COMPUTER SCIENCE & ENGINEERING
UNIVERSITY of WASHINGTON

compact adaptively secure attribute-based encryption

紧凑、适应性安全的属性加密

罗辑 / luoji@cs.washington.edu

与林蕙佳合作

IIIS Seminar / 2021 年 12 月 4 日

讲故事

No mathematical idea has ever been published ***in the way it was discovered***. Techniques have been developed and are used, if a problem has been solved, to turn the solution procedure ***upside down***, or if it is a larger complex of statements and theories, to turn definitions into propositions, and propositions into definitions, ***the hot invention into icy beauty***.

—— Hans Freudenthal [Fre02]

“数学家收起了**火热的思考**，留下了冰冷的美丽。”

——杨晶

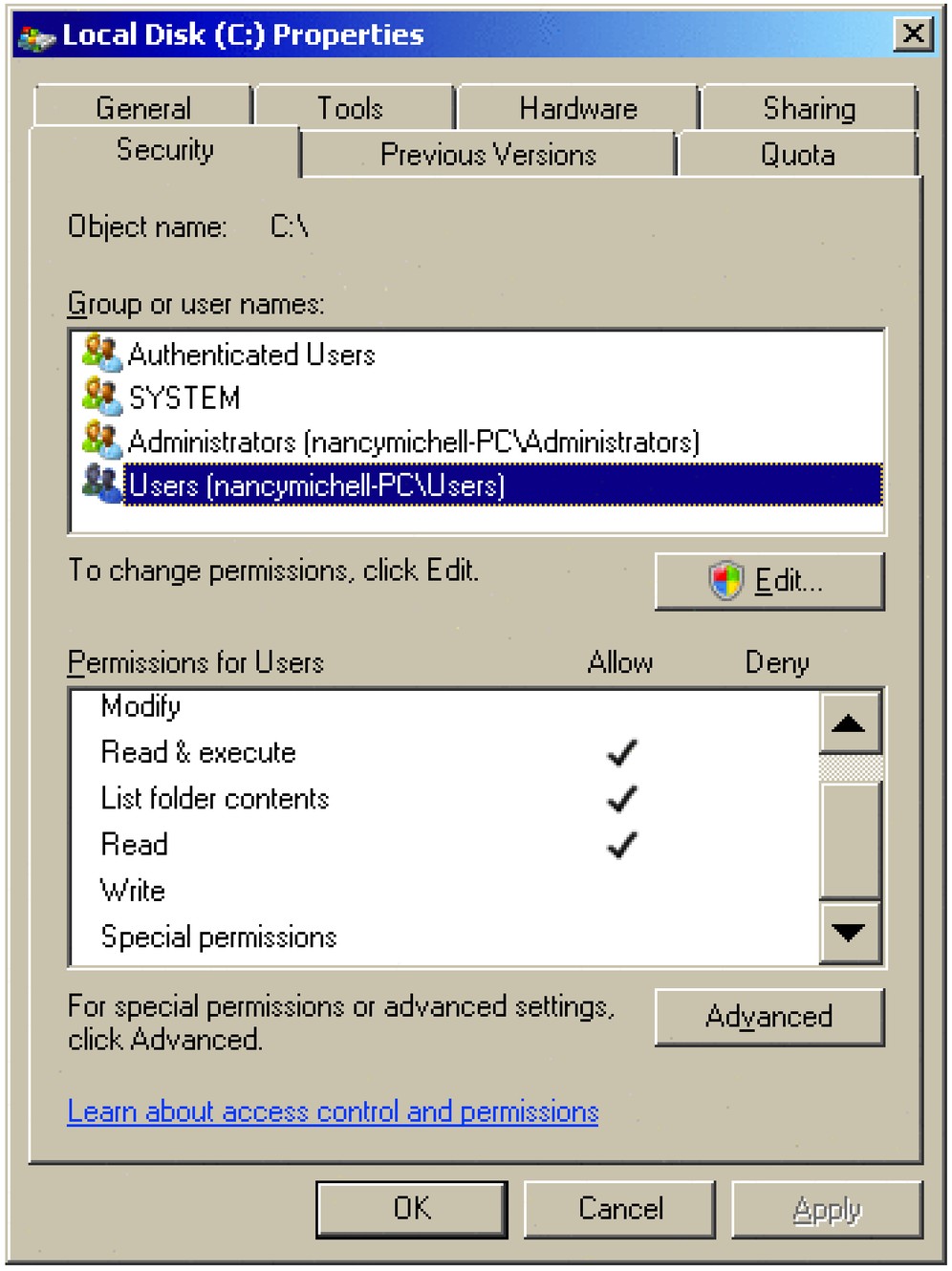
文件权限 / 访问控制

- 一家公司
- 内网服务器，存内部文件
- 不同员工，访问权限不同
- **防止越权访问?**

business logic

“业务逻辑”法：

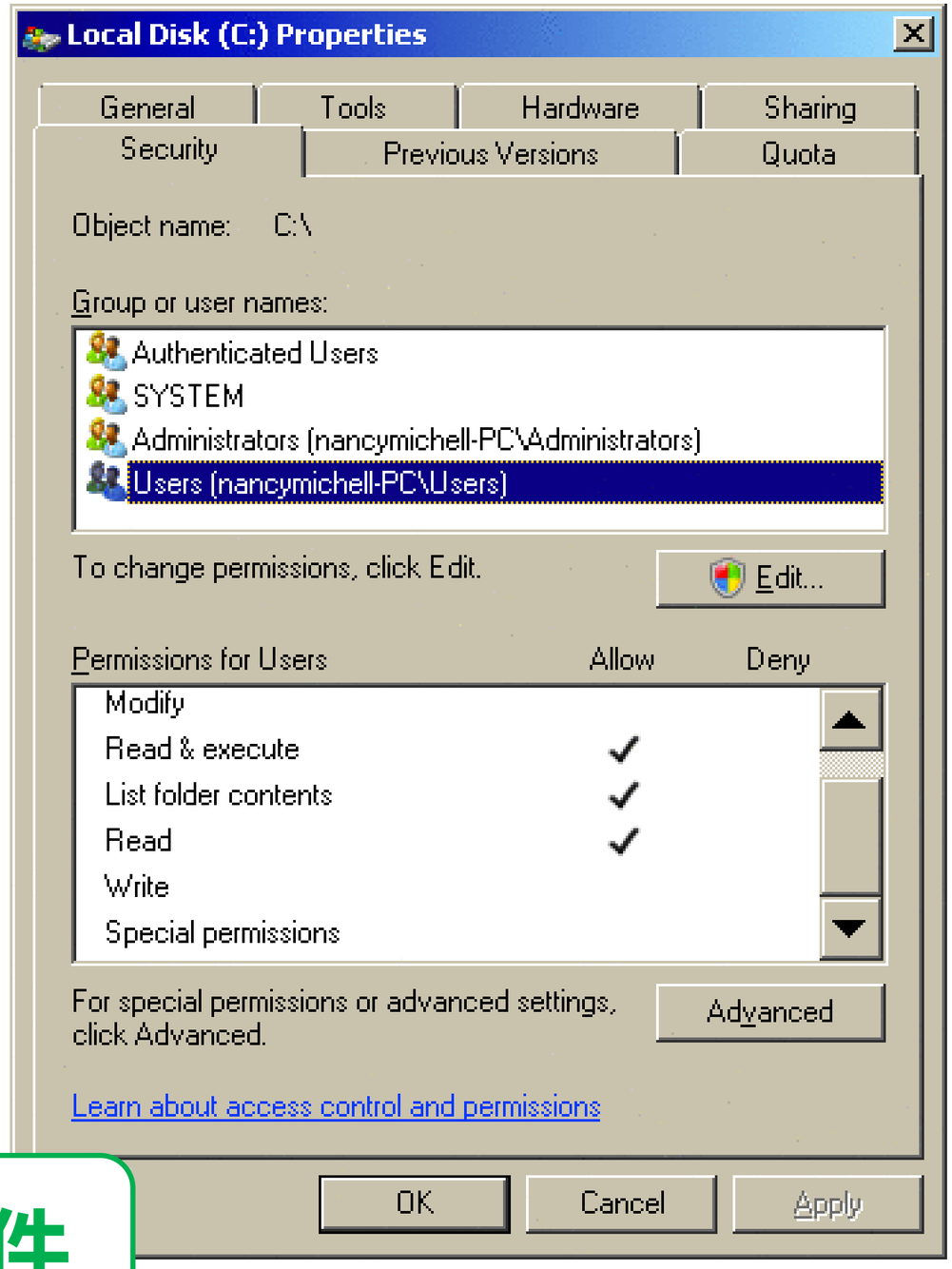
- 建立员工账户
- 设置文件权限
- 服务器判断权限



文件权限 / 访问控制

攻击“业务逻辑”保护的文件

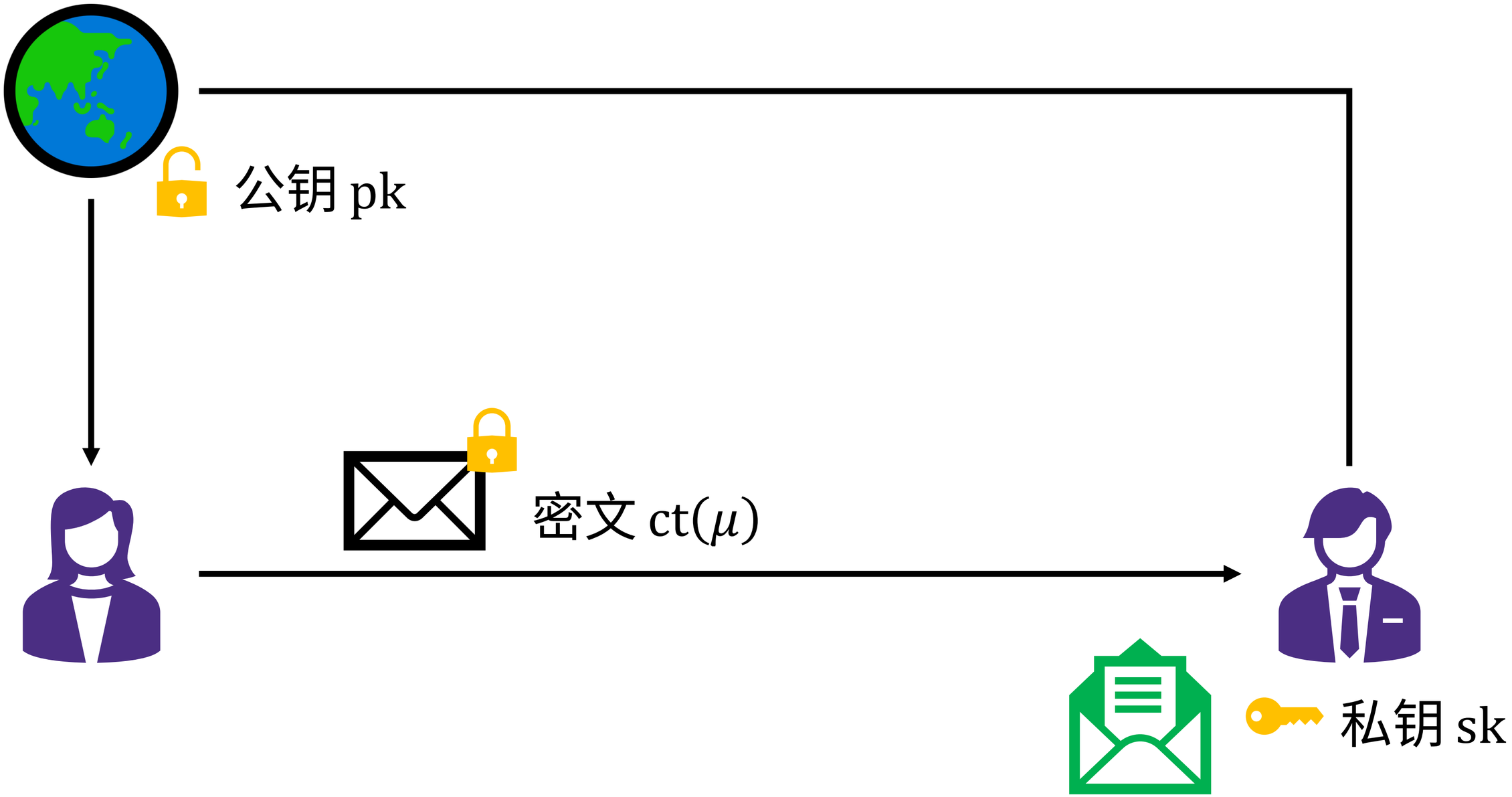
- 寻找漏洞（某个 if 写错了）
- 绕过业务逻辑
  - 劫持服务器网络处理程序
  - 劫持服务器操作系统
  - 直接查看硬盘上的数据



