

安全多方计算 (Secure Multi-Party Computation) 简称为MPC

- 安全多方计算基础讲义 (1) —— MPC基本概念及基础组件
- 安全多方计算基础讲义 (2) —— 基于秘密分享方法的 MPC 协议
- 安全多方计算基础讲义 (3) —— 基于混淆电路方法的 MPC 协议

安全多方计算基本概念及基础组件

● 冯登国 ●



一、安全多方计算定义与应用

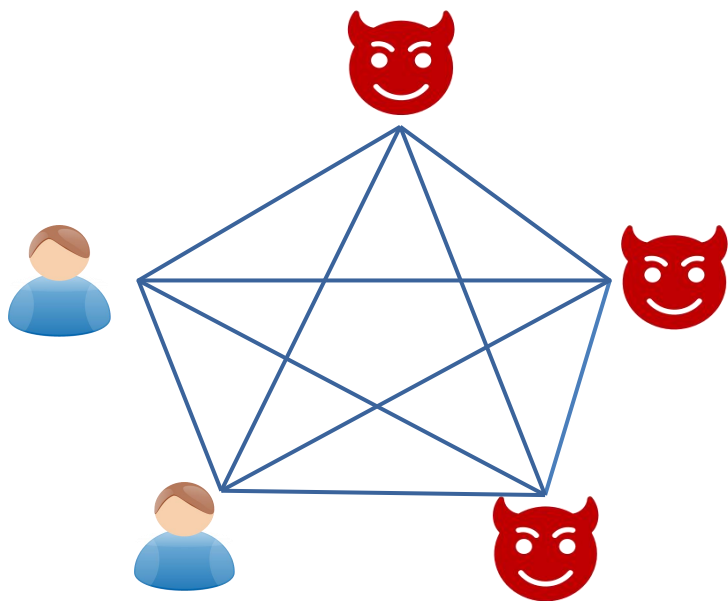
二、安全多方计算分类

三、安全多方计算基础组件

四、不经意传输及其算术变形

安全多方计算的概念

- 安全多方计算 (MPC) 是80年代提出的一个概念 [Yao86, GMW87]
- MPC 已成为大数据时代实现隐私计算的**核心技术**之一



$$(y_1, \dots, y_5) = f(x_1, \dots, x_5)$$

隐私性

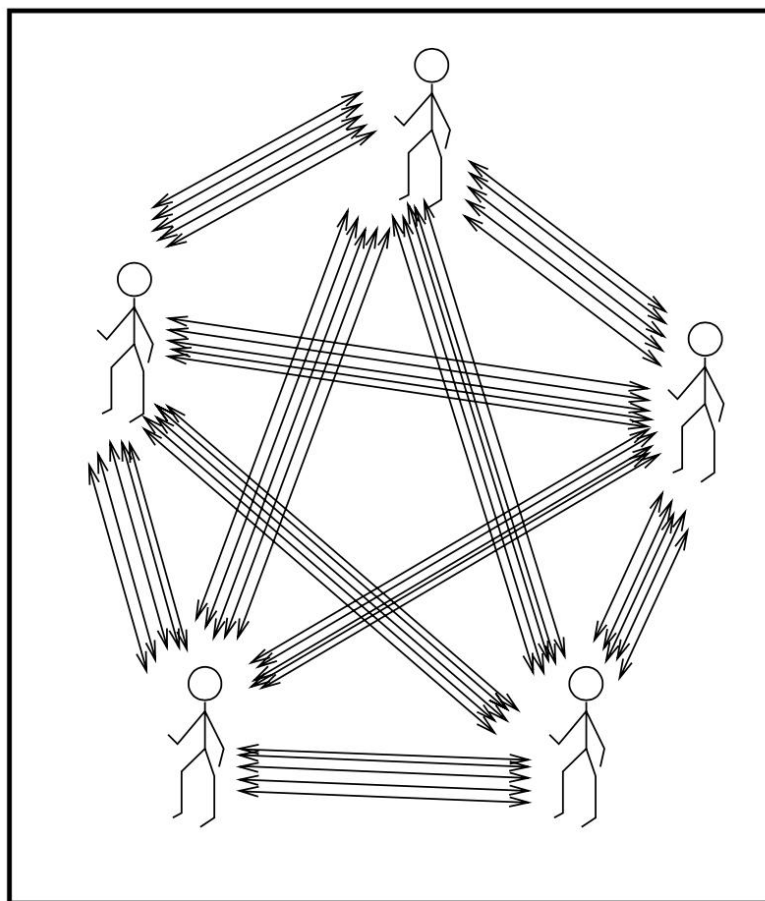
除了函数输出外，不泄漏任何输入数据的信息

正确性

保证协议输出为指定函数的计算结果

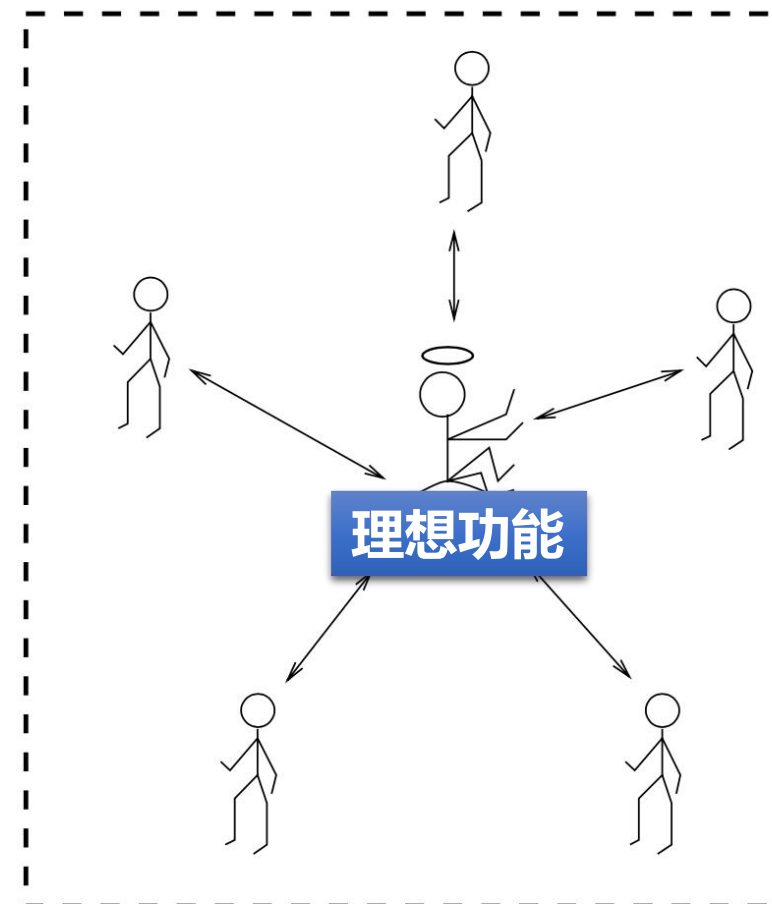
- [Yao86] Andrew Chi-Chih Yao. How to generate and exchange secrets (extended abstract). In *FOCS 1986*
- [GMW87] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game or A completeness theorem for protocols with honest majority. In *STOC 1987*

MPC 安全定义 —— 基于模拟的安全模型



REAL MODEL

\equiv



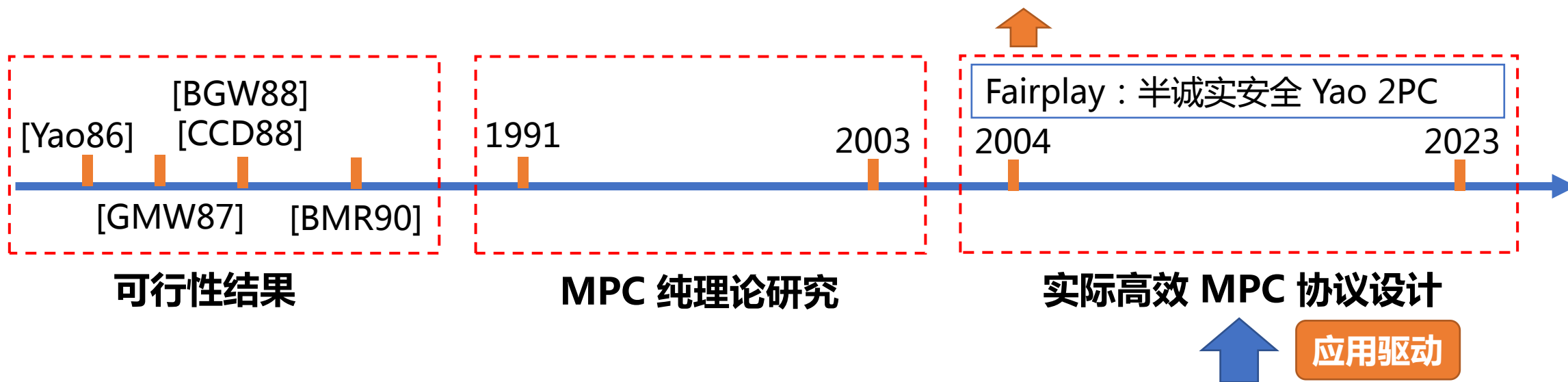
IDEAL MODEL

图来自于 Oded Goldreich 所著书籍《Foundations of Cryptography》

安全多方计算发展阶段

- [Yao86, GMW87, BGW88, CCD88, RB89, BMR90] 给出了 MPC 的可行性结果
- 80年代研究工作也给出了当前 MPC 协议的基础框架

