恶意安全性

GMW 编译器

《安全多方计算——可证明安全视角》第九章第一节

2026 年 2 月 3 日

目录

恶意敌手模型

半诚实 vs 恶意

- **半诚实 (Semi-Honest)**: 遵循协议，但试图从视图中获取隐私.
- **恶意 (Malicious)**: 可以任意偏离协议（发送错误消息、中止协议、替换输入）.

本章核心技术路线:

1. **GMW 编译器**: 适用于所有协议的通用转换（效率低）.
2. **切分选择 (Cut-and-Choose)**: 适用于姚氏混淆电路 (Yao's GC).
3. **恶意 BGW**: 适用于诚实多数 ($t < n/3$) 的信息论安全协议.
4. **BDOZ & SPDZ**: 基于 MAC 和预计算的通用 MPC ($t < n$ 或 $t < n/2$).

GMW(Goldreich-Micali-Wigderson) 编译器

核心思想: 强制恶意敌手像半诚实参与方一样行事.

转换步骤

1. **输入承诺 (Input Commitment)**: 参与方对自己的输入 x 和随机带 r 进行承诺.
2. **抛硬币 (Coin Tossing)**: 通过多方交互生成无偏的公共随机数，作为协议的随机源.
3. **零知识证明 (ZKP)**: 在协议的每一步，发送消息 m 的同时，附带一个 ZK 证明: "我是基于承诺的输入 x、随机带 r 以及历史消息，诚实地计算出 m 的."

关键工具: Sigma 协议 (3-move public-coin ZKP).

零知识证明

NP 关系与陈述

- **关系** $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$，**语言** $L_{\mathcal{R}} = \{x \mid \exists w,\ (x,w) \in \mathcal{R}\}$.
- **陈述 (statement) 为** x，**见证 (witness) 为** w.

零知识证明

- **两方协议** (P, V)：**证明者** $P(x,w)$ **与验证者** $V(x)$ **交互**.
- **目标**：P **证明** $x \in L_{\mathcal{R}}$，**要求**：
  - **完备性**：如果证明者和验证者都是诚实的，那么验证者接受证明的概率为 1；
  - **可靠性**：如果验证者接受了证明者的证明，那么 (以压倒性的概率) 陈述为真；
  - **零知识性**：验证者不能获得除了陈述为真以外的任何信息.

零知识证明示例

$$\mathcal{R} = \{((g, \mathsf{ck}, \mathsf{pk}, c, c_1, c_2), m, s, t) \mid c = g^m \mathsf{ck}^s \ \wedge \ c_1 = g^t \ \wedge \ c_2 = g^m \mathsf{pk}^t\}$$

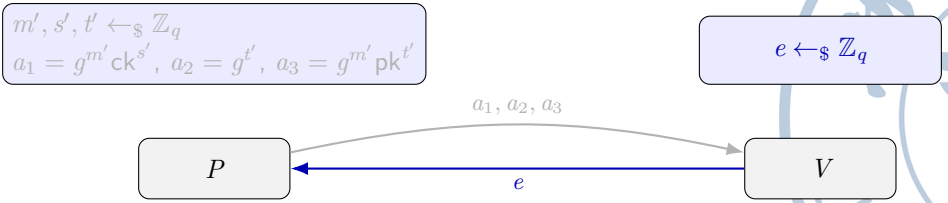
$m', s', t' \leftarrow_\$ \mathbb{Z}_q$
$a_1 = g^{m'} \mathsf{ck}^{s'}, \ a_2 = g^{t'}, \ a_3 = g^{m'} \mathsf{pk}^{t'}$

