CPA to CCA via Hinting PRG or KDM SKE

上次的小短文展示了如何利用"可随机数恢复的CPA安全的PKE"构造CCA安全的PKE. 但是为什么会想到之前那种结构:

- 外层结构：一方面，利用**集合承诺**保证解密算法**只检验B个"合法"解密即可**（其他甚至可以不加以解密，这是非常关键的，**打破死锁**的一个关键）；另一方面，加入足够多的冗余，方便为内层的加密随机数**去限制**
- 内层结构：随机数之间有限制关系，$\bigoplus_{i\in I}\mathbf{r}_i=0$，结合PKE的随机数可恢复性质，具有多种ABO结构的**等价解密模拟方式**
- 承诺结构：具有去除S独特性的一个新的启动方式Altsetup + 一个ABO类型的个数**上界**B（ABO: $tag\neq tag^*$）

我们并没有深入地追究，这篇短文我们回归到问题的出发点，思考一下从CPA实现CCA有怎样的**难点**以及这类技术的思考**切入点**，即深入讲解一下**部分加密/检验**的"Singal技术"：

- Koppula, Waters. Realizing Chosen Ciphertext Security Generically in Attribute-Based Encryption and Predicate Encryption. CRYPTO 2019. （CPA + HPRG = CCA）
- Kitagawa, Matsuda, Tanaka. CCA Security and Trapdoor Functions via Key-Dependent-Message Security, CRYPTO 2019. (CPA + one-time projection KDM SKE = CCA)

1 技术出发点：Randomness-Recovering

Randomness-Recovering指的是Dec Oracle解密的时候，可以**恢复出m和加密随机数**，在ROM下，例如经典的FO变换：

$$\mathcal{E}^{hy}_{pk}(m;r):=\mathcal{E}^{asy}_{pk}(r;H(r,c))||\mathcal{E}^{sy}_{G(r)}(m)$$
$$\text{where } c=\mathcal{E}^{sy}_{G(r)}(m)$$

Random Oracle保证了如果$H(r,c)$没有被问过，则$\mathcal{E}^{asy}_{pk}(r;H(r,c))$具有很大的熵值，基本上解密时进行重加密检验都是错的，输出$\perp$，从而在安全证明当中可以论证r被掩盖，从而安全.

但是在标准模型下，具有随机数恢复的PKE难以构造；例如一个自然的想法是：

$$\mathcal{E}^{hy}_{pk}(m;r):=\mathcal{E}^{asy}_{pk}(s;r)||\mathcal{E}^{sy}_{s}(r\mid m)$$

但是在安全证明的时候这是**相互嵌套、死锁**的，所以想要恢复所有的随机数，看上去不那么可能，那么恢复部分随机数是否可以呢?

Signaling技术 —— 恢复部分加密随机数（一部分可以恢复随机数的被称为Signal-1区域，不能则称为Signal-0区域）

恢复部分加密随机数，其实非常容易想到该怎么去构造，例如：

$$s \leftarrow \{0,1\}^n$$

$$\begin{matrix} \mathsf{Enc}(pk_0, \mathsf{Equal}(0, s_1); r_1^0) & \cdots & \mathsf{Enc}(pk_0, \mathsf{Equal}(0, s_n); r_n^0) \\ \mathsf{Enc}(pk_1, \mathsf{Equal}(1, s_1); r_1^1) & \cdots & \mathsf{Enc}(pk_0, \mathsf{Equal}(1, s_n); r_n^1) \end{matrix} \quad + \quad \mathsf{SKE.Enc}(s, \{r_i^{s_i}\}_{i\in[n]}||m)$$

即可以恢复出$\{r_i^{s_i}\}_{i\in[n]}$，但是注意到在Dec时正常流程需要解密每一个$\mathsf{Enc}(pk_b, \mathsf{Equal}(b, s_i); r_1^b)$，保证$1$的加密确实有$n$个（保证Signal恢复的$s$没有二义性），且满足相应的格式，即每一列只有一个$1$。但是，当我们只有一把私钥$sk_b$的时候，由于无法解密所有的密文，其检验并不能保证针对所有$1-b$行且$s_i=0$的元素进行重加密检验。

