- 另一类攻击方法是部分计算攻击,敌手计算客户端不同输入下的部分矩阵链乘,试图通过比较这些中间值了解 RAT 的一些信息.例如,服务器在 2 个不同客户端输入下分别计算矩阵链乘 $\boldsymbol{Z}_{i,1}$ 的前两个矩阵 $\hat{\boldsymbol{P}}_4\hat{\boldsymbol{P}}_{i,1}$ 乘积,也就是 $\boldsymbol{P}_4\boldsymbol{P}_{i,1}\boldsymbol{R}_2^{-1}$,如果上述 Kilian 矩阵编码方案是理想的,则使用部分计算攻击的敌手并不能获得任何有用信息.

定理 1. 若 DDH 假设是存在的,则上述非交互可验证计算协议对外包函数 f 是安全的.

证明:接下来,采用模拟证明技术(simulation proof technique)[13,26]证明定理 1 成立.首先,先给出引理 1 和引理 2 及其证明.

引理 1. 如果函数 f 是多项式时间可计算函数,由 $Garble(1^\lambda,C)$ 生成混淆电路的分布和由模拟器执行 $GarbleSim(1^\lambda,C)$ 生成混淆电路的分布在 DDH 假设下是计算上不可区分的.

证明:引理 1 的证明过程类似于 Lindell-Pinkas 关于 Yao 协议的证明过程[13].

首先描述模拟器的构造.模拟器执行 $GarbleSim(1^\lambda,C)$,生成一个伪造的混淆电路,过程如下所述:对于混淆电路的每个门电路 g,设其输入电线分别是 a 和 b,输出电线为 c;为电线 a 分别选择活动标签(active label)L_a 和不活动标签(inactive label) L'_a,如果标签被用于计算混淆电路则称它是活动标签,同一电线的另一标签就是不活动标签.同样地,为电线 b 分别选择活动标签和不活动标签 L_b, L'_a,为电线 c 选择活动标签和不活动标签 L_c, L'_c. 为该门电路随机选择 4 个新 2ℓ-bit 掩码 $\delta_1,\delta_2,\delta_3,\delta_4$,计算并随机排序如下 4 个密文对:

$$(Enc_{L_a}(\delta_1),Enc_{L_b}((L_c\,|\,0^\ell)\oplus\delta_1)),$$
$$(Enc_{L_a}(\delta_2),Enc_{L'_b}((L_c\,|\,0^\ell)\oplus\delta_2)),$$
$$(Enc_{L'_a}(\delta_3),Enc_{L_b}((L_c\,|\,0^\ell)\oplus\delta_3)),$$
$$(Enc_{L'_a}(\delta_4),Enc_{L'_b}((L_c\,|\,0^\ell)\oplus\delta_4)).$$

其中,所有密文对只加密输出电线的活动标签.上述所有的门电路构成了 $GarbleSim(1^\lambda,C)$ 生成的混淆电路.

为了证明模拟器产生混淆电路的分布与实际执行 $Garble(1^\lambda,C)$ 生成混淆电路的分布是计算上不可区分的,接下来使用标准的混合体论证(hybrid argument)[26],需定义一系列的混合体 $H_0,H_1,...,H_{|\Gamma|}$.

- H_0:此混合体实际执行 $Garble(1^\lambda,C)$ 生成混淆电路 Γ.
- H_i,其中,$i\in(0,|\Gamma|)$:此混合体与 H_0 的区别在于前 i 个门电路 $g_1,...,g_i$ 的 4 个密文对是由门电路输入标签所有 4 个组合加密门电路输出电线活动标签的密文组成,其他门电路与 H_0 的混淆电路门电路相同.
- $H_{|\Gamma|}$:此混合体执行 $GarbleSim(1^\lambda,C)$ 生成混淆电路,每个门电路都只有输出电线的活动标签加密.

对于所有 $i\in[1,|\Gamma|]$,任意两个连续的混合体 H_{i-1} 和 H_i 的区别在于:H_{i-1} 中门电路 g_i 输出电线不活动标签的密文在 H_i 中被输出电线活动标签的密文所取代.

假设门电路 g_i 的输入电线分别是 a 和 b,输出电线为 c.电线分别是 a 活动标签和不活标签分别是 L_{a_i}, L'_{a_i},相类似电线 b 的分别是 L_{b_i}, L'_{b_i},电线 c 的分别是 L_{c_i}, L'_{c_i}. H_{i-1} 中该门电路的 4 个密文对如下:

$$(Enc_{L_{a_i}}(\delta_{i,1}),Enc_{L_{b_i}}((L_{c_i}\,|\,0^\ell)\oplus\delta_{i,1})),$$
$$(Enc_{L_{a_i}}(\delta_{i,2}),Enc_{L'_{b_i}}((L_{i,1}\,|\,0^\ell)\oplus\delta_{i,2})),$$
$$(Enc_{L'_{a_i}}(\delta_{i,3}),Enc_{L_{b_i}}((L_{i,2}\,|\,0^\ell)\oplus\delta_{i,3})),$$
$$(Enc_{L'_{a_i}}(\delta_{i,4}),Enc_{L'_{b_i}}((L_{i,3}\,|\,0^\ell)\oplus\delta_{i,4})).$$