a_1, a_2, a_3

P → V

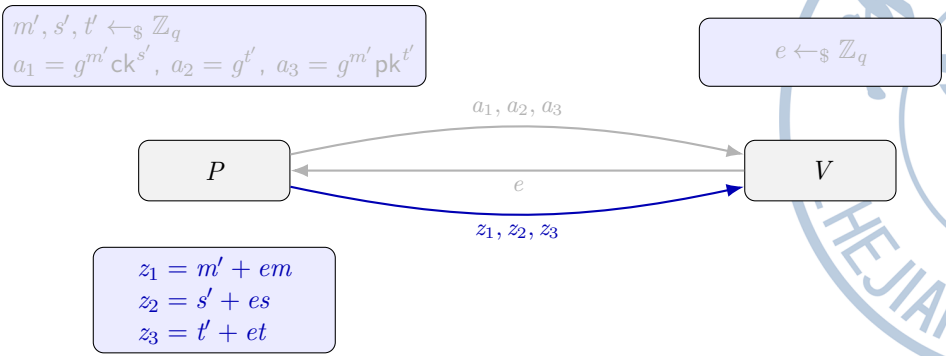
零知识证明示例

$$\mathcal{R} = \{((g, \mathsf{ck}, \mathsf{pk}, c, c_1, c_2), m, s, t) \mid c = g^m \mathsf{ck}^s \ \wedge \ c_1 = g^t \ \wedge \ c_2 = g^m \mathsf{pk}^t\}$$



零知识证明示例

$$\mathcal{R} = \{((g, \mathsf{ck}, \mathsf{pk}, c, c_1, c_2), m, s, t) \mid c = g^m \mathsf{ck}^s \ \wedge \ c_1 = g^t \ \wedge \ c_2 = g^m \mathsf{pk}^t\}$$



零知识证明示例

$$\mathcal{R} = \{((g, \mathsf{ck}, \mathsf{pk}, c, c_1, c_2), m, s, t) \mid c = g^m \mathsf{ck}^s \ \wedge \ c_1 = g^t \ \wedge \ c_2 = g^m \mathsf{pk}^t\}$$

$m', s', t' \leftarrow_\$ \mathbb{Z}_q$

$a_1 = g^{m'} \mathsf{ck}^{s'}, \ a_2 = g^{t'}, \ a_3 = g^{m'} \mathsf{pk}^{t'}$

$e \leftarrow_\$ \mathbb{Z}_q$

a_1, a_2, a_3

P e V

z_1, z_2, z_3

$z_1 = m' + em$

$z_2 = s' + es$

$z_3 = t' + et$

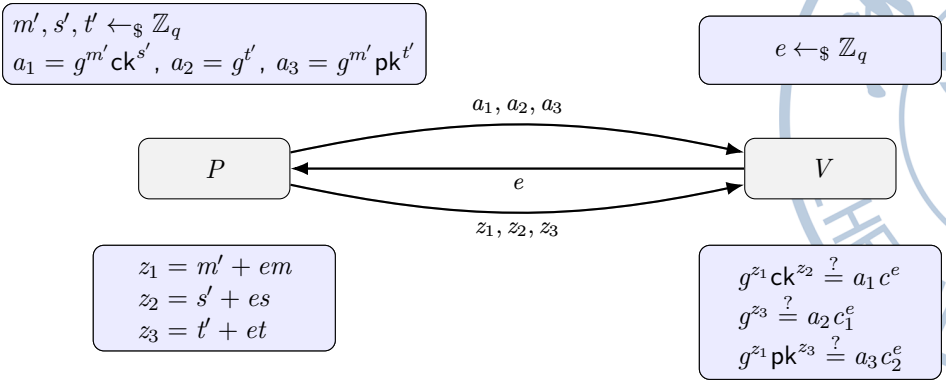
$g^{z_1} \mathsf{ck}^{z_2} \stackrel{?}{=} a_1 c^e$

$g^{z_3} \stackrel{?}{=} a_2 c_1^e$

$g^{z_1} \mathsf{pk}^{z_3} \stackrel{?}{=} a_3 c_2^e$

零知识证明示例

$$\mathcal{R} = \{((g, \mathsf{ck}, \mathsf{pk}, c, c_1, c_2), m, s, t) \mid c = g^m \mathsf{ck}^s \ \wedge \ c_1 = g^t \ \wedge \ c_2 = g^m \mathsf{pk}^t\}$$