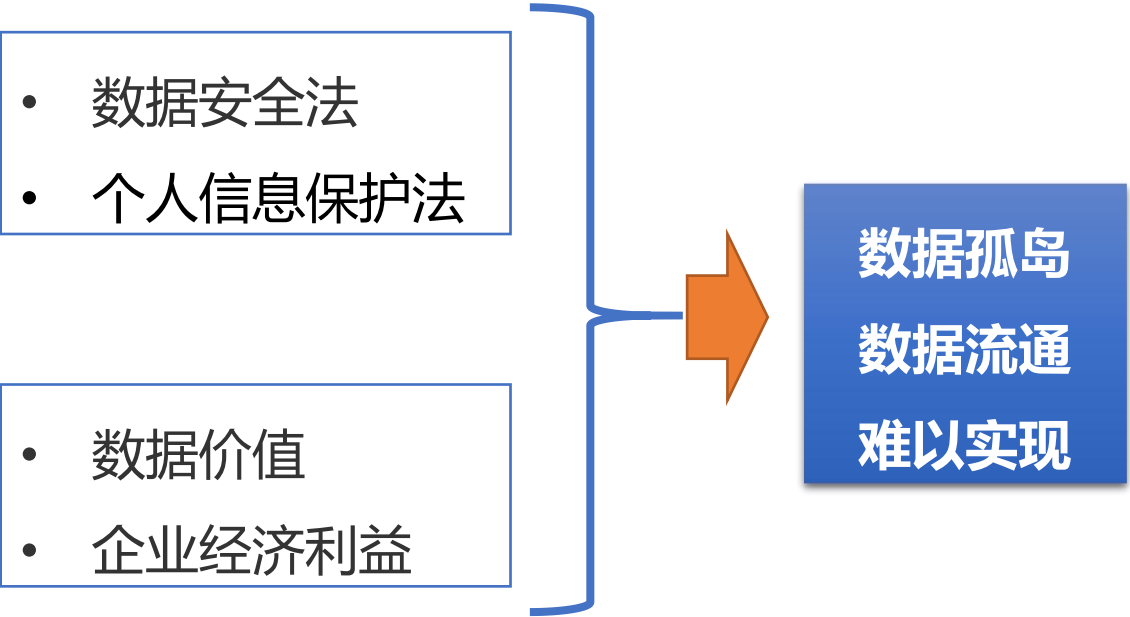
[MNPS04]: 32-bit 整数 “亿万富翁问题” 需要 1.25秒 (LAN) 和 4.01秒 (WAN)



Fairplay : 首个 MPC 实现，实现了半诚实安全的 Yao 安全两方计算 (2PC) 协议

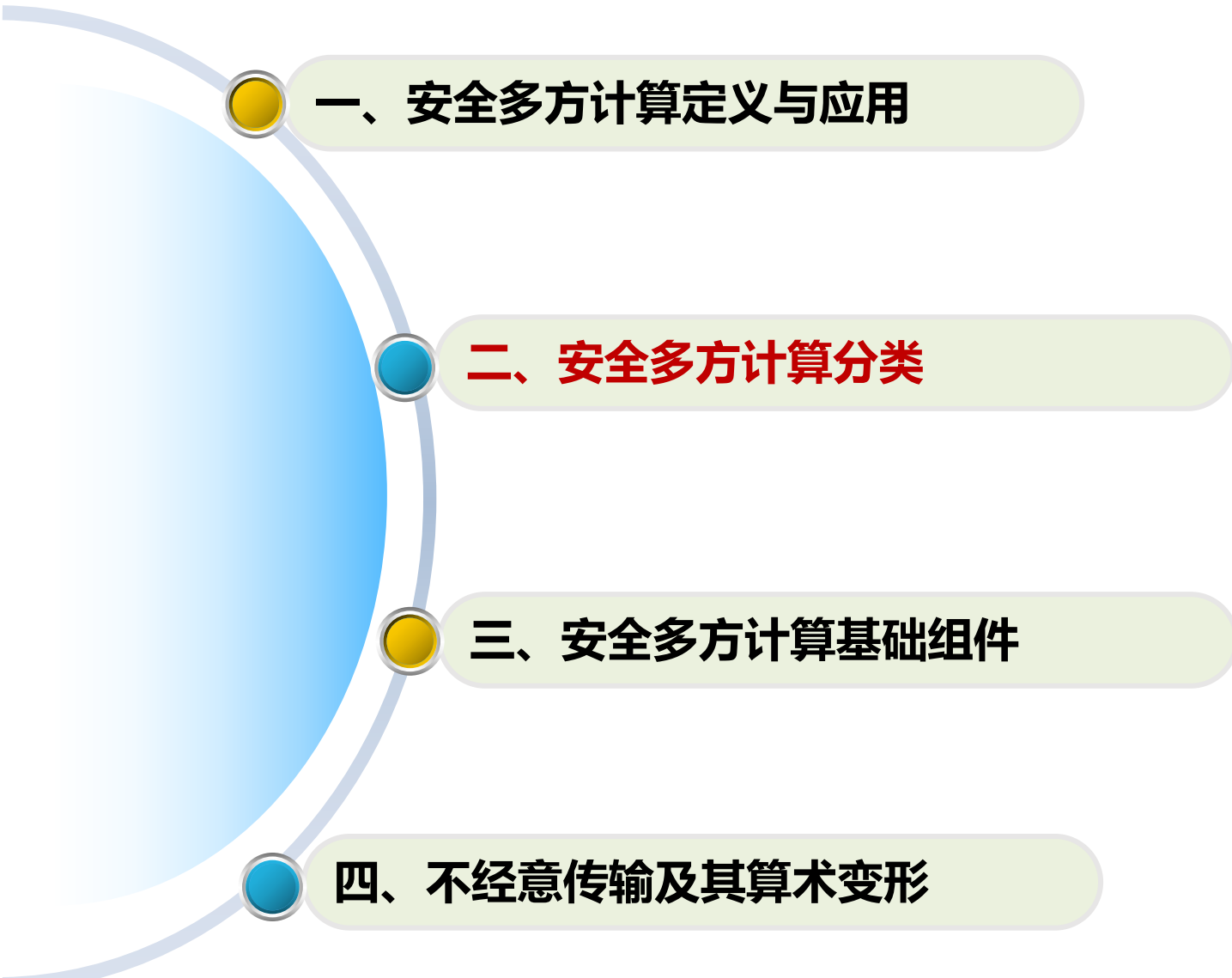
大数据时代，数据隐私计算的迫切需求

安全多方计算应用



安全多方计算 (MPC) 能打破数据孤岛，实现数据流通，保护数据隐私，已在诸多领域发现实际应用

政务、金融、医疗等领域	
联合风控	联合营销
智慧医疗	智慧园区
机器学习	区块链
数据库安全	基因分析
.....	