其中,$L_{i,1},L_{i,2},L_{i,3}$ 是 L_{c_i} 或者是 L'_{c_i}. H_i 中该门电路的 4 个密文对如下:

$$(Enc_{L_{a_i}}(\delta_{i,1}),Enc_{L_{b_i}}((L_{c_i}\,|\,0^\ell)\oplus\delta_{i,1})),$$
$$(Enc_{L_{a_i}}(\delta_{i,2}),Enc_{L'_{b_i}}((L_{c_i}\,|\,0^\ell)\oplus\delta_{i,2})),$$
$$(Enc_{L'_{a_i}}(\delta_{i,3}),Enc_{L_{b_i}}((L_{c_i}\,|\,0^\ell)\oplus\delta_{i,3})),$$
$$(Enc_{L'_{a_i}}(\delta_{i,4}),Enc_{L'_{b_i}}((L_{c_i}\,|\,0^\ell)\oplus\delta_{i,4})).$$

H_{i-1}和H_i是计算上不可区分的,这可通过归约到 BHHO 加密算法的语义安全得出.

假设存在 PPT 区分器 D 和多项式 $p(\cdot)$,对无限多的 n 有:

$$|\Pr[D(H_i)=1]-\Pr[D(H_{i-1})=1]|>\frac{1}{|\Gamma|\,p(n)}.$$

利用区分器 D 构造 PPT 敌手$\mathcal{A}$,$\mathcal{A}$接收门电路 g_i 密文对并构造混淆电路,该电路一部分是 $Garble(1^{\lambda},C)$ 真实产生的门电路,另一部分是 $GarbleSim(1^{\lambda},C)$ 伪造产生的门电路.若$\mathcal{A}$接收的门电路 g_i 密文对与 H_{i-1} 中的门电路 g_i 密文对相同,则该构造与 H_{i-1} 的混淆电路一致;若$\mathcal{A}$接收的门电路 g_i 密文对与 H_i 中的门电路 g_i 密文对相同,则该构造与 H_i 的混淆电路一致.如果敌手$\mathcal{A}$能够区分 H_{i-1} 和 H_i,就可以区分其中门电路 g_i 的密文对,这与 BHHO 加密算法的安全性相矛盾. □

引理 2. 有效算法 *reRandGb* 输入为随机数 r 和 $Garble'(1^{\lambda},C)$ 生成的混淆电路Γ',输出电路 C 的重随机化混淆电路 $\tilde{\Gamma}$,则 $Garble(1^{\lambda},C)$ 生成混淆电路Γ的分布和重随机化混淆电路 $\tilde{\Gamma}$ 的分布在 DDH 假设下是计算上不可区分的.

证明:引理 2 的证明方法与引理 1 的类似.为了证明以上两个分布是不可区分的,定义一系列的混合体 H_0,H_1,...,$H_{|T|}$,这里的 T 表示混淆电路的电线数量.

- H_0.此混合体执行 $Gb.Garble(1^{\lambda},C)$ 生成混淆电路Γ,使其输出与执行 $Gb.Garble'(1^{\lambda},C)$ 生成的混淆电路Γ'输出相同,则混淆电路Γ的分布与混淆电路Γ'的分布是一致的.
- H_i,其中,$i\in(0,|T|)$.此混合体生成的混淆电路前 i 根电线是由重随机化混淆电路Γ'的前 i 根电线得到,其他电线与混淆电路Γ'的电线相同.
- $H_{|T|}$.此混合体是执行 *reRandGb* 生成的重随机化混淆电路 $\tilde{\Gamma}$,每根电线都是重随机化混淆电路Γ'的相应电线而得到的.

对于所有 $i\in[1,|T|]$,任意两个连续的混合体 H_{i-1} 和 H_i 的区别在于第 i 根电线在 H_{i-1} 中与混淆电路Γ'的第 i 根电线相同,而 H_i 的第 i 根电线是由重随机混淆电路Γ'的第 i 根电线而得到的.H_{i-1} 和 H_i 是计算上不可区分的,这可由 BHHO 加密算法安全性得出. □

接下来,利用引理 1 和引理 2 证明定理 1 在两方计算下是成立的,那么定理 1 在可验证计算下也是成立的(我们考虑的不仅是可验证计算,还有在两方计算下的私有函数计算,见第 5 节).基于模拟的安全强于基于不可区分的安全,如果采用模拟证明技术证明协议是基于模拟的安全,则一定是计算不可区分安全(见定义 2).因此,在证明中需构造恶意的服务器和恶意的客户端,并且模拟服务器的视图(view)与客户端的视图.定义一个模拟器 $Sim=\{Sim_c,Sim_s\}$,Sim_c 模拟客户端的视图,Sim_s 模拟服务器的视图.