Sigma 协议：3 步公共币证明系统

对于关系 $\mathcal{R}$ 的 Sigma 协议由三个算法 $(\mathcal{C}, \mathcal{Z}, \mathcal{V})$ 组成：

- $a \leftarrow \mathcal{C}(x, w; r)$：输入陈述 x、见证 w 和随机硬币 r，输出初始消息 a.
- $z \leftarrow \mathcal{Z}(x, w, r, e)$：输入陈述 x、见证 w、随机硬币 r 以及挑战值 $e \in \{0,1\}^{\lambda}$，输出回答消息 z.
- $0/1 \leftarrow \mathcal{V}(x, a, e, z)$：输入陈述 x、初始消息 a、挑战值 e 以及回答消息 z，输出 0 或 1 分别代表拒绝或接受.

满足：

Sigma 协议：3 步公共币证明系统

对于关系 $\mathcal{R}$ 的 Sigma 协议由三个算法 $(\mathcal{C}, \mathcal{Z}, \mathcal{V})$ 组成：

- $a \leftarrow \mathcal{C}(x, w; r)$：输入陈述 x、见证 w 和随机硬币 r，输出初始消息 a.
- $z \leftarrow \mathcal{Z}(x, w, r, e)$：输入陈述 x、见证 w、随机硬币 r 以及挑战值 $e \in \{0,1\}^\lambda$，输出回答消息 z.
- $0/1 \leftarrow \mathcal{V}(x, a, e, z)$：输入陈述 x、初始消息 a、挑战值 e 以及回答消息 z，输出 0 或 1 分别代表拒绝或接受.

满足：

- **完备性**：对于所有 $(x, w) \in \mathcal{R}$，$e \in \{0,1\}^\lambda$，有

$$\Pr[a \leftarrow \mathcal{C}(x, w; r);\ z \leftarrow \mathcal{Z}(x, w, r, e) : \mathcal{V}(x, a, e, z) = 1] = 1$$

Sigma 协议：3 步公共币证明系统

对于关系 $\mathcal{R}$ 的 Sigma 协议由三个算法 $(\mathcal{C}, \mathcal{Z}, \mathcal{V})$ 组成：

- $a \leftarrow \mathcal{C}(x, w; r)$：输入陈述 x、见证 w 和随机硬币 r，输出初始消息 a.
- $z \leftarrow \mathcal{Z}(x, w, r, e)$：输入陈述 x、见证 w、随机硬币 r 以及挑战值 $e \in \{0,1\}^\lambda$，输出回答消息 z.
- $0/1 \leftarrow \mathcal{V}(x, a, e, z)$：输入陈述 x、初始消息 a、挑战值 e 以及回答消息 z，输出 0 或 1 分别代表拒绝或接受.

满足：

- **完备性**：对于所有 $(x, w) \in \mathcal{R}$，$e \in \{0,1\}^\lambda$，有

$$\Pr[a \leftarrow \mathcal{C}(x, w; r);\ z \leftarrow \mathcal{Z}(x, w, r, e) : \mathcal{V}(x, a, e, z) = 1] = 1$$

- **2-特殊可靠性**：存在确定性多项式时间提取器 $\mathcal{X}$，使得对任意 PPT 的敌手 $\mathcal{A}$ 都有

$$\Pr\left[\begin{array}{ll} (x, a) \leftarrow \mathcal{A}(1^\lambda);\ e, e' \leftarrow_{\$} \{0,1\}^\lambda; & \\ z, z' \leftarrow \mathcal{A}(e, e');\ w \leftarrow \mathcal{X}(x, a, \{e, z\}, \{e', z'\}) & : \begin{array}{l} e \neq e' \ \wedge\ \mathcal{V}(x, a, e, z) = 1 \wedge \\ \mathcal{V}(x, a, e', z') = 1 \ \wedge\ (x, w) \notin \mathcal{R} \end{array} \end{array}\right] = 0$$

Sigma 协议：3 步公共币证明系统

对于关系 $\mathcal{R}$ 的 Sigma 协议由三个算法 $(\mathcal{C}, \mathcal{Z}, \mathcal{V})$ 组成：

- $a \leftarrow \mathcal{C}(x, w; r)$：输入陈述 x、见证 w 和随机硬币 r，输出初始消息 a.
- $z \leftarrow \mathcal{Z}(x, w, r, e)$：输入陈述 x、见证 w、随机硬币 r 以及挑战值 $e \in \{0,1\}^\lambda$，输出回答消息 z.
- $0/1 \leftarrow \mathcal{V}(x, a, e, z)$：输入陈述 x、初始消息 a、挑战值 e 以及回答消息 z，输出 0 或 1 分别代表拒绝或接受.

