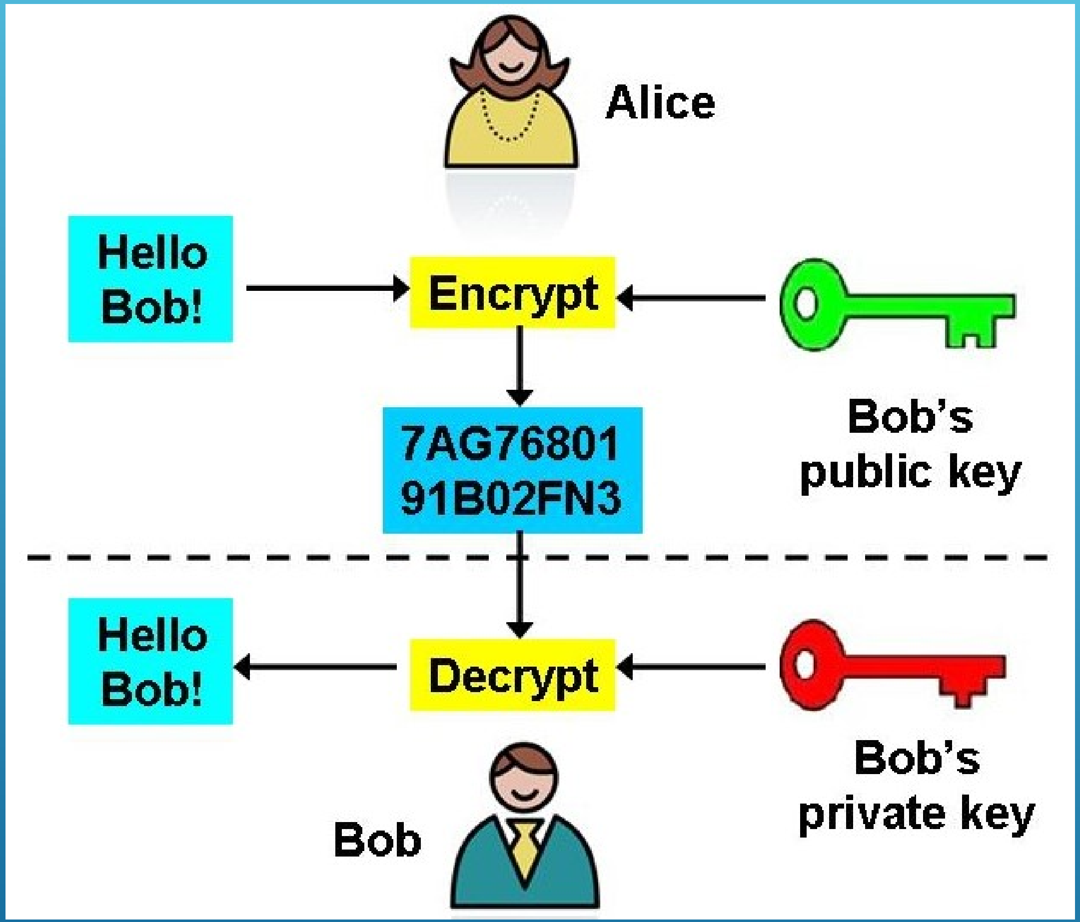
Access Control Encryption: Enforcing Information Flow with Cryptography

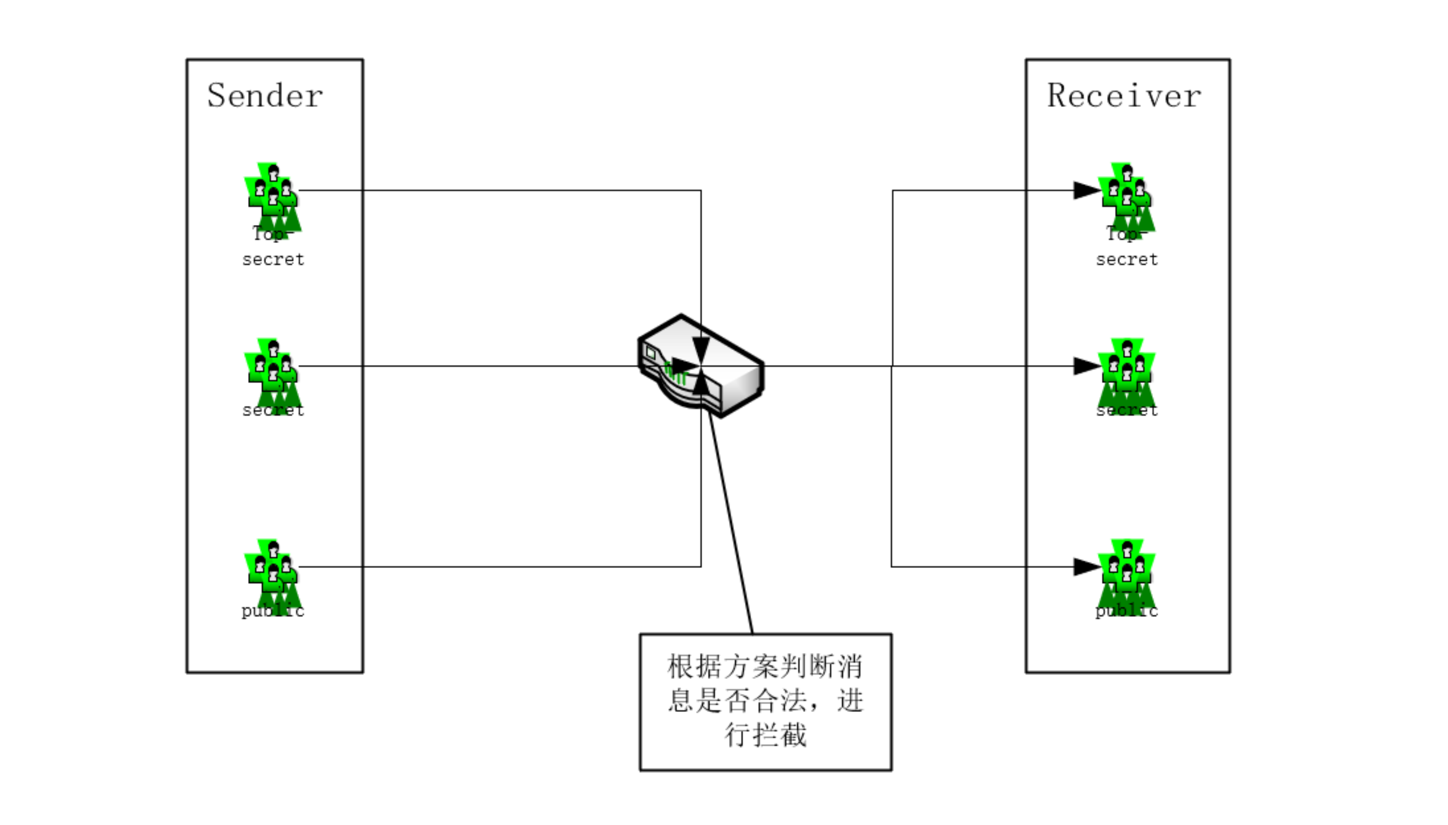
背景回顾：



传统公钥密码：两者之间靠公私钥进行加密通信

限制用户之间的通信与否？
不同权限的用户能否发送信息？能否接收信息？

回顾：



基于Bell-Lapadula模型

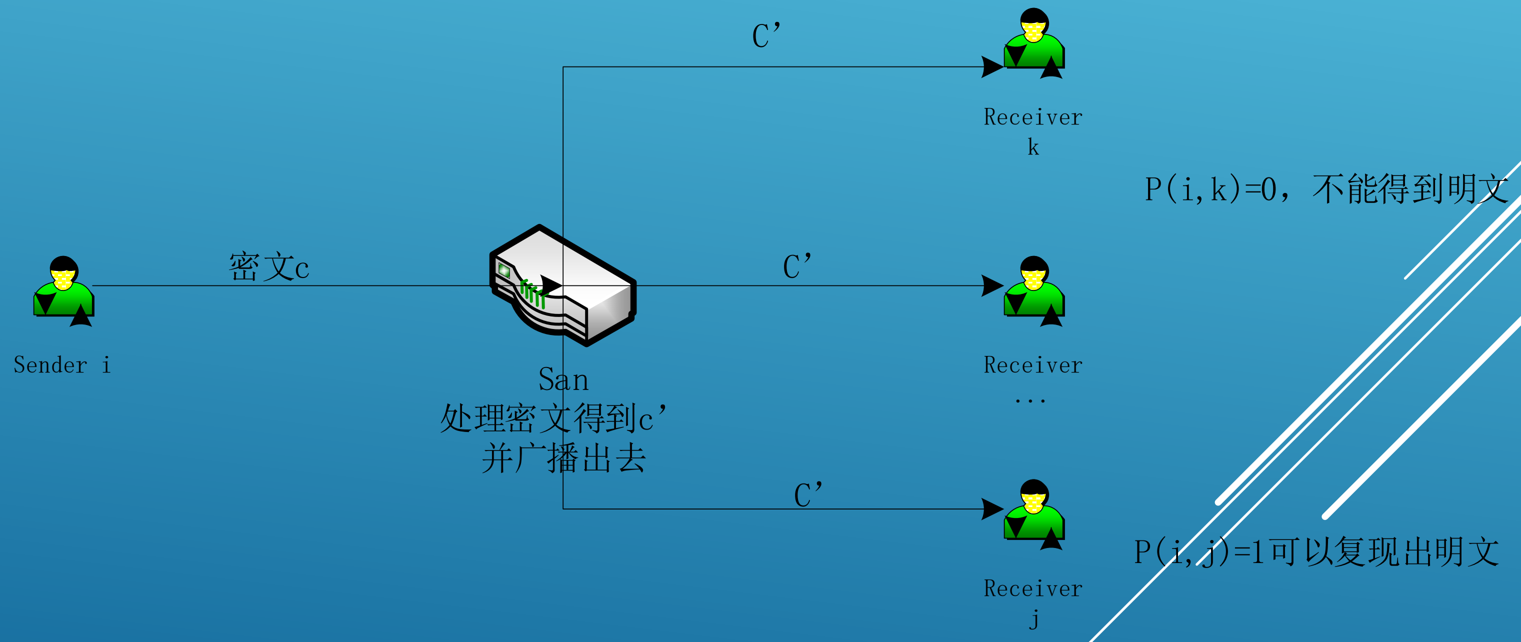
控制信息流：引入中间的消毒者 “Sanitizer”

