}(m_3))\}$ 。

由协议 1 可知, $\{view_1^\pi((x_0, x_1, y_0, y_1, b, \sigma), (j, \tau), n)\} = \{(x_0, x_1, y_0, y_1, b, \sigma, r, Enc_{pk}(j_0), Enc_{pk}(j_{\tau_0+\tau_0}), Enc_{pk}(j_{\tau_1+\tau_1}))\}$ 。

我们利用规约来证明上述两个分布不可区分。假如存在一个非均匀概率多项式时间区分器 D 和一个多项式 $p(\cdot)$, 满足对于无限多 n , 有:

$$|\Pr[D(\mathcal{S}_1(1^n, (x_0, x_1, y_0, y_1, b, \sigma)))=1] - \Pr[D(view_1^\pi((x_0, x_1, y_0, y_1, b, \sigma), (j, \tau), n))=1]| > 1/p(n),$$

则存在一个敌手 \mathcal{A}_M , 满足:

$$|\Pr[\mathcal{A}_M(Enc_{pk}(m_1), Enc_{pk}(m_2), Enc_{pk}(m_3))=1] - \Pr[\mathcal{A}_M(Enc_{pk}(j_0), Enc_{pk}(j_{\tau_0+\tau_0}), Enc_{pk}(j_{\tau_1+\tau_1}))=1]| > 1/p(n).$$

我们知道,如果公钥加密方案 M 满足 CPA 下的不可区分性,则 M 在 CPA 下是不可区分多重加密。因此,若存在敌手 \mathcal{A}_M 能够区分上述两个多重加密消息序列,则其能够攻破加密方案 M , 即 M 不具有 CPA 安全性,这和我们的假设相矛盾。因此,等式(1)成立,协议 1 在参与方 S 被腐化时是安全的。

情况 2) R 被腐化,证明等式(2)成立。

为了证明 R 被腐化时等式(2)成立,我们需要构造一个模拟器 \mathcal{S}_2 ,给定输入 (j, τ, z) 和安全参数 1^n ,输出 R 在协议 1 中的视图,并证明该视图和真实协议执行中 R 的视图 $view_2^\pi((x_0, x_1, y_0, y_1, b, \sigma), (j, \tau), n)$ 计算不可区分。

根据协议 1, R 所能看到的消息为 S 发送的密文五元组 $(w_b, w_{1-b}, w_2, w_3, w_4)$ 。由于 cut-and-choose 指示比特 j 取值不同时,由该密文五元组解密得到的输出结果不同,因此需要分情况进行讨论。

情况(1) 当 $j=1$ 时, R 解密密文得到的结果为 $z = (x_b, x_{1-b}, 1-b, y_0, y_1)$ 。给定 z 的值,模拟器 \mathcal{S}_2 可以直接用 R 的公钥 pk 对 $(x_b, x_{1-b}, 1-b, y_0, y_1)$ 进行加密,得到密文五元组 $(w'_b, w'_{1-b}, w'_2, w'_3, w'_4)$, 作为对真实协议执行中 $(w_b, w_{1-b}, w_2, w_3, w_4)$ 的模拟。

以上即是模拟器 \mathcal{S}_2 在 $j=1$ 时的构造。我们接下来证明在该情况下,等式(2)成立,即

$$\{\mathcal{S}_2(1^n, (j, \tau, z))\}_{j, \tau, z, n} \cong \{view_2^\pi((x_0, x_1, y_0, y_1, b, \sigma), (j, \tau), n)\}_{x_0, x_1, y_0, y_1, b, \sigma, j, \tau, n}.$$

由模拟器 \mathcal{S}_2 的构造可知, $\{\mathcal{S}_2(1^n, (j, \tau, z))\} = \{(j, \tau, r, w'_b, w'_{1-b}, w'_2, w'_3, w'_4)\}$ 。

由协议 1 可知, $\{view_2^\pi((x_0, x_1, y_0, y_1, b, \sigma), (j, \tau), n)\} = \{(j, \tau, r, w_b, w_{1-b}, w_2, w_3, w_4)\}$ 。