密码学方法：加密存储文件

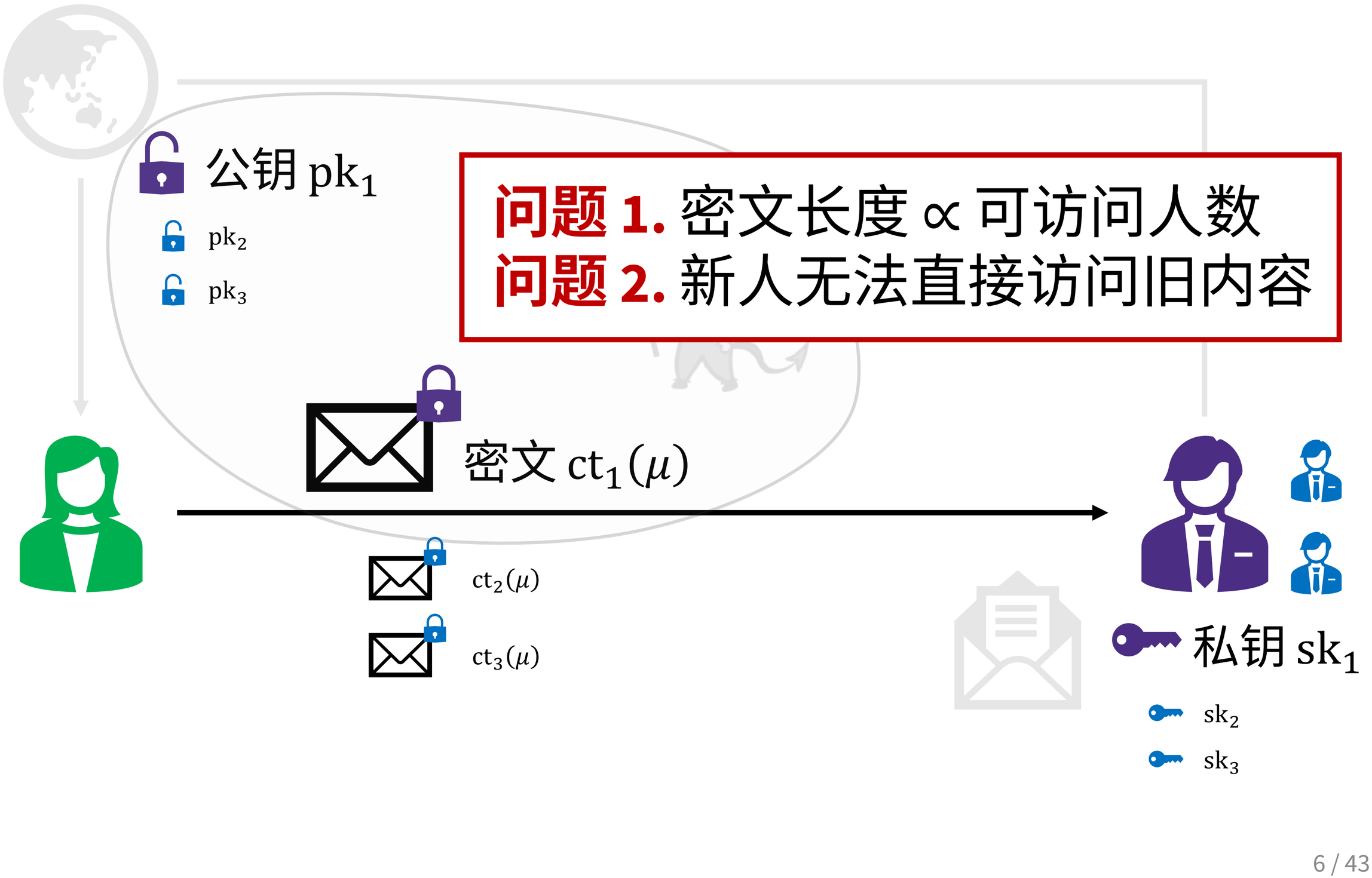
问题. 如何不依赖业务逻辑来保护文件？

公钥加密 (public key encryption)



公钥加密：安全性

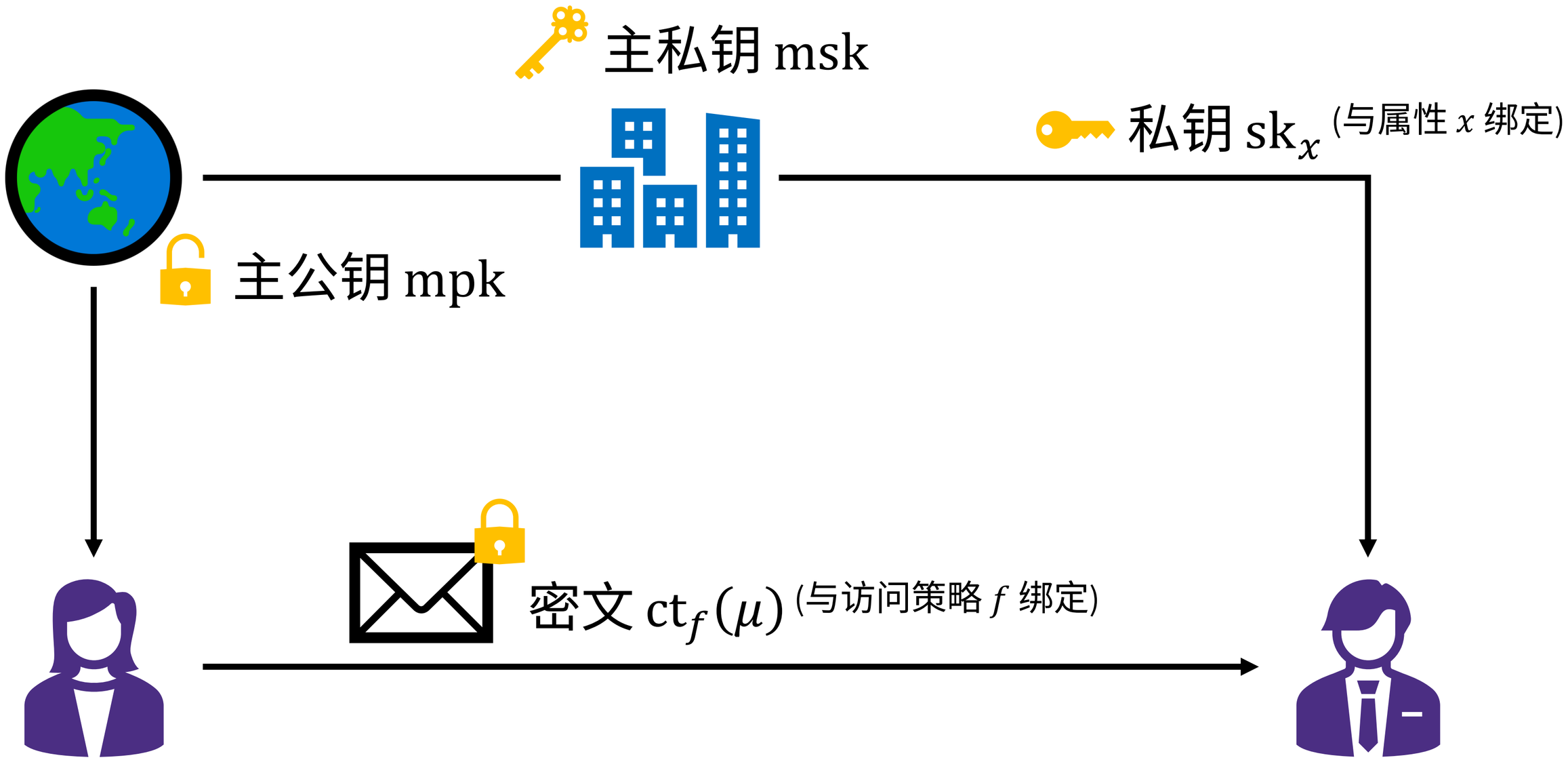
公钥加密用于访问控制



问题 1. 密文长度 $\propto$ 可访问人数

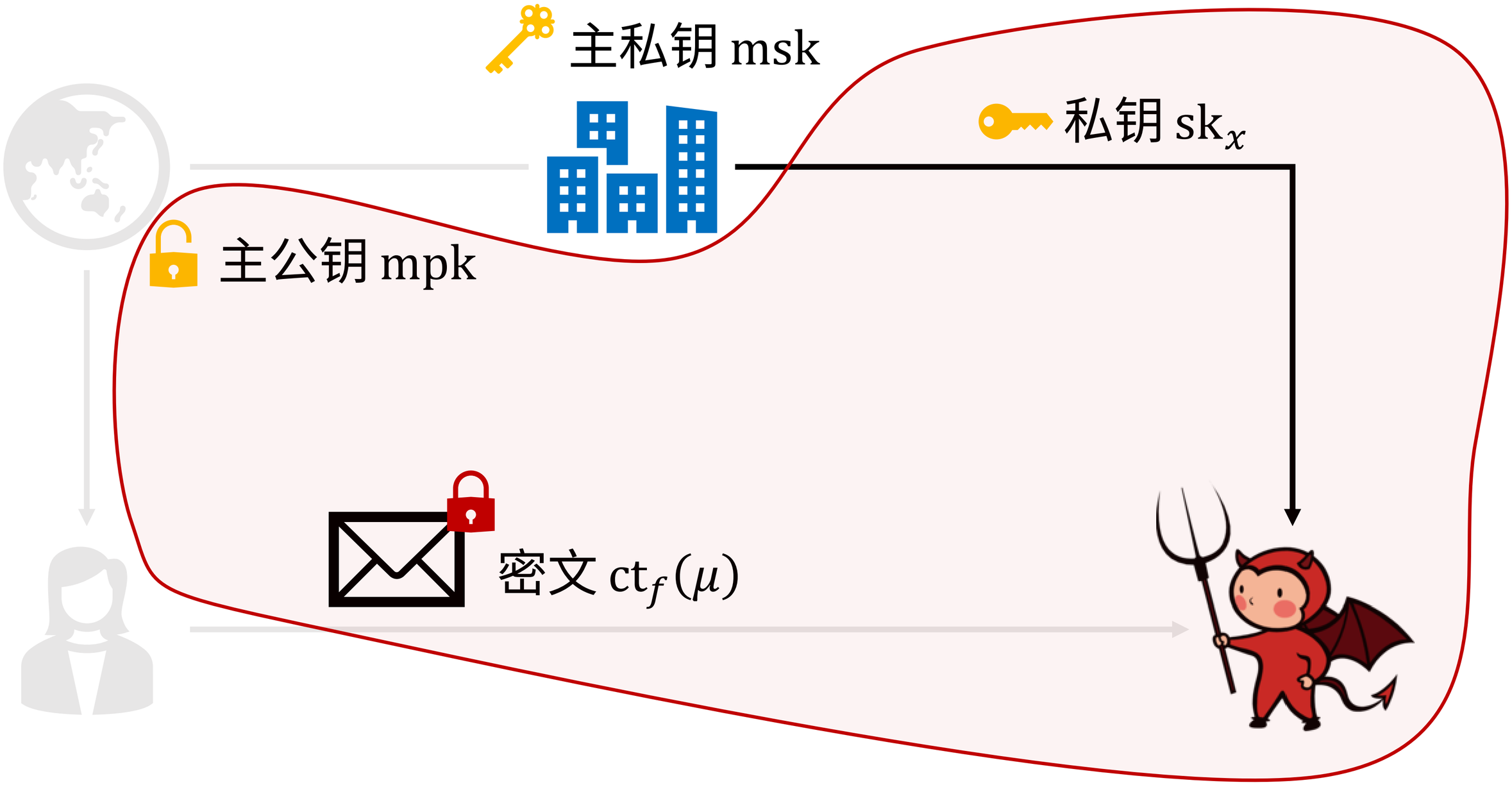
问题 2. 新人无法直接访问旧内容

基于属性的加密 (attribute-based encryption) [GPSW06]