为了防止San叛变且可以将San外包，因此减少San的权限，将判断通信与否交给接收者处理，San只对密文C进行消毒处理。

由于San的权限被削弱到只知道数据的长度和时间而不知道发送者和接收者的身份，因此消息是广播出去的

防止发送者直接丧心病狂的发送明文从而泄露信息，San的功能不能够只是转发信息，需要对信息进行处理使得处理后的信息无法区分

实现目标：用户之间的通信符合方案中的关系矩阵P的要求，当发送者s_i与接收者R_j满足$P(\mathrm{i}, j)$=0时二者不允许通信，反之可以通信。



方案的安全性与准确性要求：

(1)$Correctness$：发送者发送出去的消息，所以要满足$P(i,j)=1$ 的接收者都能够复现出明文。

(2)$No-Read\ Rule$：满足$P(i,j)=0$ 的所有用户都不能够得到关于明文的任何信息。

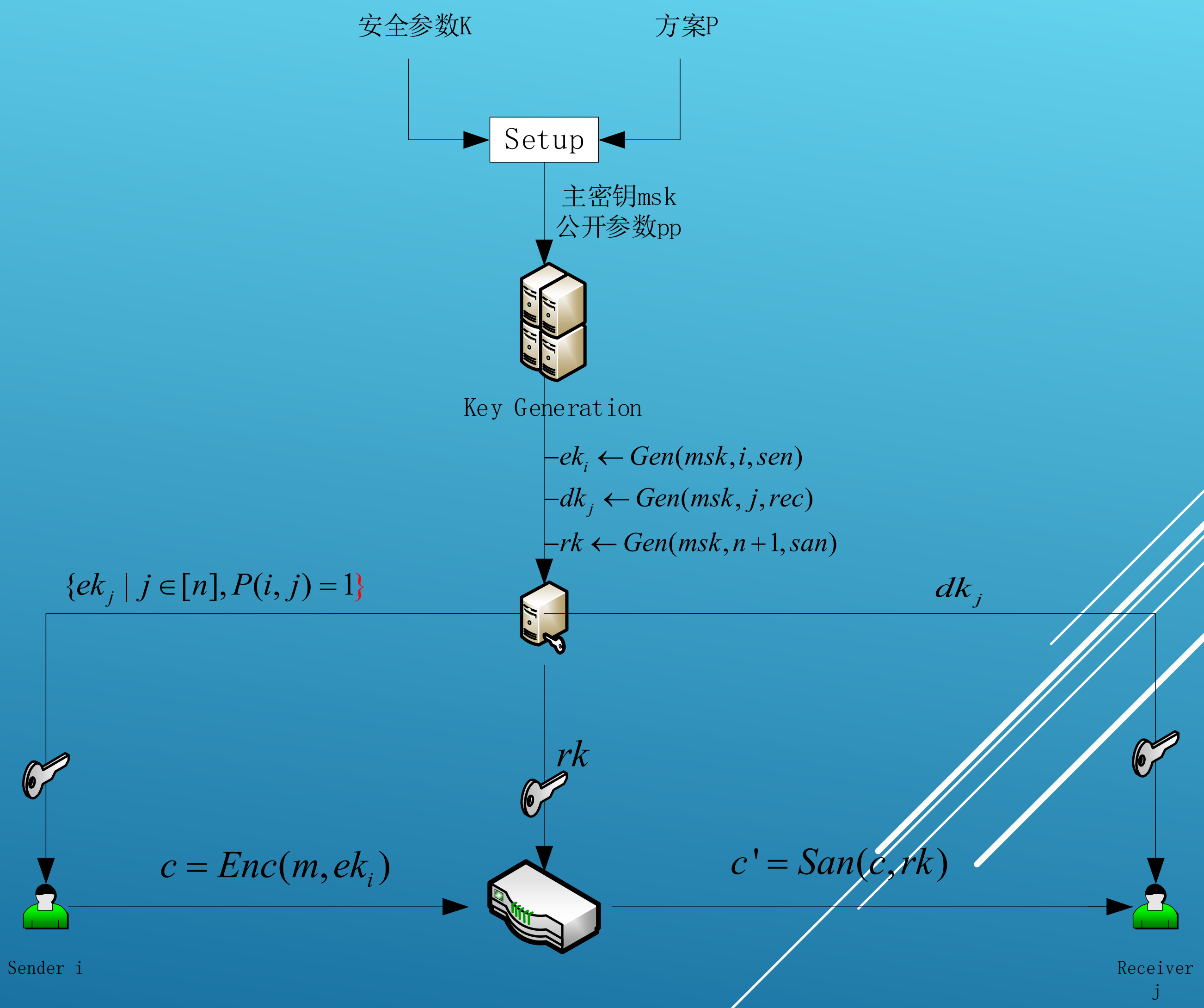
> (2b) 中间者San 不能知道发送的消息明文和发送者的身份以防止中间者叛变。

(3)$No-Write\ Rule$：如果$P(i,j)=0$ ，那么任何发送者S_i 都不应该与接收者R_j 交流任何信息。

No-Read针对接收方的角度，判断能否通信后：如果不能通信就无法得到关于明文的信息

No-Write从发送者的角度：如果P(i,j)=0，则不能够发送有效信息。问题在后续实现中可以转化为是否拥有加密密钥，因为经过了San的消毒处理，没有加密密钥的信息会转变为“随机值”从而使得接收者无法解密。

基础的ACE实现框架：



如果$P(i,j)=1$,则$m'=m$;反之m'为"随机"值。$m'=Dec(c', dk_j)$

对安全性要求的三条准则进行形式化的描述：

Definition 1 (Correctness). *For all $m \in \mathcal{M}$, $i, j \in [n]$ such that $P(i,j) = 1$:*

$$\Pr\left[\mathsf{Dec}\left(dk_j, \mathsf{San}\left(rk, \mathsf{Enc}\left(ek_i, m\right)\right)\right) \neq m\right] \leq \mathsf{negl}\left(\kappa\right)$$

with $(pp, msk) \leftarrow \mathsf{Setup}(1^\kappa, P)$, $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathsf{sen})$, $dk_j \leftarrow \mathsf{Gen}(msk, j, \mathsf{rec})$, *and* $rk \leftarrow \mathsf{Gen}(msk, n+1, \mathsf{san})$, *and the probabilities are taken over the random coins of all algorithms.*

即对于所有满足$P(i,j) = 1$的通信，解密出来的m'不等于m的概率是可以忽略的，保证方案的正确性

No-Read Rule：

Definition 2 (No-Read Rule). *Consider the following game between a challenger C and a stateful adversary A:*

No-Read Rule	
Game Definition	**Oracle Definition**
1. $(pp, msk) \leftarrow \mathsf{Setup}(1^\kappa, P)$; 2. $(m_0, m_1, i_0, i_1) \leftarrow A^{\mathcal{O}_G(\cdot),\mathcal{O}_E(\cdot)}(pp)$; 3. $b \leftarrow \{0,1\}$; 4. $c \leftarrow \mathsf{Enc}(\mathsf{Gen}(msk, i_b, \mathsf{sen}), m_b)$; 5. $b' \leftarrow A^{\mathcal{O}_G(\cdot),\mathcal{O}_E(\cdot)}(c)$;	$\mathcal{O}_G(j,t)$: 1. Output $k \leftarrow \mathsf{Gen}(msk, j, t)$; $\mathcal{O}_E(i,m)$: 1. $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathsf{sen})$; 2. Output $c \leftarrow \mathsf{Enc}(ek_i, m)$;

We say that A wins the No-Read game *if $b = b'$, $|m_0| = |m_1|$, $i_0, i_1 \in \{0, \ldots, n\}$ and one of the following holds:*

Payload Privacy: *For all queries q to $\mathcal{O}_G$ with $q = (j, \mathsf{rec})$ it holds that*

$$P(i_0, j) = P(i_1, j) = 0$$

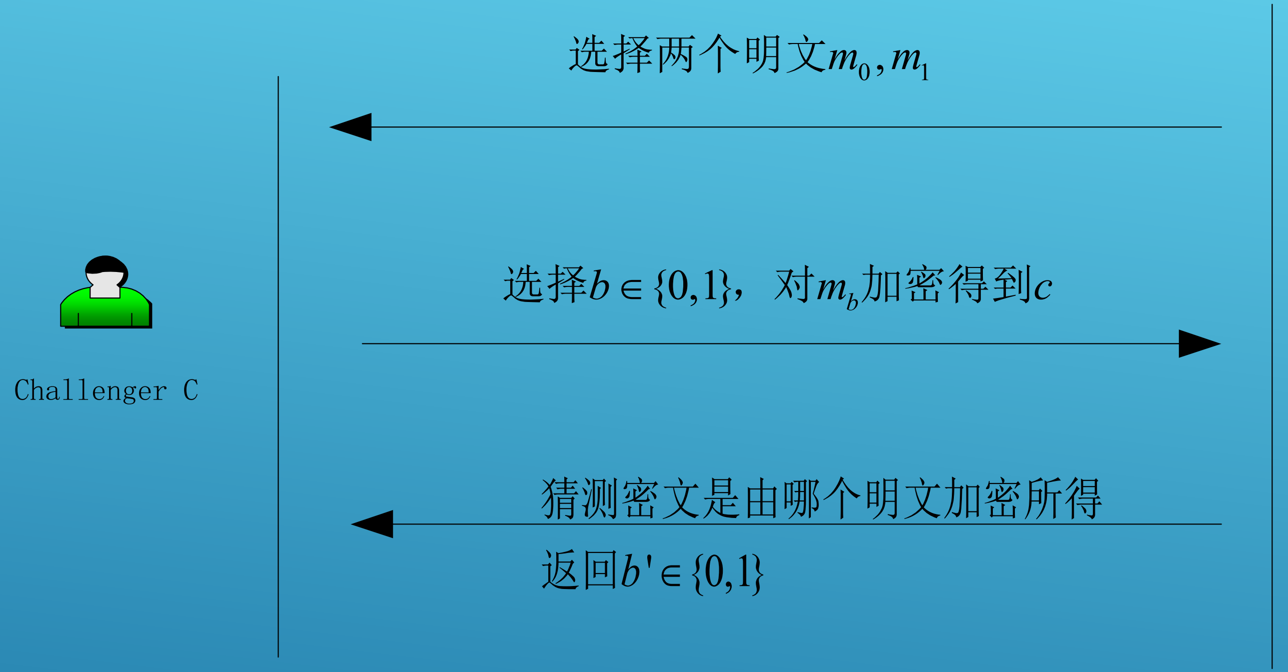
只有满足方案的接收者
才能了解信息

Sender Anonymity: *For all queries q to $\mathcal{O}_G$ with $q = (j, \mathsf{rec})$ it holds that*

$$P(i_0, j) = P(i_1, j) \text{ and } m_0 = m_1$$

无法获得发送者的身份：明文相同且都能够通信，
解密出来的明文相同

No-Read Rule：



如果$b = b', |m_0| = |m_1|$,并且以下之一满足称为A赢得$game$:

(1)$Payload\ Privacy: A$与发送者S_{i_0}和S_{i_1}都无法通信，即A无法对密文进行解密

(2)$Sender\ \ Anonymity$:明文相同且与两个身份的通信与否相同

称ACE满足No-Read $Rule$对于所有的PPT $A: adv^A = 2 \times |\Pr[A\ wins\ the\ No-Read\ game] - \frac{1}{2}| \le negl(k)$

如果两条都满足：明文相同且都无法解密，游戏没有意义。如果两条都不满足，即两个身份中有一个可以通信就可以解密出某一次密文，进而可以得到明文。

No-Write Rule：

Challenger C

Adversary A

$$(pp, msk) \leftarrow Setup(1^k, P)$$

$$(c, i') \leftarrow A^{O_E(\cdot),O_S(\cdot)}(pp)$$

获得身份i'的加密密钥$ek_{i'}$和消毒密钥$rk = Gen(msk, n+1, san)$

随机选择$b \in \{0,1\}$

$-b = 0$,则$c' \leftarrow San(rk, Enc(ek_{i'}, M))$

$-b = 1$,则$c' \leftarrow San(rk, c)$

再将c'发送给A

猜测$b' \leftarrow A^{O_E(?),O_R(?)}(c')$

敌手A可以询问身份对应的加密密钥，可以询问解密密钥但是身份j满足P（i，j）=0

使得无法区分真实信息加密与随机信息加密的结果：结果是发送者与接受者如果在不合法的前提下通信，发送者发送的消息的消毒版本与随机值的消毒版本无法区分，在没有解密密钥时无法获得关于明文的信息

即

any set of (corrupt) senders $\{S_i\}_{i \in I}$ *cannot transfer any information to any set of (corrupt) receivers* $\{R_j\}_{\{j \in J\}}$ *unless at least one of the senders in* I *is allowed communication to at least one of the receivers in* J。

No-Write Rule：

Definition 3 (No-Write Rule). *Consider the following game between a challenger C and a stateful adversary A:*

No-Write Rule	
Game Definition	**Oracle Definition**
1. $(pp, msk) \leftarrow \mathsf{Setup}(1^\kappa, P)$; 2. $(c, i') \leftarrow A^{\mathcal{O}_E(\cdot),\mathcal{O}_S(\cdot)}(pp)$; 3. $ek_{i'} \leftarrow \mathsf{Gen}(msk, i', \mathsf{sen})$; 4. $rk \leftarrow \mathsf{Gen}(msk, n+1, \mathsf{san})$; 5. $r \leftarrow \mathcal{M}$; 6. $b \leftarrow \{0,1\}$, – If $b=0$, $c' \leftarrow \mathsf{San}(rk, \mathsf{Enc}(ek_{i'}, r))$; – If $b=1$, $c' \leftarrow \mathsf{San}(rk, c)$; 7. $b' \leftarrow A^{\mathcal{O}_E(\cdot),\mathcal{O}_R(\cdot)}(c')$;	$\mathcal{O}_S(j,t)$: 1. Output $k \leftarrow \mathsf{Gen}(msk, j, t)$; $\mathcal{O}_R(j,t)$: 1. Output $k \leftarrow \mathsf{Gen}(msk, j, t)$; $\mathcal{O}_E(i,m)$: 1. $ek_i \leftarrow \mathsf{Gen}(msk, i, \mathsf{sen})$; 2. $c \leftarrow \mathsf{Enc}(ek_i, m)$; 3. Output $c' \leftarrow \mathsf{San}(rk, c)$;

Let Q_S (resp. Q) be the set of all queries $q=(j,t)$ that A issues to $\mathcal{O}_S$ (resp. both $\mathcal{O}_S$ and $\mathcal{O}_R$). Let I_S be the set of all $i \in [n]$ such that $(i, \mathsf{sen}) \in Q_S$ and let J be the set of all $j \in [n]$ such that $(j, \mathsf{rec}) \in Q$. Then we say that A wins the No-Write game *if $b' = b$ and all of the following hold:*

1. $(n+1, \mathsf{san}) \notin Q$;
2. $i' \in I_S \cup \{0\}$;
3. $\forall i \in I_S, j \in J,\ P(i,j) = 0$;

We say an ACE scheme satisfies the No-Write rule *if for all PPT A*

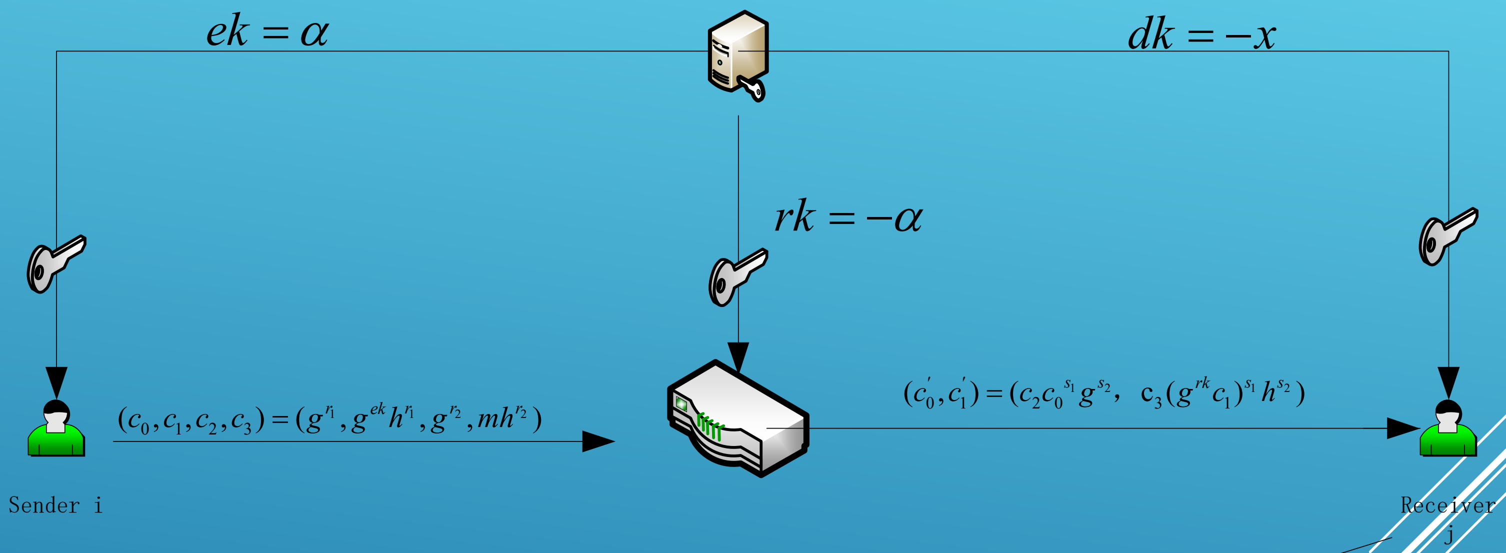
$$\mathsf{adv}^A = 2 \cdot \left| \Pr[A \text{ wins the No-Write game}] - \frac{1}{2} \right| \le \mathsf{negl}(\kappa)$$