满足：

- **完备性**：对于所有 $(x, w) \in \mathcal{R}$，$e \in \{0,1\}^\lambda$，有

$$\Pr[a \leftarrow \mathcal{C}(x, w; r);\ z \leftarrow \mathcal{Z}(x, w, r, e) : \mathcal{V}(x, a, e, z) = 1] = 1$$

- **2-特殊可靠性**：存在确定性多项式时间提取器 $\mathcal{X}$，使得对任意 PPT 的敌手 $\mathcal{A}$ 都有

$$\Pr\left[\begin{array}{ll} (x,a) \leftarrow \mathcal{A}(1^\lambda);\ e, e' \leftarrow_{\$} \{0,1\}^\lambda; & \\ z, z' \leftarrow \mathcal{A}(e, e');\ w \leftarrow \mathcal{X}(x, a, \{e, z\}, \{e', z'\}) & : \begin{array}{l} e \neq e' \ \wedge\ \mathcal{V}(x, a, e, z) = 1 \wedge \\ \mathcal{V}(x, a, e', z') = 1 \ \wedge\ (x, w) \notin \mathcal{R} \end{array} \end{array}\right] = 0$$

- **诚实验证者零知识性**：存在 PPT 的算法 $\mathcal{S}$，使得对任意 PPT 的敌手 $\mathcal{A}$ 都有

$$\begin{aligned} &\Pr[(x, e) \leftarrow \mathcal{A}(1^\lambda);\ a \leftarrow \mathcal{C}(x, w; r);\ z \leftarrow \mathcal{Z}(x, w, r, e) : \mathcal{A}(a, z) = 1] \\ \approx &\Pr[(x, e) \leftarrow \mathcal{A}(1^\lambda); (a, z) \leftarrow \mathcal{S}(x, e) : \mathcal{A}(a, z) = 1] \end{aligned}$$

其中 r 是均匀随机选取的，$\mathcal{A}$ 输出的值必须满足 $(x, w) \in \mathcal{R}$ 和 $e \in \{0,1\}^\lambda$.

安全性证明

- **完备性**：当证明者和验证者均诚实执行协议时，有

$$
\begin{aligned}
a_1\, c^e &= g^{m'}\mathsf{ck}^{s'} \cdot (g^m \mathsf{ck}^s)^e = g^{m'+em}\mathsf{ck}^{s'+es} = g^{z_1}\mathsf{ck}^{z_2}, \\
a_2\, c_1^e &= g^{t'} \cdot (g^t)^e = g^{t'+et} = g^{z_3}, \\
a_3\, c_2^e &= g^{m'}\mathsf{pk}^{t'} \cdot (g^m \mathsf{pk}^t)^e = g^{m'+em}\mathsf{pk}^{t'+et} = g^{z_1}\mathsf{pk}^{z_3}.
\end{aligned}
$$

安全性证明

- **2-特殊可靠性**：如果对于初始消息 a_1, a_2, a_3，有 (e, z_1, z_2, z_3) 和 (e', z_1', z_2', z_3') 都能通过验证，根据验证等式，有

$$g^{z_1}\mathsf{ck}^{z_2} = a_1 \cdot c^e \qquad g^{z_3} = a_2 \cdot c_1^e \qquad g^{z_1}\mathsf{pk}^{z_3} = a_3 \cdot c_2^e$$
$$g^{z_1'}\mathsf{ck}^{z_2'} = a_1 \cdot c^{e'} \qquad g^{z_3'} = a_2 \cdot c_1^{e'} \qquad g^{z_1'}\mathsf{pk}^{z_3'} = a_3 \cdot c_2^{e'}$$