由于 R 掌握私钥 sk , 因此 $(w'_b, w'_{1-b}, w'_2, w'_3, w'_4)$ 和 $(w_b, w_{1-b}, w_2, w_3, w_4)$ 在 R 看来都是对 $(x_b, x_{1-b}, 1-b, y_0, y_1)$ 的加密。也就是说,上述两个分布对 R 来说是等同的。因此,等式(2)在 $j=1$ 时成立。

情况(2) 当 $j=0$ 时, R 解密密文得到的结果为 $z = (x_\sigma, y_\tau, \sigma \oplus b)$ 。给定 (j, τ, z) 的值,模拟器 \mathcal{S}_2 可以如下构造 R 的视图:

对 $(w_b, w_{1-b}, w_2, w_3, w_4)$ 中前两个元素 w_b 和 w_{1-b} , \mathcal{S}_2 用 pk 加密 x_σ 作为对其中第 $\sigma \oplus b + 1$ 个元素的模拟,用 pk 加密 0 作为对另一个元素的模拟;对第 3 个元素 w_2 , \mathcal{S}_2 用 pk 加密 0 作为对 w_2 的模拟;对最后两个元素 w_3 和 w_4 , \mathcal{S}_2 用 pk 加密 y_τ 作为对其中第 $\tau + 1$ 个元素的模拟,用 pk 加密 0 作为对另一个元素的模拟。

以上即是模拟器 \mathcal{S}_2 在 $j=0$ 时的构造。在掌握私钥 sk 的 R 看来,上述构造的密文和真实协议执行中接收到的密文 $(w_b, w_{1-b}, w_2, w_3, w_4)$ 具有等同的分布。因此,等式(2)在 $j=0$ 时成立。

这样就完成了 R 被腐化时的证明。协议 1 在参与方 R 被腐化时是安全的。

综上,协议 1 在定义 2 下安全计算功能函数 \mathcal{F}_{cbot} 。 \square

3 应用

Cut-and-choose 双向不经意传输在基于 cut-and-choose 技术的密码协议中具有重要意义.本文以安全两方计算为例,对其应用及所带来的优势进行阐述.在详细介绍如何将 cut-and-choose 双向不经意传输应用到安全两方计算之前,首先简单介绍基于 Yao 混乱电路的安全两方计算协议以及 cut-and-choose 范例.

安全两方计算是参与方 P_1 和 P_2 之间的交互式计算任务.双方各自持有自己的秘密输入 x 和 y ,互不信任,交互式地协同计算一个目标函数 $f(\cdot, \cdot)$.任务完成后,双方各自获得相应的秘密输出,整个计算过程不泄露其他任何额外信息.20世纪80年代,姚期智先生提出采用“混乱电路”的方法来实现安全两方计算^[15].混乱电路实际上是对布尔电路的一种加密形式,其拥有如下性质:

(1) 每条电路线对应两个混乱密钥,一个密钥对应比特 0,另一个密钥对应比特 1;

(2) 当每条电路输入线给定一个密钥时,可以逐层计算该混乱电路,并最终获得电路输出值,除此之外得不到其他任何额外信息.

基于混乱电路和不经意传输,文献[15]给出了安全两方计算的第 1 个通用解决方案——Yao 协议.在该协议中, P_1 和 P_2 分别为混乱电路构造方和混乱电路计算方.双方首先将要计算的函数表示为布尔电路,参与方输入的每一位在布尔电路中都有对应的电路输入线.然后, P_1 独立构造该布尔电路对应的混乱电路,并将该混乱电路以及对应自己输入的密钥发送给 P_2 .对 P_2 输入所对应的每条输入线,双方执行一次不经意传输协议,使 P_2 获取自己输入对应的密钥.具体来说, P_1 作为发送方,输入为该条输入线上的两个密钥; P_2 作为接收方,输入为该条输入线对应的输入比特.协议执行完成后, P_2 获得其输入比特对应的密钥.最后, P_2 使用已获得的所有电路输入线上的密钥,在不知道 P_1 秘密输入和其他额外信息的情况下,茫然地对混乱电路逐层计算,获得电路的输出值.