那么重加密检验只有部分，我们**希望部分检验仍然和整体的检验等价**，得到正确Signal。为此，我们还需要**引入集合承诺**，一方面，我们希望该承诺保证对于每列的两个元素，**至多一个**是对应着承诺的，即保证了**Signal-1区域的上界是一半**：

$$s \leftarrow \{0,1\}^n \text{ (若此时}s = (11\cdots 0))$$

$$\begin{matrix} \mathsf{Enc}(pk_0, 0; r_1^0) & \cdots & \mathsf{Enc}(pk_0, 1|v_n; r_n^0) \\ \mathsf{Enc}(pk_1, 1|v_1; r_1^1) & \cdots & \mathsf{Enc}(pk_0, 0; r_n^1) \\ com(pp, v_1) & \cdots & com(pp, v_n) \end{matrix} \quad + \quad \mathsf{SKE.Enc}(s, \{r_i^{s_i}\}_{i\in[n]}||m)$$

另外一方面，为了证明的方便，加强该集合承诺的性质，使其具有ABO结构，其**针对特定的挑战密文的有摸棱两可的性质**，而对其他的密文均有上述的上界限制。例如：

$$\begin{matrix} \mathsf{Enc}(pk, 1|v_1; r_1^0) \\ \mathsf{Enc}(pk, 1|w_1; r_1^1) \\ com(pp, v_1) \end{matrix}$$

这里，我们可以定义com的规则使得其即可以Signal 1，也可以Signal 0. 从而使得其与s_1无关。

为了这样Hybrid方案安全，我们应该想办法把s和"+"号左边的部分解耦合，在上面的**承诺的两种性质**下以及PKE的CPA安全性下可以完成，但是由于"+"右边要做的重加密检验永远包含Signal-1区域，即s有关，例如使用的加密随机数$\{r_i^{s_i}\}_{i\in[n]}$，所以如何都会要牵扯到**"正式进行加密的部分包含signal-1区域的信息"**这一问题，如何运用s的熵是一个需要考虑的问题，本短文将介绍KW19和KMT19的两种解决方式：

- KMT19 - SKE的KDM安全（依赖函数族仅需Projection函数族）：$\mathsf{Enc}(s, \{r_i^{s_i}\}_{i\in[n]}|k)$与$\mathsf{Enc}(s, 0)$计算不可区分
- KW19 - HPRG的类KDM安全: 在其安全性定义中，s会用作index的hints使用：

$\mathsf{Setup}(1^\lambda, 1^\ell)$: The setup algorithm takes as input the security parameter λ, and length parameter ℓ, and outputs public parameters pp and input length $n = n(\lambda, \ell)$.

$\mathsf{Eval}(\mathsf{pp}, s \in \{0,1\}^n, i \in [n] \cup \{0\})$: The evaluation algorithm takes as input the public parameters pp, an n bit string s, an index $i \in [n] \cup \{0\}$ and outputs an ℓ bit string y.

Definition 3.1. A hinting PRG scheme $(\mathsf{Setup}, \mathsf{Eval})$ is said to be secure if for any PPT adverasry $\mathcal{A}$, polynomial $\ell(\cdot)$ there exists a negligible function $\mathsf{negl}(\cdot)$ such that for all $\lambda \in \mathbb{N}$, the following holds:

$$\left| \Pr\left[1 \leftarrow A\left(\mathsf{pp}, \left(y_0^\beta, \left\{y_{i,b}^\beta\right\}_{i\in[n],b\in\{0,1\}}\right)\right) : \begin{array}{c} (\mathsf{pp}, n) \leftarrow \mathsf{Setup}(1^\lambda, 1^{\ell(\lambda)}), s \leftarrow \{0,1\}^n, \\ \beta \leftarrow \{0,1\}, y_0^0 \leftarrow \{0,1\}^\ell, y_0^1 = \mathsf{Eval}(\mathsf{pp}, s, 0), \\ y_{i,b}^0 \leftarrow \{0,1\}^\ell \ \forall\ i \in [n], b \in \{0,1\}, \\ y_{i,s_i}^1 = \mathsf{Eval}(\mathsf{pp}, s, i), y_{i,\overline{s_i}}^1 \leftarrow \{0,1\}^\ell \ \forall\ i \in [n] \end{array} \right] - \frac{1}{2} \right| \leq \mathsf{negl}(\cdot)$$