- Sim_c.给定客户端的输入 x 和计算结果 $y=f(x)$,构造模拟客户端的 Sim_c.首先,一致随机选择矩阵 $\{\hat{\boldsymbol{P}}_{i,1},\hat{\boldsymbol{P}}_{i,2},\hat{\boldsymbol{P}}_{i,3}\}_{i=1}^{i=|\Gamma|}$,$\boldsymbol{A}$,$\boldsymbol{B}$;接下来计算矩阵乘积 $\{\boldsymbol{Z}_{i,1}=\hat{\boldsymbol{P}}_{i,1}\cdot\hat{\boldsymbol{P}}_{i,3},\boldsymbol{Z}_{i,2}=\hat{\boldsymbol{P}}_{i,2}\cdot\hat{\boldsymbol{P}}_{i,3}\}_{i=1}^{i=|\Gamma|}$,为了生成混淆电路$\Gamma$,执行 $Gb.GarbleSim(1^{\lambda},C)\rightarrow(\Gamma,\{L_i^0,L_i^1\}_{i=1}^{2n})$;然后选择客户端输入 x 相对应的输入电线活动标签 $\{c_i\}_{i=1}^{n}$;对输入电线 $i\in[n]$,执行 $\boldsymbol{reRandGb}.\boldsymbol{InLabelSim}(\gamma_i,c_i)\rightarrow\gamma_i(c_i)$,其中,$\gamma_i=A\cdot\hat{\boldsymbol{P}}_{i,1}$ 或 $\gamma_i=B\cdot\hat{\boldsymbol{P}}_{i,2}$. Sim_c 剩下的步骤与客户端执行过程相同.
- Sim_s.给定客户端输入 x 和计算结果 $y=f(x)$,Sim_s 模拟服务器构造混淆电路,其计算结果等于 $f(x)$.具体来说,执行 $Gb.GarbleSim(1^{\lambda},C)\rightarrow(\Gamma,\{L_i^0,L_i^1\}_{i=1}^{2n})$, 在 $\{L_i^0,L_i^1\}_{i=1}^{n}$ 中活动标签与 $\{\gamma_i(c_i)\}_{i=1}^{n}$ 相关联.考虑输入电线是 a,b 和输出电线为 c 的门电路,可用如等式(1)的 4 个密文对表示.然而,此时 4 个密文对只包含门电路输出电线的活动标签密文.Sim_s 剩下的步骤与服务器执行过程相同.

为了证明协议执行过程的模拟器视图与协议实际执行是计算上不可区分的,需定义一系列的混合体,它们开始于客户端与服务器协议的真实执行,结束于充当客户端与服务器角色的模拟器理想执行.

- H_0.此混合体中的客户端和服务器都遵循协议的实际执行(见第 4.1 节).
- H_1.此混合体与 H_0 的区别仅在于 $\hat{\boldsymbol{P}}_{i,1},\hat{\boldsymbol{P}}_{i,2},\hat{\boldsymbol{P}}_{i,3}$ 的计算方法.模拟器在计算中不采用 Kilian 的随机化技术,

而是为客户端和服务器端都选择随机矩阵 $\hat{\boldsymbol{P}}_{i,1},\hat{\boldsymbol{P}}_{i,2},\hat{\boldsymbol{P}}_{i,3}$.

- H_2.此混合体与 H_1 的区别仅在于 $\boldsymbol{Z}_{i,1},\boldsymbol{Z}_{i,2}$ 的计算方法.Sim_c 在计算中不采用数学伪装方法,而是直接计算 $\boldsymbol{Z}_{i,1}=\hat{\boldsymbol{P}}_{i,1}\cdot\hat{\boldsymbol{P}}_{i,3},\boldsymbol{Z}_{i,2}=\hat{\boldsymbol{P}}_{i,2}\cdot\hat{\boldsymbol{P}}_{i,3}$. 再者,$Sim_s$ 将 $\boldsymbol{Z}_{i,1},\boldsymbol{Z}_{i,2}$ 作为输入,执行 $reRandGb.GateSim\{g_i,\boldsymbol{Z}_{i,1},\boldsymbol{Z}_{i,2},(\delta'_{i,1},\delta'_{i,2},\delta'_{i,3},\delta'_{i,4})\}$, 生成重随机化混淆电路.
- H_3.此混合体与 H_2 的区别仅在于由模拟器新构建混淆电路取代重随机化的混淆电路.具体来说,模拟器模拟客户端执行 $Gb.Garble(1^{\lambda},C)$,生成混淆电路;接下来为客户端输入 x 选取电路输入电线标签,执行 $\boldsymbol{Gb}.\boldsymbol{Enc}(gsk,x)$,生成输入的编码.相类似地,模拟器模拟服务器执行 $Gb.Garble(1^{\lambda},C)$, 生成混淆电路.
- H_4.此混合体与 H_3 的区别仅在于由模拟器执行 $Gb.GarbleSim(1^{\lambda},C)$, 新构建混淆电路取代原有电路.具体来说,模拟器模拟客户端执行 $Gb.GarbleSim(1^{\lambda},C)$, 生成混淆电路;接下来为客户端输入 x 选取电路输入电线标签,执行 $\boldsymbol{reRandGb}.\boldsymbol{InLabelSim}(\gamma_i,c_i)$,重随机化输入.相类似地,模拟器模拟服务器执行 $Gb.GarbleSim(1^{\lambda},C)$,生成电路混淆.

