通用 MPC 协议构造

混淆电路基本构造

张聪
zhangcong@iie.ac.cn

中国科学院信息工程研究所国家重点实验室

2023 年 2 月 4 日

1. GC 定义
2. GC 构造
3. 参考文献

1. GC 定义
2. GC 构造
3. 参考文献

GC 背景

混淆电路 (Garbled Circuit, GC) 最早由姚期智 [Yao86] 提出，用于构造通用两方计算协议，其特点是轮数为常数，通信量大，且只能用于两方计算。直到 2004 年，Lindell 和 Pinkas[LP04] 才给出 GC 协议的正式描述与安全证明。随后，Bellare 等人 [BHR12] 将 GC 看作一个密码组件，而不是协议，给出了 GC 的基本定义，以及需要满足的安全性质。

后续发展：

- 半诚实情况：点置换技术 (point-and-permute) [BMR90]，Free-XOR 技术 [KS08]，行约减技术 (grable row reduction,GRR)[NPS99; Pin+09]，半门技术 (half-gate) [ZRE15]，分片切割技术 (slicing-and-dicing) [RR21]。
- 恶意情况：Cut-and-Choose：[LP07; LP11; Lin13; LR14; LR15; RR16; WMK17]；认证 GC：[WRK17; Kat+18]。
- 算术电路：[AIK11; BMR16; Ben18; Bal+19; MW19]。

电路定义

定义 (电路)

一个电路是指一个六元组 $f = (n, m, q, A, B, G)$。这里 $n \geq 2$ 是输入个数，$m \geq 1$ 是输出个数，$q \geq 1$ 是门的个数。令 $r = n + q$ 表示线的个数。令 $Inputs := \{1, \ldots, n\}, Wires := \{1, \ldots, n+q\}, OutputWires := \{n+q-m+1, \ldots, n+q\}, Gates := \{n+1, \ldots, n+q\}$。定义
$A : Gates \to Wires/OutputWires$ 是一个识别门的第一个输入线的函数，
$B : Gates \to Wires/OutputWires$ 是一个识别门的第二个输入线的函数，
$G : Gates \times \{0,1\}^2 \to \{0,1\}$ 是计算每个门的函数。我们要求 $\forall g \in Gates, A(g) < B(g) < g$。用 $Topo(f) = (n, m, q, A, B)$ 表示电路 f 的拓扑。

电路定义

- 每个非输入线都是某个门的输出线
- 输出线不能是输入线，也不能是其他门的输入线
- 同一条输出线在输出中不得使用两次
- 要求 $A(g) < B(g) < g$ 确保对应于 f 的有向图是无环的，并且没有线两次输入给同一个门

电路定义

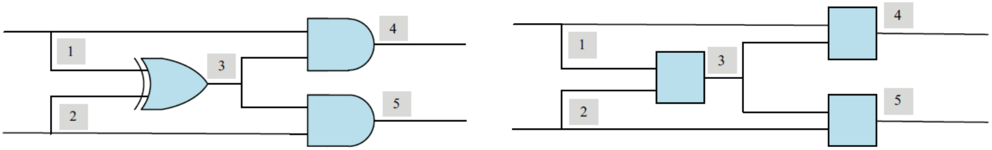

图 1: 左： 电路 $f = (n, m, q, A, B, G)$。它有 $n = 2$ 个输入， $m = 2$ 个输出， $q = 3$ 个门。门根据输出线编号为 $3, 4, 5$。映射函数 $A(3) = 1, B(3) = 2, A(4) = 1, B(4) = 3, A(5) = 2, B(5) = 3$。门符号表示 $G_1(\cdot, \cdot) = XOR, G_2(\cdot, \cdot) = G_3(\cdot, \cdot) = AND$。右： 与左侧电路相对应的电路拓扑 $Topo(f)$。

电路定义

电路求值：定义电路求值算法 ev 如下，该算法以电路 f，和输入 $x = x_1x_2\ldots x_n$ 为输入：

1. $(n, m, q, A, B, G) \leftarrow f$
2. 对 $g \in [n+1, n+q] : a \leftarrow A(g), b \leftarrow B(g), x_g \leftarrow G_g(x_a, x_b)$
3. 输出 $x_{n+q-m+1}x_{n+q-m+2}\ldots x_{n+q}$

GC 定义

一个混淆方案 (Garbling Scheme) 是指算法五元组 $\mathcal{G} = (\mathrm{Gb}, \mathrm{En}, \mathrm{Ev}, \mathrm{De}, \mathrm{ev})$，第一个算法是概率算法，其他都是确定性算法:

- $(F, e, d) \leftarrow \mathrm{Gb}(1^\kappa, f)$
- $X \leftarrow \mathrm{En}(e, x)$
- $Y \leftarrow \mathrm{Ev}(F, X)$
- $y \leftarrow \mathrm{De}(d, Y)$
- $y \leftarrow \mathrm{ev}(f, x)$

正确性：对任意电路 f，任意输入 x，$(F, e, d) \leftarrow \mathrm{Gb}(1^\kappa, f)$，除去可忽略的概率，有 $\mathrm{De}(d, \mathrm{Ev}(F, \mathrm{En}(e, x))) = \mathrm{ev}(f, x)$.