在这种方法下，KDM安全似乎不可避免，个人认为KDM安全的SKE，即**[KMT19]对如何在Signal技术下使用s的熵这一内容的捕捉更准确**。HPRG也可以看作一种KDM安全！

2 构造 1：CCA PKE via Hinting PRG

使用组件：带有Recover算法的CPA安全的PKE（被CPA安全的PKE蕴涵） + KDM安全的HPRG + 0/1 区分承诺

构造如下：

- Setup: 产生Hinting PRG的公开参数pp，产生n对CPA安全PKE的密钥对$\{(sk_i^b, pk_i^b)\}_{i\in[n],b\in\{0,1\}}$；选择承诺的公开参数，即随机数$\{a_i\}_{i\in[n]}$和$B$
- Enc: 产生一次性签名密钥对(ssk, vk)；产生Hinting PRG的种子$s = (s_1, \cdots, s_n)$，$\mathsf{HPRG}(s)$产生了$1 + 2n$个数$(r_0, \{r_i^b\}_{i\in[n],b\in\{0,1\}})$；选择承诺随机参数$v_i$进行加密
  - Signal**检验**部分：
    - 1-Signal 部分：$\mathsf{ct}_i^{s_i} = \mathsf{Enc}(pk_i^{s_i}, 1|v_i; r_i^{s_i})$
    - 0-Signal 部分：$\mathsf{ct}_i^{1-s_i} = \mathsf{Enc}(pk_i^{1-s_i}, 0; r_i^{1-s_i})$
    - **0/1 区分承诺**部分：$com_i = \mathsf{PRG}(v_i) + s_i \cdot (a_i + B \cdot vk)$
  - **加密**部分：$\mathsf{ct} = m \oplus r_0$
  - 签名部分：对上述进行ssk下的签名，得到Σ
  - 输出：$[vk, (\mathsf{ct}_i^0, \mathsf{ct}_i^1, com_i), ct, \Sigma]$
- Dec:
  - 验证签名
  - 仅仅使用sk_i^0对ct_i^0进行解密，得到封装的$1 \mid v_i$值，如果不是此种格式，**直接**设置$s_i = 1$
  - 检验承诺$com_i = \mathsf{PRG}(v_i)$是否通过，若通过则$s_i = 0$，否则$s_i = 1$，恢复出$s$
  - 恢复出所有的$\mathsf{HPRG}(s)$的输出$(r_0, \{r_i^b\}_{i\in[n],b\in\{0,1\}})$

- 利用CPA的PKE的$\mathsf{Recover}$算法与s相关的$\mathsf{ct}_i^{s_i}$（**只关注1-Signal下的重加密验证！**），并再次验证相关的承诺（第一次验证承诺是为了恢复出**关键的随机数**s，第二次则是进行**关键位置的重加密检验**）

安全性证明：整体想法是先去除s信息在挑战密文中的体现（包括在Enc和com中），去除com中的信息可以利用承诺的性质（类似于上一篇短文中将Setup换为AltSetup），而去除Enc信息则使用CPA安全性（注意保证Dec的模拟，正如上篇短文利用 具有随机数恢复的Dec以及随机数的限制【这里由HPRG来扮演】 + 承诺的安全上界性质【仍然由承诺来扮演】来进行模拟），最后利用HPRG的性质论证ct计算不可区分（对应到上次短文中去除随机数的关联性后，用Enc的安全性来论证），即上一篇短文是【对集合承诺性质的更为精准的提炼】 + 【对随机数恢复结构（部分$\to$整体）的重新安排：$\bigoplus$求和限制（后续去除S的信息用哈希剩余定理） $\leftrightarrow$ HPRG恢复（去除s的信息用承诺的摸棱两可性）】 + 【对消息m加密的重新放置：Signal作为Trigger，产生随机数作为对称加密 $\leftrightarrow$ Signal和消息m被放置在一起被加密】