下面证明每个混合体与相邻的混合体是不可区分的.

引理 3. 对所有的概率多项式时间敌手$\mathcal{A}$, $H_0\overset{C}{\equiv}H_1$.

证明:根据 Kilian 的随机化技术正确性,返回给敌手$\mathcal{A}$的矩阵 $\hat{\boldsymbol{P}}_{i,1},\hat{\boldsymbol{P}}_{i,2},\hat{\boldsymbol{P}}_{i,3}$ 分布上并没有变化.因此,敌手$\mathcal{A}$赢得 H_1 的概率仍然与赢得 H_0 的概率相同. □

引理 4. 对所有的概率多项式时间敌手$\mathcal{A}$, $H_1\overset{C}{\equiv}H_2$.

证明:敌手$\mathcal{A}$赢得 H_1 的概率与赢得 H_2 的概率相同,因为否则就存在一个攻击者对数学伪装方法具有不可忽略的优势. □

引理 5. 对所有的概率多项式时间敌手$\mathcal{A}$, $H_2\overset{C}{\equiv}H_3$.

证明:根据引理 2 的证明可以得出:给定一个由 $Gb.Garble(1^{\lambda},C)$ 生成的混淆电路以及由 $reRandGb$ 生成的重随机化混淆电路,没有计算能力受限的敌手能够区分这两个电路.这意味敌手$\mathcal{A}$在 H_3 中的概率与在 H_2 中的概率相同. □

引理 6. 对所有的概率多项式时间敌手$\mathcal{A}$, $H_3\overset{C}{\equiv}H_4$.

证明:根据引理 1 的证明可得出:由 $Gb.GarbleSim(1^{\lambda},C)$ 得到的电路与由 $Gb.Garble(1^{\lambda},C)$ 得到的电路,没有计算能力受限的敌手能够区分这两个电路.这意味敌手$\mathcal{A}$在 H_4 中的概率与在 H_3 中的概率相同. □

接下来说明在基于模拟的安全下客户端不会接受敌手$\mathcal{A}$伪造的计算结果.若敌手$\mathcal{A}$不能使验证算法接受一个不正确的输出,则可验证计算方案是安全的(见定义 2 关于可验证计算方案是计算不可区分安全的定义).为了说明客户端不会接受敌手$\mathcal{A}$伪造的计算结果,需要给定输入 x 和计算结果 $y=f(x)$,构造客户端模拟器 Sim'_c, 具体过程如下.

1. Sim'_c 生成 $(pk,sk)\leftarrow\boldsymbol{KeyGen}(1^{\lambda},f)$.
2. 敌手$\mathcal{A}$向 Sim'_c 发起 **ProbGen** 查询, Sim'_c 执行 $\boldsymbol{ProbGen}(sk,x)$,并将$\sigma_x$返回敌手$\mathcal{A}$.
3. 敌手$\mathcal{A}$向预言机 **PVerify** 发起查询,预言机返回 acc,当且仅当 $\boldsymbol{Verify}(sk,\sigma_x,\tau_x)\rightarrow(acc,y)$.预言机 **Pverify** 仅返回接收/拒绝比特 acc.
4. 步骤 2 和步骤 3 可重复多项式次数.
5. 给定敌手$\mathcal{A}$的σ_y, Sim'_c 生成$(acc,y)\leftarrow\boldsymbol{Verify}(sk,\sigma_y,\tau_x)$.如果$\sigma_y$ 是伪造的计算结果, Sim'_c 将不能映射出正确的混淆电路输出标签,则 acc=0(拒绝),输出$\perp$;否则,acc=1(接收),并输出 y.

Sim'_c 的输出与在真实协议中客户端的输出消息是不可区分的,这可从引理 1~引理 6 的证明过程得出.

综上所述,定理 1 得证. □

定理 2. 若 DDH 假设是存在的,则上述非交互可验证计算协议对服务器是隐私的.

证明:该证明与定理 1 的证明都具有类似形式的混合体和证明过程.服务器的隐私性可由 RAT 的隐私、可重随机化的混淆电路安全性和 Yao 的混淆电路安全性得出. □

4.3 适应性安全

静态安全的混淆电路是指输入不依赖于混淆电路[13,27](见定义 7).与之相反,适应性安全的混淆电路是指,如果敌手在查看混淆电路以后还允许适应性地选择输入,此时混淆电路仍是安全的(见定义 8).Yao 的混淆电路只能做到静态安全的,这是因为为了提高在线阶段的效率,混淆电路的生成和发送通常都在离线阶段,恶意的敌手就有可能根据混淆电路自己选择输入,使得混淆电路不再安全.文献[3]的可验证计算方案使用了 Yao 的混淆电路,为了保证方案的安全,必须在发送混淆电路之前确定输入,因此该方案是静态安全的.