(1)San不能作为A请求的一员，因为A可以通过O_S来获得$rk = Gen(msk, n+1, san)$

(2)0也是身份i'的一种，因为$ek_0 = pp$,也可以产生类似于"随机"的密文

(3)$P(i,j)=0$；否则A对得到的密文进行解密便可以判断出被消毒的是否是自己传过去的密文

线性ACE（基于ElGamal）:简略



$h=g^x;s_1$和s_2为Z_q的随机值 $\quad m'=c_1'(c_0')^{dk}$

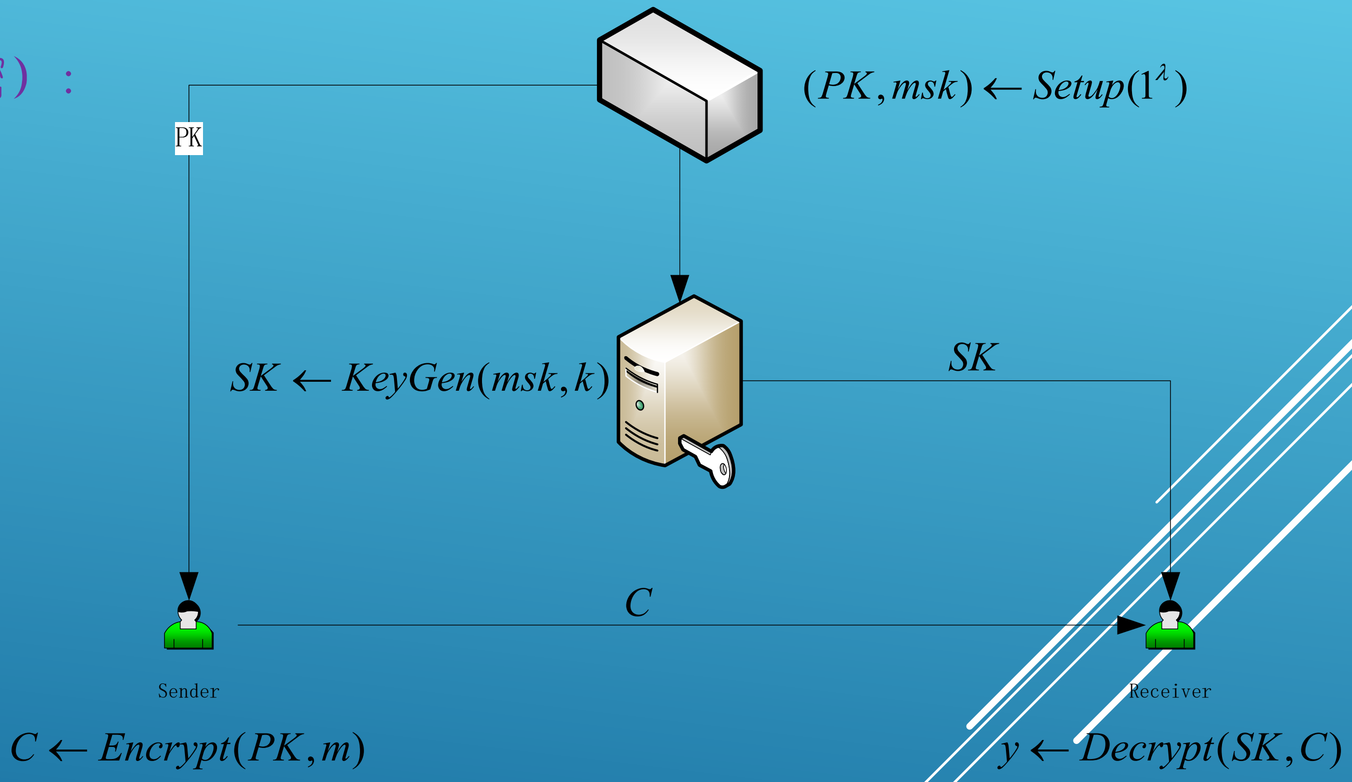
用户数量为n的ACE是1ACE的n个副本运行：加密密钥是能够通信的对方的ek的集合，密文是明文的n份加密，能通信的用对方的ek来加密，否则用对方的pp来加密。

安全性依赖于DDH假设：区分元组(g, g^a, g^b, g^{ab})与(g, g^a, g^b, g^z)是困难的

Polylogarithmic ACE：多重对数复杂性

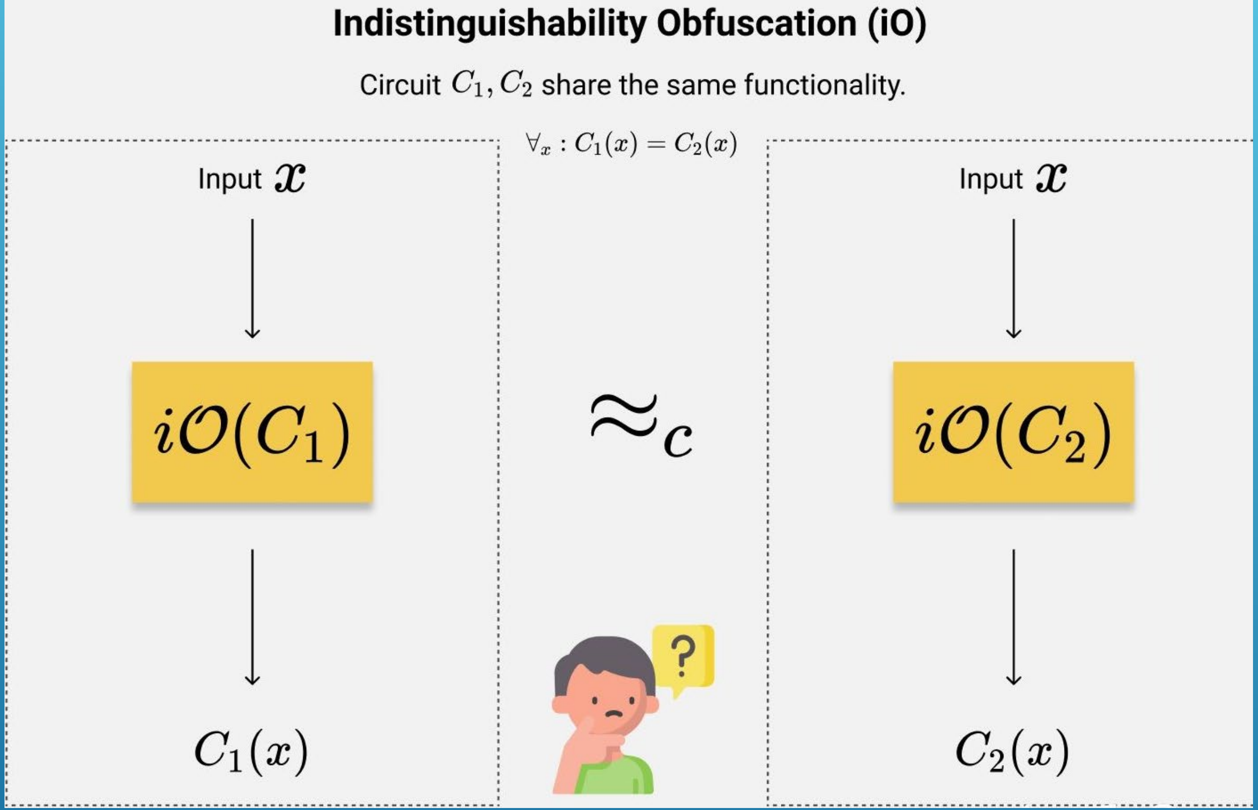
首先介绍FE（Functional Encryption）、IO以及NIZK，然后引出sFE构造，再是ACE的构造

泛函加密FE（函数加密）：



泛函加密的好处在于当你拥有函数f_k的密钥时只能够获得$f_k(m)$ 的值而不能够获得m 的值（f(x)=x除外）。传统的公钥加密使得你看到的要么是全部明文要么便是非明文的随机值。泛函加密可以让你获得部分明文。

IO（不可区分性混淆）：



$i\mathcal{O}$ 是一个PPT Algorithm（概率多项式时间的算法），通俗的说就是一个比较高效率的程序。这个程序的输入是一个待混淆的程序/电路 C，而输出则是这个输入程序的不可区分混淆 $i\mathcal{O}(C)$。

$i\mathcal{O}$ 主要需要满足三大属性：

1. 首先，我们需要满足Functionality Preserving（保持功能性），即被混淆的程序需要和混淆之前一样，保持完全一样的功能性。

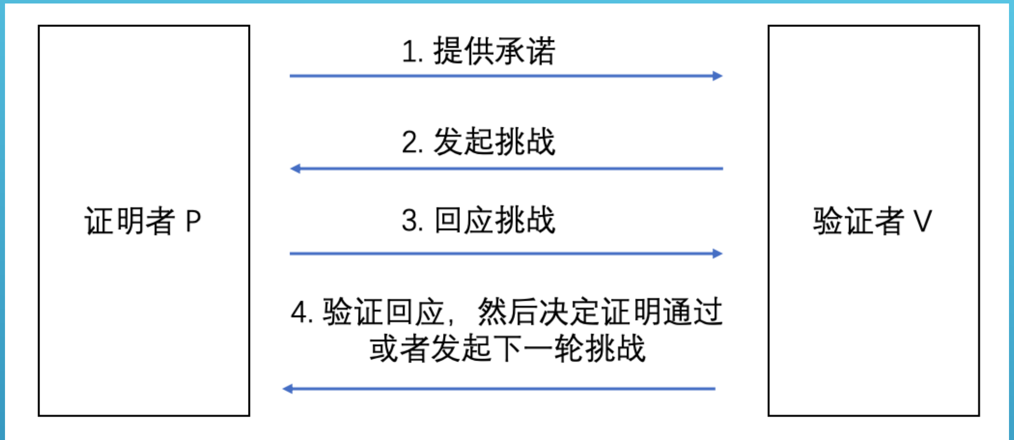
$$\forall_x : C(x) = i\mathcal{O}(C)(x)$$

1. 其次就是Indistinguishability，即不可区分性啦。对于两个功能性完全一样的程序 C_1, C_2，即 $\forall_x : C_1(x) = C_2(x)$，就算这两个程序的实现方法完全不同，我们也无法有效的分辨他们的混淆结果。换句话来说的话，如果有两个相同功能的程序，而我们看到了只看到了一个未知的 $i\mathcal{O}$ 结果的话，我们无法分辨到底是哪个程序被混淆了。

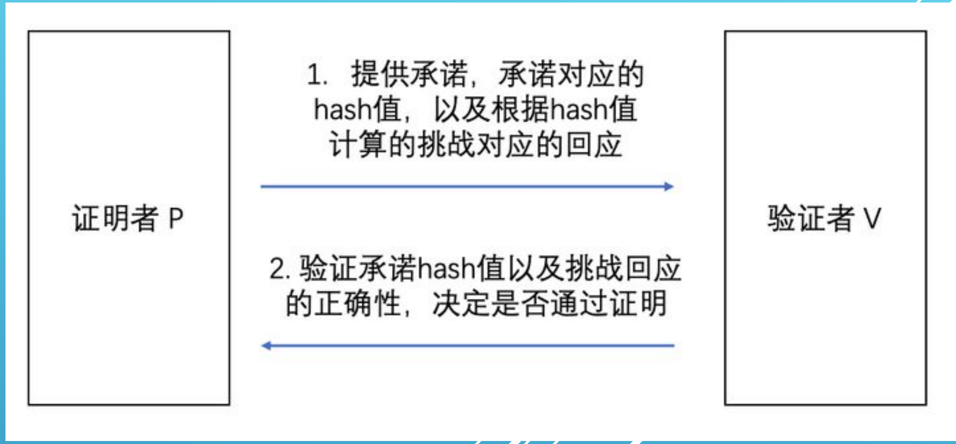
$$|Pr[A(i\mathcal{O}(C_1)) = 1] - Pr[A(i\mathcal{O}(C_2)) = 1]| = negl$$