- Game 0: 原始的安全模型
- Game 1: Dec Oracle询问vk^*则拒绝
- Game 2: 为区分承诺变得**摸棱两可**作**准备**
  - 产生n个随机数w_i
  - 根据挑战密文的参数，重新设置a_i:
    - 当$s_i^* = 0$: $a_i = \mathsf{PRG}(v_i^*) - \mathsf{PRG}(w_i) - vk^* \cdot B$
    - 当$s_i^* = 1$: $a_i = \mathsf{PRG}(w_i) - \mathsf{PRG}(v_i^*) - vk^* \cdot B$
  - $com_i^* = \mathsf{PRG}(v_i^*) + s_i \cdot ((-1)^{s_i}(\mathsf{PRG}(v_i^*) - \mathsf{PRG}(w_i)))$
    - $s_i^* = 0$: $com_i^* = \mathsf{PRG}(v_i^*)$ 【本身】
    - $s_i^* = 1$: $com_i^* = \mathsf{PRG}(w_i)$ 【另一个】
- Game 3: 更换Dec Oracle中检验恢复s的方式，原来是全用n个密钥sk_i^0进行检验，现在使用n个密钥$sk_i^{s_i^*}$进行检验，这一步是为了消除挑战信息Enc中关于s的信息。类似于之前的证明，由Recover算法的正确性以及加密重检验的保证，这样的Dec的模拟也是完成正确的（可以被之前Dec Oracle通过的情况，即合法【承诺通过】1-Signal部件的个数**恰为一半**，会在这一修改的情况下通过，而被之前的Dec Oracle拒绝的情况，即1-Signal部件的个数**少于一半**，也会以类似的方式被拒绝；另外承诺保证了在$vk \neq vk^*$的条件下，任意$(\mathsf{ct}_i^0, \mathsf{ct}_i^1, com_i)$就算用所有的$sk_i^b$全局地被Check，**合法的也不会多于一半**，这一安全上界的保证也是非常重要的！）
- Game 4: 去除挑战密文Enc当中关于s的信息，即将$\mathsf{ct}_i^{1-s_i^*}$从$\mathsf{Enc}(pk_i^{1-s_i^*}, 0; r_i^{1-s_i^*})$切换为$\mathsf{Enc}(pk_i^{1-s_i^*}, 1|w_i; r_i^{1-s_i^*})$
  - 注意到此时，在不考虑$r_i^{s_i}$和$r_i^{1-s_i}$分布的不同下，对于**挑战密文**中**承诺解释的摸棱两可性**，假定$s_i = 0$:
    - $\mathsf{ct}_i^0 = \mathsf{Enc}(pk_i^0, 1|v_i; r_i^{s_i}), \mathsf{ct}_i^1 = \mathsf{Enc}(pk_i^1, 1|w_i; r_i^{1-s_i}), com_i = \mathsf{PRG}(v_i)$, $a_i = \mathsf{PRG}(v_i) - \mathsf{PRG}(w_i) - vk \cdot B$
    - 显然，这可以理解为关于$s_i = 0$的Signal