Yao 协议非常高效,但其仅在半诚实模型下安全.在恶意模型下,潜在的恶意参与方可以任意偏离协议执行,协议构造更加困难,也更低效.一个显然的问题是,恶意的 P_1 可能会构造错误的混乱电路,使 P_2 计算出的函数结果出错.与利用零知识证明强迫潜在的恶意对手诚实地执行协议相比,使用 cut-and-choose 技术可以获得恶意模型下非常高效的安全两方计算协议^[12,16,17].该技术的基本想法是使参与方 P_2 能够以很高的概率发现恶意 P_1 的作弊行为.首先, P_1 被要求构造 s 份混乱电路而不是 1 份(s 为统计学安全参数). P_2 随机选择其中一部分(称为检测电路),要求 P_1 打开,以检测混乱电路是否正确构造;若全部检测通过,则以很高的概率说明剩余混乱电路中的大多数都是正确构造的.随后,双方像 Yao 协议一样计算剩余的每个混乱电路(称为计算电路),并取计算结果中占大多数的值作为协议输出.通过这种方法,恶意构造方作弊成功的概率被限制得非常小.

在所有基于 cut-and-choose 范例的两方计算协议中,文献[12]提供的解决方案较为典型、高效.该协议主要基于 cut-and-choose 不经意传输进行构造.注意到该功能函数仅能用来传输单个混乱电路中单条 P_2 输入线上的密钥.为了使其能够适用于外层的两方计算协议,需要将其扩展为一个可以进行批处理的版本,称为批处理单选择 cut-and-choose 不经意传输,记为 $\mathcal{F}_{cbot}^{B,S}$,以用来传输 s 个混乱电路中所有 P_2 输入线上的密钥.关于 $\mathcal{F}_{cbot}^{B,S}$ 的详细描述可参见文献[12].基于 cut-and-choose 不经意传输,我们可以获得如下的安全两方计算协议框架.

基于 Cut-and-Choose 不经意传输的安全两方计算协议框架.

步骤 1. P_1 构造 s 份混乱电路.

步骤 2. 双方调用批处理单选择 cut-and-choose 不经意传输功能函数 $\mathcal{F}_{cbot}^{B,S}$.对于所有混乱电路中每条 P_2 输入线:在检测电路中, P_2 获得全部两个密钥;在计算电路中, P_2 获得与其输入比特相对应的密钥.

步骤 3. P_1 将所有混乱电路发送给 P_2 .

步骤 4. P_2 公开 $\mathcal{F}_{cbot}^{B,S}$ 中用到的 cut-and-choose 指示比特串 J ,并向 P_1 证明 J 的正确性.

步骤 5. 对每个检测电路, P_1 发送自己每条输入线上的两个密钥.

步骤 6. P_2 对所有检测电路进行检测.若检测出现问题,则 P_2 中止协议;否则继续执行.

步骤 7. 对每个计算电路, P_1 将自己输入对应的密钥发送给 P_2 ,并证明在所有计算电路中, P_1 输入一致.

步骤 8. P_2 对所有计算电路进行计算,并取计算结果中占大多数的值作为协议输出.

可以看出,上述协议框架中的密钥传输被分割成为几个部分,导致参与方之间的交互变得非常复杂,从轮复杂度的角度考虑协议较为低效;同时,复杂的交互使协议框架非常混乱,不利于协议的理解和应用。

为了降低参与方的交互复杂度并简化协议框架,我们将上述协议中的 cut-and-choose 不经意传输替换为新原语 cut-and-choose 双向不经意传输.利用该原语的双向性,可以将所有涉及到的密钥在一个阶段全部传输完成.同样,由于功能函数 \mathcal{F}_{cobot} 仅能用来传输单个混乱电路中单条输入线上的密钥,我们将其扩展为可用于传输 s 个混乱电路中所有输入线密钥的功能函数,称为批处理单选择 cut-and-choose 双向不经意传输,记为 $\mathcal{F}_{cobot}^{B,S}$.基于 cut-and-choose 双向不经意传输,安全两方计算协议框架可以简化为如下过程.