GC 定义

对电路 $f=(n,m,q,A,B,G)$，用 $\Phi(f)$ 表示泄漏的电路信息，主要有以下三种：

- $\Phi_{size}(f)=(n,m,q)$
- $\Phi_{topo}(f)=Topo(f)=(n,m,q,A,B)$
- $\Phi_{circ}(f)=f$

GC 定义

隐私性 (Privacy)：存在模拟器 $\mathcal{S}$，对任意电路 f 和输入 x，有下述分布不可区分：

$$\{(F,X,d)|(F,e,d) \leftarrow \mathrm{Gb}(1^\kappa,f), X := \mathrm{En}(e,x)\} \approx \{(F,X,d)|(F,X,d) \leftarrow \mathcal{S}(1^\kappa,\Phi(f),\mathrm{ev}(f,x))\}$$

不经意性 (Obliviousness)：存在模拟器 $\mathcal{S}$，对任意电路 f 和输入 x，有下述分布不可区分：

$$\{(F,X)|(F,e,d) \leftarrow \mathrm{Gb}(1^\kappa,f), X := \mathrm{En}(e,x)\} \approx \{(F,X)|(F,X) \leftarrow \mathcal{S}(1^\kappa,\Phi(f))\}$$

认证性 (Authenticity)：对任意电路 f 和输入 x，任意 PPT 的敌手 $\mathcal{A}$，下述概率是可忽略的：

$$Pr[\mathrm{De}(d,Y) \notin \{\mathrm{ev}(f,x),\perp\}|(F,e,d) \leftarrow \mathrm{Gb}(1^\kappa,f), X := \mathrm{En}(e,x), Y \leftarrow \mathcal{A}(F,d,X)]$$

1 GC 定义

2 GC 构造

3 参考文献

加密方案

定义 ([LP04])

令 $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ 是一个 SKE 方案，令 $\mathrm{Range}_\kappa(k) := \{\mathrm{Enc}_k(x)\}_{x\in\{0,1\}^\kappa}$ 表示在密钥 k 下密文的值域，则：

1. 称 $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ 具有 elusive range，如果对每个 PPT 算法 A，有 $Pr_{k\leftarrow\mathrm{KeyGen}(1^\kappa)}[A(1^\kappa) \in \mathrm{Range}_\kappa(k)] = \mathrm{negl}(\kappa)$
2. 称 $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Dec})$ 具有 efficiently verifiable range，如果存在 PPT 算法 M 使得 $M(1^\kappa, c, k) = 1$ 当且仅当 $c \in \mathrm{Range}_\kappa(k)$

当 $c \notin \mathrm{Range}_\kappa(k)$ 时，定义 $\mathrm{Dec}_k(c) = \perp$.

加密方案

方案构造：令 $\{f_k : \{0,1\}^\kappa \to \{0,1\}^{2\kappa}\}_{k\in\{0,1\}^\kappa}$ 是伪随机函数族：

- KeyGen(1^κ)：随机选择 $k \in \{0,1\}^\kappa$.
- $\mathrm{Enc}_k(m)$:
  1. $r \leftarrow \{0,1\}^\kappa$
  2. 计算 $f_k(r), f_k(r) \oplus (m||0^\kappa)$
  3. 输出 $c := (c_1, c_2) = (r, f_k(r) \oplus (m||0^\kappa))$
- $\mathrm{Dec}_k(c)$:
  1. $c = (c_1, c_2)$.
  2. $m = (m_1||m_2) := f_k(c_1) \oplus c_2$.
  3. 如果 $m_2 = 0^\kappa$，输出 m_1，否则输出 $\perp$.

容易验证，此方案满足 elusive range 和 efficiently verifiable range。

加密方案

为了证明 GC 方案的安全性，考虑双重加密实验如下：

$\mathrm{Expt}_{\mathcal{A}}^{double}(\kappa, \sigma)$：

1. 敌手 $\mathcal{A}$ 输入 1^κ，输出两个密钥 k_0, k_1 和两组消息 (x_0, y_0, z_0) 和 (x_1, y_1, z_1)。
2. 挑战者随机选两个密钥 $k_0', k_1' \leftarrow \mathrm{KeyGen}(1^\kappa)$。
3. $\mathcal{A}$ 收到挑战密文 $\langle \mathrm{Enc}_{k_0}(\mathrm{Enc}_{k_1'}(x_\sigma)), \mathrm{Enc}_{k_0'}(\mathrm{Enc}_{k_1}(y_\sigma)), \mathrm{Enc}_{k_0'}(\mathrm{Enc}_{k_1'}(z_\sigma)) \rangle$。并且可以访问 oracle: $\mathrm{Enc}_{(\cdot)}(\mathrm{Enc}_{k_1'}(\cdot)), \mathrm{Enc}_{k_0'}(\mathrm{Enc}_{(\cdot)}(\cdot))$
4. $\mathcal{A}$ 输出一个 σ 的猜测 b 作为实验的输出。

加密方案

定义

我们称加密方案 $(\text{KeyGen}, \text{Enc}, \text{Dec})$ 满足选择双重加密 (chosen double encryption) 攻击下的安全性，如果对每个非一致 PPT 敌手 $\mathcal{A}$，每个多项式 $p(\cdot)$，所有足够大的 κ，有

$$|Pr[\text{Expt}_{\mathcal{A}}^{double}(\kappa, 0) = 1] - Pr[\text{Expt}_{\mathcal{A}}^{double}(\kappa, 1) = 1]| < \frac{1}{p(n)}$$

定理 ([LP04])

如果加密方案 $(\text{KeyGen}, \text{Enc}, \text{Dec})$ 满足 *IND-CPA* 安全性，则该方案满足选择双重加密 *(chosen double encryption)* 攻击下的安全性。

GC 描述

$\mathrm{Gb}(1^\kappa, f)$:

1. $(n,m,q,A,B,G) \leftarrow f$
2. 对每条线 $i \in [n+q]$： $k_i^0, k_i^1 \leftarrow \mathrm{KeyGen}(1^\kappa)$
3. 对每个门 $l \in [n+1, n+q]$： $i \leftarrow A(l), j \leftarrow B(l)$，计算

$$c_{0,0} = Enc_{k_i^0}(Enc_{k_j^0}(k_l^{G_l(0,0)}))$$
$$c_{0,1} = Enc_{k_i^0}(Enc_{k_j^1}(k_l^{G_l(0,1)}))$$
$$c_{1,0} = Enc_{k_i^1}(Enc_{k_j^0}(k_l^{G_l(1,0)}))$$
$$c_{1,1} = Enc_{k_i^1}(Enc_{k_j^1}(k_l^{G_l(1,1)}))$$