一、安全多方计算定义与应用

二、安全多方计算分类

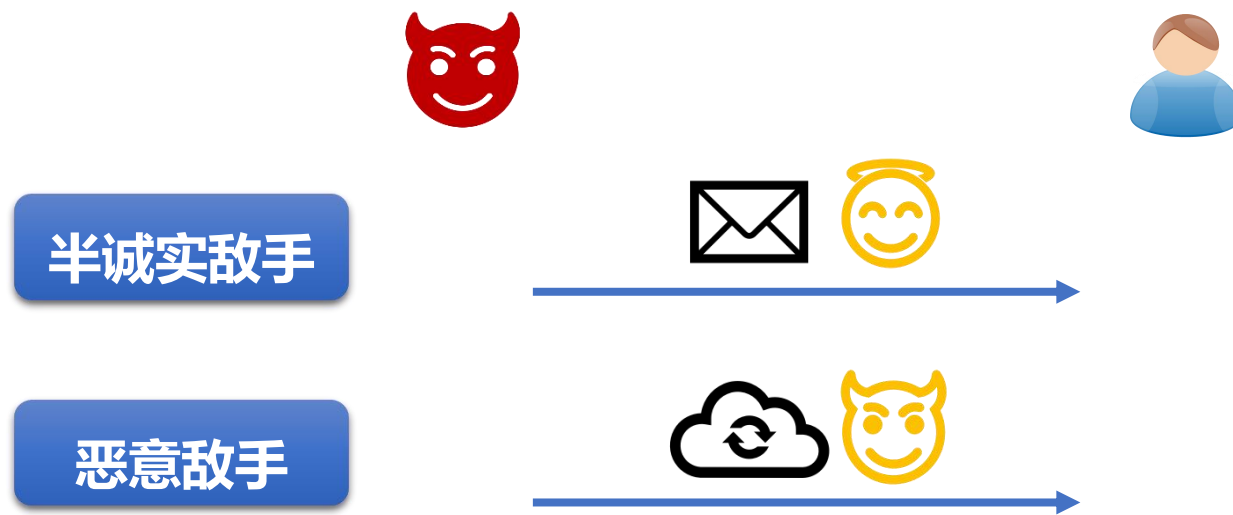
三、安全多方计算基础组件

四、不经意传输及其算术变形

安全多方计算分类（1）

（1）敌手行为

- 半诚实敌手（按照协议描述执行，但试图从协议记录中获取信息）
- 恶意敌手（可以执行任何攻击，发送任意的消息）

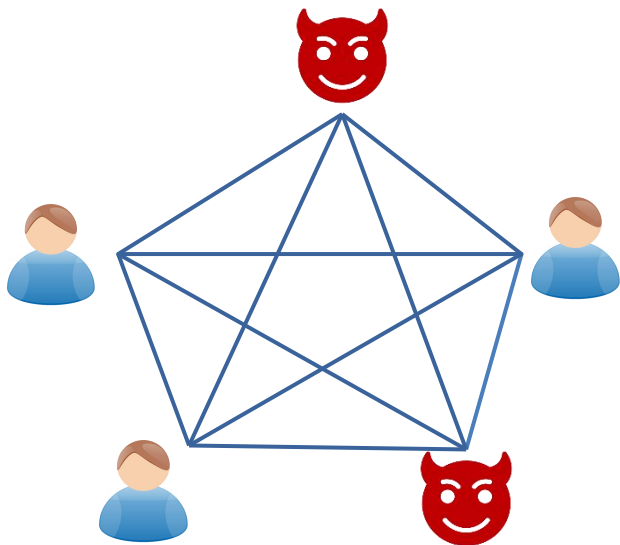


安全多方计算分类（2）

（2）腐化门限 t

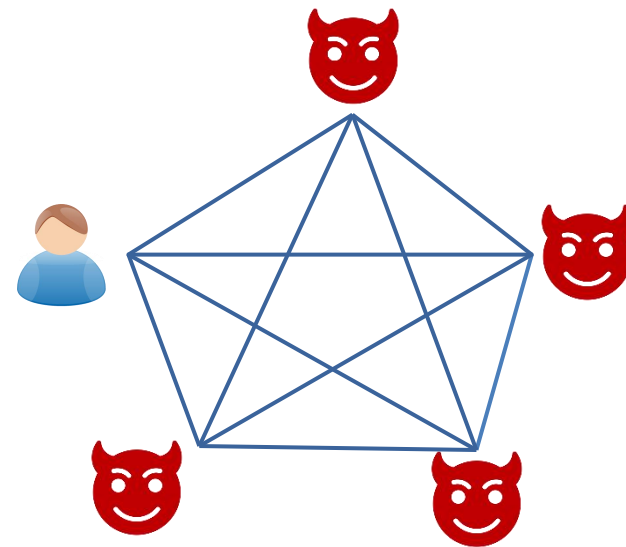
$$t < n/2$$

诚实
大多数



$$n/2 \leq t < n$$

不诚实
大多数



通常 $t = n - 1$

n 表示参与方总数， t 表示不诚实参与方数量的上界

安全多方计算分类 (3)

(3) 输出可达性

- **中止安全 (Security with abort)** : 腐化实体获得输出后, 可以阻止诚实实体获得输出
- **公平性 (Fairness)** : 要么腐化实体和诚实实体均获得输出, 要么他们均没有输出
- **保证输出传送 (Guaranteed output delivery)** : 所有诚实实体总是获得输出

一般而言

- 不诚实大多数情况 ($n/2 \leq t < n$) 仅能达到中止安全 [Cle86] (Coin-tossing 协议不能满足公平性)
- 诚实大多数情况 ($t < n/2$) 能达到公平性和保证输出传送

安全多方计算分类（4）

（4）计算模型

- **布尔电路**：由 AND、XOR、NOT 等逻辑门组成
- **算术电路**：由 ADD、MULT 等运算组成，通常定义在域 \mathbb{F} 上或环 \mathbb{Z}_{2^k} （即 $\text{mod } 2^k$ ）上
- **RAM程序**：由 read、write 等指令组成的程序

- 大部分 MPC 协议考虑布尔电路模型或算术电路模型
- 目前，只有少数 MPC 协议针对 RAM 计算模型设计，适合于输入为数据库的应用场景

安全多方计算分类（5）

（5）敌手计算能力

- **概率多项式时间（PPT）**：任意PPT敌手不能打破协议安全性
- **无限计算能力（信息论安全或无条件安全）**：即使无限计算能力的敌手也不能打破协议安全性（抵抗量子计算机攻击）

信息论安全 MPC 协议需要在诚实大多数模型下设计

安全多方计算分类（6）

（6）腐化策略

- **静态腐化（Static Corruption）**：在协议运行前，敌手决定腐化（Corrupt）哪些实体
- **自适应腐化（Adaptive Corruption）**：在协议运行过程中，敌手能自适应决定腐化哪些实体