1. 最后一点是Efficiency，即高效率。这一条属性指出了，$i\mathcal{O}(C)$ 的输出一定不能太大，最多是原本程序大小的多项式倍数，即 $|i\mathcal{O}(C)| \leq poly(|C|)$。为什么需要这一条约束呢？这是因为如果没有这一条效率的约束，整个混淆操作就失去了实际使用的意义。我们甚至可以构造出非常简单的 $i\mathcal{O}$ 结构：假如一个程序 C 的输入是 n 位 $\{0,1\}^n$ 而输出是 m 位 $\{0,1\}^m$，那我们只需要输出一个 2^{n+m} 大小的表格，把所有 C 输入值和输出值一一对应。这样一个巨大的表格也是不可区分混淆了，因为只要功能性不变，那么输入和输出的一一对应也不会改变！为了排除这一类无意义的 $i\mathcal{O}$ 构造，我们需要规定一个理想的 $i\mathcal{O}$ 算法输出的程序大小相比起输入的程序大小来说不会变得过大。

NIZK（非交互式零知识证明）： 在不泄露任何信息的前提下证明命题的正确性



交互式零知识证明



非交互式零知识证明

首先参与方必须同时在线，并且只有验证者才会相信证明的正确性，
因为只有验证者才知道提出的挑战是随机的，还是和证明者串通作弊的。

相比减少了重复的挑战和回应过程，证明者只需要发送一次数据供验证者验证。

1. **如何保证一次证明被多个验证者验证？去掉交互过程，证明者直接公开验证所需的数据。**
2. **所有验证者如何相信挑战包含随机数的随机性？该随机性由hash函数的随机性保证。**
3. **证明者能否先知道挑战再更改承诺？计算上不可行，hash函数是单向的，也就是说用户需要通过承诺计算挑战，知道挑战后更改承诺会导致hash结果改变，挑战需要重新生成，足够多次计算才能得到用户希望的hash结果，这普遍认为是计算不可行的。**

本篇论文中用到的基于[GGH+13]中的NIZK的构造：

The content of this subsection is taken almost verbatim from [GGH$^+$13]. Let L be a language and R a relation such that $x \in L$ if and only if there exists a witness w such that $(x, w) \in R$. A non-interactive proof system [BFM88] for a relation R is defined by the following PPT algorithms

Setup: The Setup algorithm takes as input the security parameter κ and outputs common reference string crs.

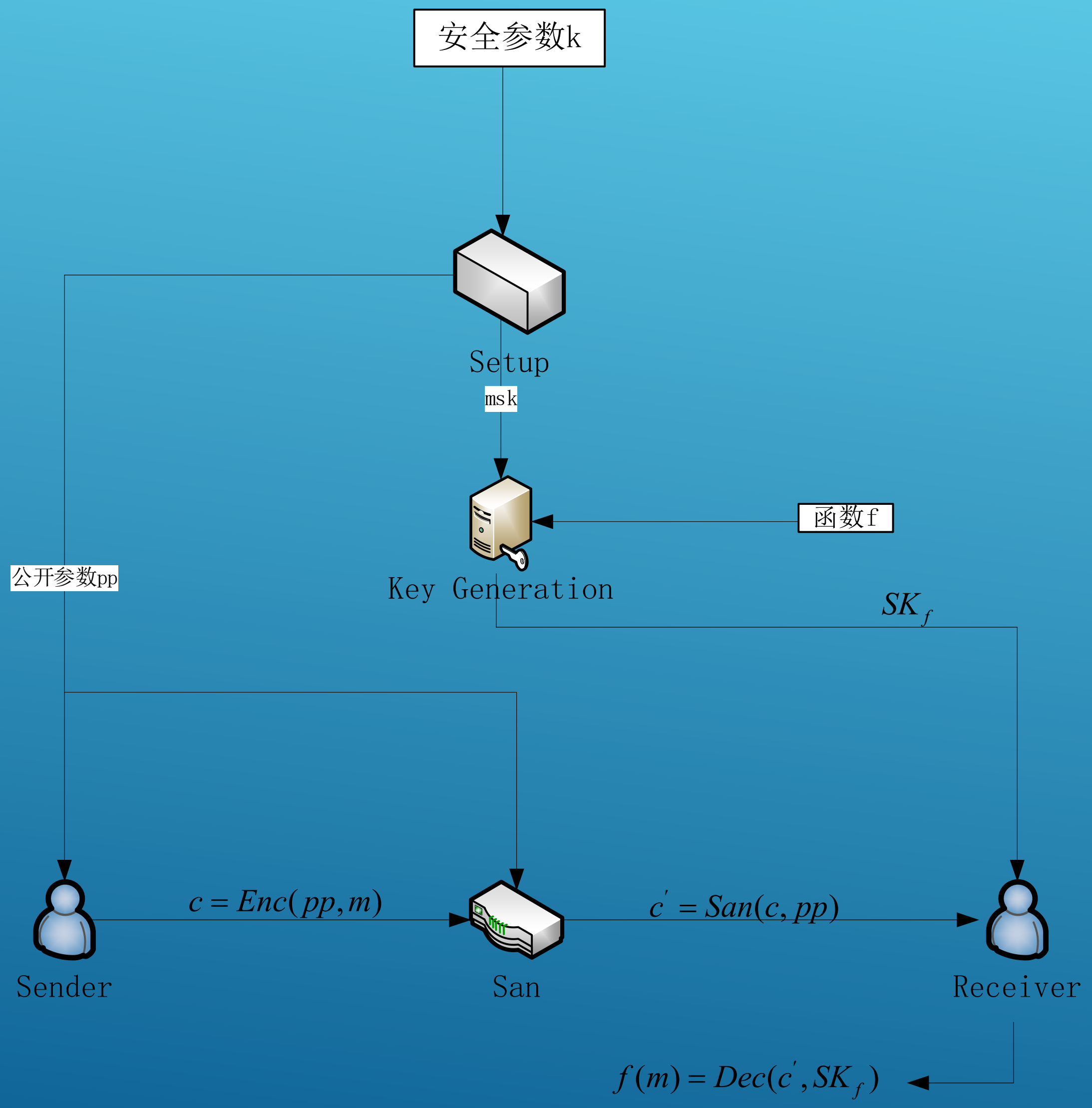
Prove: The Prove algorithm takes as input the common reference string crs, a statement x, and a witness w, and outputs a proof π.

Verify: The Verify algorithm takes as input the common reference string crs, a statement x, and a proof π. It outputs 1 if it accepts the proof, and 0 otherwise.

The non-interactive proof system must be complete, meaning that if $R(x, w) = 1$ and $crs \leftarrow \mathsf{Setup}(1^\kappa)$ then

$$\mathsf{Verify}(crs, x, \mathsf{Prove}(crs, x, w)) = 1$$

sFE的定义：



与前面的泛函加密的区别在于加了一个消毒过程

这里的$SK_f = Gen(msk, f)$,解密得到的为$f(m)$。

另外又定义了一个$m \leftarrow MDec(Gen(msk, f_{id}), San(pp, c))$

这个MDec能够得到明文m

sFE的安全性：

Definition 4 (Correctness for sFE). *Given a function family $\mathcal{F}$. For all $f \in \mathcal{F}$ and all messages $m \in \mathcal{M}$:*

$$\Pr\left[\mathsf{Dec}(\mathsf{Gen}(msk, f), \mathsf{San}(pp, \mathsf{Enc}(pp, m))) \neq f(m)\right] \leq \mathsf{negl}(\kappa)$$

where $(pp, msk) \leftarrow \mathsf{Setup}(1^\kappa)$ *and the probabilities are taken over the random coins of all algorithms.*

正确性，保证能够解密出的结果是正确的

IND-CPA安全性：语义安全

发送两个明文 m_0, m_1

Challenger C

选择$b \in \{0,1\}$，发送$c = Enc(pp, m_b)$

Adversary A

发送对于密文猜测的b'

We say that A wins the IND-CPA game if $b = b'$, $|m_0| = |m_1|$, *and that* $f_i(m_0) = f_i(m_1)$ *for all oracle queries* f_i. *We say a sFE scheme satisfies the IND-CPA security property if for all PPT A*

$$\mathsf{adv}^A = 2 \cdot \left| \Pr[A \textit{ wins the IND-CPA game}] - \frac{1}{2} \right| \leq \mathsf{negl}(\kappa)$$

A具有询问SK_{f_i}的权限，可以解密获得 $f(m)$

但是对于明文的要求是对于所有的 $f_i : f_i(m_0) = f_i(m_1)$

消毒过程的安全性：

We say that A wins the sanitizer game *if* $b = b'$. *We say a sFE scheme is sanitizable if for all PPT A*

$$\mathsf{adv}^A = 2 \cdot \left| \Pr[A \textit{ wins the sanitizer game}] - \frac{1}{2} \right| \leq \mathsf{negl}(\kappa)$$