   选随机置换 π，令门 l 的garbled table$P_l = (c_0, c_1, c_2, c_3) := \pi(c_{0,0}, c_{0,1}, c_{1,0}, c_{1,1})$
4. $F := (n,m,q,A,B,\{P_{n+i}\}_{i\in[q]}), e := \{k_i^0, k_i^1\}_{i\in[n]}, d := \{(b, k_{n+q-m+i}^b)\}_{i\in[m], b\in\{0,1\}}$
5. 输出 (F, e, d)

GC 描述

$\mathrm{En}(e, x)$:

1. $\{k_i^0, k_i^1\}_{i\in[n]} \leftarrow e$
2. $x_1 x_2 \ldots x_n \leftarrow x$
3. 输出 $X := \{k_i^{x_i}\}_{i\in[n]}$

GC 描述

Ev(F, X):

1. $(n, m, q, A, B, \{P_l\}) \leftarrow F$
2. $(k_1, \ldots, k_n) \leftarrow X$
3. 对每个门 $l \in [n+1, n+q]$: $i \leftarrow A(l), j \leftarrow B(l), (c_0, c_1, c_2, c_3) \leftarrow P_l$.
   对 $t \in [0, 3]$:
   1. $m_t := \mathrm{Dec}_{k_i}(\mathrm{Dec}_{k_j}(c_t))$
   2. 如果 $m_t \neq \perp$，令 $k_l := m_t$
4. 输出 $Y := \{k_{n+q-m+i}\}_{i \in [m]}$

GC 描述

$\mathrm{De}(d, Y)$:

1. $\{(b, k^b_{n+q-m+i})\}_{i\in[m],b\in\{0,1\}} \leftarrow d$
2. $\{k_{n+q-m+i}\}_{i\in[m]} \leftarrow Y$
3. 对 $i \in [m], b \in \{0,1\}$: 如果 $k_{n+q-m+i} = k^b_{n+q-m+i}$，令 $y_i := b$.
4. 输出 $y := y_1 \ldots y_m$

GC 协议构造

$P_1(x \in \{0,1\}^n)$ $\qquad$ $P_2(y \in \{0,1\}^n)$

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

$\{k_i^0, k_i^1\}_{i\in[2n]} \leftarrow e$

$\xrightarrow{F, d, \{k_i^{x_i}\}_{i\in[n]}}$

$i \in [n]$:

$\xrightarrow{(k_{n+i}^0, k_{n+i}^1)} \boxed{\text{OT}} \xleftarrow{y_i}$

$\boxed{\text{OT}} \xrightarrow{k_{n+i}^{y_i}}$

$X := \{k_i^{x_i}, k_{n+i}^{y_i}\}_{i\in[n]}$

$f(x,y) := \mathrm{De}(d, \mathrm{Ev}(F, X))$

$\xleftarrow{f(x,y)}$

安全证明

定义 (半诚实安全)

设 f 是确定性函数。我们说在半诚实模型下，协议 π 安全地计算 $f(x,y)$，如果存在 PPT 的 S_1, S_2 使得

$$\{S_1(x, f(x,y))\}_{x,y\in\{0,1\}^*} \equiv^c \{view_1^\pi(x,y)\}_{x,y\in\{0,1\}^*}$$
$$\{S_2(y, f(x,y))\}_{x,y\in\{0,1\}^*} \equiv^c \{view_2^\pi(x,y)\}_{x,y\in\{0,1\}^*}$$

安全证明

证明:

P_1 是半诚实的:

P_1 在协议 π 执行中的 view 只包括了在 OT 中收到的消息和最后一条 P_2 发的 $f(x,y)$,即:

$\{view_1^\pi(x,y)\} = \{(x, r_C, R_1^{OT}((k_{n+1}^0, k_{n+1}^1), y_1), \ldots, R_1^{OT}((k_{2n}^0, k_{2n}^1), y_n), f(x,y))\}$

这里的 $R_1^{OT}((k_{n+j}^0, k_{n+j}^1), y_j)$ 表示,P_1 在执行第 j 次 OT 时所看到的真实的 view,即 $view_1^{OT}((k_{n+j}^0, k_{n+j}^1), y_j)$,$j \in [n]$。

为了生成模拟的 view,可以直接调用 n 次 OT 的模拟器 $S_1^{OT}(k_{n+j}^0, k_{n+j}^1)$,再在随后加上输出 $f(x,y)$ 就可以生成模拟的 P_1 的 view。

安全证明

下面描述模拟器 $S_1(x,f(x,y))$ 的构造如下：

1. 随机选 P_1 的随机带 r_C，并用 r_C 生成混淆电路。用 $k^0_{n+1},k^1_{n+1},\dots,k^0_{2n},k^1_{2n}$ 表示生成的对应 P_2 输入线的 label。
2. 对 $i\in[n]$，调用 OT 的模拟器 $S_1^{OT}(k^0_{n+i},k^1_{n+i})$ 得到 P_1 在第 i 次 OT 中 view 的模拟。
3. 将输出 $f(x,y)$ 放到模拟的 view 中，即输出：

$\{(x,r_C,S_1^{OT}(k^0_{n+1},k^1_{n+1}),\dots,S_1^{OT}(k^0_{2n},k^1_{2n}),f(x,y))\}$

下面说明 $\{S_1(x,f(x,y))\}_{x,y\in\{0,1\}^*}\equiv^c\{view_1^\pi(x,y)\}_{x,y\in\{0,1\}^*}$，即

$\{(x,r_C,R_1^{OT}((k^0_{n+1},k^1_{n+1}),y_1),\dots,R_1^{OT}((k^0_{2n},k^1_{2n}),y_n),f(x,y))\}$

$\equiv^c\{(x,r_C,S_1^{OT}(k^0_{n+1},k^1_{n+1}),\dots,S_1^{OT}(k^0_{2n},k^1_{2n}),f(x,y))\}$