已知实际高效的 MPC 协议均考虑静态腐化

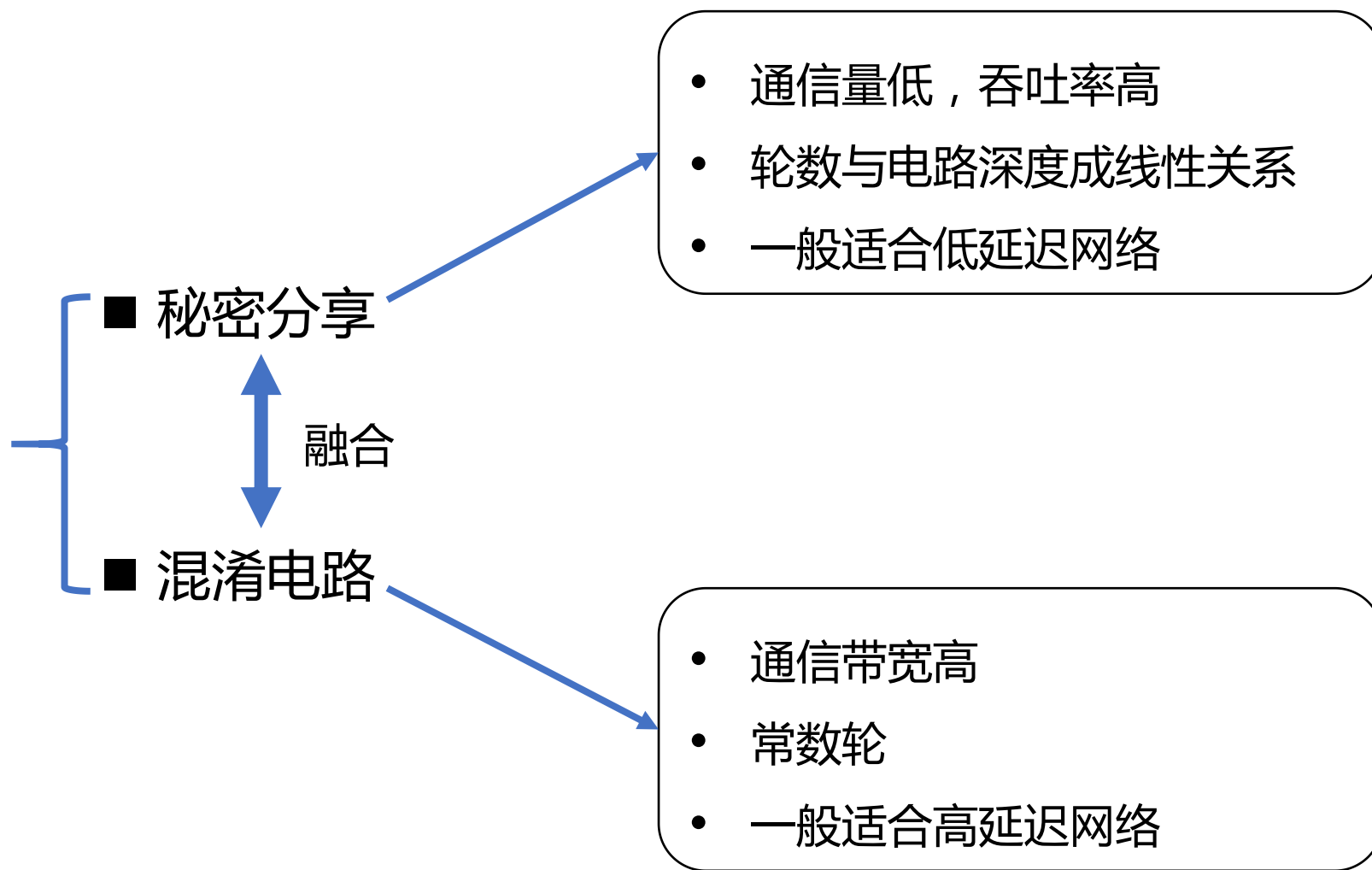
安全多方计算分类（7）

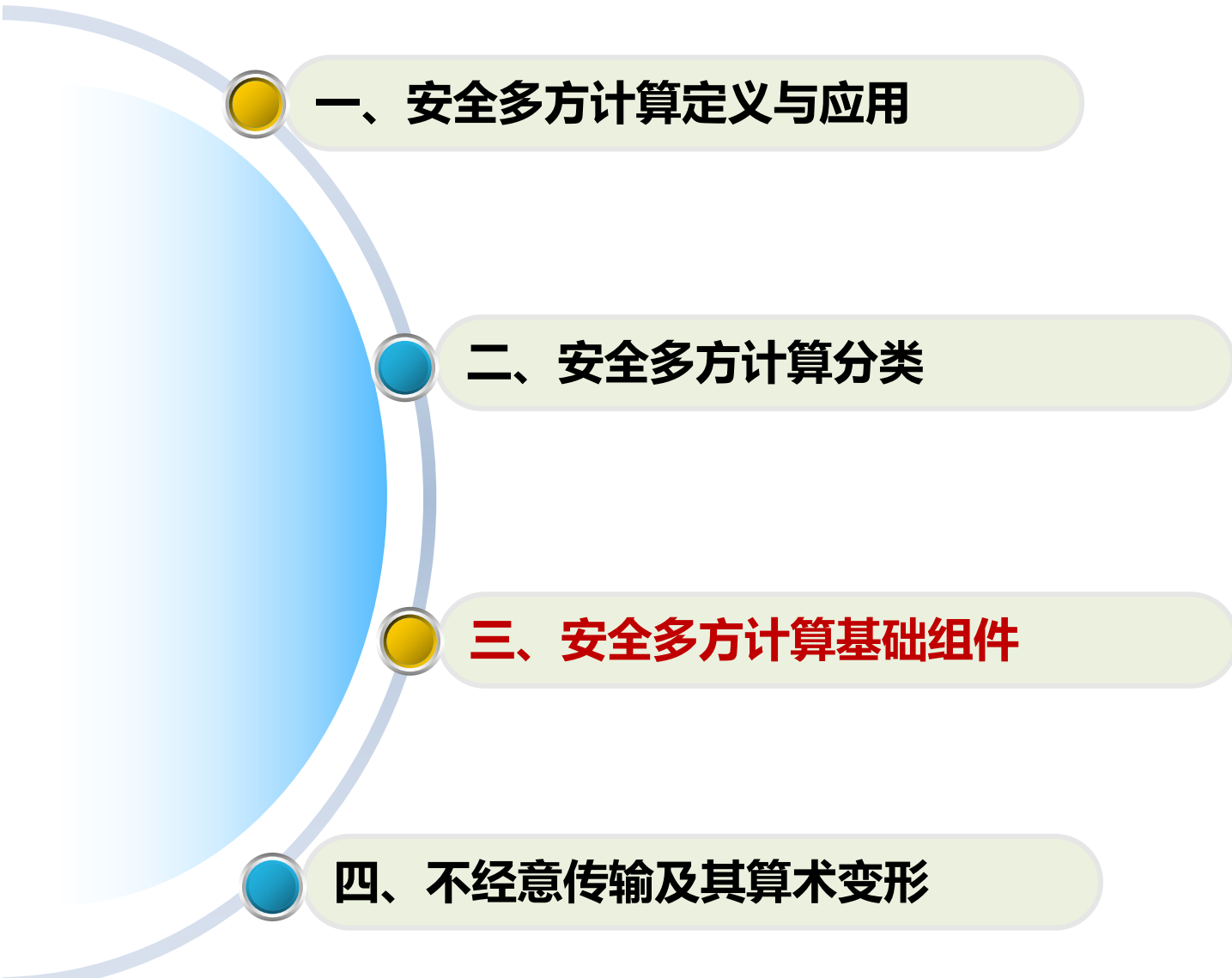
（7）网络模型

- **同步（Synchronous）网络模型**：同一个交互轮的消息在固定延时 Δ 内一定到达
- **异步（Asynchronous）网络模型**：没有要求同步时钟，也没有要求预先固定的网络延时，更加现实的网络假设（通常要求 $t < n/3$ ）

已知高效的 MPC 协议均考虑同步网络模型

MPC 基本设计方法





一、安全多方计算定义与应用

二、安全多方计算分类

三、安全多方计算基础组件

四、不经意传输及其算术变形

MPC 基础组件

实际高效 MPC 协议基础组件

- 线性秘密分享
- 信息论消息认证码
- 零知识证明
- 混淆电路
- 承诺方案
- 投币协议
- 同态加密
- 不经意传输 (OT)
- 相关不经意传输 (COT)
- 不经意线性计算 (OLE)
- 向量不经意线性计算 (VOLE)



本讲最后一节

(n, t) 门限线性秘密分享 (1)

LSSS定义

- 分享算法 $[x] \leftarrow \text{Share}(x)$: P_i 获得份额 x^i , 其中共有 n 个参与方 P_1, \dots, P_n
- 重构算法 $x \leftarrow \text{Rec}([x], i)$: P_i 获得秘密值 x (要求至少 $t + 1$ 个份额)
- 打开算法 $x \leftarrow \text{Open}([x])$: 所有参与方获得 x

线性性质

- $[z] = [x] + [y]$, $[z] = [x] + c$, $[z] = c \cdot [x]$ 均能够**本地计算** , **无需通信** , 其中 c 为公开的常数
- 应用于MPC协议设计中, 加法等线性门是 **free** (即无需通信)

本次讲义主要关注定义在有限域 \mathbb{F} 上的 LSSS, 其也能扩展到环 (如 \mathbb{Z}_{2^k}) 上

(n, t) 门限线性秘密分享 (2)

加法秘密分享

- 主要用在不诚实大多数MPC协议中 ($t = n - 1$)
- $x = x^1 + x^2 + \dots + x^n \in \mathbb{F}$, 其中 x 为秘密 (secret), x^i 为份额 (share)