基于 Cut-and-Choose 双向不经意传输的安全两方计算协议框架.

步骤 1. P_1 构造 s 份混乱电路.

步骤 2. 双方调用批处理单选择 cut-and-choose 双向不经意传输功能函数 $\mathcal{F}_{cobot}^{B,S}$. 对于所有混乱电路中的每条输入线:在检测电路中, P_2 获得全部两个密钥;在计算电路中, P_2 获得与参与方输入比特相对应的密钥.

步骤 3. P_1 将所有混乱电路发送给 P_2 .

步骤 4. P_2 对所有检测电路进行检测,检测通过后计算所有计算电路,并取计算结果中占大多数的值作为协议输出.

通过对比该协议框架与基于 cut-and-choose 不经意传输的协议框架,可以看出,将 cut-and-choose 不经意传输替换为 cut-and-choose 双向不经意传输后,协议框架明显得到简化,新框架更清晰、更易于理解;参与方之间的交互轮数明显得到降低.具体来说,原有协议框架中的公开 cut-and-choose 指示比特串(步骤 4)、发送检测电路中 P_1 输入线密钥(步骤 5)以及发送计算电路中 P_1 输入对应密钥(步骤 7)等诸多步骤都被省去.参与方 P_1 仅需在构造完混乱电路后和 P_2 运行一次批处理单选择 cut-and-choose 双向不经意传输协议,然后将所有混乱电路发送给 P_2 即可.剩余阶段不再需要 P_1 参与, P_2 可以本地完成所有检测和计算操作,从而极大地降低了协议的交互复杂度.

4 结束语

本文提出了一种称为 cut-and-choose 双向不经意传输的密码学原语,并基于同态加密给出了仅需 1 轮交互的协议构造.该构造被形式化证明在半诚实模型下满足标准的安全性定义.此外,本文详细阐述了该原语在基于 cut-and-choose 范例的安全两方计算中的应用和优势.在下一步的研究中,我们将重点关注如何构造可抵抗恶意对手攻击的 cut-and-choose 双向不经意传输协议,使其能够真正应用到基于 cut-and-choose 范例的安全协议中.

References:

- [1] Rabin MO. How to exchange secrets with oblivious transfer. TR-81: Cambridge & Boston, Massachusetts, Aiken Computation Laboratory, Harvard University, 1981.
- [2] Crépeau C. Equivalence between two flavours of oblivious transfers. In: Pomerance C, ed. Advances in Cryptology-CRYPTO'87. Berlin: Springer-Verlag, 1988. 350–354. [doi: 10.1007/3-540-48184-2_30]
- [3] Naor M, Pinkas B. Efficient oblivious transfer protocols. In: Kosaraju SR, ed. Proc. of the 12th Annual ACM-SIAM Symp. on Discrete Algorithms. Philadelphia: Society for Industrial and Applied Mathematics, 2001. 448–457.
- [4] Camenisch J, Neven G, Shelat A. Simulatable adaptive oblivious transfer. In: Naor M, ed. Proc. of the EUROCRYPT 2007. Berlin: Springer-Verlag, 2007. 573–590. [doi: 10.1007/978-3-540-72540-4_33]
- [5] Peikert C, Vaikuntanathan V, Waters B. A framework for efficient and composable oblivious transfer. In: Wagner D, ed. Proc. of the CRYPTO 2008. Berlin: Springer-Verlag, 2008. 554–571. [doi: 10.1007/978-3-540-85174-5_31]
- [6] Guo YB, Zhang ZN, Yang KW. Oblivious transfer based on physical unclonable function system. Journal on Communications, 2013,34(Z1):38–43 (in Chinese with English abstract).
- [7] Garay JA, MacKenzie P, Yang K. Efficient and universally composable committed oblivious transfer and applications. In: Naor M, ed. Proc. of the TCC 2004. Berlin: Springer-Verlag, 2004. 297–316. [doi: 10.1007/978-3-540-24638-1_17]
- [8] Kiraz MS, Schoenmakers B, Villegas J. Efficient committed oblivious transfer of bit strings. In: Garay JA, Lenstra AK, Mambo M,