安全证明

我们用 hybrid argument 的方法证明。用 $H_i(x, y, r_C)$ 表示如下分布：

$\{(x, r_C, S_1^{OT}(k_{n+1}^0, k_{n+1}^1), \ldots, S_1^{OT}(k_{n+i}^0, k_{n+i}^1), R_1^{OT}((k_{n+i+1}^0, k_{n+i+1}^1), y_{i+1}), \ldots, R_1^{OT}((k_{2n}^0, k_{2n}^1), y_n)$

则 $H_0(x, y, r_C) \equiv view_1^{\pi}(x, y)$,$H_n(x, y, r_C) \equiv S_1(x, f(x, y))$

要证 $H_0(x, y, r_C) \equiv^c H_n(x, y, r_C)$，用反证法，假设存在 PPT 的 D 可以区分，即

$|Pr[D(H_0(x, y, r_C)) = 1] - Pr[D(H_n(x, y, r_C)) = 1]| > \frac{1}{p(n)}$，则存在 i 使得

$|Pr[D(H_{i-1}(x, y, r_C)) = 1] - Pr[D(H_i(x, y, r_C)) = 1]| > \frac{1}{np(n)}$

这说明 D 可以区分 $H_{i-1}(x, y, r_C)$ 和 $H_i(x, y, r_C)$。但 $H_{i-1}(x, y, r_C)$ 和 $H_i(x, y, r_C)$ 的唯一的不同是第 i 个 OT 的副本是 $S_1^{OT}(k_{n+i}^0, k_{n+i}^1)$ 还是 $R_1^{OT}((k_{n+i}^0, k_{n+i}^1), y_i)$。但是由 OT 的安全性我们知 $\{S_1^{OT}(k_{n+i}^0, k_{n+i}^1)\} \equiv^c \{R_1^{OT}((k_{n+i}^0, k_{n+i}^1), y_{i+1})\}$。也就是说上述 D 是可以区分 $S_1^{OT}(k_{n+i}^0, k_{n+i}^1)$ 和 $R_1^{OT}((k_{n+i}^0, k_{n+i}^1), y_i)$ 的，矛盾。

(注：我们需要区分器是 non-uniform 的是因为需要辅助输入 x, y, r_C, i)

安全证明

P_2 是半诚实的：
我们首先看一下真实协议中 P_2 的 view 是什么，包括了 P_1 发的一个混淆电路 (F, d)，还有对应 P_1 输入线的 label，$k_1^{x_1}, \ldots, k_n^{x_n}$，还有在 OT 中看到的消息 $R_2^{OT}((k_{n+i}^0, k_{n+i}^1), y_i)$。即：
$\{view_2^\pi(x, y)\} = \{(y, F, d, \{k_i^{x_i}\}_{i\in[n]}, R_2^{OT}((k_{n+1}^0, k_{n+1}^1), y_1), \ldots, R_2^{OT}((k_{2n}^0, k_{2n}^1), y_n))\}$
要想模拟这个 view，可以看一下各个部分的如何模拟，首先 OT 部分比较简单，可以直接调用 OT 的模拟器 $S_2^{OT}(y_i, k_{n+i}^{y_i})$ 模拟 $R_2^{OT}((k_{n+i}^0, k_{n+i}^1), y_i)$，而对于 P_1 输入线的 label，由于这就是一些独立随机选取的密钥，可以直接用加密方案的密钥生成算法生成一系列独立随机的密钥即可，即 $k_i \leftarrow \mathrm{KeyGen}(1^\kappa), i \in [n]$。但接下来的问题是如何模拟 (F, d) 使得在模拟器不知道 P_1 的输入 x 的情况下，用这些随机的密钥 $k_1, \ldots, k_n$ 还能让 P_2 求出来正确的输出 $f(x, y)$。这就需要构造一个模拟的混淆电路 $(\tilde{F}, \tilde{d})$，这个模拟的 $(\tilde{F}, \tilde{d})$ 不管在任何密钥下，总是能求得 $f(x, y)$，因此可以不需要知道 x。

安全证明

为了做到这一点，可以考虑对每个门的 garbled table 的四个密文，全都加密同一个子密钥，这样输入密钥不会影响输出密钥的值。问题是，如何证明这个电路 $\tilde{F}$ 与真实的 F 不可区分。我们还是用 hybrid argument 的方法，先把 OT 中的真实 view，$R_2^{OT}((k_{n+i}^0, k_{n+i}^1), y_i)$ 替换成模拟的 $S_2^{OT}(y_i, k_{n+i}^{y_i})$，记为 $H_{OT}(x, y)$，然后再考虑一系列的 $H_i(x, y)$ 逐个门替换真实的 garbled table，使得 $H_0(x, y)$ 包含了真实的 F，$H_q(x, y)$ 包含了模拟的 $\tilde{F}$。

安全证明

下面描述 $\tilde{F}$ 的构造方法：

对于电路 f 的每条线 i，随机生成 2 个独立的随机密钥 $k_i, k_i', i \in [n+q]$。接下来对每个门 l，设输入线是 $i = A(l), j = B(l)$，输出线是 l，对应密钥是 $k_i, k_i', k_j, k_j', k_l, k_l'$。我们令这个门的 garbled table 全都只加密 k_l（密钥 k_l' 完全不出现在密文中），即计算如下的密文：

$c_{0,0} = Enc_{k_i}(Enc_{k_j}(k_l))$
$c_{0,1} = Enc_{k_i'}(Enc_{k_j}(k_l))$
$c_{1,0} = Enc_{k_i}(Enc_{k_j'}(k_l))$
$c_{1,1} = Enc_{k_i'}(Enc_{k_j'}(k_l))$

然后再对这 4 个密文做一个随机置换作为这个门的模拟的 garbled table。也就是说，模拟的 garbled table 和真实的 garbled table 的区别在于，真实的 garbled table 中加密的密钥是根据这个门对输入密钥的计算 (即 $G_l(x_i, x_j)$) 得到的，而模拟的 garbled table 不管输入密钥是什么，都只加密一个密钥。