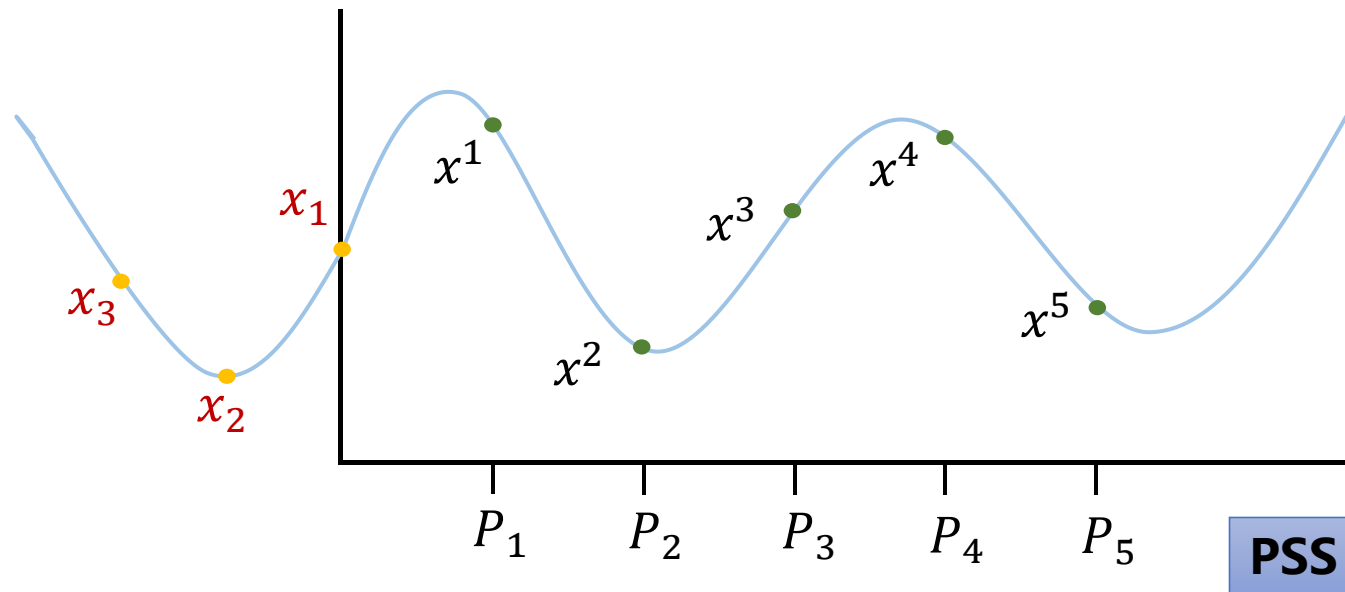
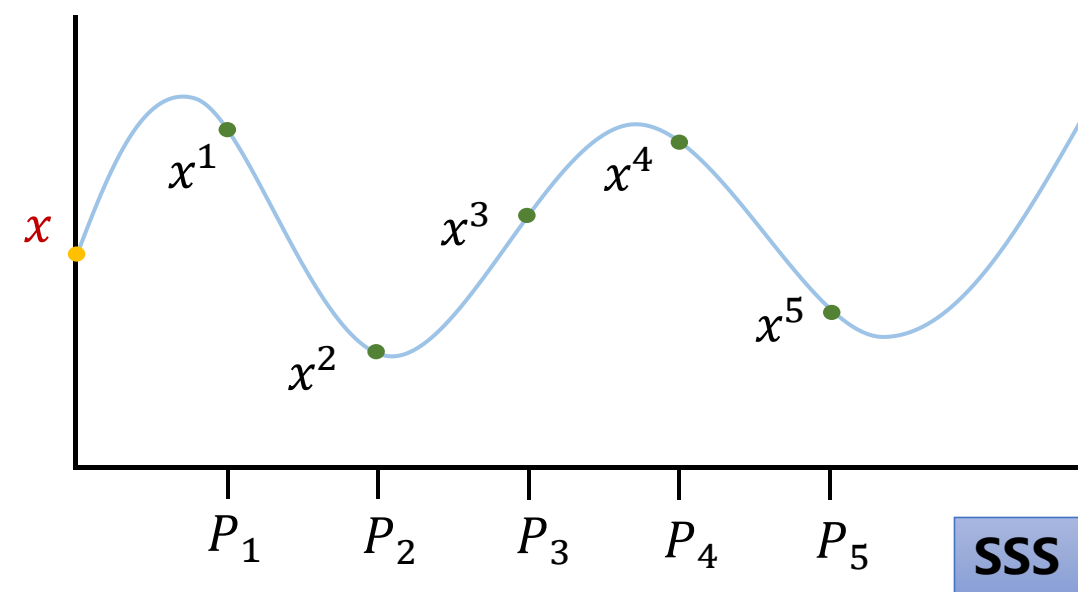
重复秘密分享

- 主要用在诚实大多数MPC协议中 ($t < n/2$, n 较小); 以 $n = 3$, $t = 1$ 为例
- $x = x^1 + x^2 + x^3 \in \mathbb{F}$: P_1 持有 (x^2, x^3) , P_2 持有 (x^1, x^3) , P_3 持有 (x^1, x^2)

(n, t) 门限线性秘密分享 (3)

Shamir 秘密分享及其推广

- 主要用在诚实大多数MPC协议中 ($t + 1 \leq (n + 1)/2$)
- $x = f(0)$, $x^i = f(\alpha_i)$, 其中 f 为次数 t 的随机多项式, $\alpha_1, \dots, \alpha_n \in \mathbb{F}$ 为不同的非零域元素
- 推广形式 (打包秘密分享, packed secret sharing) : $x_i = f(\beta_i)$, $x^i = f(\alpha_i)$, 一个多项式分享 k 个秘密 x_1, \dots, x_k , 其中 $\beta_1, \dots, \beta_k, \alpha_1, \dots, \alpha_n \in \mathbb{F}$ 为不同的域元素 (更小的腐化门限 $t + k \leq (n + 1)/2$)



(n, t) 门限线性秘密分享 (4)

3种秘密分享的共性

- 均满足线性性质（加法同态性），均可用于设计高效的MPC协议

3种秘密分享的区别

- 针对恶意敌手，加法秘密分享需要结合信息论消息认证码（**IT-MAC**）实现秘密的认证，Shamir 及重复秘密分享直接保证了秘密的认证性（由于大部分实体是诚实的，**重构需要 $t+1/n$ 个分享**）
- 重复秘密分享适合于**少量参与方**（分享尺寸与参与方数量成指数关系，例如：3-9 个参与方），Shamir 秘密分享适合于**大量参与方**

秘密重构 $Rec([x], i)$

- P_i 从**安全通道**收到 x 的分享 $[x]$ （记 $[x]$ 为秘密 x 的分享）
 - 加法秘密分享： $x = \sum_{i=1}^n x^i$ （认证性：用 IT-MAC 验证）
 - Shamir 秘密分享：拉格朗日插值方法计算 $x = f(0)$ （认证性：验证所有分享落在多项式 f 上）
 - 重复秘密分享： $x = x^1 + x^2 + x^3$ （认证性：验证分享的一致性）

信息论消息认证码 (IT-MAC)

BDOZ

$$M = K + x \cdot \Delta \in \mathbb{F}$$

- Δ : 全局密钥
- K : 本地密钥, 仅用一次
- x : 消息
- M : 消息认证码 (MAC)

SPDZ

$$M = x \cdot \Delta \in \mathbb{F}$$

- Δ : 全局密钥
- x : 消息
- M : 消息认证码 (MAC)
- 更紧致

伪造不同消息的 MAC 等价于 获得了 Δ , 发生概率 $1/|\mathbb{F}|$

- [BDOZ11] Rikke Bendlin, Ivan Damgård, Claudio Orlandi, and Sarah Zakarias. Semi-homomorphic encryption and multiparty computation. In *EUROCRYPT 2011*
- [DPSZ12] Ivan Damgård, Valerio Pastro, Nigel P. Smart, and Sarah Zakarias. Multiparty computation from somewhat homomorphic encryption. In *CRYPTO 2012*

信息论消息认证码 (IT-MAC)

BDOZ

$$M = K + x \cdot \Delta$$



$$M_j[x^i] = K_j[x^i] + x^i \cdot \Delta_j (\text{相互认证})$$

P_i 的份额 x^i 被 P_j 的密钥认证

- 适用于设计恶意敌手模型下常数轮的 MPC 协议
- 通常用 Hash 函数实现**批量认证**
(认证通信开销常数小)

更多细节将在第二讲中介绍

SPDZ

$$M = x \cdot \Delta$$



$$\sum_{i \in [1, n]} M^i = \sum_{i \in [1, n]} x^i \cdot \sum_{i \in [1, n]} \Delta^i$$

认证码、秘密和全局密钥均被加法分享

- 适用于设计 SPDZ 类 MPC 协议，用 IT-MAC 转化半诚实 GMW 协议到恶意安全
- 通常用随机线性组合方法实现**批量认证**
(认证通信开销常数小)

更多细节将在第二讲中介绍

零知识证明

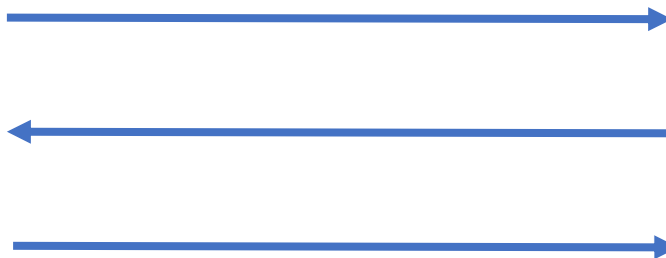


证明者
 (x, w)

$(x, w) \in R$

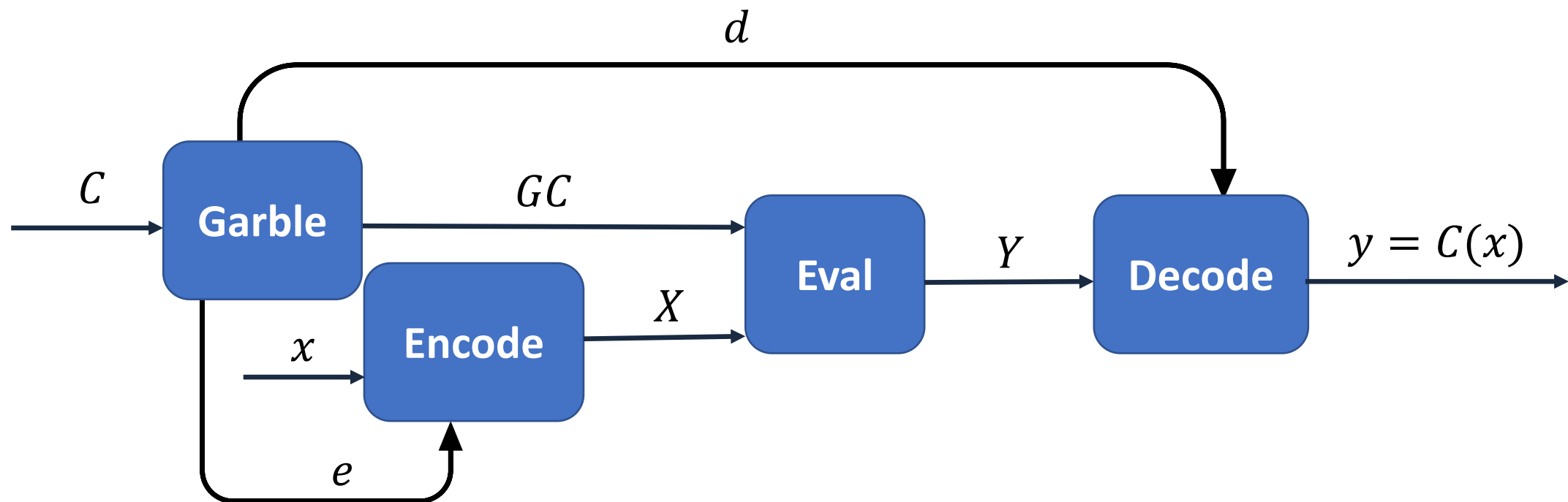


验证者
 x



- 完备性 (Completeness)
- 可靠性 (Soundness)
- 零知识性 (Zero Knowledge)