敌手A无法区分消毒后的密文是由发送的密文还是随机的密文进行消毒得到的，
说明消毒过程产生的结果是“随机”的且无法预测的

前面的MDec(Gen(msk,fid),c)才能得到明文，因此Enc(pp,MDec(msk,c))产生的密文相当于“随机”密文

接下来根据sFE的定义加入IO和NIZK后的构造：

安全参数k

$(pk_1, sk_1) \leftarrow sPKE.Setup(1^k)$

$(pk_2, sk_2) \leftarrow sPKE.Setup(1^k)$

$crs_E \leftarrow NIZK.Setup(1^k, R_E)$

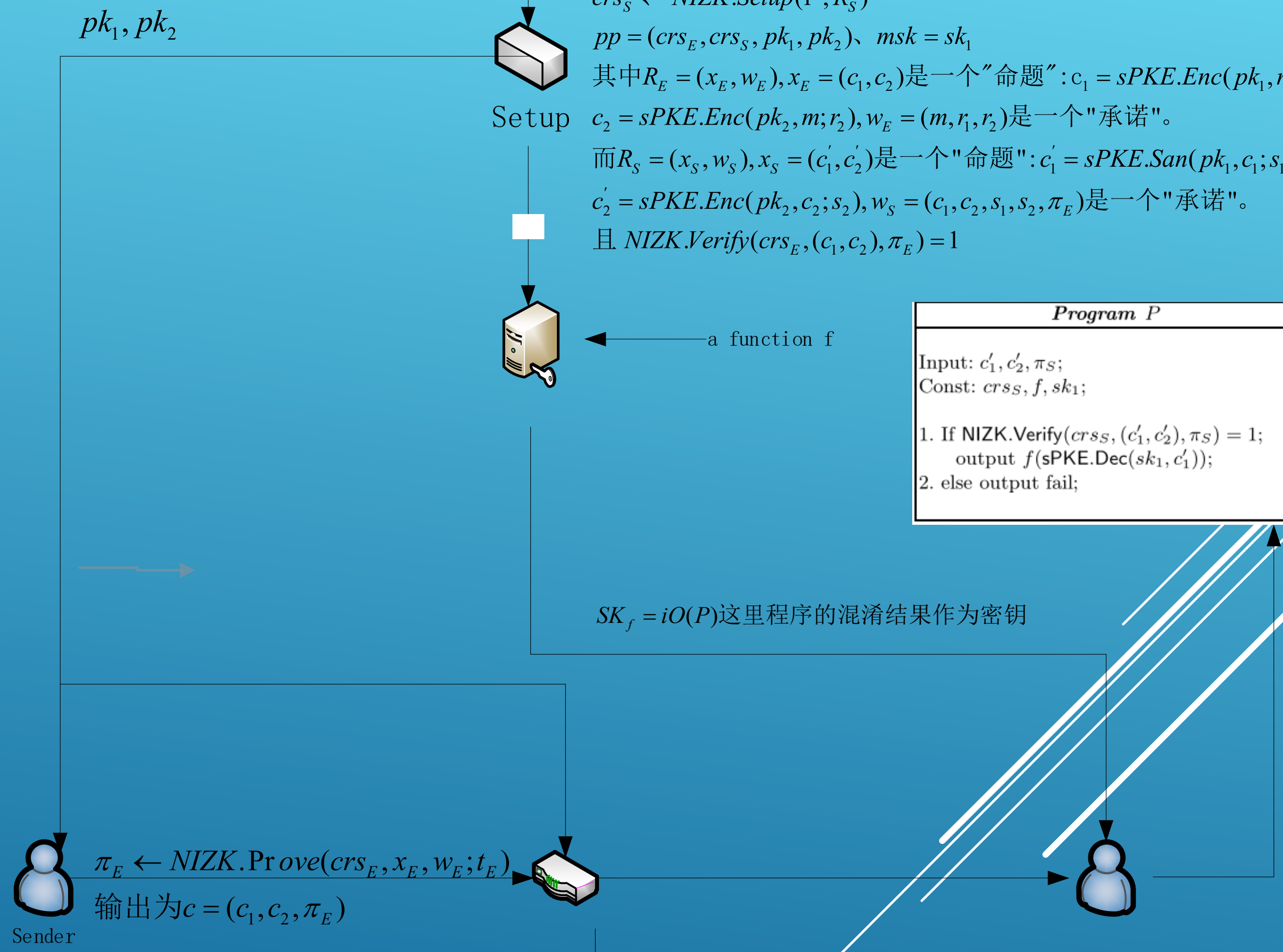
$crs_S \leftarrow NIZK.Setup(1^k, R_S)$

$pp = (crs_E, crs_S, pk_1, pk_2)$、$msk = sk_1$

其中$R_E = (x_E, w_E), x_E = (c_1, c_2)$是一个"命题"：$c_1 = sPKE.Enc(pk_1, m; r_1)$，$c_2 = sPKE.Enc(pk_2, m; r_2), w_E = (m, r_1, r_2)$是一个"承诺"。

而$R_S = (x_S, w_S), x_S = (c_1', c_2')$是一个"命题"：$c_1' = sPKE.San(pk_1, c_1; s_1)$，$c_2' = sPKE.Enc(pk_2, c_2; s_2), w_S = (c_1, c_2, s_1, s_2, \pi_E)$是一个"承诺"。

且 $NIZK.Verify(crs_E, (c_1, c_2), \pi_E) = 1$



Program P

Input: c_1', c_2', π_S;
Const: crs_S, f, sk_1;

1. If NIZK.Verify$(crs_S, (c_1', c_2'), \pi_S) = 1$;
   output f(sPKE.Dec(sk_1, c_1'));
2. else output fail;

Sanitizer: On input the public parameter pp and a ciphertext $c = (c_1, c_2, \pi_E) \in \mathcal{C}$ compute the following

1. If NIZK.Verify$(crs_E, x_E, \pi_E) = 1$ then
   $c_1' \leftarrow$ sPKE.San$(pk_1, c_1; s_1)$
   $c_2' \leftarrow$ sPKE.San$(pk_2, c_2; s_2)$
   $\pi_2 \leftarrow$ NIZK.Prove$(crs_S, x_S, w_S; t_S)$
   Output $c' = (c_1', c_2', \pi_2)$
2. Else
   Output $c' \leftarrow$ sFE.San$(pp$, sFE.Enc$(pp, \perp))$

Polylog ACE构造：sFE=(Setup,Gen,Enc,San,Dec).

F_1和F_2是伪随机函数

$K \leftarrow \{0,1\}^k$是 F_1的一个 key

$(pp^{sFE}, msk^{sFE}) \leftarrow sFE.Setup(1^k)$

$pp = pp^{sFE}, msk = (msk^{sFE}, K)$

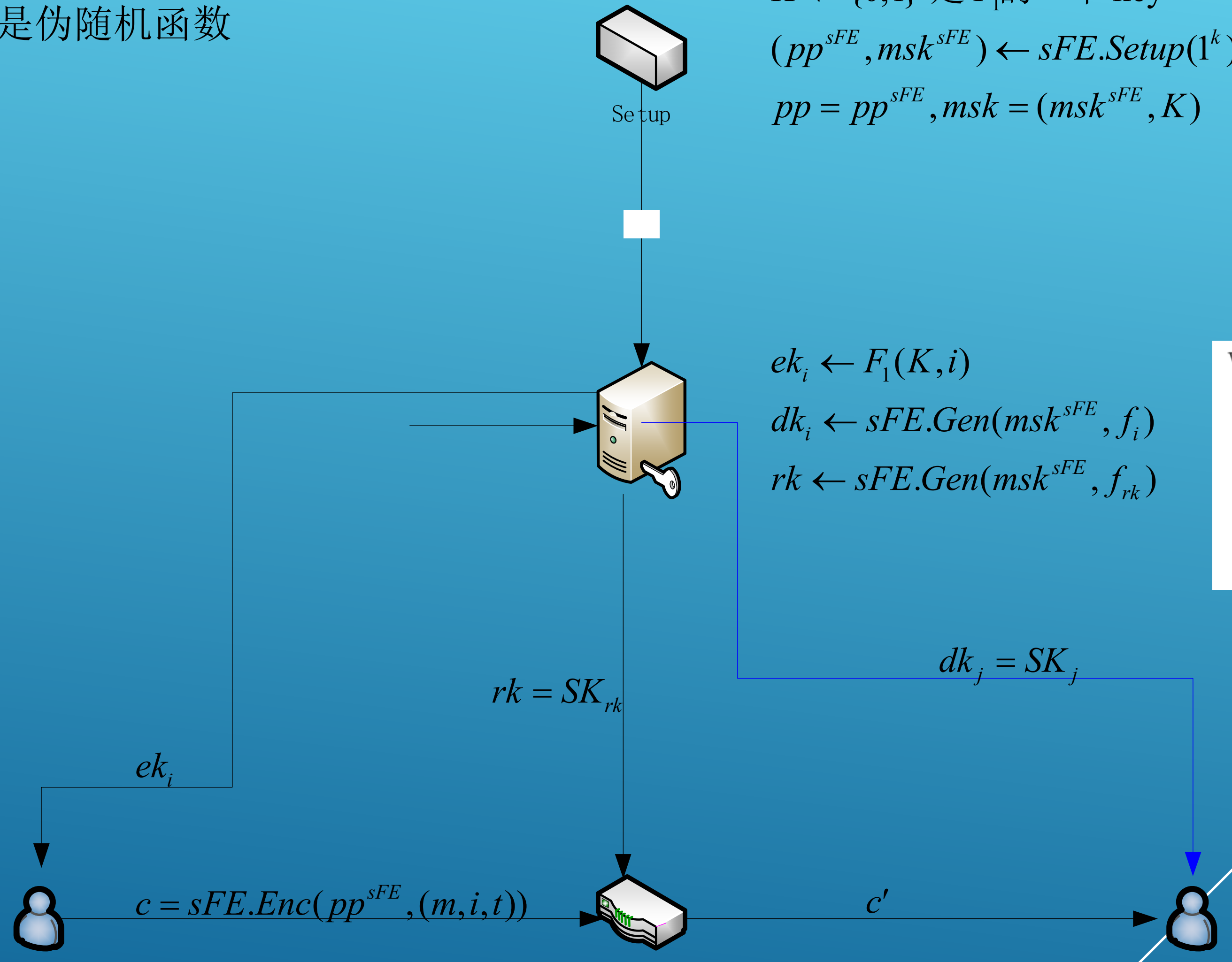
$ek_i \leftarrow F_1(K, i)$

$dk_i \leftarrow sFE.Gen(msk^{sFE}, f_i)$

$rk \leftarrow sFE.Gen(msk^{sFE}, f_{rk})$

where the functions f_i and f_{rk} are defined as follows

Decryption function	*Sanitizer function*
$f_i(m,j,t)$: 1. If $P(j,i)=1$: output m; 2. Else output $\perp$;	$f_{rk}(m,j,t)$: 1. $ek_j = F_1(K,j)$; 2. If $t = F_2(ek_j, m)$: output 1; 3. Else output 0;



$t = F_2(ek_i, m)$相当于明文的MAC

检查MAC

$1. c' = sFE.San(pp^{sFE}, c)$

$2. if\ \ sFE.Dec(SK_{rk}, c') = 1: output\ c'$

$3. Else\ output\ San(rk, Enc(ek_0, \perp))$

$m' = sFE.Dec(SK_j, c')$

正确性：

Lemma 5. *Construction 4 is a* correct *ACE scheme*

Proof. Let $P(i,j) = 1$ for some i, j. Let c' be a honest sanitization of a honest generated encryption o message m under identity i:

$$c' = \mathsf{San}(rk, \mathsf{Enc}(ek_i, m)) = \mathsf{sFE.San}(pp^{\mathsf{sFE}}, \mathsf{sFE.Enc}(pp^{\mathsf{sFE}}, (m, i, F_2(ek_i, m))))$$

Given the decryption key $dk_j = SK_j \leftarrow \mathsf{sFE.Gen}(msk, f_j)$. Then the correctness property of the sFE scheme gives

$$\Pr[\mathsf{Dec}(dk_j, c') = m] = \Pr[\mathsf{sFE.Dec}(SK_j, c') = m] \leq \mathsf{negl}(\kappa)$$