第 4.1 节的可验证计算协议虽然也是静态安全的,但可采用两种方法实现适应性安全:一种方法是将该协议运用复杂性杠杆(complexity leveraging)实现适应性安全;另一种方法是将该协议作细微的调整,即可做到适应性安全.为了使恶意的敌手在离线阶段不能根据混淆电路自己选择输入,客户端只需将所有门电路的密文对的前半部分和后半部分别与一个大小合适的随机可逆矩阵 $\boldsymbol{R}_5^{-1}$ 相乘.相应的,构造矩阵 $\hat{\boldsymbol{P}}_4=\boldsymbol{R}_0\boldsymbol{P}_4\boldsymbol{R}_1^{-1}$ 修改为 $\hat{\boldsymbol{P}}_4=\boldsymbol{R}_5\boldsymbol{P}_4\boldsymbol{R}_1^{-1}$,计算 $\boldsymbol{Z}_{i,1}=\boldsymbol{R}_5\boldsymbol{P}_4\boldsymbol{P}_{i,1}\boldsymbol{P}_5\boldsymbol{P}_4\boldsymbol{P}_{i,3}\boldsymbol{P}_5$ 和 $\boldsymbol{Z}_{i,2}=\boldsymbol{R}_5\boldsymbol{P}_4\boldsymbol{P}_{i,2}\boldsymbol{P}_5\boldsymbol{P}_4\boldsymbol{P}_{i,3}\boldsymbol{P}_5$,再将现在的 $\boldsymbol{Z}_{i,1},\boldsymbol{Z}_{i,2}$ 用于随机化门电路 i.上述过程既没有将 RAT 泄露给敌手,也保证敌手不能尝试混淆电路的输入,即实现了适应性安全.为了说明这一点,可采用文献[27]中静态安全转换为适应性安全的构造方法.具体来说,因为定理 1 成立,故存在静态安全 PPT 模拟器$\mathcal{S}$,对任意 PPT 敌手$\mathcal{A}$(见定义 7),

$$\mathrm{pr}[\boldsymbol{Exp}_{\mathcal{A},s}^{static}(1^{\lambda},0)=1]-\mathrm{pr}[\boldsymbol{Exp}_{\mathcal{A},s}^{static}(1^{\lambda},1)=1]\leqslant negl(\lambda).$$

给定任意适应性安全 PPT 敌手$\mathcal{A}_1$,构建静态安全敌手$\mathcal{A}$.若由模拟器$\mathcal{S}$能够构建适应性安全 PPT 模拟器$\mathcal{S}_1$:

$$\mathrm{pr}[\boldsymbol{Exp}_{\mathcal{A}_1,\mathcal{S}_1}^{adaptive}(1^{\lambda},0)=1]-\mathrm{pr}[\boldsymbol{Exp}_{\mathcal{A}_1,\mathcal{S}_1}^{adaptive}(1^{\lambda},1)=1]\leqslant negl(\lambda) \tag{3}$$

则适应性安全成立(见定义 8).

适应性安全实验 $\boldsymbol{Exp}_{\mathcal{A}_1,\mathcal{S}_1}^{adaptive}(1^{\lambda},b_1)$ 定义如下.

1. 挑战者随机选择比特 b_1.
2. 敌手$\mathcal{A}_1$ 向挑战者提交 C,挑战者返回 $\tilde{\Gamma}$..如果 b_1=0,挑战者返回随机混淆电路 $\tilde{\Gamma}$;如果 b_1=1,挑战者也返回随机混淆电路 $\tilde{\Gamma}$.
3. 敌手$\mathcal{A}_1$ 向挑战者提交 x.
4. 如果 b_1=0,则挑战者调用定义 7 的步骤 3,生成 $(\tilde{\Gamma}',gsk)\leftarrow \boldsymbol{Gb}.\boldsymbol{Garble}(1^{\lambda},C)$ 和 $\tilde{c}\leftarrow \boldsymbol{Gb}.\boldsymbol{Enc}(gsk,x)$, 并将 $\tilde{\Gamma}'$ 所有门电路的密文对的前半部分和后半部分别与 $\boldsymbol{R}_5^{-1}$ 相乘生成 $\tilde{\Gamma}$, 并返回 $\tilde{\Gamma}$ 和 $\tilde{c}$.
5. 如果 b_1=1,则挑战者调用定义 7 的步骤 4,生成 $(\tilde{\Gamma}'',state)\leftarrow S(1^{\lambda},T(C))$ 和 $\tilde{c}\leftarrow S(C(x),state)$,并将 $\tilde{\Gamma}''$ 所有门电路的密文对的前半部分和后半部分别与 $\boldsymbol{R}_5^{-1}$ 相乘生成 $\tilde{\Gamma}$, 并返回 $\tilde{\Gamma}$ 和 $\tilde{c}$.
6. 最后,敌手$\mathcal{A}_1$ 输出猜测比特 b_1',如果 $b_1'=b_1$,敌手$\mathcal{A}_1$ 获胜.

设 b 是定义 7 的挑战比特,从上述适应性安全实验可以得出:

$$pr[\boldsymbol{Exp}_{\mathcal{A},s}^{static}(1^{\lambda},1)=1]=pr[\boldsymbol{Exp}_{\mathcal{A}_1,\mathcal{S}_1}^{adaptive}(1^{\lambda},1)=1],$$

$$pr[\boldsymbol{Exp}_{\mathcal{A},s}^{static}(1^{\lambda},0)=1]=pr[\boldsymbol{Exp}_{\mathcal{A}_1,\mathcal{S}_1}^{adaptive}(1^{\lambda},0)=1].$$

所以,等式(3)成立.