安全证明

对所有的门都这样做，现在还差输出解密表 d 没有构造。假设 $f(x,y)=z_1z_2\dots z_m\in\{0,1\}^m$，设输出线是 $w_{n+q-m+1},\dots,w_{n+q}$，对应 garbled table 里加密的密钥为 $k_{n+q-m+1},\dots,k_{n+q}$。则要想让这个模拟的电路求得 $f(x,y)$，只需令输出解密表为 $[(z_i,k_{n+q-m+i})(\bar{z}_i,k'_{n+q-m+i})]_{i\in[m]}$。

以上就是 $(\tilde{F},\tilde{d})$ 构造的完整描述。

下面描述模拟器 $S_2(y,f(x,y))$ 的构造如下：

1. 按照上述方式生成 $(\tilde{F},\tilde{d})$（即所有门的 garbled table 和输出解密表）。
2. 对 $i\in[n]$，调用 OT 的模拟器 $S_2^{OT}(y_i,k_{n+i})$ 得到 P_2 在第 i 次 OT 中 view 的模拟。
3. 输出 $\{(y,\tilde{F},\tilde{d},k_1,\dots,k_n,S_2^{OT}(y_1,k_{n+1}),\dots,S_2^{OT}(y_n,k_{2n}))\}$

下证 $\{S_2(y,f(x,y))\}_{x,y\in\{0,1\}^*}\equiv^c\{view_2^\pi(x,y)\}_{x,y\in\{0,1\}^*}$

先把 OT 用模拟的替换掉真实的，即令

$H_{OT}(x,y)=\{(y,F,d,k_1^{x_1},\dots,k_n^{x_n},S_2^{OT}(y_1,k_{n+1}),\dots,S_2^{OT}(y_n,k_{2n}))\}$

则根据刚刚 P_1 半诚实的情况可知 $\{H_{OT}(x,y)\}_{x,y\in\{0,1\}^*}\equiv^c\{view_2^\pi(x,y)\}_{x,y\in\{0,1\}^*}$

安全证明

下面只需证 $\{S_2(y,f(x,y))\}_{x,y\in\{0,1\}^*} \equiv^c \{H_{OT}(x,y)\}_{x,y\in\{0,1\}^*}$
考虑 hybrid 实验 $H_i(x,y)$ 表示逐个门用模拟的 garbled table 替换真实的 garbled table。在此之前，我们首先考虑一种新的描述构造 $(\tilde{F},\tilde{d})$ 的方法，这种新的构造方式需要知道真实的输入 x,y 的值，但是构造的结果与只知道 $y,f(x,y)$ 的构造方式是相同的。也就是说这是一个思想实验，可以把它看成另一种构造 $(\tilde{F},\tilde{d})$ 的描述。这样新的描述的好处是可以更好地描述我们的证明。
新的描述是这样的，首先当电路 f(没有 garble) 进行求值时，对于输入 (x,y)，每条线 w_a 上都会有一个对应的值 α。则构造 garbled circuit 时，这条线上会有两个密钥 k_a^0,k_a^1 对应真值 0，1，我们就称 k_a^α 为 active key，另一个密钥 $k_a^{1-\alpha}$ 为 inactive key。然后构造 $(\tilde{F},\tilde{d})$ 时，首先利用已知的 (x,y) 计算 $f(x,y)$，得到每条线上的真值，根据这个真值把每条线上的两个密钥分为 active key 和 inactive key。然后构造 garbled table 时，每个模拟的 garbled table 中加密的都是 active key。

安全证明

可以看出，上述新的构造 $(\tilde{F}, \tilde{d})$ 的描述和之前的描述构造出来的 $(\tilde{F}, \tilde{d})$ 的分布是完全相同的。这是因为，首先，在两种构造中，所有的门都只加密一个随机密钥，其次，在两种构造中，每个门的 garbled table 的四条密文的顺序都是随机的，最后，两种构造中，输出解密表的构造方法一致，都只能解密得到 $f(x, y)$。

对每个电路门 $i \in [q]$, 描述实验 $H_i(x, y)$:

首先 P_2 在 OT 中的 view 构造和 $H_{OT}(x, y)$ 中一样，对于混淆电路的构造，首先利用 (x, y) 构造真实的 (F, d)，并把每条线上的密钥根据真值标记为 active key 和 inactive key。然后，对前 i 个门 $n+1, \dots, n+i$ 的 garbled table 修改为新的描述中的构造，即这些门的 garbled table 只加密 active key。对剩下的门 $n+i+1, \dots, n+q$ 的 garbled table 保持不变，即和真实的 (F, d) 中一致。

安全证明

因此根据 $H_i(x,y)$ 的描述，有 $H_0(x,y) \equiv H_{OT}(x,y)$，$H_q \equiv \{S_2(y, f(x,y))\}$。因此只需证 $\{H_0(x,y)\}_{x,y\in\{0,1\}^*} \equiv^c \{H_q(x,y)\}_{x,y\in\{0,1\}^*}$
直觉上，我们需要用加密方案的安全性来说明这一点。
假设存在 PPT 的 D 可以区分 $H_0(x,y)$ 和 $H_q(x,y)$，即
$|Pr[D(H_0(x,y)) = 1] - Pr[D(H_q(x,y)) = 1]| > \frac{1}{p(n)}$，则存在 i 使得
$|Pr[D(H_{i-1}(x,y)) = 1] - Pr[D(H_i(x,y)) = 1]| > \frac{1}{qp(n)}$
下面我们构造调用 D 的 PPT 敌手 $\mathcal{A}$ 来攻击加密方案 ($\mathcal{A}$ 的辅助输入是 (x,y,i))

安全证明

首先回忆一下 $\mathrm{Expt}_{\mathcal{A}}^{double}(\kappa,\sigma)$：

1. 敌手 $\mathcal{A}$ 输入安全参数 1^κ，输出两个密钥 k_0,k_1 和两组消息 (x_0,y_0,z_0) 和 (x_1,y_1,z_1)，所有消息长度相同。