混淆电路定义

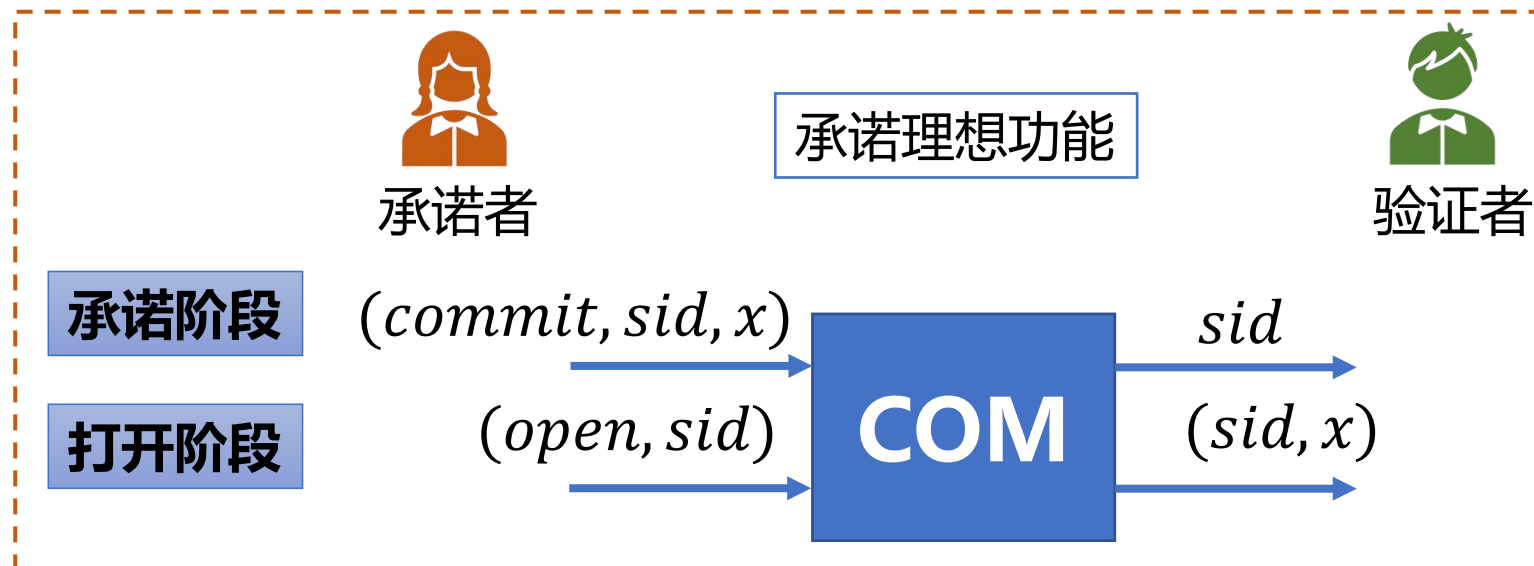


- C 为电路, GC 为混淆电路
- e 编码信息, d 解码信息
- x 电路输入, y 电路输出
- X 输入编码, Y 输出编码

- Garble 混淆算法
- Encode 编码算法
- Decode 解码算法
- Eval 计算算法

混淆电路构造见第三讲

承诺方案



隐藏性

承诺阶段，不泄漏消息 x 的任何信息

绑定性

打开阶段，承诺不能打开为不同消息 x'

- 常用于设计不诚实大多数恶意敌手模型下 MPC 协议，也用在零知识证明协议设计中
- 实际高效 MPC常用承诺方案： $\text{Commit}(x, r) = H(x, r)$ ，其中 H 为随机预言机， r 为随机数

投币 (coin-tossing) 协议



P_i



P_j

r

不诚实大多数 MPC 协议



P_i



P_j

$Commit(r_i)$

$Open\ r_i$

$\bigoplus_{i \in [1, n]} r_i$

- 多个参与方共同生成一个随机数 r
- 不诚实参与方 (即使合谋) 不能控制随机数 , 使其偏离既定分布
- 例如 : 产生均匀随机串 $r \in \{0,1\}^n$