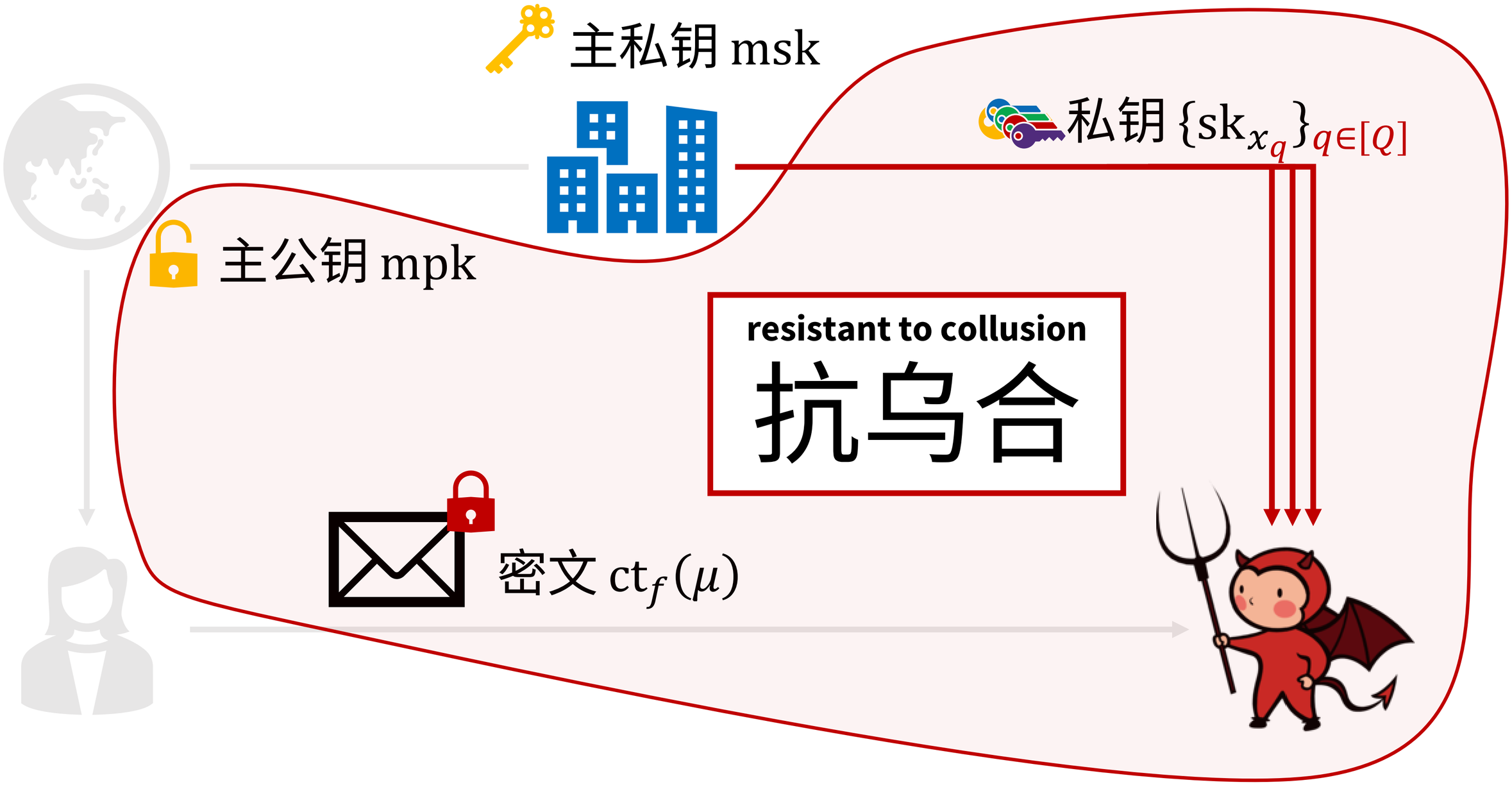
正确性. 若 $f(x) = 1$ 则可以解密得到 μ

基于属性的加密：安全性



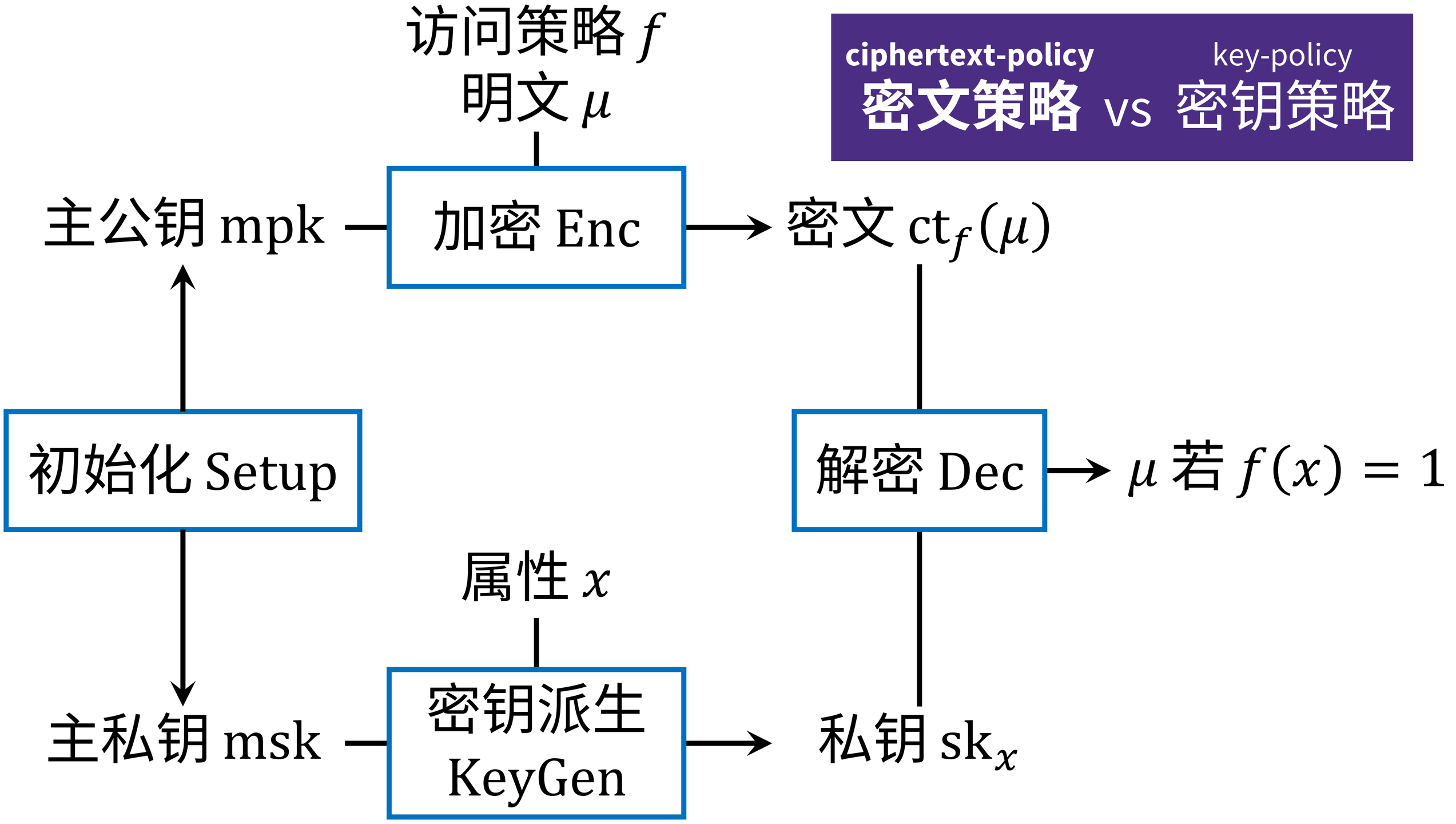
安全性. 若 $f(x) = 0$
则无法获得 μ 的信息

基于属性的加密：安全性



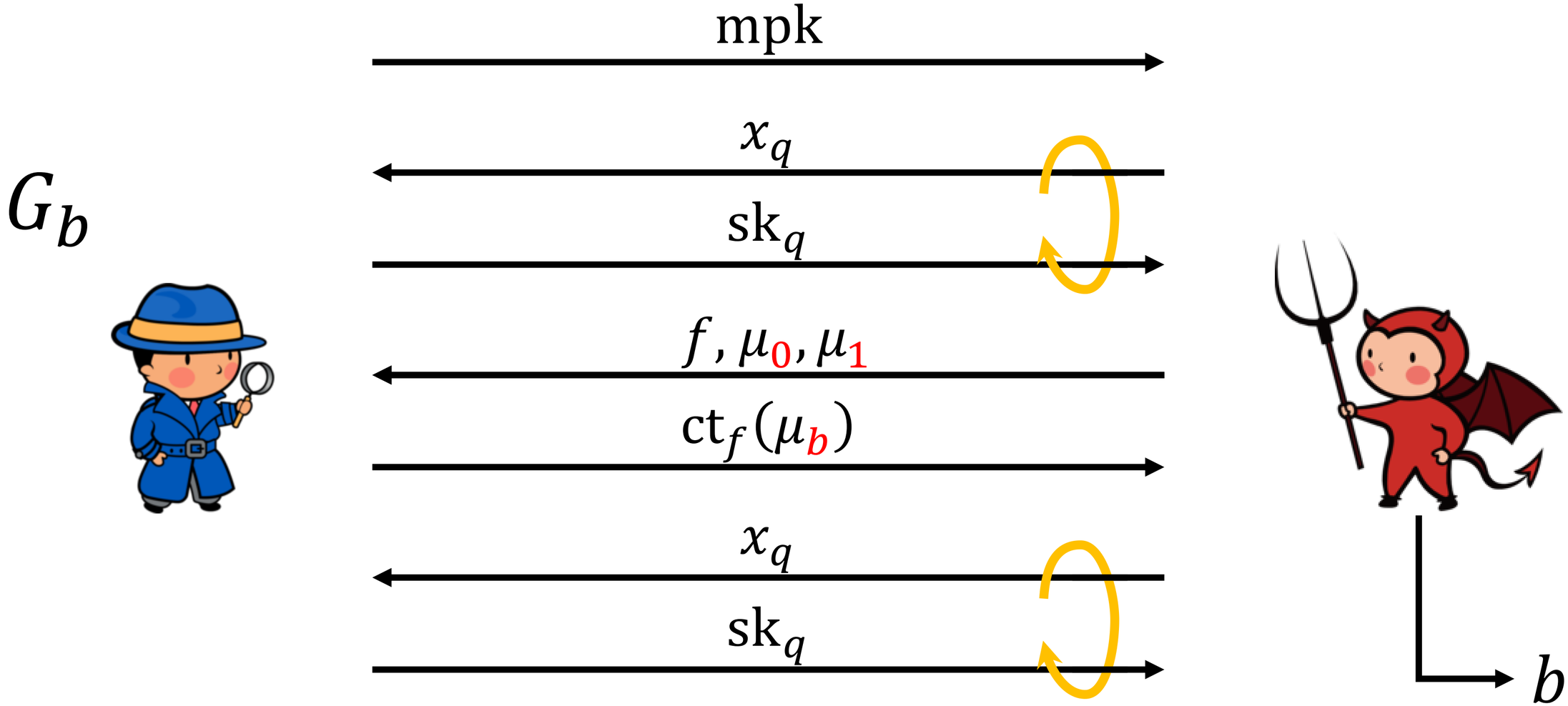
安全性. 若 $f(x_q) = 0$ 对任意 q 成立
则无法获得 μ 的信息

属性加密：形式化定义



ciphertext-policy **密文策略** vs key-policy 密钥策略

属性加密：**适应性**安全的定义



“$G_0 \approx G_1$”：　∀高效$\mathcal{A}$, 若 $f(x_q) = 0$ 对任意 q 成立, 则

$$\Pr_{G_0 \leftrightarrow \mathcal{A}}[b' = 1] - \Pr_{G_1 \leftrightarrow \mathcal{A}}[b' = 1]$$ 绝对值很小.

属性加密：~~选择性~~安全的定义

目标. 适应性安全

G_b

f

mpk

x_q

sk_q

μ_0, μ_1

$\mathrm{ct}_f(\mu_b)$

x_q

sk_q

b'

“$G_0 \approx G_1$”: $\forall$高效$\mathcal{A}$, 若 $f(x_q) = 0$ 对任意 q 成立, 则

$\Pr_{G_0 \leftrightarrow \mathcal{A}}[b' = 1] - \Pr_{G_1 \leftrightarrow \mathcal{A}}[b' = 1]$ 绝对值很小.