2. 挑战者随机选两个密钥 $\boldsymbol{k}'_0,\boldsymbol{k}'_1 \leftarrow \mathrm{KeyGen}(1^\kappa)$。

3.$\mathcal{A}$ 收到挑战密文 $\langle \mathrm{Enc}_{k_0}(\mathrm{Enc}_{\boldsymbol{k}'_1}(x_\sigma)),\mathrm{Enc}_{\boldsymbol{k}'_0}(\mathrm{Enc}_{k_1}(y_\sigma)),\mathrm{Enc}_{\boldsymbol{k}'_0}(\mathrm{Enc}_{\boldsymbol{k}'_1}(z_\sigma))\rangle$。并且可以访问 oracle: $\mathrm{Enc}_{(\cdot)}(\mathrm{Enc}_{k'_1}(\cdot)),\mathrm{Enc}_{k'_0}(\mathrm{Enc}_{(\cdot)}(\cdot))$

4.$\mathcal{A}$ 输出一个 σ 的猜测 b 作为实验的输出。

加密方案的安全性定义为

$$|Pr[\mathrm{Expt}_{\mathcal{A}}^{double}(\kappa,0)=1]-Pr[\mathrm{Expt}_{\mathcal{A}}^{double}(\kappa,1)=1]|<\frac{1}{p(n)}$$

由于 $\mathcal{A}$ 要利用 D 的能力，$\mathcal{A}$ 要构造 D 的输入

$$\{(y,\tilde{F}',\tilde{d}',k_1^{x_1},\ldots,k_n^{x_n},S_2^{OT}(y_1,k_{n+1}),\ldots,S_2^{OT}(y_n,k_{2n}))\}$$

安全证明

如果用 $(\tilde{F}_i, \tilde{d}_i)$ 表示如 H_i 描述的模拟的构造，则希望当 $\mathcal{A}$ 处在两种实验中所得到的的输入分别对应构造的 $(\tilde{F}', \tilde{d}')$ 正好是 $(\tilde{F}_{i-1}, \tilde{d}_{i-1})$ 和 $(\tilde{F}_i, \tilde{d}_i)$。
由于 $\mathcal{A}$ 知道 (x, y, i)，则 $\mathcal{A}$ 知道所有线上的 active 值和 inactive 值。对于门 i，设输入线为 $A(i) = a, B(i) = b$，输出线 i，输入线上的 active 值为 α, β，则 $\mathcal{A}$ 令在实验中选择的密钥为 k_a^α, k_b^β（即 active key），并记挑战者选的密钥为 $\boldsymbol{k_a^{1-\alpha}}, \boldsymbol{k_b^{1-\beta}}$（即 inactive key）。那么门 i 真实的 garbled table 为 (顺序随机)：

$$\mathrm{Enc}_{k_a^\alpha}(\mathrm{Enc}_{k_b^\beta}(k_i^{G_i(\alpha,\beta)}))$$

$$\mathrm{Enc}_{k_a^\alpha}(\mathrm{Enc}_{\boldsymbol{k_b^{1-\beta}}}(k_i^{G_i(\alpha,1-\beta)}))$$

$$\mathrm{Enc}_{\boldsymbol{k_b^{1-\alpha}}}(\mathrm{Enc}_{k_b^\beta}(k_i^{G_i(1-\alpha,\beta)}))$$

$$\mathrm{Enc}_{\boldsymbol{k_b^{1-\alpha}}}(\mathrm{Enc}_{\boldsymbol{k_b^{1-\beta}}}(k_i^{G_i(1-\alpha,1-\beta)}))$$

安全证明

模拟的 garbled table 为 (顺序随机):

$\mathrm{Enc}_{k_a^\alpha}(\mathrm{Enc}_{k_b^\beta}(k_i^{G_i(\alpha,\beta)}))$

$\mathrm{Enc}_{k_a^\alpha}(\mathrm{Enc}_{\boldsymbol{k_b^{1-\beta}}}(k_i^{G_i(\alpha,\beta)}))$

$\mathrm{Enc}_{\boldsymbol{k_b^{1-\alpha}}}(\mathrm{Enc}_{k_b^\beta}(k_i^{G_i(\alpha,\beta)}))$

$\mathrm{Enc}_{\boldsymbol{k_b^{1-\alpha}}}(\mathrm{Enc}_{\boldsymbol{k_b^{1-\beta}}}(k_i^{G_i(\alpha,\beta)}))$

$\mathcal{A}$ 可以自己计算 $\mathrm{Enc}_{k_a^\alpha}(\mathrm{Enc}_{k_b^\beta}(k_i^{G_i(\alpha,\beta)}))$，而对于后面的三条密文，$\mathcal{A}$ 可以令

$(x_0, y_0, z_0) := (k_i^{G_i(\alpha,1-\beta)}, k_i^{G_i(1-\alpha,\beta)}, k_i^{G_i(1-\alpha,1-\beta)})$，令

$(x_1, y_1, z_1) := (k_i^{G_i(\alpha,\beta)}, k_i^{G_i(\alpha,\beta)}, k_i^{G_i(\alpha,\beta)})$。

这样，当 $\sigma = 0$ 时，门 i 的 garbled table 就是真实的 garbled table，对应于 D 的输入就是 $H_{i-1}(x, y)$；当 $\sigma = 1$ 时，门 i 的 garbled table 就是模拟的 garbled table，对应于 D 的输入就是 $H_i(x, y)$。

安全证明

这里还有一个问题，就是 $\mathcal{A}$ 不知道门 i 输入线 (即 a, b) 的 inactive key，因为这是挑战者生成的，其他的线的 active key 和 inactive key 敌手 $\mathcal{A}$ 都可以知道，因为可以自己生成。那么不知道 i 输入线 (以 b 为例) 的 inactive key 的情况下，$\mathcal{A}$ 可以构造出 H_{i-1} 或 H_i 的其他部分吗？分两种情况讨论:

(a). 当 b 是输入线时，H 中包含了 P_2 的输入密钥 $k_i^{x_i}$ 或 $k_{n+i}^{y_i}$，但由于密钥都是 active key，所以 $\mathcal{A}$ 可以构造出来。

(b). 当 b 不是输入线时，前面的包含 b 的密钥的部分只有以 b 作为输出线的门 $t, t < i$ 的 garbled table，但由 H_{i-1} 和 H_i 中的门 t 都是模拟的 garbled table，即都加密的 active key，没有涉及到 inactive key，所以 $\mathcal{A}$ 是可以模拟的。

安全证明

解决这个问题之后还有另一个问题，就是如果 i 的输入线 a,b 不止进入门 i，还进入了别的门如 ${i_1}^a,\ldots,{i_j}^a,{i_1}^b,\ldots,{i_l}^b$，这样的话，这些门的 garbled table 加密时是需要用到 inactive key 进行加密的，这时候由于 $\mathcal{A}$ 不知道 a,b 的 inactive key，所以无法直接构造出这些门的 garbled table。但是加密方案的实验中，$\mathcal{A}$ 可以访问两个 oracle，即 $\mathrm{Enc}_{(\cdot)}(\mathrm{Enc}_{\boldsymbol{k}_b^{1-\beta}}(\cdot)),\mathrm{Enc}_{\boldsymbol{k}_a^{1-\alpha}}(\mathrm{Enc}_{(\cdot)}(\cdot))$，而这两个 oracle 正好是对应 $\mathcal{A}$ 所有使用 inactive key 的需求，因此 $\mathcal{A}$ 也可以模拟这些门的 garbled table。

综上，A 可以完全模拟 D 的输入，当 $\sigma=0$ 时，D 的输入是 $H_{i-1}(x,y)$，当 $\sigma=1$ 时，D 的输入是 $H_i(x,y)$。

即 $|Pr[\mathrm{Expt}_{\mathcal{A}}^{double}(\kappa,0)=1]-Pr[\mathrm{Expt}_{\mathcal{A}}^{double}(\kappa,1)=1]|=|Pr[D(H_{i-1}(x,y))=1]-Pr[D(H_i(x,y))=1]|>\frac{1}{qp(n)}$

与加密方案的安全性矛盾。

安全证明

因此 $\{H_{OT}(x,y)\}_{x,y\in\{0,1\}^*} \equiv \{H_0(x,y)\}_{x,y\in\{0,1\}^*} \equiv^c \{H_q(x,y)\}_{x,y\in\{0,1\}^*} \equiv \{S_2(y,f(x,y))\}_{x,y\in\{0,1\}^*}$

再由 $\{H_{OT}(x,y)\}_{x,y\in\{0,1\}^*} \equiv^c \{view_2^{\pi}(x,y)\}_{x,y\in\{0,1\}^*}$

有 $\{S_2(y,f(x,y))\}_{x,y\in\{0,1\}^*} \equiv^c \{view_2^{\pi}(x,y)\}_{x,y\in\{0,1\}^*}$

秘密分享角度

回忆两方情况下的 GC 方案：

- $(F,e,d) \leftarrow \mathrm{Gb}(1^\kappa,f)$
- $X||Y \leftarrow \mathrm{En}(e,x||y)$
- $Z \leftarrow \mathrm{Ev}(F,X||Y)$
- $z \leftarrow \mathrm{De}(d,Z)$

可以将 $f(x,y)$ 的分享看成 (d,Z)，一方持有 d，一方持有 Z。如何让一方只知道 d 不知道 Z 呢？靠输入分割自然能保证这一点，只知道 x 不知道 y 或只知道 y 不知道 x 均无法得到 Z。那如何让另一方得到 Z 呢？用 OT 执行 En 算法，一方输入 e，一方输入 y 并得到 Y。因此这里 OT 就起到了秘密分享的作用。

1. GC 定义
2. GC 构造
3. 参考文献

主要参考文献

1. [LP04] Yehuda Lindell and Benny Pinkas. A Proof of Yao's Protocol for Secure Two-Party Computation. Electronic Colloquium on Computational Complexity (ECCC) 2004.
2. [BHR12] Mihir Bellare , Viet Tung Hoang and Phillip Rogaway. Foundations of garbled circuits. CCS 2012.

参考文献 I

[AIK11] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. "How to Garble Arithmetic Circuits". In: *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*. 2011, pp. 120–129. DOI: 10.1109/FOCS.2011.40. URL: https://doi.org/10.1109/FOCS.2011.40.

[BMR16] Marshall Ball, Tal Malkin, and Mike Rosulek. "Garbling Gadgets for Boolean and Arithmetic Circuits". In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. 2016, pp. 565–577. DOI: 10.1145/2976749.2978410. URL: https://doi.org/10.1145/2976749.2978410.

参考文献 II

[Bal+19] Marshall Ball et al. “Garbled Neural Networks are Practical”. In: *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 338. URL: https://eprint.iacr.org/2019/338.

[BMR90] Donald Beaver, Silvio Micali, and Phillip Rogaway. “The Round Complexity of Secure Protocols (Extended Abstract)”. In: *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*. 1990, pp. 503–513. DOI: 10.1145/100216.100287. URL: https://doi.org/10.1145/100216.100287.

参考文献 III

[BHR12] Mihir Bellare, Viet Tung Hoang, and Phillip Rogaway. “Foundations of garbled circuits”. In: *the ACM Conference on Computer and Communications Security, CCS’12, Raleigh, NC, USA, October 16-18, 2012*. 2012, pp. 784–796. DOI: 10.1145/2382196.2382279. URL: https://doi.org/10.1145/2382196.2382279.

[Ben18] Aner Ben-Efraim. “On Multiparty Garbling of Arithmetic Circuits”. In: *Advances in Cryptology - ASIACRYPT 2018 - 24th International Conference on the Theory and Application of Cryptology and Information Security, Brisbane, QLD, Australia, December 2-6, 2018, Proceedings, Part III*. 2018, pp. 3–33. DOI: 10.1007/978-3-030-03332-3_1. URL: https://doi.org/10.1007/978-3-030-03332-3_1.