5 可重用的密码转置防火墙

接下来,将第 4.1 节的协议应用于密码转置防火墙[23],从而提供一种密码转置防火墙的新模式,即可重用的密码转置防火墙.密码转置防火墙聚焦于用户与外部通讯的密码安全,可解释为一个状态算法,应用于用户发送和接收已由某些密码算法处理的消息.两方计算下的私有函数计算 PFE 协议有一对参与方分别是 Alic 和 Bob, Alice 拥有私有的函数 f,Bob 拥有私有的输入 x,双方计算 $f(x)$并保证输入 x 和函数 f 的隐私性和结果的正确性.

PFE 协议的密码转置防火墙主要技术工具是可重随机化的不经意传输和可重随机化的混淆电路.Bob(Bob 的密码转置防火墙)和 Alice(Alice 的密码转置防火墙)执行可重随机化的不经意传输,Alice 向 Bob 透露电路输入电线的两个重随机化标签其中之一,而不了解确切是哪一个.接下来,Alice 的密码转置防火墙将重随机化的混淆电路发送给 Bob,Bob 就可以根据重随机化标签计算电路.然而,Alice 的密码转置防火墙重用用于重随机化电路的 Yao 的混淆电路是不安全的,这是因为重复的重随机化等同于混淆电路的重用.所以,我们提出一种可重用的密码转置防火墙,用户一次生成混淆电路,接下来,密码转置防火墙可安全地重随机化多次.

5.1 不经意传输

Naor-Pinks/Aiello-Ishai-Reingold 证明了不经意传输协议可在诚实但好奇模式下是安全的[28,29].不经意传输协议有两个参与方:发送方 Alice 和接收方 Bob.Alice 的输入为 $a_0,a_1\in\{0,1\}$;Bob 的输入是选择的比特 $b\in\{0,1\}$.Alice 和 Bob 之间协议实现如图 3 所示,其中,$\mathbb{G}$是阶为 q 的群.

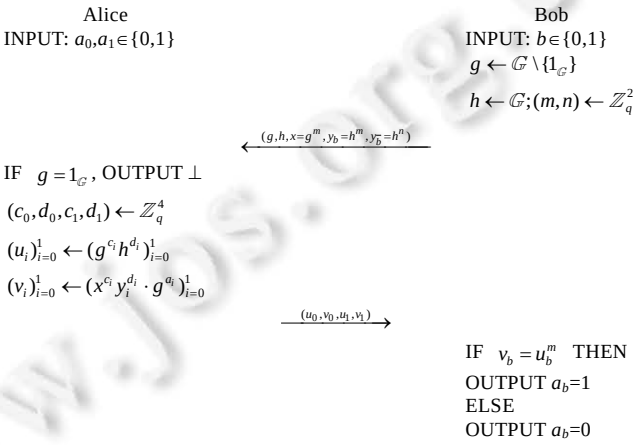

Fig.3 Oblivious transfer (OT) protocol

图 3 不经意传输协议 OT

定义 10(可重随机化的不经意传输). 如图 3 所示两消息的不经意传输协议,设 $msg_1=(g,h,x,y_0,y_1)$是第 1 轮的消息,$msg_2=(u_0,v_0,u_1,v_1)$是第 2 轮的消息.

不经意传输协议是可重随机化的,若存在输入为 msg_1,msg_2 和随机数 r 的有效算法 reRandOT,即使给定 r,b,a_0,a_1,msg_1,msg_2,以下两个分布是计算上不可区分的.

$$reRandOT(msg_1,msg_2,r)\overset{C}{\equiv}(u_0,v_0,u_1,v_1).$$

5.2 私有函数计算的可重用密码转置防火墙

Alice 的 PFE 可重用密码转置防火墙 W_A 的构造与可验证计算协议的构造非常类似,技术上的区别在于,W_A 需要定义如何构造不经意传输的重随机化.因为 Bob 的可重用密码转置防火墙 W_B 与文献[23]的不经意传输协议重随机化是一致的,故此处省略.构造 Alice 的 PFE 可重用转置密码防火墙如图 4 所示.本质上该构造与可验证计算协议相同,安全要求相同,证明也类似.

定理 3. 若 DDH 假设成立,如图 4 所示的 Alice 的可重用密码转置防火墙对于函数 f 是正确的和安全的.

证明:Alice 的可重用密码转置防火墙的正确性证明如下.

$\mathbb{G}$是素数阶 q 的循环群,定义 $g\leftarrow\mathbb{G}\setminus\{1_{\mathbb{G}}\},k=\log_g^h.(g,h,x,y_0,y_1)$是 Bob 发送给 Alice 的初始消息,Bob 的防火墙安全性可直接由 DDH 假设得出.恶意模式下的可重随机化不经意传输安全和可重随机化混淆电路安全确保 Alice 的防火墙是安全的.接下来证明可重随机化不经意传输的安全性.