计算上不可区分 (computational indistinguishability)

如何理解

$$\varepsilon \overset{\text{def}}{=} \Pr_{G_0 \leftrightarrow \mathcal{A}}[b' = 1] - \Pr_{G_1 \leftrightarrow \mathcal{A}}[b' = 1]?$$

- 假设 $b \overset{\$}{\leftarrow} \{0,1\}$ 均匀随机
- $\mathcal{A}$ 参与 G_0 或 G_1 中随机的一个
- 把 b' 看作 $\mathcal{A}$ 对 b 的猜测

$$\Pr[\mathcal{A} \text{ 猜对}] = \Pr[b = 0]\Pr[b' = 0 | b = 0] + \Pr[b = 1]\Pr[b' = 1 | b = 1]$$

计算上不可区分 (computational indistinguishability)

如何理解

$$\varepsilon \overset{\text{def}}{=} \Pr_{G_0 \leftrightarrow \mathcal{A}}[b' = 1] - \Pr_{G_1 \leftrightarrow \mathcal{A}}[b' = 1]?$$

- 假设 $b \overset{\$}{\leftarrow} \{0,1\}$ 均匀随机
- $\mathcal{A}$ 参与 G_0 或 G_1 中随机的一个
- 把 b' 看作 $\mathcal{A}$ 对 b 的猜测

$$\begin{aligned}\Pr[\mathcal{A}\text{ 猜对}] &= \Pr[b = 0]\Pr[b' = 0 | b = 0] \\ &\quad + \Pr[b = 1]\Pr[b' = 1 | b = 1] \\ &= \frac{1}{2}\left(1 - \Pr_{G_0 \leftrightarrow \mathcal{A}}[b' = 1]\right) + \frac{1}{2}\Pr_{G_1 \leftrightarrow \mathcal{A}}[b' = 1]\end{aligned}$$

计算上不可区分 (computational indistinguishability)

如何理解

$$\varepsilon \stackrel{\text{def}}{=} \Pr_{G_0 \leftrightarrow \mathcal{A}}[b' = 1] - \Pr_{G_1 \leftrightarrow \mathcal{A}}[b' = 1]?$$

- 假设 $b \xleftarrow{\$} \{0,1\}$ 均匀随机
- $\mathcal{A}$ 参与 G_0 或 G_1 中随机的一个
- 把 b' 看作 $\mathcal{A}$ 对 b 的猜测

ε 绝对值很小
$\Leftrightarrow \Pr[\mathcal{A}\text{ 猜对}] \approx 1/2$
$\Leftrightarrow$ 再怎么努力都是瞎猜

$$\begin{aligned}\Pr[\mathcal{A}\text{ 猜对}] &= \Pr[b=0]\Pr[b'=0|b=0] \\ &\quad + \Pr[b=1]\Pr[b'=1|b=1] \\ &= \frac{1}{2}\left(1 - \Pr_{G_0 \leftrightarrow \mathcal{A}}[b'=1]\right) + \frac{1}{2}\Pr_{G_1 \leftrightarrow \mathcal{A}}[b'=1] \\ &= \frac{1}{2} - \frac{\varepsilon}{2}.\end{aligned}$$

过渡证明法 (hybrid argument)

a.k.a. 三角不等式

- 令 $p_G = \Pr_{G \leftrightarrow \mathcal{A}}[b' = 1]$
- “$G_0 \approx G_1$” $\Leftrightarrow \left|p_{G_0} - p_{G_1}\right|$ 很小

假设 $G_0 \approx H$ 且 $H \approx G_1$，则

$$\left|p_{G_0} - p_{G_1}\right| \leq \left|p_{G_0} - p_H\right| + \left|p_H - p_{G_1}\right|$$

也很小

假设 $G_0 \equiv H_0 \approx H_1 \approx \cdots \approx H_n \equiv G_1$ 且 n **不大**，
则 $G_0 \approx G_1$（形式化细节略）

算术公式 (arithmetic formula)

是指形如

$$(x_1 - x_3 + 2)(x_2 + 5x_4) + x_1$$

的表达式

- $\mathbb{F}$ 为域 (也可对交换环定义)
- $x_1, \dots, x_n$ 为 n 个未定元 (变量)

仿射形式. 是指次数不超过 1 的多项式

$$a_0 + a_1 x_1 + \cdots + a_n x_n, \qquad a_i \in \mathbb{F}$$

算术公式. 是指用**有限个**减号、乘号连接有限个仿射形式所得到的表达式 (它是多项式的**一种写法**)

算术公式 (arithmetic formula)

算术公式.

- 可以写成二叉树
  - 叶子节点 = 仿射形式，其他节点 = 减号、乘号
  - **规模 (size) = 叶子数目**
  - **深度 (depth)**
- 这里考虑 $\mathbb{F} = \mathbb{Z}/(p)$，其中 p 是质数

注意. 算术公式 $5x^{10}$ 的规模是 10
该公式符合算术公式定义的写法是

$$(5x)\underbrace{xxxxxxxxx}_{9}$$

支持算术公式的属性加密

令 f 为算术公式，可定义

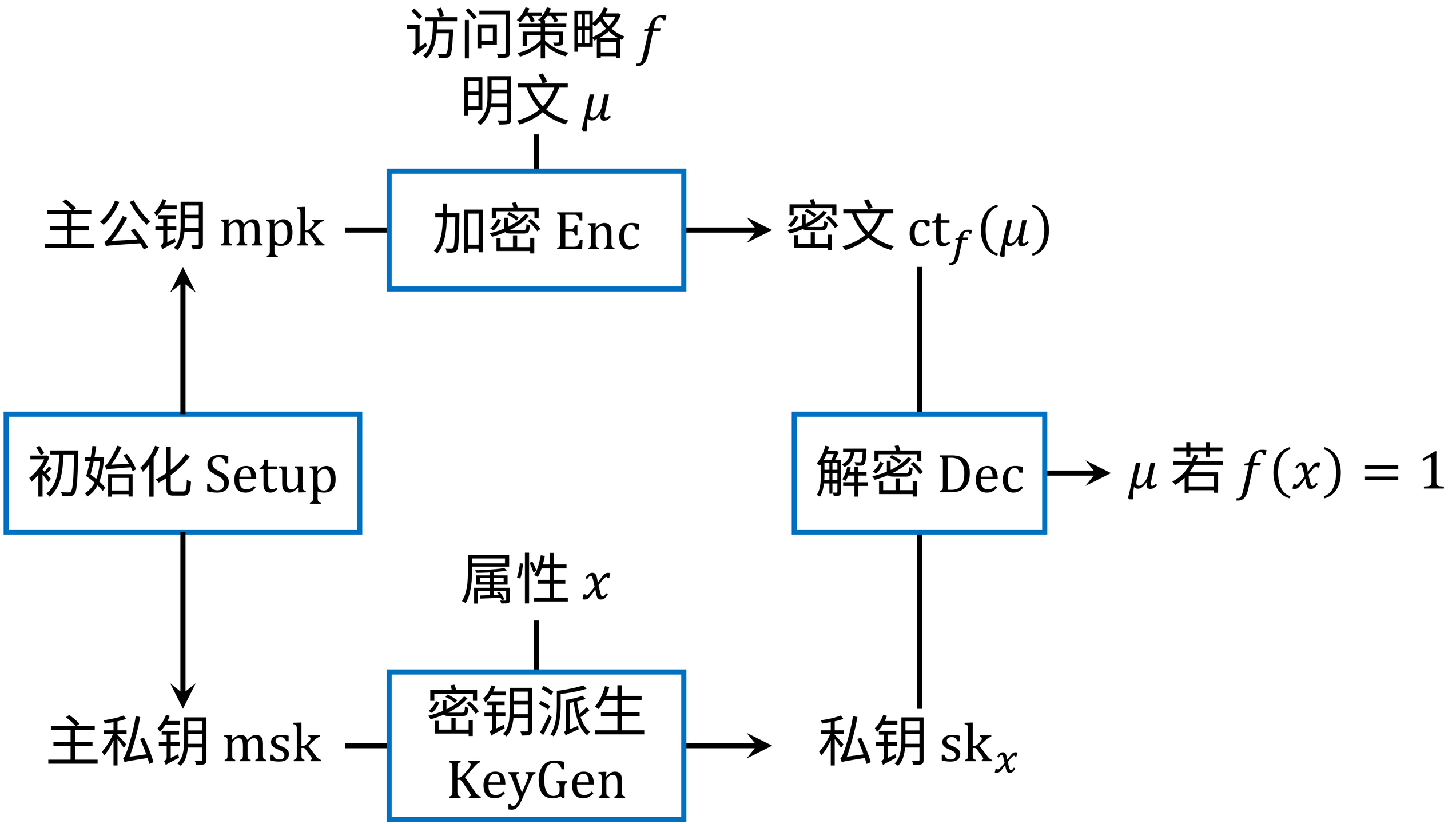
$$f_{\neq 0}(\boldsymbol{x}) = \begin{cases} 1, & f(\boldsymbol{x}) \neq 0; \\ 0, & f(\boldsymbol{x}) = 0. \end{cases}$$

可类似定义 $f_{=0}$

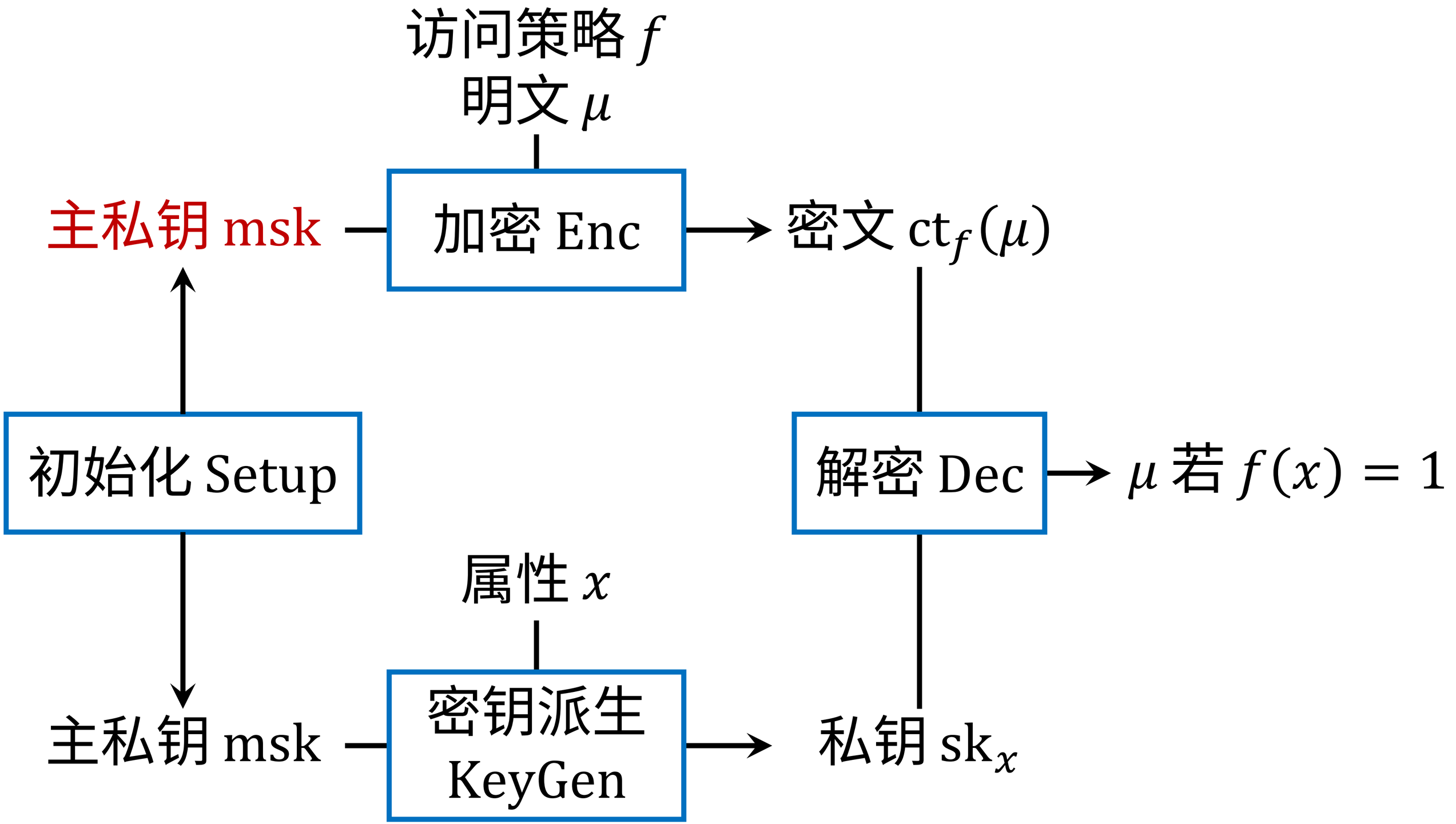
目标. 构造访问策略可以是任意 $f_{\neq 0}, f_{=0}$ 的属性加密