- Peralta R, eds. Proc. of the ISC 2007. Berlin: Springer-Verlag, 2007. 130–144. [doi: 10.1007/978-3-540-75496-1_9]
- [9] Ishai Y, Kilian J, Nissim K, Petrank E. Extending oblivious transfers efficiently. In: Boneh D, ed. Proc. of the CRYPTO 2003. Berlin: Springer-Verlag, 2003. 145–161. [doi: 10.1007/978-3-540-45146-4_9]
- [10] Nielsen JB, Nordholt PS, Orlandi C, Burra SS. A new approach to practical active-secure two-party computation. In: Safavi-Naini R, Canetti R, eds. Proc. of the CRYPTO 2012. Berlin: Springer-Verlag, 2012. 681–700. [doi: 10.1007/978-3-642-32009-5_40]
- [11] Asharov G, Lindell Y, Schneider T, Zohner M. More efficient oblivious transfer and extensions for faster secure computation. In: Sadeghi AR, ed. Proc. of the 20th ACM Conference on Computer and Communications Security. New York: ACM, 2013. 535–548. [doi: 10.1145/2508859.2516738]
- [12] Lindell Y, Pinkas B. Secure two-party computation via cut-and-choose oblivious transfer. In: Ishai Y, ed. Proc. of the TCC 2011. Berlin: Springer-Verlag, 2011. 329–346. [doi: 10.1007/978-3-642-19571-6_20]
- [13] Kiraz MS, Schoenmakers B. A protocol issue for the malicious case of Yao's garbled circuit construction. In: Legendijk RL, Weber JH, eds. Proc. of the 27th Symp. on Information Theory in the Benelux. Eindhoven: Werkgemeenschap voor Informatie-en Communicatietheorie, 2006. 283–290.
- [14] Goldreich O. Foundations of Cryptography: Volume II, Basic Applications. New York: Cambridge University Press, 2004. [doi: 10.1017/CBO9780511721656]
- [15] Yao ACC. How to generate and exchange secrets. In: Hopcroft J, ed. Proc. of the 27th Annual IEEE Symp. on Foundations of Computer Science. Washington: IEEE Computer Society, 1986. 162–167. [doi: 10.1109/SFCS.1986.25]
- [16] Lindell Y, Pinkas B. An efficient protocol for secure two-party computation in the presence of malicious adversaries. In: Naor M, ed. Proc. of the EUROCRYPT 2007. Berlin: Springer-Verlag, 2007. 52–78. [doi: 10.1007/978-3-540-72540-4_4]
- [17] Lindell Y. Fast cut-and-choose based protocols for malicious and covert adversaries. In: Canetti R, Garay JA, eds. Proc. of the CRYPTO 2013. Berlin: Springer-Verlag, 2013. 1–17. [doi: 10.1007/978-3-642-40084-1_1]

附中文参考文献:

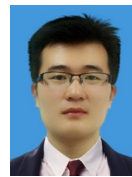
- [6] 郭渊博,张紫楠,杨奎武.基于 PUFs 的不经意传输协议.通信学报,2013,34(Z1):38–43.



赵川(1989—),男,山东泰安人,博士,讲师,主要研究领域为安全多方计算,云计算安全.



蒋瀚(1974—),男,博士,讲师,主要研究领域为密码学与信息安全,公钥密码学,安全多方计算.



魏晓超(1990—),男,博士生,主要研究领域为实用安全多方计算,安全外包计算.



徐秋亮(1960—),男,博士,教授,博士生导师,主要研究领域为公钥密码算法设计与分析,密码协议设计与分析,多方安全计算协议研究,网络信息安全核心技术与开发.