- 事实上，这也可以理解为关于$s_i = 1$的Signal，因为v_i, w_i不过是变量叫法的不同，并且注意到在Game 2中a_i的设置是适应性的，所以我们也可以看作是关于$s_i = 1$的Signal，$\mathsf{ct}_i^0 = \mathsf{Enc}(pk_i^0, 1|v_i; r_i^{s_i}), \mathsf{ct}_i^1 = \mathsf{Enc}(pk_i^1, 1|w_i; r_i^{1-s_i})$, $com_i = \mathsf{PRG}(v_i), a_i = (-1)^1 \cdot (\mathsf{PRG}(w_i) - \mathsf{PRG}(v_i)) - vk \cdot B$，除了随机数$r_i^{s_i}$和$r_i^{1-s_i}$的分布不同，解释是完全相同的，这种对称性保证我们在不知道s的情况下即可以进行加密，并且总可以解释为按照s加密的规则
- **也许会问**，这种摸棱两可性不和之前Dec Oracle模拟等价性中承诺所带来的安全上界保证矛盾吗？事实上，摸棱两可性是仅仅针对vk^*成立的，而安全上界是在上述a_i的设置下对$vk \neq vk^*$成立的

- Game 5: 利用摸棱两可性，挑战密文当中s的信息完全被去除，用Hinting PRG的伪随机性可以完成证明
  - 一个变体：我们可以将相应的消息m塞入Enc，即$\mathsf{ct}_i^{s_i} = \mathsf{Enc}(pk_i^b, 1|v_i|m; r_i^{s_i})$，**HPRG的替换消除了随机数的限制**，使得可以直接再度Enc的CPA性质消除m的信息

3 构造 2：CCA KEM via KDM SKE

相较于PKE可以用$\mathsf{Enc}'(pk, m; r) := \mathsf{Enc}(pk, r_1; r_2)|m \oplus r_1$实现$\mathsf{Recover}$算法；KEM 天然就具有$\mathsf{Recover}$算法，即调用$\mathsf{Encap}(pk; r)$，并且直接就等价于执行了一次重加密检验，而无需像PKE那样还需要检验是否有 $\mathsf{Enc}'(pk, \mathsf{Recover}(\mathsf{ct}, r); r) = \mathsf{ct}$. 但本质上没有太大的区别，【KDM SKE】是对KW19方法signal方法在最后**如何利用s这件事情上更好的抽象**。

使用组件：CPA安全的KEM + 单次KDM安全的SKE + 0/1 区分承诺

构造如下：

- Setup: 产生两组CPA安全的KEM的公私钥 $(pk_0, sk_0; pk_1, sk_1)$；产生哈希函数族的标识 hk，产生 n 个 4λ 比特长的随机数 A_i，以及一个 λ 比特长度的 B，公钥是 $(pk_0, pk_1, hk, \{A_i\}_{i\in[n]}, B)$，私钥是 sk_0 （只有一半）
- Encap: SKE产生私钥$s = (s_1, \cdots, s_n)$，产生n对λ比特长随机数$\{r_i^0, r_i^1\}_{i\in[n]}$，产生随机数$k$
  - Signal**检验**部分：产生n组$(\mathsf{ct}_i^v, k_i^v) \leftarrow \mathsf{Encap}(pk_v; r_i^v)$
  - **加密**部分随机数 + 消息：$\mathsf{ct}_{\mathsf{SKE}} \leftarrow \mathsf{SKE.Enc}(s, (r_i^{s_i})_{i\in[n]} \mid k)$
  - 0/1 **区分承诺**：$h \leftarrow H(hk, \mathsf{ct}_i^v || \mathsf{ct}_{\mathsf{SKE}})$作为$tag$，生成承诺$T_i = k_i^{s_i} + s_i \cdot (A_i + B \cdot h)$
  - 返回：$(\mathsf{ct}_i^v, T_i, \mathsf{ct}_{\mathsf{SKE}})$
- Decap:
  - $k_i^0 \leftarrow \mathsf{Decap}(sk_0, \mathsf{ct}_i^0)$，得到第0组$n$个封装的$k_i^0$值
  - 验证承诺是否有$T_i = k_i^0$，若有则令$s_i = 0$，否则$s_i = 1$，恢复出s
  - $(r_i^{s_i})_{i\in[n]} \mid k \leftarrow \mathsf{SKE.Dec}(s, \mathsf{ct}_{\mathsf{SKE}})$