这里只考虑 $f_{\neq 0}$，另一种类似

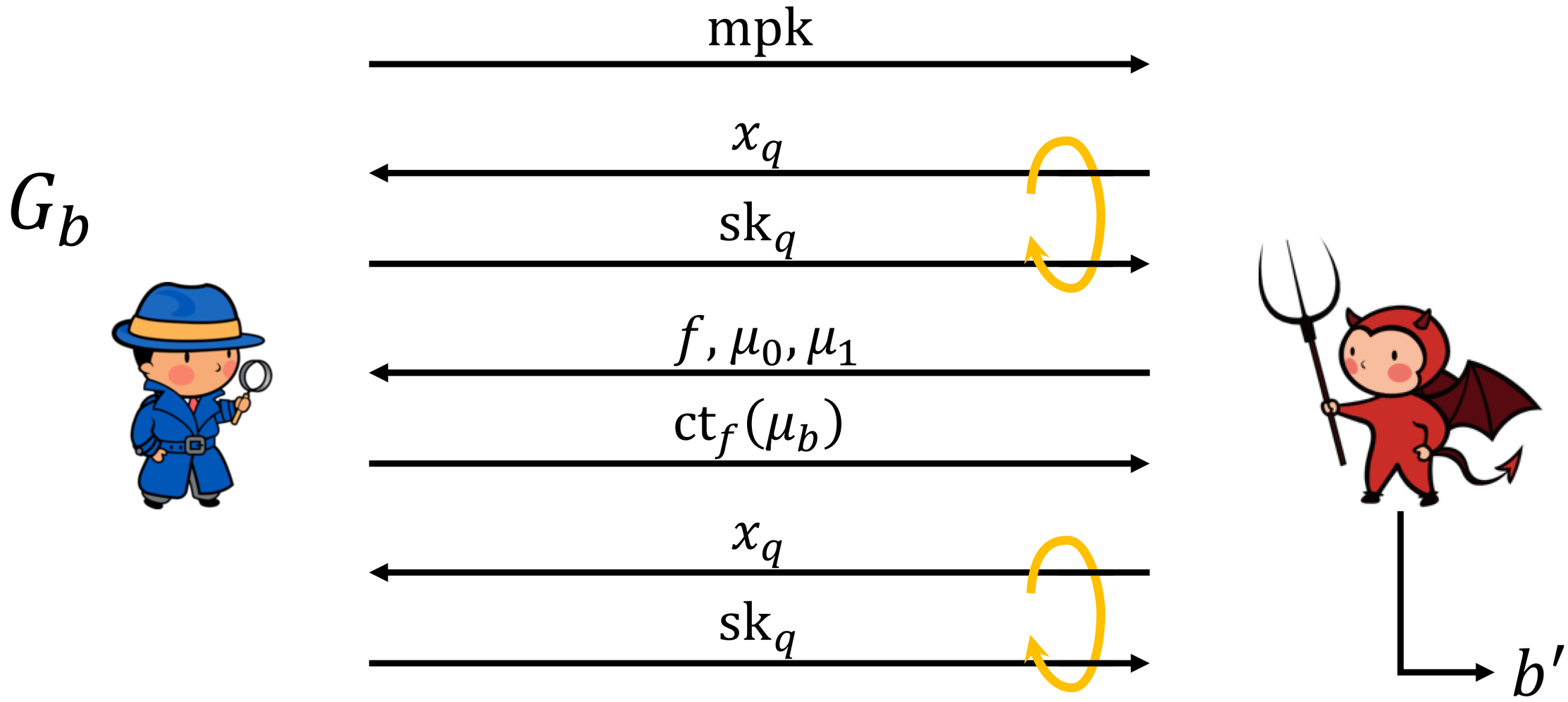
属性加密


访问策略 f
明文 μ
主公钥 mpk
加密 Enc
密文 $\mathrm{ct}_f(\mu)$
初始化 Setup
解密 Dec
μ 若 $f(x)=1$
属性 x
主私钥 msk
密钥派生
KeyGen
私钥 sk_x

属性加密：简化版

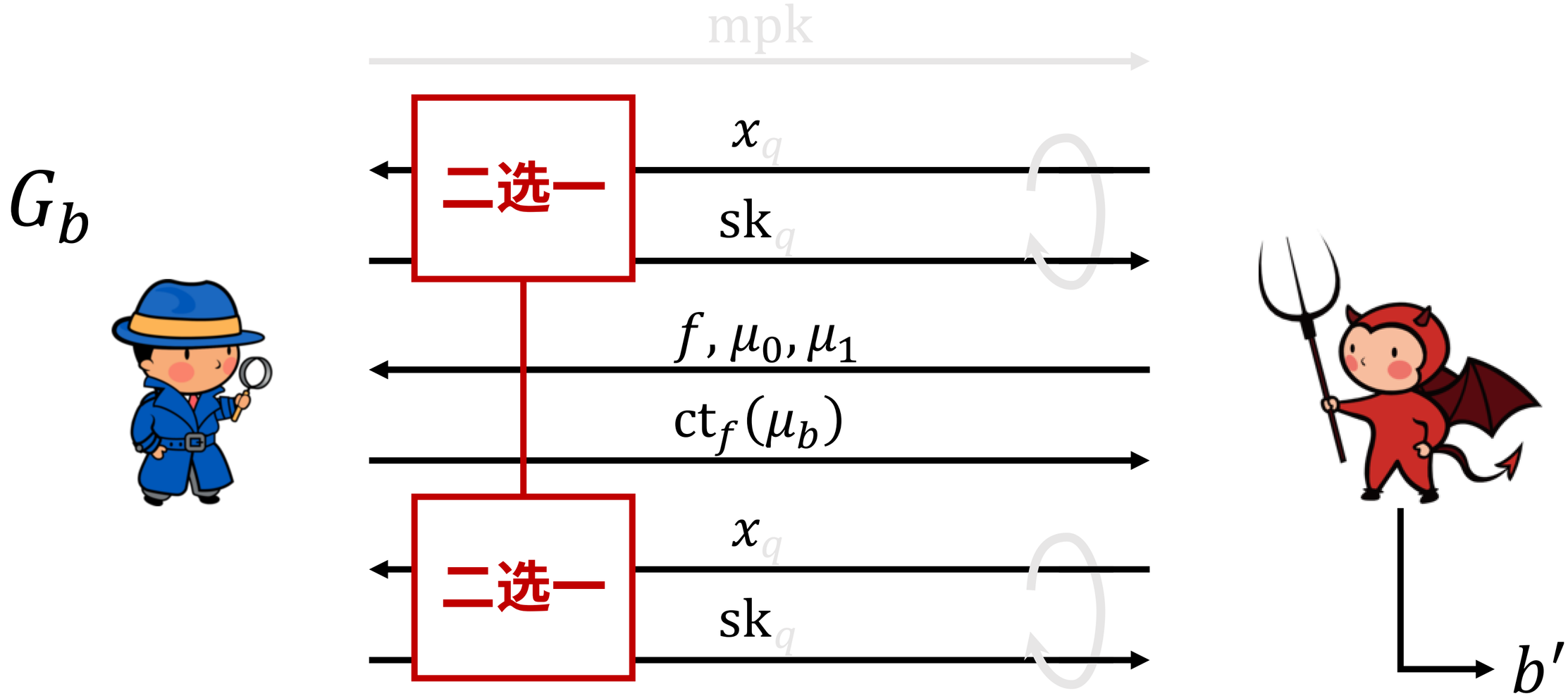

访问策略 f
明文 μ
主私钥 msk
加密 Enc
密文 $\mathrm{ct}_f(\mu)$
初始化 Setup
解密 Dec
μ 若 $f(x)=1$
属性 x
主私钥 msk
密钥派生
KeyGen
私钥 sk_x

属性加密：适应性安全



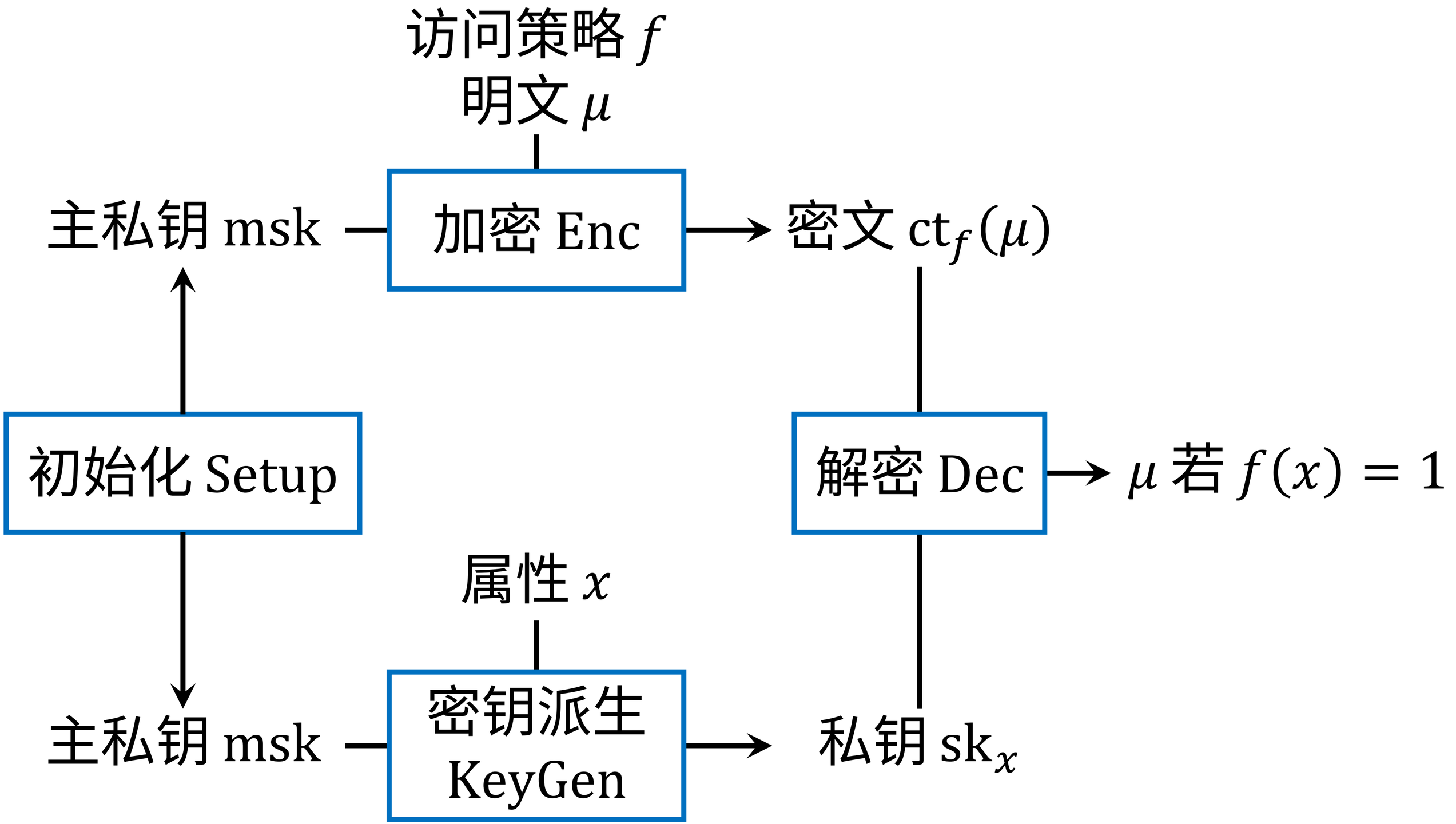
“$G_0 \approx G_1$”，要求 $f(x_q) = 0$ 对任意 q 成立