因此

$$c = g^{(z_1 - z_1')(e-e')^{-1}}\mathsf{ck}^{(z_2 - z_2')(e-e')^{-1}}$$

$$c_1 = g^{(z_3 - z_3')(e-e')^{-1}}$$

$$c_2 = g^{(z_1 - z_1')(e-e')^{-1}}\mathsf{pk}^{(z_3 - z_3')(e-e')^{-1}}$$

可以提取见证 $m = (z_1 - z_1')(e-e')^{-1}, s = (z_2 - z_2')(e-e')^{-1}, t = (z_3 - z_3')(e-e')^{-1}$ 满足 $c = g^m\mathsf{ck}^s \ \wedge \ c_1 = g^t \ \wedge \ c_2 = g^m\mathsf{pk}^t$.

安全性证明

- **诚实验证者零知识性**：模拟器 $\mathcal{S}(e)$ 随机取 $z_1, z_2, z_3 \leftarrow_{\$} \mathbb{Z}_q$，置

$$a_1 = \frac{g^{z_1}\mathsf{ck}^{z_2}}{c^e}, \quad a_2 = \frac{g^{z_3}}{c_1^e}, \quad a_3 = \frac{g^{z_1}\mathsf{pk}^{z_3}}{c_2^e},$$

则 (a, e, z) 与真实会话分布一致.

GMW 编译器

将任何半诚实安全的协议转化为恶意安全的协议:

GMW 编译器

将任何半诚实安全的协议转化为恶意安全的协议:

1. **输入承诺**：每个参与方对自己的输入值进行承诺. 即参与方 P_i 计算承诺值 $c_i = \mathsf{Com}(x_i, \rho_i)$，其中 x_i 是 P_i 的输入，ρ_i 是随机数. P_i 将 c_i 公开.

GMW 编译器

将任何半诚实安全的协议转化为恶意安全的协议:

1. **输入承诺**：每个参与方对自己的输入值进行承诺. 即参与方 P_i 计算承诺值 $c_i = \mathsf{Com}(x_i, \rho_i)$，其中 x_i 是 P_i 的输入，ρ_i 是随机数. P_i 将 c_i 公开.

2. **抛硬币**：每个参与方 P_i 均匀选择随机数 $r_{i,i}$ 并承诺，将承诺公开. 其他参与方 $P_j,\ j \neq i$ 均匀选择随机数 $r_{i,j}$ 并承诺. 所有参与方完成承诺后，$P_j,\ j \neq i$ 将 $r_{i,j}$ 打开. 协议执行过程中，P_i 使用 $r_i = \bigoplus_{j=1}^{n} r_{i,j}$ 作为随机数. 这样的构造使得只要有一个参与方是诚实的，那么 r_i 就是均匀随机的.

GMW 编译器

将任何半诚实安全的协议转化为恶意安全的协议:

1. **输入承诺**：每个参与方对自己的输入值进行承诺. 即参与方 P_i 计算承诺值 $c_i = \mathsf{Com}(x_i, \rho_i)$，其中 x_i 是 P_i 的输入，ρ_i 是随机数. P_i 将 c_i 公开.

2. **抛硬币**：每个参与方 P_i 均匀选择随机数 $r_{i,i}$ 并承诺，将承诺公开. 其他参与方 P_j, $j \neq i$ 均匀选择随机数 $r_{i,j}$ 并承诺. 所有参与方完成承诺后，P_j, $j \neq i$ 将 $r_{i,j}$ 打开. 协议执行过程中，P_i 使用 $r_i = \bigoplus_{j=1}^{n} r_{i,j}$ 作为随机数. 这样的构造使得只要有一个参与方是诚实的，那么 r_i 就是均匀随机的.

3. **执行协议**：在协议执行过程中，当参与方 P_i 发送消息时，通过零知识证明协议证明该消息是使用 x_i 为输入，r_i 为随机数，诚实地遵循协议执行得到的.

Q & A