□

正确性转化为sFE的正确性，进而为iO、PKE、SSS-NIZK的正确性以及对函数的检查等

Lemma 2. *Construction 3 is a correct functional encryption scheme.*

Proof. Correctness follows from the correctness of the iO, PKE, and SSS-NIZK schemes, and from inspection of the algorithms. □

No-Read Rule：

Theorem 1. *For any adversary A that breaks the No-Read Rule of Construction 4, there exists an adversary B for the IND-CPA security of the sanitizable functional encryption scheme, such that the advantage of A is*

$$\mathsf{adv}^{\mathsf{ACE},A} \leq \mathsf{adv}^{\mathsf{sFE},B}$$

反证法：假如任何敌手赢得sFE 中的IND-CPA 游戏：最大优势为ε。则假设存在一个敌手A 赢得ACE 中 No-Read 游戏的优势大于ε ，接下来证明存在敌手B 赢得IND-CPA 优势大于ε。

$A: ACE的No-Read\ game,$可以询问O_G和O_E

$B: sFE的IND-CPA\ game,$ 可以询问O_{f_i} 获得SK

B首先可以生成 F_1的$Key: K$，并收到公开参数 pp^{sFE}然后将其发给 A

对 $A:(j,sen)$可以得到$ek_j = F_1(K,j),(j,rec)$可以得到来自于B的$dk_j = SK_j$

(j,san)可以得到来自于B的SK_{rk}（通过$O_{f_{rk}}$），(i,m)可以通过B得到密文

即B可以：$c \leftarrow sFE.Enc(pp^{sFE},(m,i,F_2(ek_i,m)))$

对B：收到来自于A的m_0,m_1,i_0,i_1，计算出ek_i后就能够计算出m_l^{sFE},将其发送给$Challenger$得到c'

B将密文发送给A，如果A赢得 $game$，则B也可以赢得$game$

则此时B赢得IND-CPA $game$的优势就会大于ε,与假设不符

因此A赢得$No-Read\ game$的优势不会大于B赢得$sFE: IND-CPA$的优势

Lemma 3. *For any adversary A that breaks the IND-CPA security property of Construction 3, there exists an adversary B for the computational zero-knowledge property of the NIZK scheme, an adversary C for the IND-CPA security of the PKE scheme, and an adversary D for iO such that the advantage of adversary A is*

$$\mathsf{adv}^{\mathsf{sFE},A} \leq 4|\mathcal{M}|\left(\mathsf{adv}^{\mathsf{NIZK},B} + \mathsf{adv}^{\mathsf{sPKE},C} + q \cdot \mathsf{adv}^{iO,C}(1-2p_{sss})\right)$$

where q is the number of secret key queries adversary A makes during the game, and p_{sss} is the negligible soundness error of the SSS-NIZK scheme.

对于sFE中赢得IND-CPA的优势进行了量化

No-Write Rule：

Theorem 2. *For any adversary A that breaks the No-Write Rule of Construction 4, there exists an adversary B for the PRF security, an adversary C for the sanitizer property of the sFE scheme, and an adversary D for the IND-CPA security of the sFE scheme, such that the advantage of A is*

$$\mathsf{adv}^{\mathsf{ACE},A} \leq 3 \cdot \mathsf{adv}^{\mathsf{PRF},B} + \mathsf{adv}^{\mathsf{sFE},C} + \mathsf{adv}^{\mathsf{sFE},D} + 2^{-\kappa}$$

通过一系列的Hybrid序列来证明不可区分：

$Hybrid\ 0: No-Write\ game\ \ for\ b=1$即$c' = San(rk,c)$

$Hybrid\ 1$:同0,除了：挑战者收到请求(i,sen)后，会保存身份i和加密密钥$ek_i = F_1(K,i)$

收到挑战(c,i')后，用sFE的主解密得到$(m^*,i^*,t^*) = sFE.MDec(msk^{sFE},c)$

如果$i^* \notin I_S$,计算ek_{i^*}。然后检查$t^* = F_2(ek_{i^*},m^*)$,检查通过则$c^* = sFE.San(pp^{sFE},c)$,否则$c^* = San(rk,Enc(ek_0,\perp))$

首先，Hybrid 0和Hybrid 1是相同的，这由San的消毒过程可得，对c用pp消毒，用SK进行验证

$Hybrid\ 1$:同0,除了：挑战者收到请求 (i,sen) 后，会保存身份i和加密密钥$ek_i = F_1(K,i)$

收到挑战(c,i')后，用sFE的主解密得到$(m^*,i^*,t^*) = sFE.MDec(msk^{sFE},c)$

如果$i^* \notin I_S$,计算ek_{i^*}。然后检查 $t^* = F_2(ek_{i^*},m^*)$,检查通过则 $c^* = sFE.San(pp^{sFE},c)$,否则 $c^* = San(rk,Enc(ek_0,\perp))$

$Hybrid\ 2$:同1，除了：加密密钥是随机选择的，包括 ek_{i^*}也是随机选择的

Claim 2. For any adversary A that can distinguish Hybrid 1 and Hybrid 2, there exists an adversary B for the security of PRF F_1 such that the advantage of A is $\mathsf{adv}^A \leq \mathsf{adv}^{\mathsf{PRF},B}$.

仍旧是反证法，假设对于伪随机函数任何敌手A 赢得的优势最大为ε,假设能够区分 $Hybrid\ 1$和 $Hybrid\ 2$ 的优势大于ε，然后构建对于PRF的优势大于ε的敌手B。

首先B正确生成 pp,将其发送给A。

对于B，收到查询 (i,sen) 时，将i 发给 PRF得到y_i，设置$ek_i = y_i$并返回

当收到 (i,m)时，询问$Challenger$加密密钥并加密明文。其他敌手的请求用构造的算法正确返回。

当收到 (c,i')时，主解密得到i^*。如果$i^* \notin I_S$，然后B通过询问$Challenger$计算正确的ek_i^*。

B通过将敌手猜测的 b'转发给挑战者结束 $game$。

观察如果$y_i \leftarrow F_1(K,i)$则在$Hybrid\ 1$,如果y_i随机则在$Hybrid\ 2$.

因此，如果A能够区分两种混合，那么B 就可以打破PRF约束。

$Hybrid\ 2$:同1，除了：加密密钥是随机选择的，包括 ek_{i^*} 也是随机选择的

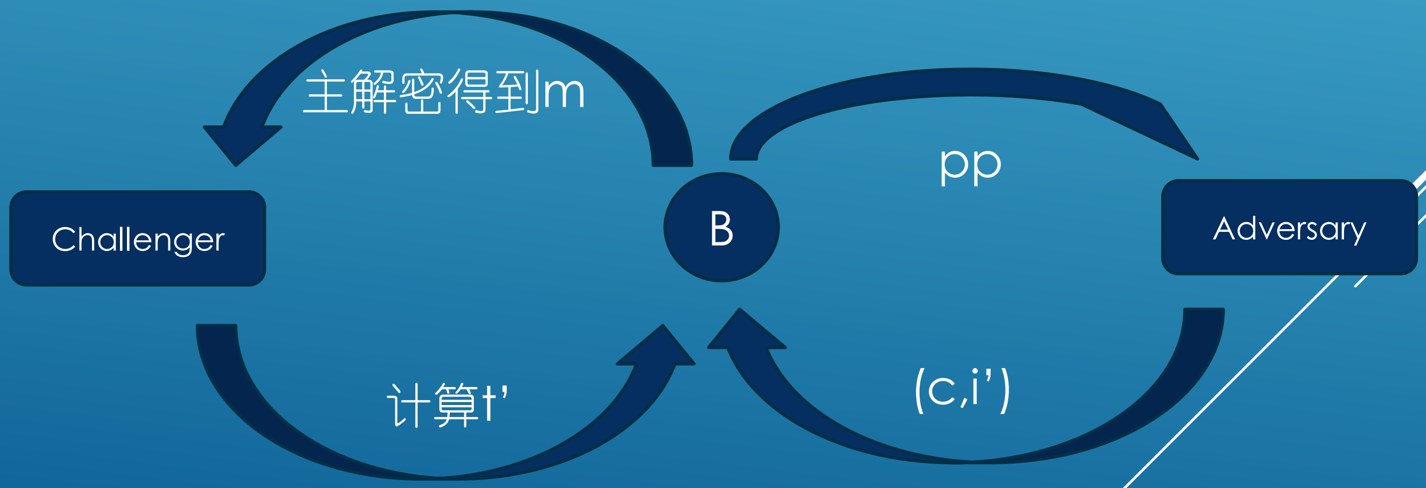
即 $c^* = San(rk, Enc(ek_i, \perp))$，$ek_i$ 是随机的

$Hybrid\ 3$:同2,除了要检查 $i^* \in I_S$,如果满足则像1一样检查MAC，否则就计算

$c^* = San(rk, Enc(ek_0, \perp))$

Claim 3. For any adversary A that can distinguish Hybrid 2 and Hybrid 3, there exists an adversary B' for the security of PRF F_2 such that the advantage of A is $\mathsf{adv}^A \leq \mathsf{adv}^{\mathsf{PRF},B'} + 2^{-\kappa}$.