属性加密：适应性安全，简化版

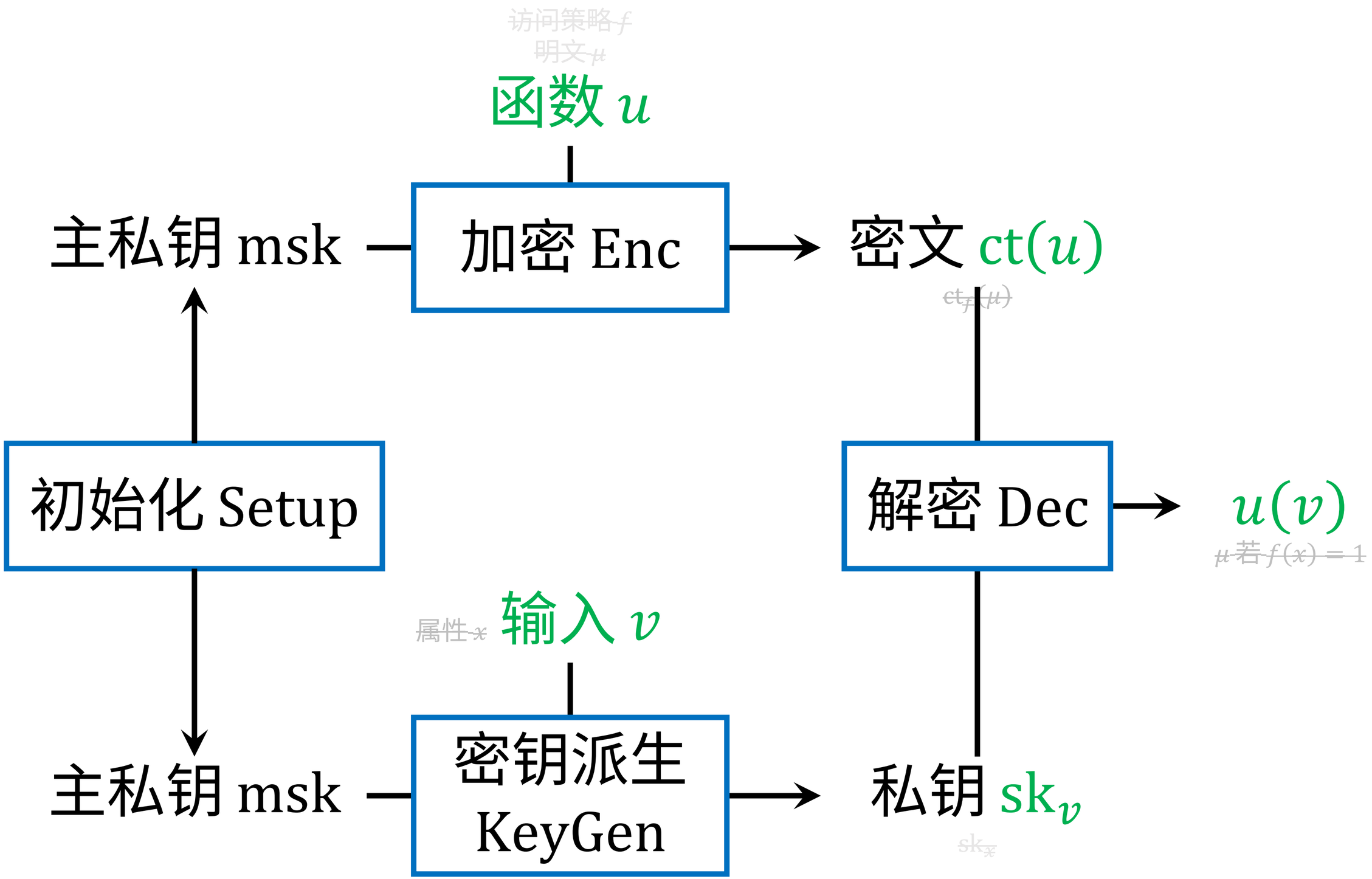


“$G_0 \approx G_1$”，要求 $f(x) = 0$ 且不能有多个 x

属性加密 (attribute-based encryption)



泛函加密 (**functional** encryption) [BSW10]



用泛函加密 (FE) 实现属性加密 (ABE)

- ABE 密钥 = FE 密钥 + $\boldsymbol{x}$ 本身，令 $v \stackrel{\text{def}}{=} \boldsymbol{x}$
- ABE 密文 = FE 密文 + f 本身，令 $u(v) \stackrel{\text{def}}{=} \mu \cdot f(\boldsymbol{x})$

ABE 解密：

1. 从 ABE 密文、密钥中读取 $f, \boldsymbol{x}$，计算 $A = f(\boldsymbol{x})$
2. 若 $A = 0$ 则放弃
3. 否则用 FE 解密算法计算 $B = \mu \cdot f(\boldsymbol{x})$
4. 计算 $\mu = B/A$

问题. FE 反而更**难**构造 (吐槽：所有的归约都是这个意思吧！)

线性泛函加密 (inner-product functional encryption)

令 N 为维数

- $\boldsymbol{v} = (v_1, \ldots, v_N) \in \mathbb{F}^N$ 是 N 维向量
- $u \in (\mathbb{F}^N)^*$ 是对偶空间元素，即线性泛函

说人话：$\boldsymbol{u} = (u_1, \ldots, u_N) \in \mathbb{F}^N$ 是系数

好处. 容易构造!

$$u(\boldsymbol{v}) \stackrel{\text{def}}{=} \boldsymbol{u}^\top \boldsymbol{v} = \sum_{i=1}^{N} u_i v_i$$

也叫“内积加密”(IPFE)

内积加密：双重保密性

对于 IPFE，可以做到同时、多次保护 $\boldsymbol{u}, \boldsymbol{v}$

考虑两组向量

$$\boldsymbol{u}_1^{(0)}, \ldots, \boldsymbol{u}_I^{(0)}, \boldsymbol{v}_1^{(0)}, \ldots, \boldsymbol{v}_J^{(0)}$$
$$\boldsymbol{u}_1^{(1)}, \ldots, \boldsymbol{u}_I^{(1)}, \boldsymbol{v}_1^{(1)}, \ldots, \boldsymbol{v}_J^{(1)}$$

满足内积分别相等

$$\left(\boldsymbol{u}_i^{(0)}\right)^\top \boldsymbol{v}_j^{(0)} = \left(\boldsymbol{u}_i^{(1)}\right)^\top \boldsymbol{v}_j^{(1)} \qquad \forall i \in [I], j \in [J]$$

好处. 容易构造！

问题. 有什么用？

则安全性保证

$$\{\mathrm{ict}(\boldsymbol{u}_i^{(0)})\}_{i\in[I]}, \{\mathrm{isk}(\boldsymbol{v}_j^{(0)})\}_{j\in[J]} \approx \{\mathrm{ict}(\boldsymbol{u}_i^{(1)})\}_{i\in[I]}, \{\mathrm{isk}(\boldsymbol{v}_j^{(1)})\}_{j\in[J]}$$

实际上需要适应性安全（交互式博弈）

算术密钥乱码化 (arithmetic key garbling)

partially hiding randomized encoding

—种特殊的 **部分隐藏　随机化编码** [IW14]　　garbled circuits

（渊源可追溯到 [Yao86] 乱码电路）

设 f 是算术公式，$\alpha, \beta \in \mathbb{F}, \boldsymbol{x} \in \mathbb{F}^n$，想计算 $\alpha f(\boldsymbol{x}) + \beta$

- α, β 需要保护
- $\boldsymbol{x}$ 是公开的
- **例.** 设置 $\alpha = \mu, \beta = 0$，即计算 $\mu \cdot f(\boldsymbol{x})$

需求. 用 IPFE 计算 $\alpha f(\boldsymbol{x}) + \beta$

问题. f 本身很复杂、次数很高，无法用 IPFE 计算

解法. 降次，用**一次函数**表达 $\alpha f(\boldsymbol{x}) + \beta$

算术公式的 AKGS

主要是利用

$$\alpha(f_1 - f_2) + \beta = (\alpha f_1 + \beta + r) - (\alpha f_2 + r)$$
$$\alpha f_1 f_2 + \beta = (r f_1 + \beta) - (-\alpha f_2 + r) f_1$$

其中 r 是任意数 (这里设置为随机数)

- 若 f **规模**是 1，则 $\alpha f + \beta$ 已经是 $\boldsymbol{x}$ 的**仿射函数**
- 递归过程中算术公式**规模**降低
- 该过程将 $\alpha f + \beta$ 分解为数个仿射函数 (规模那么多个)

算术公式的 AKGS：例子

例.

$$f(\boldsymbol{x}) = \overbrace{(x_1 - x_3 + 2)}^{f_1}\overbrace{(x_2 + 5x_4)}^{f_2} - \overbrace{(-x_1)}^{f_3}$$

则

$$\begin{aligned}\alpha f + \beta &= (\alpha f_1 f_2 + \beta + r_1) - (\alpha f_3 + r_1) \\ &= (r_2 f_1 + \beta + r_1) - (-\alpha f_2 + r_2) f_1 - (\alpha f_3 + r_1)\end{aligned}$$

令

$$\begin{aligned}\ell_1 &= L_1(\boldsymbol{x}) = r_2 x_1 - r_2 x_3 + (2r_2 + \beta + r_1) \\ \ell_2 &= L_2(\boldsymbol{x}) = -\alpha x_2 - 5\alpha x_4 + r_2 \\ \ell_3 &= L_3(\boldsymbol{x}) = -\alpha x_1 + r_1\end{aligned}$$

则

$$\alpha f + \beta = \ell_1 - \ell_2 f_1 - \ell_3$$

算术公式的 AKGS：术语

例.

$$f(\boldsymbol{x}) = \overbrace{(x_1 - x_3 + 2)}^{f_1}\overbrace{(x_2 + 5x_4)}^{f_2} - \overbrace{(-x_1)}^{f_3}$$

则

$$\alpha f + \beta = f, \alpha, \beta; \boldsymbol{r} \quad + r_1) - (\alpha f_3 + r_1)$$

$$= \quad - (-\alpha f_2 + r_2) f_1 - (\alpha f_3 + r_1)$$

Garble

乱码化过程

（标签函数的**值**）

标签　　标签函数（$\boldsymbol{x}$ 的**仿射函数**，系数是 $\alpha, \beta, \boldsymbol{r}$ 的**线性函数**）

$$\ell_1 = L_1(\boldsymbol{x}) = r_2 x_1 - r_2 x_3 + (2r_2 + \beta + r_1)$$

$$\ell_2 = L_2(\boldsymbol{x}) = -\alpha x_2 - 5\alpha x_4 + r_2$$

$$\ell_3 = L_3(\boldsymbol{x}) = -\alpha x_1 + r_1$$

则

Eval

求值过程（关于标签 ℓ **线性**，关于 $\boldsymbol{x}$ 的次数可以很高）

$$\alpha f + \beta = \ell_1 - \ell_2 f_1 - \ell_3$$

AKGS 安全定义

想法.

- 计算 $\alpha f + \beta$ 归结为先计算标签 ℓ 再求值
- 标签 $\ell = L(\boldsymbol{x})$ 是 $\boldsymbol{x}$ 的仿射函数，可以用 IPFE 安全计算

自然的问题.

- 显然 ℓ 蕴含着 $\alpha f(\boldsymbol{x}) + \beta$ 的值
- 万一 ℓ 蕴含着 α, β 的其他信息怎么办?

AKGS 安全定义. $\exists \mathcal{S}$ 使 $\forall f, \boldsymbol{x}, \alpha, \beta$ 有

$$\{\ell\} \equiv \mathcal{S}(f, \boldsymbol{x}, \alpha f(\boldsymbol{x}) + \beta; \boldsymbol{s})$$

- 左侧是利用 $f, \alpha, \beta, \boldsymbol{r}, \boldsymbol{x}$ 所算出的标签的分布
- 右侧的分布不含除 $\alpha f(\boldsymbol{x}) + \beta$ 外任何关于 α, β 的信息

算术公式的 AKGS 安全证明

刚刚的 AKGS 满足安全定义吗?

- 可以用结构化归纳证明

归纳奠基. 对于仿射形式 f，有

$$\alpha f + \beta = \ell$$

注意 $\alpha f + \beta$ 值已知，$\mathcal{S}$ 可直接返回 $\alpha f + \beta$ 作为 ℓ

算术公式的 AKGS 安全证明

刚刚的 AKGS 满足安全定义吗?