诚实大多数 MPC 协议



P_i



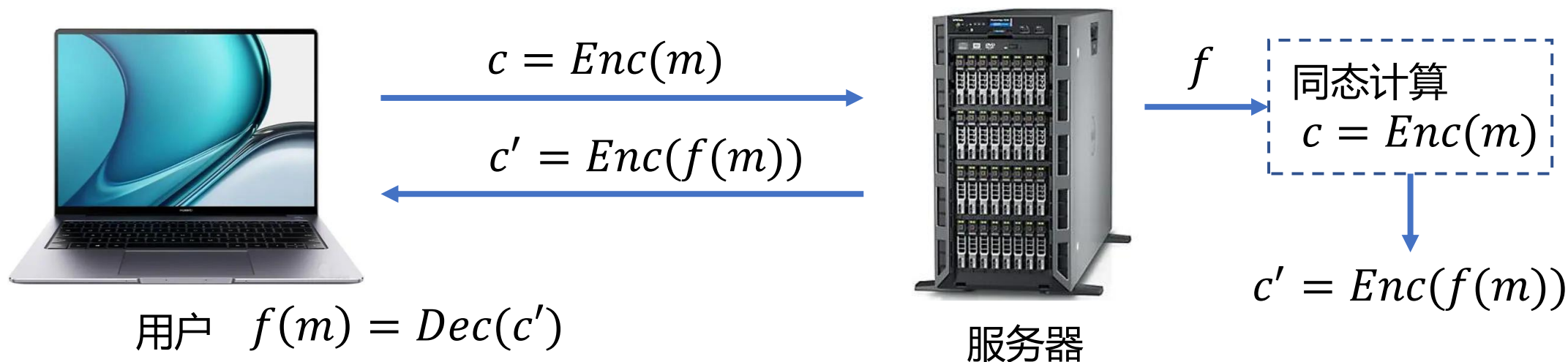
P_j

$[r]$

$r \leftarrow Open([r])$

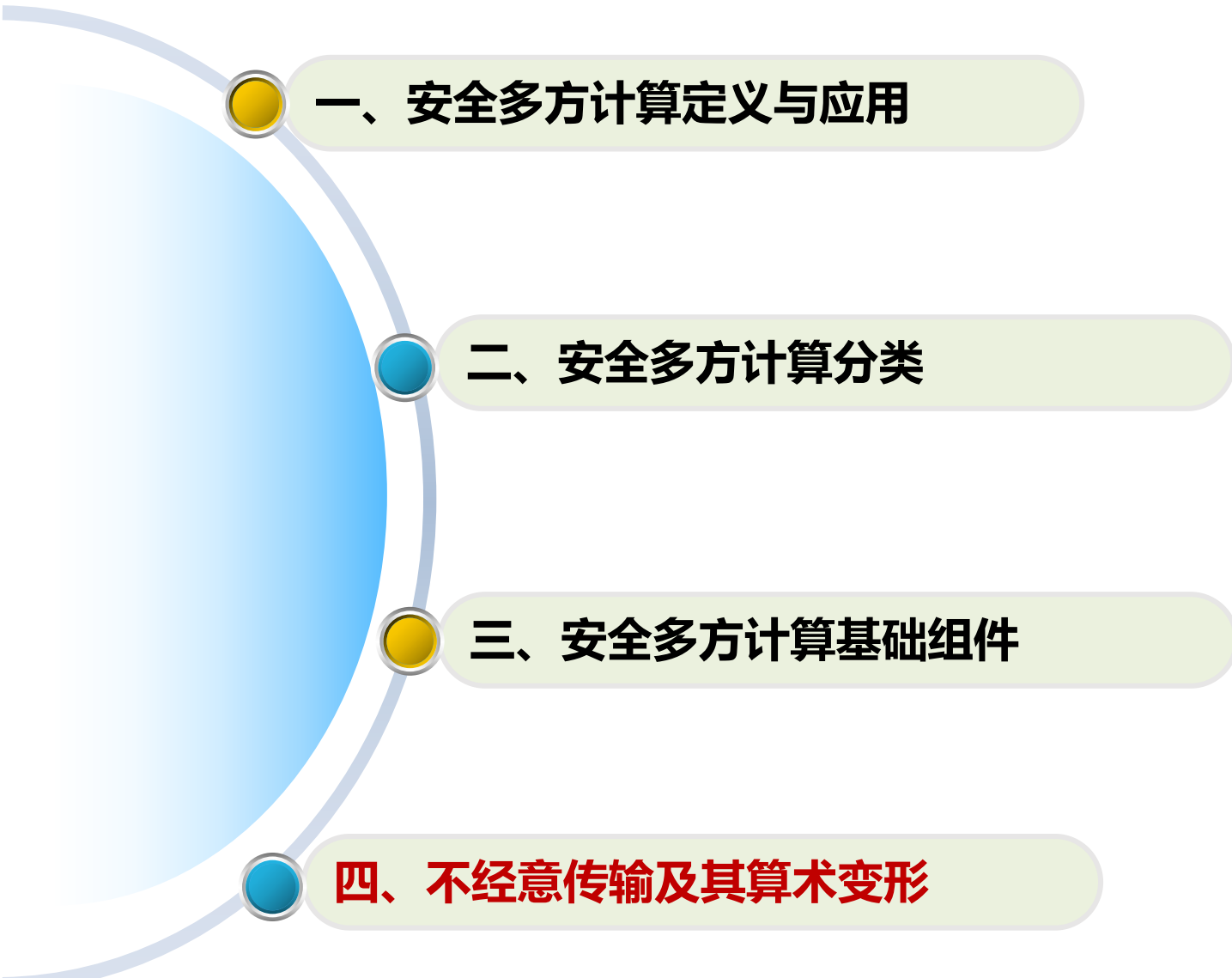
生成随机分享 $[r]$ 方法见第二讲

同态加密 (HE)



- 全同态加密 (FHE) : f 为任意电路 (包括乘法、加法等运算)
- 层级同态加密 (Leveled FHE) : f 为指定深度的任意电路
- 类同态加密 (SHE) : f 为深度浅的任意电路
- 加法同态加密 (AHE) : f 为只包括加法等线性操作的电路

实际高效 MPC 协议主要采用



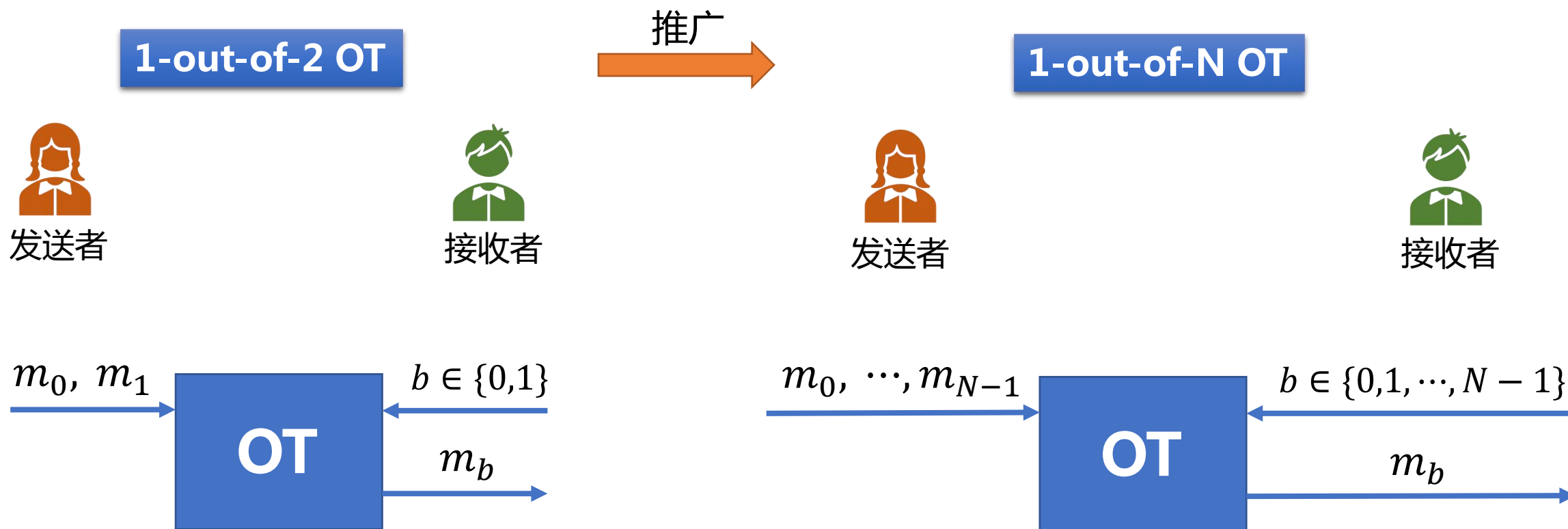
一、安全多方计算定义与应用

二、安全多方计算分类

三、安全多方计算基础组件

四、不经意传输及其算术变形

不经意传输 (OT) 基本定义



- **OT 完备性** : OT 能用于建立关于任何函数的 MPC 协议 [STOC:Kilian88,C:CGT95]
- **1-out-of-2 OT** \Rightarrow 1-out-of-N OT \Rightarrow k-out-of-N OT [JC:NP05]

不经意传输 (OT) 背景

- OT 依赖于公钥密码组件 [C:IR88]
- OT 能基于不同困难假设构造
 - Enhanced trapdoor permutation [EGL85]
 - DDH [C:PVW08,CCS:MR19,CCS:MRR20,AC:MRR21]
 - CDH [SODA:NP01,LATINCRYPT:CO15,PKC:CSW20,EC:DGHMW20]
 - LWE [C:PVW08,IMACC:BDGM19,SCN:Quach20]
 - LPN [EC:DGHMW20]
 - SIDH [AFRICACRYPT:Vitse19,EC:LGG21]

基础 OT 协议举例：Naor-Pinkas 协议



发送者
 (m_0, m_1)



接收者
 $b \in \{0, 1\}$

循环群 (\mathbb{G}, q, g, C)

PK_0

$PK_b = g^k, PK_{1-b} = C/PK_b$

• $r \leftarrow \mathbb{Z}_q, g^r, C^r$ ➡ 预计算

• $(PK_0)^r, (PK_1)^r = C^r / (PK_0)^r$

• $E_0 = H((PK_0)^r, 0) \oplus m_0$

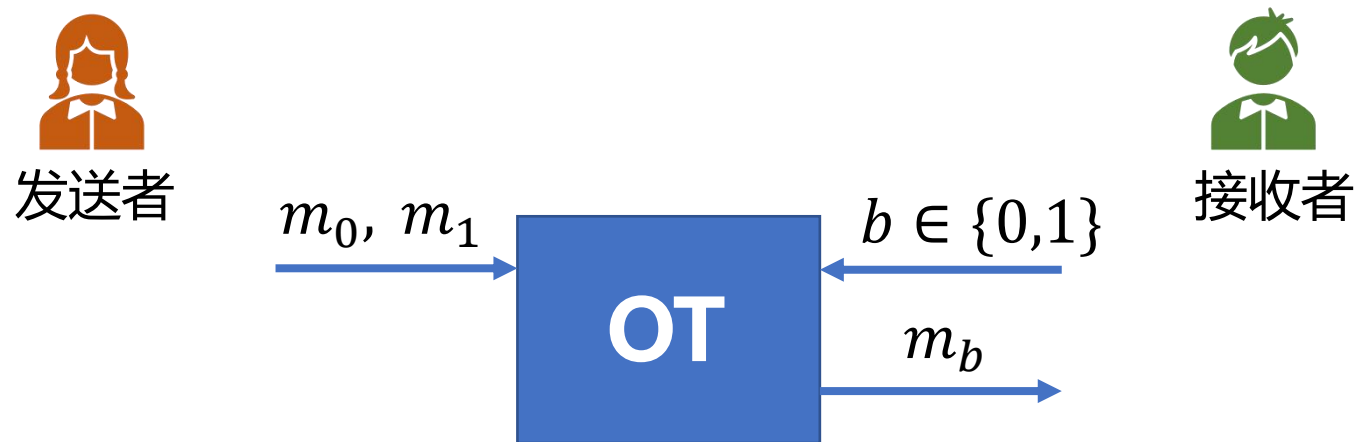
• $E_1 = H((PK_1)^r, 1) \oplus m_1$

(g^r, E_0, E_1)