假设区分PRF：ε-1/(2^k)，区分2与3:大于ε,构建B对于PRF:区分优势大于ε-1/(2^k)



$Adversray$能够以大于 ε 区分2与3

B验证 t' 与 t^*，相等则说明 $Challenger$ 用的为 F_2 函数

否则用的为其他函数。

当 t' 是用函数 F_2 生成的，B输出PRF的概率是 ε

不是用 F_2 生成但相等的概率是 2^{-k}。

因此 B 的优势大于 $\varepsilon - 2^{-k}$，矛盾

Hybrid 4与 *Hybrid* 3的区别：检查$i^* \in I_S$后检查MAC，都通过的情况下，

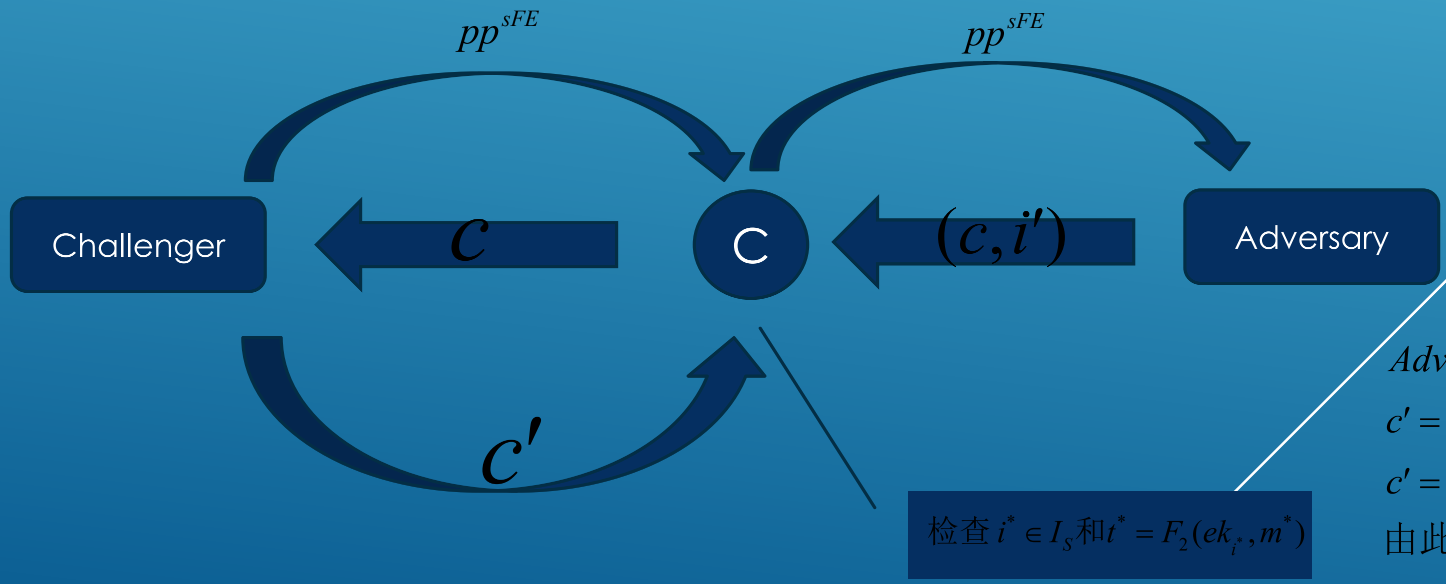
Hybrid 3计算$c^* = sFE.San(pp^{sFE}, c)$

Hybrid 4计算$c^* = sFE.San(pp^{sFE}, sFE.Enc(pp^{sFE}, (m^*, i^*, t^*)))$

Claim 4. For any adversary A that can distinguish Hybrid 3 and Hybrid 4, there exists an adversary C for the sanitizer property of the sFE scheme such that the advantage of A is $\mathsf{adv}^{A} \leq \mathsf{adv}^{\mathsf{sFE},C}$.

假设Adversary能够区分3与4，任何敌手对于sFE的消毒安全性游戏赢得优势为ε，设立矛盾Adversary区分3与4的优势大于ε，构建敌手C对于sFE的消毒优势大于ε

根据sFE消毒安全性游戏定义，C可以拿到主密钥msk，也就意味着C与Adversary勾结可以主解密。



*Adversray*拿到C拿到的c'区分出Hybrid 3与*Hybrid* 4，猜测发给C

$c' = sFE.San(pp^{sFE}, c)$ *or*

$c' = sFE.San(pp^{sFE}, sFE.Enc(pp^{sFE}, sFE.MDec(msk^{sFE}, c)))$

由此C具有的优势同Adversary大于ε，矛盾

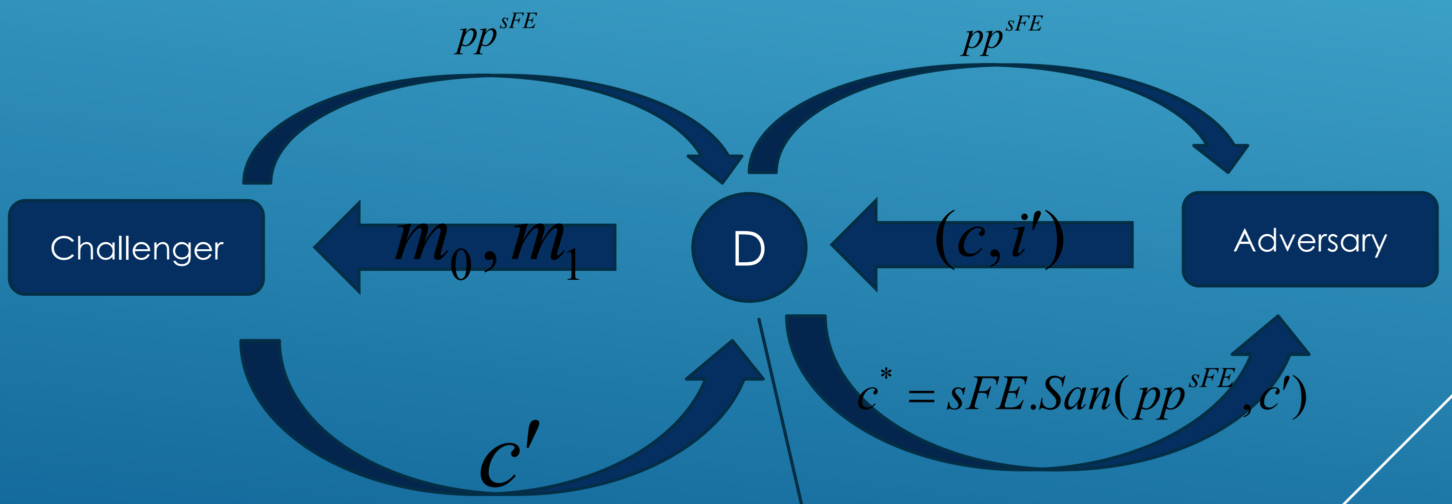
$Hybrid\ 5$与$Hybrid\ 4$的区别，其他相同在计算 c^*时：

$$\text{Hybrid } 4: c^* = sFE.San(pp^{sFE}, sFE.Enc(pp^{sFE}, (m^*, i^*, t^*)))$$

$$Hybrid\ 5: c^* = San(rk, Enc(ek_{i'}, r)) \quad where\ r \leftarrow_\$ M, rk \leftarrow Gen(msk, n+1, san)$$

Claim 5. For any adversary A that can distinguish Hybrid 4 and Hybrid 5, there exists an adversary D for the IND-CPA security of the sFE scheme such that the advantage of A is $\mathsf{adv}^A \leq \mathsf{adv}^{\mathsf{sFE},D}$.

假设任意敌手对于sFE的IND-CPA优势为ε，敌手A可以以大于ε的优势区分Hybrid 4与Hybrid 5，构建敌手D可以以大于ε的优势赢得sFE的IND-CPA



检查 $i^* \in I_S$和$t^* = F_2(ek_{i^*}, m^*)$,通过检查设置 $m_0 = (m^*, i^*, t^*)$

否则$m_0 = (\perp, 0, \perp)$

另外 $m_1 = (r, i', F_2(ek_{i'}, r))$

Definition 5 (IND-CPA Security for sFE). *Consider the following game between a challenger C and a stateful adversary A:*

IND-CPA Security	
Game Definition	Oracle Definition
1. $(pp, msk) \leftarrow \mathsf{Setup}(1^\kappa)$; 3. $(m_0, m_1) \leftarrow A^{\mathcal{O}(\cdot)}(pp)$; 4. $b \leftarrow \{0,1\}$; 5. $c^* \leftarrow \mathsf{Enc}(pp, m_b)$ 6. $b' \leftarrow A^{\mathcal{O}(\cdot)}(c^*)$;	$\mathcal{O}(f_i)$: 1. Output $SK_{f_i} \leftarrow \mathsf{Gen}(msk, f_i)$;

Adversary能够区分$Hybrid\ 3$与$Hybrid\ 4$,可以给出c^*是用发出的密文c生成或是随机值生成，猜测发给D

对于D而言，处于IND-CPA $game$中，来自于随机值就对应m_1反之对应m_0。

因此D具有同A的优势大于ε，矛盾

$Hybrid\ 6$与$Hybrid\ 5$的区别：$Hybrid\ 6$的加密密钥是诚实生成的即$ek_i = F_1(K,i)$

对于$Hybrid\ 5, c^* = San(rk, Enc(ek_{i'}, r))$中$ek_{i'}$是随机生成的

对于$Hybrid\ 6, c^* = San(rk, Enc(ek_{i'}, r))$中$ek_{i'}$是正确生成的$\rightarrow b=0$的$No-Write\ game$

Claim 6. For any adversary A that can distinguish Hybrid 5 and Hybrid 6, there exists an adversary B for the security of PRF F_1 such that the advantage of A is $\mathsf{adv}^A \leq \mathsf{adv}^{\mathsf{PRF},B}$.

证明同Claim 2

综上，对于任意能够区分$Hybrid\ 0(b=1$的$No-Write\ game)$~$Hybrid\ 6(b=0$的$No-Write\ game)$

其优势满足：

$$adv^{ACE,A} \leq 3 \times adv^{PRF,B} + adv^{sFE,C} + adv^{sFE,D} + 2^{-k}$$

Lemma 3. *For any adversary A that breaks the IND-CPA security property of Construction 3, there exists an adversary B for the computational zero-knowledge property of the NIZK scheme, an adversary C for the IND-CPA security of the PKE scheme, and an adversary D for iO such that the advantage of adversary A is*

$$\mathsf{adv}^{\mathsf{sFE},A} \leq 4|\mathcal{M}|\left(\mathsf{adv}^{\mathsf{NIZK},B} + \mathsf{adv}^{\mathsf{sPKE},C} + q \cdot \mathsf{adv}^{iO,C}(1-2p_{sss})\right)$$

where q is the number of secret key queries adversary A makes during the game, and p_{sss} is the negligible soundness error of the SSS-NIZK scheme.

Lemma 4. *For any adversary A that breaks the sanitizer property of Construction 3, there exists an adversary B for the computational zero-knowledge property of the NIZK scheme such that the advantage of adversary A is*

$$\mathsf{adv}^{\mathsf{sFE},A} \leq 2|\mathcal{M}|\mathsf{adv}^{\mathsf{NIZK},B}$$

谢谢！