Fig.4 Alice's reusable cryptographic rerverse firewall for the private function evaluation protocol

图 4 私有函数计算协议中 Alice 的可重用密码转置防火墙

W_B·(W_A·*Alice*)对来自 Bob 的任意有效消息(g,h,x,y_0,y_1)以不可忽略的概率做出一致随机有效响应(u_0',v_0',u_1',v_1'). 若$(g,h,x,y_b,y_{\bar{b}}) \neq (g,g^k,g^m,g^{km},g^{kn})$, 则$(u_0',v_0',u_1',v_1')$是一致随机群元素.为了说明这一点,需要利用等式$\log_g^{u_i} = c_i + kd_i$和$\log_g^{v_i/g^{a_i}} = mc_i + kmd_i$(或者$\log_g^{v_i/g^{a_i}} = mc_i + knd_i$).若 $y_b \neq g^{km}$(或 $y_b \neq g^{kn}$), (u_0',v_0',u_1',v_1')的分布是一致和独立的.

Alice 的可重用密码转置防火墙安全意味已知$(g,h,x,y_b,y_{\bar{b}})$可以模拟(u_0',v_0',u_1',v_1').为了证明这一点,需要给定$(g,h,x,y_b,y_{\bar{b}})$和有效计算结果a_b,构造模拟器Sim_A.模拟器的操作如下.

1. 输入比特σ,Sim_A产生$(g,h,x,y_\sigma,y_{\bar{\sigma}})$.
2. 随机选择$(a_0,a_1) \leftarrow \{0,1\}$和$(c_0,c_1,d_0,d_1) \leftarrow \mathbb{Z}_q^4$.
3. 计算$(u_i,v_i)_{i=0}^1 \leftarrow \{g^{c_i}h^{d_i}, x^{c_i}y_i^{d_i} \cdot g^{a_i}\}_{i=0}^1$.
4. 同样地,选择$(c_0',c_1',d_0',d_1') \leftarrow \mathbb{Z}_q^4$.
5. 同样地,计算$(u_i',v_i')_{i=0}^1 \leftarrow \{g^{c_i'}h^{d_i'} \cdot u_i, x^{c_i'}y_i^{d_i'} \cdot v_i\}_{i=0}^1$.
6. 输出$(\sigma,u_0',v_0',u_1',v_1')$.

Sim_A的输出与在真实协议中 Bob 接收来自 Alice 的可重用密码防火墙的消息是不可区分的,这可从 DDH 假设得出. □

References:

[1] Kilian J. A note on efficient zero-knowledge proofs and arguments (extended abstract). In: Proc. of the 24th Annual ACM Symp. on Theory of Computing. ACM Press, 1992. 723−732.

[2] Micali S. CS proofs (extended abstract). In: Proc. of the 35th Annual Symp. on Foundations of Computer Science. IEEE Press, 1994. 436−453.

[3] Gennaro R, Gentry C, Parno B. Non-interactive verifiable computing: Outsourcing computation to untrusted workers. In: Rabin T, ed. Proc. of the 30th Annual Cryptology Conf. (CRYPTO 2010). Heidelberg: Springer-Verlag, 2010. 465−482.

[4] Applebaum B, Ishai Y, Kushilevitz E. From secrecy to soundness: efficient verification via secure computation. In: Abramsky S, *et al*., eds. Proc. of the 37th Int'l Colloquium on Automata, Languages, and Programming (ICALP 2010). Heidelberg: Springer-Verlag, 2010. 152−163.

[5] Asharov G, Jain A, Lopez-Alt A, Tromer E, Vaikuntanathan V, Wichs D. Multiparty computation with low communication, computation and interaction via threshold FHE. In: Pointcheval D, Johansson T, eds. Proc. of the 31st Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2012). Heidelberg: Springer-Verlag, 2012. 483−501.

[6] Ben-Sasson E, Chiesa A, Genkin D, Tromer E, Virza M. SNARKs for C: Verifying program executions succinctly and in zero knowledge. In Canetti R, Garay JA, eds. Proc. of the 33rd Annual Cryptology Conf. (CRYPTO 2013). Heidelberg: Springer-Verlag, 2013. 99−108.

[7] Choi SG, Katz J, Kumaresan R, Cid C. Multi-client non-interactive verifiable computation. In Sahai A, ed. Proc. of the 10th Theory of Cryptography Conf. (TCC 2013). Heidelberg: Springer-Verlag, 2013. 499−518.

[8] Chung KM, Kalai YT, Liu FH, Raz R. Memory delegation. In Rogaway P, ed. Proc. of the 31st Annual Cryptology Conf. (CRYPTO 2011). Heidelberg: Springer-Verlag, 2011. 151−168.

[9] Chung KM, Kalai Y, Vadhan S. Improved delegation of computation using fully homomorphic encryption. In: Rabin T, ed. Proc. of the 30th Annual Cryptology Conf. (CRYPTO 2010). Heidelberg: Springer-Verlag, 2010. 483−501.

[10] Fiore D, Gennaro R, Pastro V. Efficiently verifiable computation on encrypted data. In: Proc. of the 21st ACM Conf. on Computer and Communications Security (CCS 2014). ACM Press, 2014. 844−855.