参考文献 IV

[Kat+18] Jonathan Katz et al. “Optimizing Authenticated Garbling for Faster Secure Two-Party Computation”. In: *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part III*. 2018, pp. 365–391. DOI: 10.1007/978-3-319-96878-0_13. URL: https://doi.org/10.1007/978-3-319-96878-0_13.

参考文献 V

[KS08] Vladimir Kolesnikov and Thomas Schneider. "Improved Garbled Circuit: Free XOR Gates and Applications". In: *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*. 2008, pp. 486–498. DOI: `10.1007/978-3-540-70583-3\_40`. URL: `https://doi.org/10.1007/978-3-540-70583-3\_40`.

参考文献 VI

[Lin13] Yehuda Lindell. "Fast Cut-and-Choose Based Protocols for Malicious and Covert Adversaries". In: *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part II*. 2013, pp. 1–17. DOI: 10.1007/978-3-642-40084-1_1. URL: https://doi.org/10.1007/978-3-642-40084-1_1.

[LP04] Yehuda Lindell and Benny Pinkas. "A Proof of Yao's Protocol for Secure Two-Party Computation". In: *Electronic Colloquium on Computational Complexity (ECCC)* 063 (2004). URL: http://eccc.hpi-web.de/eccc-reports/2004/TR04-063/index.html.

参考文献 VII

[LP07] Yehuda Lindell and Benny Pinkas. "An Efficient Protocol for Secure Two-Party Computation in the Presence of Malicious Adversaries". In: *Advances in Cryptology - EUROCRYPT 2007, 26th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Barcelona, Spain, May 20-24, 2007, Proceedings*. 2007, pp. 52–78. DOI: 10.1007/978-3-540-72540-4_4. URL: https://doi.org/10.1007/978-3-540-72540-4_4.

参考文献 VIII

[LP11] Yehuda Lindell and Benny Pinkas. "Secure Two-Party Computation via Cut-and-Choose Oblivious Transfer". In: *Theory of Cryptography - 8th Theory of Cryptography Conference, TCC 2011, Providence, RI, USA, March 28-30, 2011. Proceedings*. 2011, pp. 329–346. DOI: 10.1007/978-3-642-19571-6_20. URL: https://doi.org/10.1007/978-3-642-19571-6_20.

[LR15] Yehuda Lindell and Ben Riva. "Blazing Fast 2PC in the Offline/Online Setting with Security for Malicious Adversaries". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*. 2015, pp. 579–590. DOI: 10.1145/2810103.2813666. URL: https://doi.org/10.1145/2810103.2813666.

参考文献 IX

[LR14] Yehuda Lindell and Ben Riva. “Cut-and-Choose Yao-Based Secure Computation in the Online/Offline and Batch Settings”. In: *Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part II*. 2014, pp. 476–494. DOI: 10.1007/978-3-662-44381-1_27. URL: https://doi.org/10.1007/978-3-662-44381-1_27.

[MW19] Eleftheria Makri and Tim Wood. “Full-Threshold Actively-Secure Multiparty Arithmetic Circuit Garbling”. In: *IACR Cryptol. ePrint Arch.* 2019 (2019), p. 1098. URL: https://eprint.iacr.org/2019/1098.

参考文献 X

[NPS99] Moni Naor, Benny Pinkas, and Reuban Sumner. "Privacy preserving auctions and mechanism design". In: *Proceedings of the First ACM Conference on Electronic Commerce (EC-99), Denver, CO, USA, November 3-5, 1999*. 1999, pp. 129–139. DOI: 10.1145/336992.337028. URL: https://doi.org/10.1145/336992.337028.

参考文献 XI

[Pin+09] Benny Pinkas et al. “Secure Two-Party Computation Is Practical”. In: *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*. 2009, pp. 250–267. DOI: 10.1007/978-3-642-10366-7_15. URL: https://doi.org/10.1007/978-3-642-10366-7_15.

参考文献 XII

[RR16] Peter Rindal and Mike Rosulek. “Faster Malicious 2-Party Secure Computation with Online/Offline Dual Execution”. In: *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016*. 2016, pp. 297–314. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/rindal.

参考文献 XIII

[RR21] Mike Rosulek and Lawrence Roy. “Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits”. In: *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*. 2021, pp. 94–124. DOI: 10.1007/978-3-030-84242-0_5. URL: https://doi.org/10.1007/978-3-030-84242-0_5.

参考文献 XIV

[WMK17] Xiao Wang, Alex J. Malozemoff, and Jonathan Katz. “Faster Secure Two-Party Computation in the Single-Execution Setting”. In: *Advances in Cryptology - EUROCRYPT 2017 - 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30 - May 4, 2017, Proceedings, Part III*. 2017, pp. 399–424. DOI: 10.1007/978-3-319-56617-7_14. URL: https://doi.org/10.1007/978-3-319-56617-7_14.

参考文献 XV

[WRK17] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. "Authenticated Garbling and Efficient Maliciously Secure Two-Party Computation". In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*. 2017, pp. 21–37. DOI: 10.1145/3133956.3134053. URL: https://doi.org/10.1145/3133956.3134053.

[Yao86] Andrew Chi-Chih Yao. "How to Generate and Exchange Secrets (Extended Abstract)". In: *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*. 1986, pp. 162–167. DOI: 10.1109/SFCS.1986.25. URL: https://doi.org/10.1109/SFCS.1986.25.

参考文献 XVI

[ZRE15] Samee Zahur, Mike Rosulek, and David Evans. "Two Halves Make a Whole - Reducing Data Transfer in Garbled Circuits Using Half Gates". In: *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*. 2015, pp. 220–250. DOI: 10.1007/978-3-662-46803-6_8. URL: https://doi.org/10.1007/978-3-662-46803-6_8.

Thanks!