- 可以用结构化归纳证明

归纳 (减号). 对于

$$\alpha(f_1 - f_2) + \beta = \overbrace{(\alpha f_1 + \beta + r)}^{\gamma_1} - \overbrace{(\alpha f_2 + r)}^{\gamma_2}$$

可以看出 γ_2 均匀随机，而 γ_1 是惟一满足

$$\alpha(f_1 - f_2) + \beta = \gamma_1 - \gamma_2$$

的值

注意 $\alpha(f_1 - f_2) + \beta$ 值已知，因此 S:

1. 选择随机的 γ_2，然后算出 γ_1
2. 对 γ_1, γ_2 递归

算术公式的 AKGS 安全证明

刚刚的 AKGS 满足安全定义吗?

- 可以用结构化归纳证明

归纳 (乘号). 对于

$$\alpha f_1 f_2 + \beta = \overbrace{(r f_1 + \beta)}^{\gamma_1} - \overbrace{(-\alpha f_2 + r)}^{\gamma_2} f_1$$

可以看出 γ_2 均匀随机，而 γ_1 是惟一满足

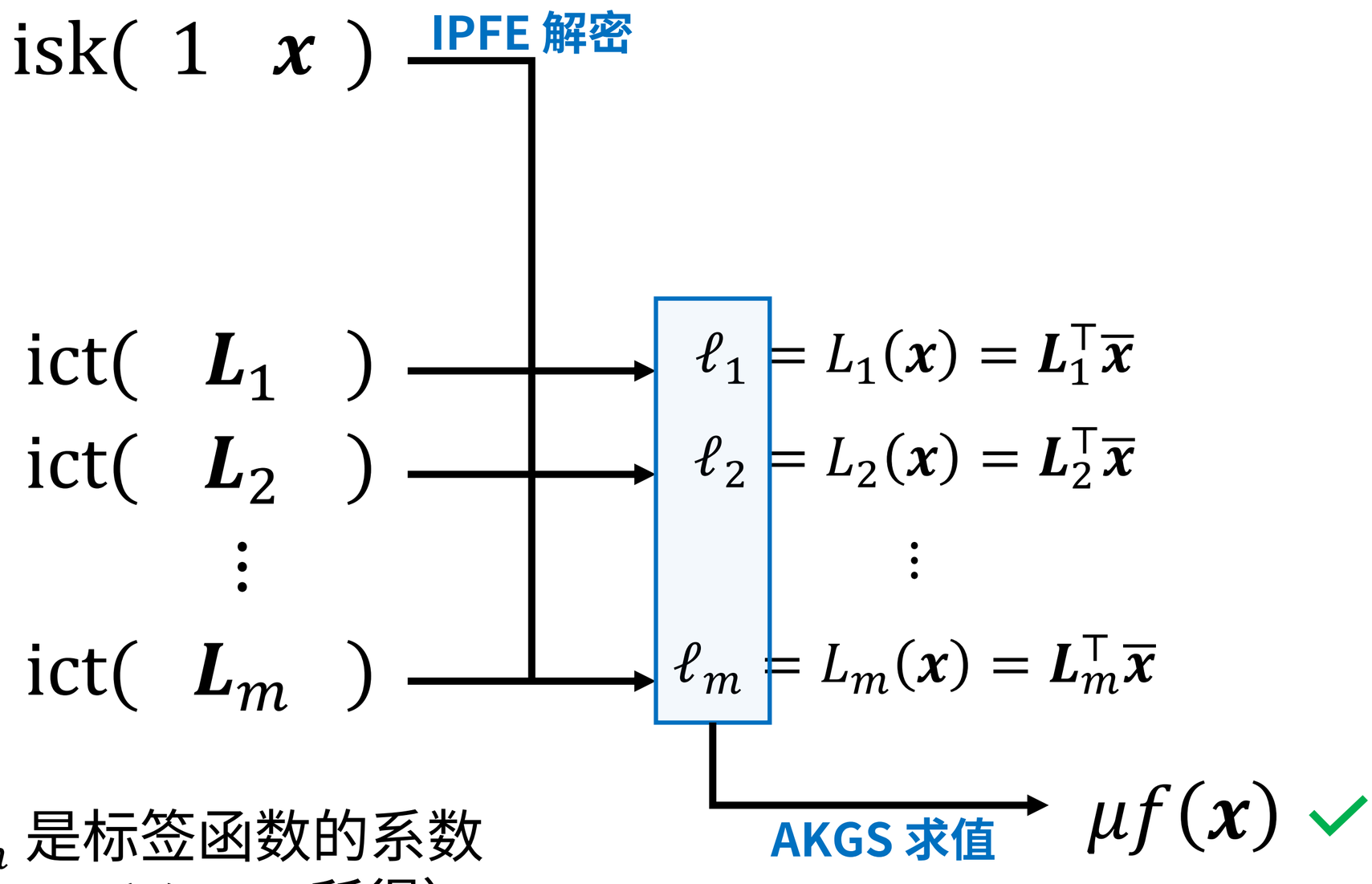
$$\alpha f_1 f_2 + \beta = \gamma_1 - \gamma_2 f_1$$

的值

注意 $\alpha f_1 f_2 + \beta$ 值已知，因此 S:

1. 选择随机的 γ_2，然后算出 γ_1
2. 对 γ_1, γ_2 递归

用 IPFE 和 AKGS 构造 ABE



$\boldsymbol{L}_1, \dots, \boldsymbol{L}_m$ 是标签函数的系数
（乱码化 $\underbrace{\mu}_{\alpha} f(\boldsymbol{x}) + \underbrace{0}_{\beta}$ 所得）

ABE 安全证明

isk(1 $\boldsymbol{x}$)

G_b

ict($\boldsymbol{L}_1$)

ict($\boldsymbol{L}_2$)

$\vdots$

ict($\boldsymbol{L}_m$)

IPFE $\approx$

isk(1 $\boldsymbol{x}$)

ict(ℓ_1 **0**)

ict(ℓ_2 **0**)

$\vdots$

ict(ℓ_m **0**)

AKGS $\equiv$

isk(1 $\boldsymbol{x}$)

H

ict(ℓ_1 0)

ict(ℓ_2 0)

$\vdots$

ict(ℓ_m 0)

$\boldsymbol{L}_1, \dots, \boldsymbol{L}_m$ 是标签函数的系数
（乱码化 $\underbrace{\mu_b}_{\alpha} f(\boldsymbol{x}) + \underbrace{0}_{\beta}$ 所得）

$\ell_j = L_j(\boldsymbol{x}) = \boldsymbol{L}_j^\top \overline{\boldsymbol{x}}$

$\{\ell\} \xleftarrow{\$} \mathcal{S}\left(f, \boldsymbol{x}, \mu_b f(\boldsymbol{x})\right)$

$= 0$

约束. $f(\boldsymbol{x}) = 0$

目标. 证明 $G_0 \approx H \approx G_1$ 且 H 不含 μ_0, μ_1 的信息

完美大结局，谢谢！

你诈和了！

ABE 安全证明：问题所在

$$\text{isk}(\ 1\quad \boldsymbol{x}\ ) \qquad \text{isk}(\ 1\quad \boldsymbol{x}\ ) \qquad \text{isk}(\ 1\quad \boldsymbol{x}\ )$$

$$G_b \overset{\text{IPFE}}{\approx} \qquad \overset{\text{AKGS}}{\equiv} H$$

$$\begin{array}{lll} \text{ict}(\ \boldsymbol{L}_1\ ) & \text{ict}(\ \ell_1\quad \mathbf{0}\ ) & \text{ict}(\ \ell_1\quad 0\ ) \\ \text{ict}(\ \boldsymbol{L}_2\ ) & \text{ict}(\ \ell_2\quad \mathbf{0}\ ) & \text{ict}(\ \ell_2\quad 0\ ) \\ \vdots & \vdots & \vdots \\ \text{ict}(\ \boldsymbol{L}_m\ ) & \text{ict}(\ \ell_m\quad \mathbf{0}\ ) & \text{ict}(\ \ell_m\quad 0\ ) \end{array}$$

$$\ell_j = L_j(\boldsymbol{x}) = \boldsymbol{L}_j^{\top}\overline{\boldsymbol{x}} \qquad \{\ell\} \overset{\$}{\leftarrow} \mathcal{S}\left(f, \boldsymbol{x}, \mu_b f(\boldsymbol{x})\right)$$

隐含假设. 先 $\boldsymbol{x}$ 后 f

ABE 安全证明：先 f 后 $\boldsymbol{x}$

$$\text{ict}(\ \boldsymbol{L}_1\ ) \qquad \text{ict}(\ \ell_1\ \ \mathbf{0}\ )$$
$$\text{ict}(\ \boldsymbol{L}_2\ ) \qquad \text{ict}(\ \ell_2\ \ \mathbf{0}\ )$$
$$\vdots \qquad\qquad \vdots$$
$$\text{ict}(\ \boldsymbol{L}_m\ ) \qquad \text{ict}(\ \ell_m\ \ \mathbf{0}\ )$$

不可能在知道 x 之前
hardwire
把 $\{\ell\}$ 写死

G_b

$$\text{isk}(\ 1\quad \boldsymbol{x}\ ) \qquad \text{isk}(\ 1\quad \boldsymbol{x}\ )$$

线性方程

$\{\ell\}$ 必须满足求值约束

$$\text{Eval}(\{\ell\}, f, \boldsymbol{x}) = \mu_b f(\boldsymbol{x})^{=0}$$

$\{\ell\}$ 的系数是 x 的 (高次) 函数

ABE 安全证明：先 f 后 $\boldsymbol{x}$，坏思路

$$\text{ict}\begin{pmatrix} \boldsymbol{L}_1 & \mathbf{0} \\ \boldsymbol{L}_2 & \mathbf{0} \\ \vdots & \\ \boldsymbol{L}_m & \mathbf{0} \end{pmatrix} \qquad \text{ict}\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

G_b

$$\text{isk}\begin{pmatrix} 1 & \boldsymbol{x} & \mathbf{0} \end{pmatrix} \qquad \text{isk}\begin{pmatrix} \ell_1 & \ell_2 & \cdots & \ell_m \end{pmatrix}$$

programming space
“编程空间”

好处. 证明走通了
坏处. 公式规模 m 被 IPFE 维数限制
思路. 大多数 $\{\ell\}$ 写死到 ict
少部分 $\{\ell\}$ 写死在 isk

算术公式的 AKGS：有趣性质

$\{\ell\}$ 在满足求值约束下均匀随机

并且求值约束方程里 ℓ_1 的系数永远是 1（和 $\boldsymbol{x}$ 无关）

- 设置 $\ell_2, \ldots, \ell_m$ 为均匀随机
  - ✓ 和 $\boldsymbol{x}$ 无关，可以写死到 ict 里
- 已知 $f, \boldsymbol{x}$ 及 $\alpha f(\boldsymbol{x}) + \beta$ 时可**反解** ℓ_1
  - ✓ 只有一个，写死在 isk 里不会过度增加 IPFE 维数