[11] Ananth P, Chandran N, Goyal V, Kanukurthi B, Ostrovsky R. Achieving privacy in verifiable computation with multiple servers—Without FHE and without pre-processing. In: Krawczyk H, ed. Proc. of the 20th Int'l Conf. on Practice and Theory of Public-Key Cryptography (PKC 2014). Heidelberg: Springer-Verlag, 2014. 149−166.

[12] Badrinarayanan S, Goyal V, Jain A, Sahai A. Verifiable functional encryption. In: Cheon J, Takagi T, eds. Proc. of the 22nd Annual Int'l Conf. on the Theory and Applications of Cryptology and Information Security (ASIACRYPT 2016). Heidelberg: Springer-Verlag, 2016. 557−587.

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

[14] Yao AC. Protocols for secure computations (extended abstract). In: Shamir A, ed. Proc. of the 23rd Annual Symp. on Foundations of Computer Science. IEEE Press, 1982. 160−164.

[15] Goldwasser S, Kalai Y, Popa RA, Vaikuntanathan V, Zeldovich N. Reusable garbled circuits and succinct functional encryption. In: Proc. of the 45th Annual ACM Symp. on Theory of Computing (STOC 2013). ACM Press, 2013. 555−564.

[16] Agrawal S. Stronger security for reusable garbled circuits, general definitions and attacks. In: Katz J, Shacham H, eds. Proc. of the 37th Annual Cryptology Conf. (CRYPTO 2017). Heidelberg: Springer-Verlag, 2017. 3−35.

[17] Gentry C. Fully homomorphic encryption using ideal lattices. In: Proc. of the 41st Annual ACM Symp. on Theory of Computing (STOC 2009). ACM Press, 2009. 169−178.

[18] Brakerski Z, Vaikuntanathan V. Lattice-based FHE as secure as PKE. In: Proc. of the 5th Conf. on Innovations in Theoretical Computer Science. ACM Press, 2014. 1−12.

[19] Gentry C, Halevi S, Vaikuntanathan V. I-hop homomorphic encryption and re-randomizable Yao circuits. In: Rabin T, ed. Proc. of the 30th Annual Cryptology Conf. (CRYPTO 2010). Heidelberg: Springer-Verlag, 2010. 155−172.

[20] Halevi S, Lindell Y, Pinkas B. Secure computation on the Web: Computing without simultaneous interaction. In: Rogaway P, ed. Proc. of the 31st Annual Cryptology Conf. (CRYPTO 2011). Heidelberg: Springer-Verlag, 2011. 132−150.

[21] Atallah MJ, Pantazopoulos KN, Rice JR. Secure outsourcing of scientific computations. Advances in Computers, 2002,(54): 215−272.

[22] Kilian J. Founding cryptography on oblivious transfer. In: Proc. of the 20th Annual ACM Symp. on Theory of Computing. ACM Press, 1988. 20−31.

[23] Mironov I, Stephens-Davidowitz N. Cryptographic reverse firewalls. In: Oswald E, Fischlin M, eds. Proc. of the 34th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2015). Heidelberg: Springer-Verlag, 2015. 657−686.

[24] Boneh D, Halevi S, Hamburg M, Ostrovsky R. Circular-secure encryption from decision Diffie-Hellman. In: Wagner D, ed. Proc. of the 28th Annual Cryptology Conf. (CRYPTO 2008). Heidelberg: Springer-Verlag, 2008. 108−125.

[25] Barak B, Garg S, Kalai YT, Paneth O, Sahai A. Protecting obfuscation against algebraic attacks. In: Nguyen PQ, Oswald E, eds. Proc. of the 33rd Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2014). Heidelberg: Springer-Verlag, 2014. 221−238.

[26] Lindell Y. How to simulate it—A tutorial on the simulation proof technique. IACR Cryptology ePrint Archive: Report 2016/046 (2016), 2016. http://eprint.iacr.org/2016/046.pdf

[27] Bellare M, Hoang VT, Rogaway P. Adaptively secure garbling with applications to one-time programs and secure outsourcing. In: Wang X, Sako K, eds. Proc. of the 18th Int'l Conf. on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2012). Heidelberg: Springer-Verlag, 2012. 134−153.

[28] Aiello W, Ishai Y, Reingold O. Priced oblivious transfer: How to sell digital goods. In: Pfitzmann B, ed. Proc. of the 20th Annual Int'l Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2011). Heidelberg: Springer-Verlag, 2001. 119−135.

[29] Naor M, Pinkas B. Efficient oblivious transfer protocols. In: Proc. of the 12th Annual ACM-SIAM Symp. on Discrete Algorithms. ACM Press, 2001. 448−457.

赵青松(1973－),男,江苏连云港人,博士,讲师,CCF 专业会员,主要研究领域为信息安全和隐私,公钥密码学.

刘西蒙(1988－),男,博士,教授,CCF 专业会员,主要研究领域为应用密码学,数据安全,安全计算,公钥密码学.

曾庆凯(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为信息安全,分布计算.

徐焕良(1963－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布计算,数据科学与计算.