$m_b = H((g^r)^k, b) \oplus E_b$

- 半诚实敌手模型下可证明安全
- CDH困难假设，随机预言机模型

不经意传输 (OT)



- 需要公钥密码操作 —— **计算效率低** (与对称密码操作比较)
- MPC 协议需要大量的OTs (如亿级数量), 使得基于公钥操作的OT协议效率过低

不经意传输扩展 (OT Extension)

不经意传输扩展可以用于解决需要大量 OTs 的问题 [Beaver96, IKNP03]



- IKNP类 — **PRG** : [IKNP03, KK13, ALSZ13, ALSZ15, KOS15, OOS17, PSS17, Scholl18, Roy22] , 利用伪随机生成器实现扩展
- PCG类 — **LPN** : [BCGI18, BCGIKS19, BCGIKRS19, SGRR19, YWLZX20, CRR21, GYWZXZL22] , 基于LPN问题中噪音的稀疏性实现扩展
- PCF类 — **LPN/DCR/DDH** : [BCGIKS20, OSY21, BCGIKRS22, ADOS22] , 目前实际效率低

不经意传输扩展协议比较

OT 类型	通信效率	计算效率	通信轮数	困难假设
IKNP类	线性, 低	线性, 高	$O(1)$	OWF
PCG类	亚线性, 高	线性, 低	$O(1)$	LPN
PCF类	亚线性, 高	线性, 更低	$O(1)$	LPN/DCR/DDH的变形

大部分 OT 扩展协议：

COT



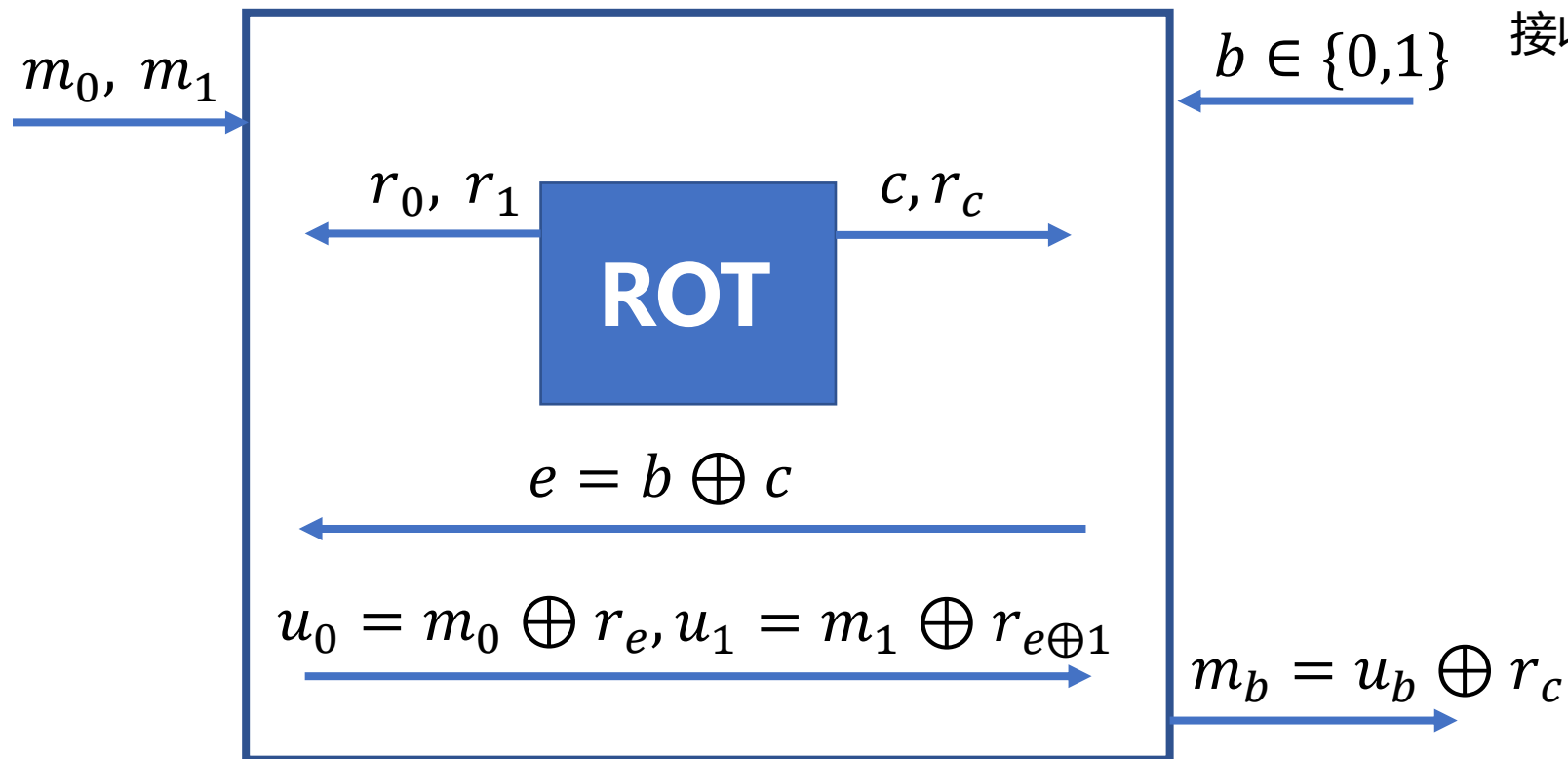
ROT



OT

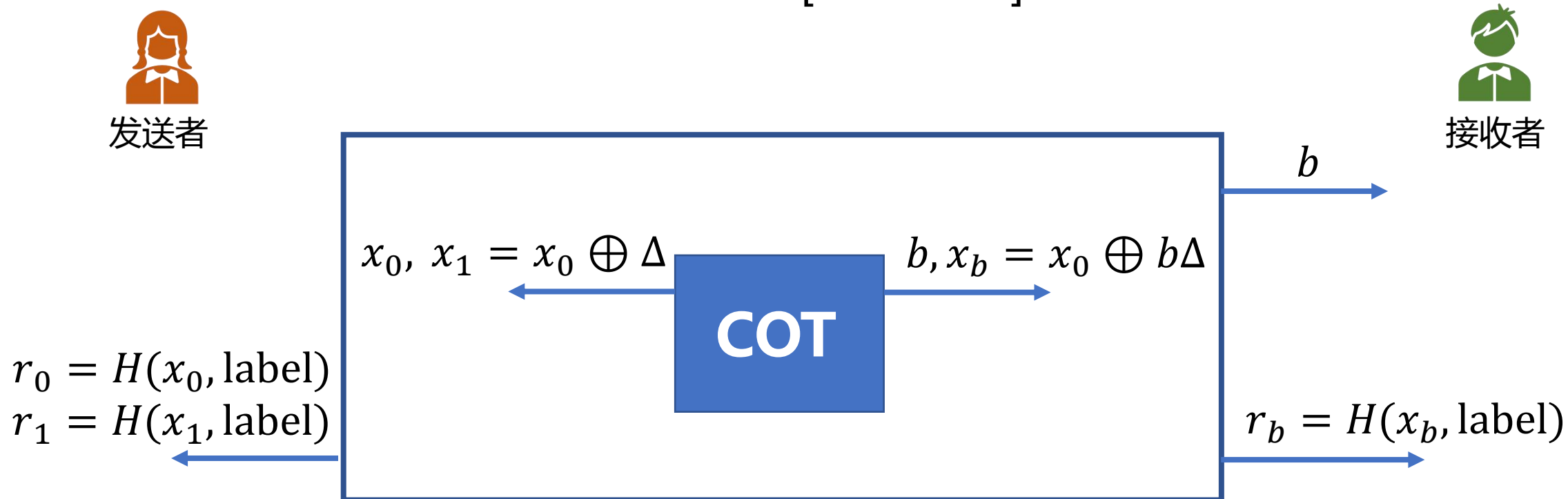
随机不经意传输 (ROT)

ROT \Rightarrow OT [C:Beaver95]



相关不经意传输 (COT)

COT \Rightarrow ROT [C:IKNP03]



- 相关强健 Hash 函数 : **Correlation Robust Hash Function (CRHF)**
- 杂凑函数 (RO) : SHA256, SM3, SHA3, ...
- 密钥固定的分组密码 (RPM) : AES, SM4, ... [SP:BHKR13, SP:GKWWY20]

向量不经意线性函数计算 (VOLE)



发送者



接收者

$\Delta \in \mathbb{F}, v \in \mathbb{F}^n$



- 向量不经意线性计算 (VOLE), COT 算术变形, 常定义在大域 \mathbb{F} 上
- 可用于生成 IT-MACs

不经意线性函数计算 (OLE)



发送者



接收者



- 不经意线性计算 (OLE) , OT 算术变形 , 常定义在大域 \mathbb{F} 上
- 可用于生成 Beaver 三元组