读者习题. 证明这些性质

递归性质. 这些性质对公式树里的**每棵子树**都成立

ℓ_1 对应**最左叶子**，其系数（相对于**该子树**的求值约束）是 1

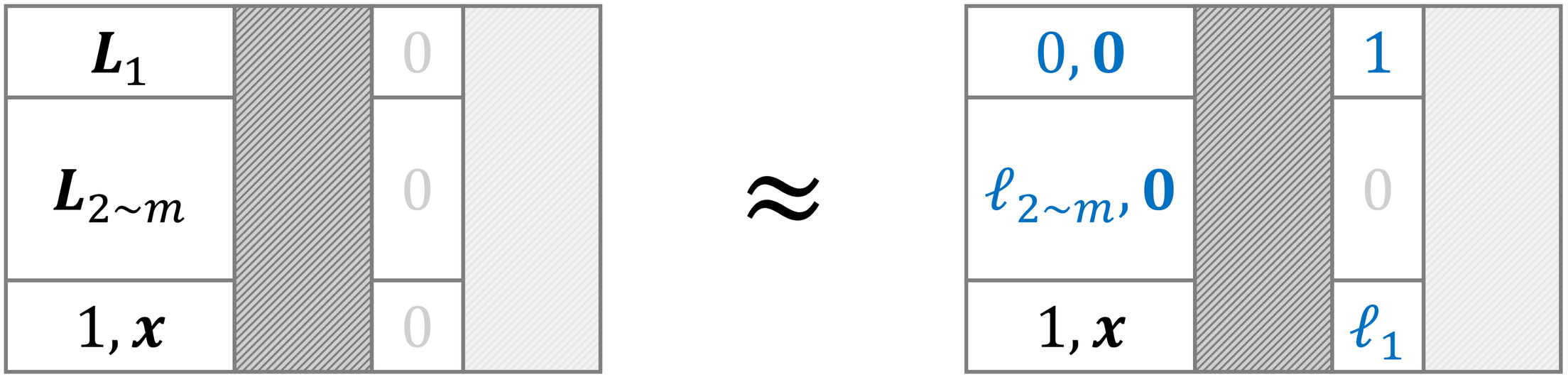
ABE 安全证明：先 f 后 $\boldsymbol{x}$，新思路

$$
\begin{array}{ccc}
\mathrm{ict}(\ \boldsymbol{L}_1\ \ 0\ ) & & \mathrm{ict}(\ 0\ \ 0\ \ 1\ ) \\
\mathrm{ict}(\ \boldsymbol{L}_2\ \ 0\ ) & & \mathrm{ict}(\ \ell_2\ \ 0\ \ 0\ ) \\
\vdots & & \vdots \\
\mathrm{ict}(\ \boldsymbol{L}_m\ \ 0\ ) & ? & \mathrm{ict}(\ \ell_m\ \ 0\ \ 0\ ) \\
G_b & \approx & H \\
\mathrm{isk}(\ 1\ \ \boldsymbol{x}\ \ 0\ ) & & \mathrm{isk}(\ 1\ \ \boldsymbol{x}\ \ \ell_1\ )
\end{array}
$$

直觉. 左右两边的内积“同分布”(模拟算法保证)

形式化. 内积相等才能用 IPFE 安全性
同分布是否仍然安全，不明确

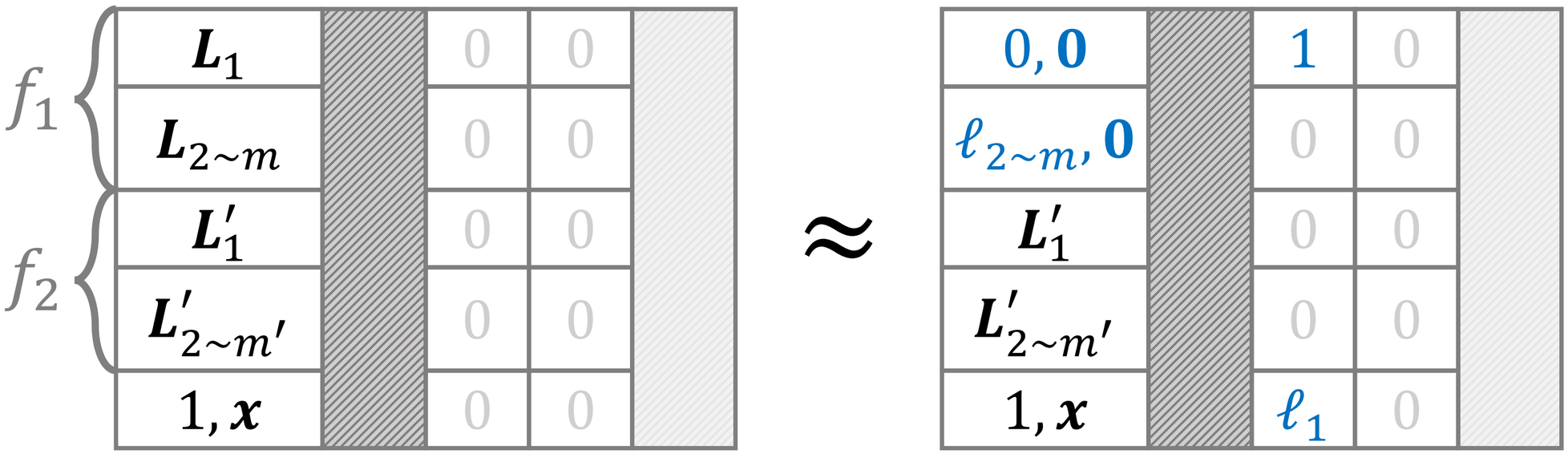
ABE 安全证明：先 f 后 $\boldsymbol{x}$，递归策略



递归出口. f 规模是 1

ABE 安全证明：先 f 后 $\boldsymbol{x}$，递归策略

递归 (减号). $\alpha(f_1 - f_2) + \beta = (\alpha f_1 + \beta + r) - (\alpha f_2 + r)$

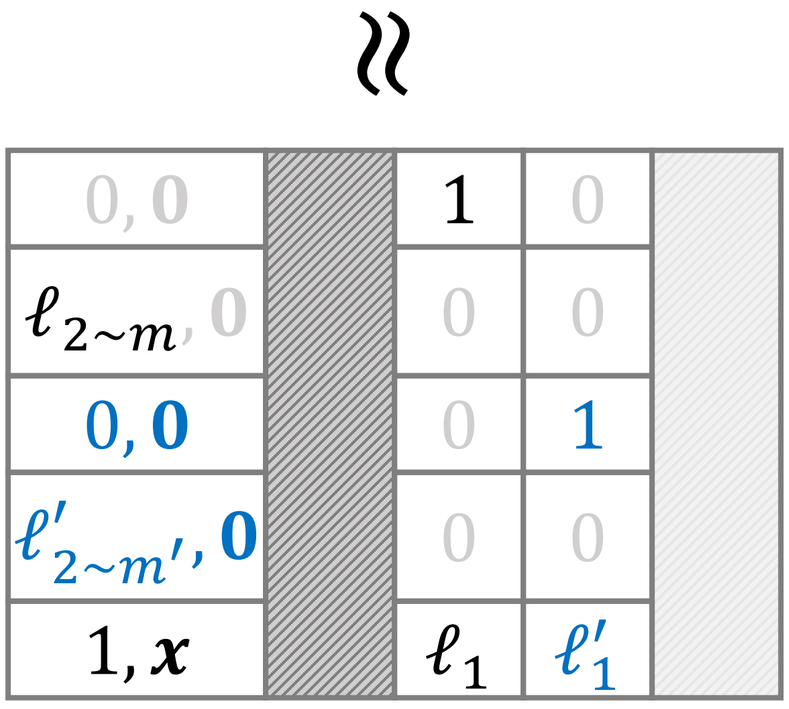


$$\ell_1 = (\alpha f_1 + \beta + r) - \cdots$$
$$\ell'_1 = (\alpha f_2 + r) - \cdots$$

(换元 $r = \ell'_1 - \alpha f_2 + \cdots$**)**

$$\ell_1 = (\alpha f_1 + \beta + \ell'_1 - \alpha f_2) - \cdots$$
$$= (\alpha(f_1 - f_2) + \beta) + \ell'_1 - \cdots$$

$\ell'_1 \xleftarrow{\$} \mathbb{F}$



ABE 安全证明：先 f 后 $\boldsymbol{x}$，递归策略

递归 (减号). $\alpha(f_1 - f_2) + \beta = (\alpha f_1 + \beta + r) - (\alpha f_2 + r)$

f_1

$\boldsymbol{L}_1$		0	0
$\boldsymbol{L}_{2\sim m}$		0	0

f_2

$\boldsymbol{L}'_1$		0	0
$\boldsymbol{L}'_{2\sim m'}$		0	0
$1, \boldsymbol{x}$		0	0

$\approx$

$0, \mathbf{0}$		1	0
$\ell_{2\sim m}, \mathbf{0}$		0	0
$\boldsymbol{L}'_1$		0	0
$\boldsymbol{L}'_{2\sim m'}$		0	0
$1, \boldsymbol{x}$		ℓ_1	0

$\approx$

$0, \mathbf{0}$		1	0
$\ell_{2\sim m}, \mathbf{0}$		0	0
$\ell'_1, \mathbf{0}$		0	0
$\ell'_{2\sim m'}, \mathbf{0}$		0	0
$1, \boldsymbol{x}$		ℓ_1	0

$\approx$

$0, \mathbf{0}$		1	0
$\ell_{2\sim m}, \mathbf{0}$		0	0
$0, \mathbf{0}$		0	1
$\ell'_{2\sim m'}, \mathbf{0}$		0	0
$1, \boldsymbol{x}$		ℓ_1	ℓ'_1

需要多少编程空间?

可以归纳证明编程空间只需要 $O(d)$
d 是公式深度

情况很糟糕. d 最多可以是 $\Omega(m)$

定理 (算术公式平衡化 [Bre74]).
可以在多项式时间内
把规模是 m 的算术公式
改写为深度是 $O(\log m)$ 的等价公式

设置 $\max m = 2^{128}$ 即可处理所有（有意义的）情况

故事时间

平衡树不一定最佳

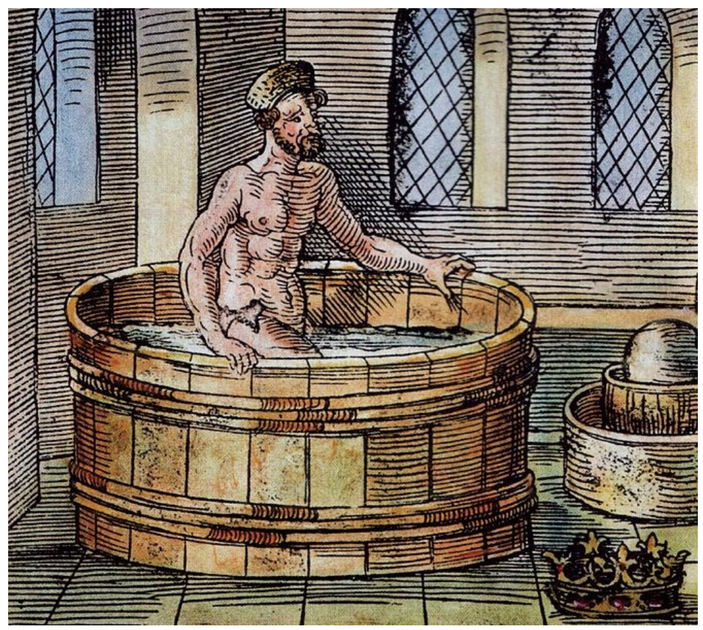
全是乘号或者全是减号：左偏链需要的空间是 $O(1)$

问题 1. 是否应该采用**交替公式**表示?

问题 2. 树形递归是**必要之繁**，还是**技巧匮乏**?

加深体会 + 验证：写程序打印证明过程

后来还新写了一个动态演示网页



AKGS 的分段安全性 (反解 + 条件随机性)

一般地，算术公式的 AKGS 中，
对任意 $1 < j \le m$，
标签函数 L_j 常数项具有某 r，
且该 r 在 $L_{j+1}, \dots, L_m$ 中不出现。

L_j 对应仿射形式 f_j
f_j 是子树 T 的右子树的最左叶子
递归到 T 时产生的 r 对应 L_j

$\Rightarrow$

标签的条件随机性. 对任意 $1 < j \le m$ 和 $\boldsymbol{x}$，要求

$$\left(L_j(\boldsymbol{x}), \boldsymbol{L_{j+1}}, \dots, \boldsymbol{L_m}\right) \equiv \left(\$, \boldsymbol{L_{j+1}}, \dots, \boldsymbol{L_m}\right)$$

$$\ell_1 = L_1(\boldsymbol{x}) = r_2 x_1 - r_2 x_3 + (2r_2 + \beta + r_1)$$
$$\ell_2 = L_2(\boldsymbol{x}) = -\alpha x_2 - 5\alpha x_4 + r_2$$
$$\ell_3 = L_3(\boldsymbol{x}) = -\alpha x_1 + r_1$$

$$\alpha f + \beta = \ell_1 - \ell_2 f_1 - \ell_3$$

分段安全性的后续

用分段安全性做证明，编程空间的维数是 2

1. 把 ℓ_1 挪入 isk，并改为反解 (反解性质)
2. 依次分别把 ℓ_j $(j = 2, \ldots, m)$
   挪入 isk
   替换为随机数 (条件随机性)
   再挪回 ict

扩大适用范围

算术分支程序 (arithmetic branching program)

NFA、NL 图灵机

有趣的刻画

f 分段安全 AKGS 的标签个数 $\approx$ 能表达 f 的最小 ABP 的规模

ia.cr/2020/318

ia.cr/2020/1139