- 重加密验证：（与上面的相同，只要有s对应位置的检验通过即可，一方面，这是充分的，有合法的**承诺保证稳健性**；另一方面，这是必要的，留出**自由度在密钥切换时模拟仍然等价**）
    - $\mathsf{Encap}(pk^{s_i}; r_i^{s_i}) = (\mathsf{ct}_i^{s_i}, T_i - s_i \cdot (A_i + B \cdot h))$
- 若上述检验都通过，输出：k

安全性证明：仍然是相同的思路，不再赘叙High-Level Ideas

- Game 0: 原安全实验
- Game 1: 若有$H(hk, (\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]}||\mathsf{ct}_{\mathsf{SKE}}) = h^*$直接拒绝
    - $(\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]}||\mathsf{ct}_{\mathsf{SKE}} = (\mathsf{ct}_i^{*0}, \mathsf{ct}_i^{*1})_{i\in[n]}||\mathsf{ct}^*_{\mathsf{SKE}}$：此时必然有某个$i$，使得$T_i \neq T_i^*$，这是不可能的
    - $(\mathsf{ct}_i^0, \mathsf{ct}_i^1)_{i\in[n]}||\mathsf{ct}_{\mathsf{SKE}} \neq (\mathsf{ct}_i^{*0}, \mathsf{ct}_i^{*1})_{i\in[n]}||\mathsf{ct}^*_{\mathsf{SKE}}$：违背了Hash函数抗碰撞性 (Collision-Resistant)
- Game 2: CPA-secure KEM as a PRG，为后续T_i的摸棱两可性做准备
    - 重新产生$A_i = (k_i^{*0} - k_i^{*1}) - B \cdot h^*$ 当$s_i^* = 0$
        - 由于当前的Decap算法只使用sk_0，所以可以利用k_i^{*1}条件在ct_i^{*1}上的伪随机性
- Game 3: Decap算法只使用sk_1与原来的等价；要证明其等价性，需要说明承诺所保证的安全上界仍然是一半
- Game 4: 重新产生$A_i = (k_i^{*0} - k_i^{*1}) - B \cdot h^*$ 当$s_i^* = 1$，此时区分承诺在h^*上已经完全摸棱两可了，$(\mathsf{ct}_i^{s_i}, T_i)$已经完全不包含$s$的信息了
- Game 5: 利用P-KDM安全性，证明$\mathsf{SKE.Enc}(s, (r_i^{s_i^*}) \mid k)$对$k$的掩盖，定义函数$f_{r_i^b, k}(s) = (r_i^{s_i}) \mid k$

Q1. 为什么上面需要$2n$对密钥，这里仅仅需要2对密钥?

事实上，我们也可以尝试对KW19的方案化简如下：

- $\mathsf{ct}_i^{s_i} = \mathsf{Enc}(pk^{s_i}, 1|v_i; r_i^{s_i})$
- $\mathsf{ct}_i^{1-s_i} = \mathsf{Enc}(pk^{1-s_i}, 0; r_i^{1-s_i})$
- $com_i = \mathsf{PRG}(v_i) + s_i \cdot (a_i + B \cdot vk)$
- $\mathsf{ct} = m \oplus r_0$

针对上面Game 3和Game 4需要进行变形，无需做Game 3，Game 4修改如下：

- Game 4.1: 修改挑战密文中$s_i^* = 0$的部分的ct_i^1为$\mathsf{Enc}(pk^1, 1|w_i; r_i^{1-s_i^*})$
- Game 4.2: 修改Dec Oracle的模拟方式，使用sk^1进行检验
- Game 4.3: 修改挑战密文中$s_i^* = 1$的部分的ct_i^0为$\mathsf{Enc}(pk^0, 1|w_i; r_i^{1-s_i^*})$

4 反思

本次的两个方案的技术流：

$$\text{利用Signal技术表征随机数}s \Rightarrow \text{为了解除死锁引入摸棱两可承诺（保证使用部分随机数亦是等价）}$$
$$\Rightarrow \text{利用类KDM安全使用}s\text{的随机性（辅助恢复signal-1加密随机数 + 加密消息}m\text{）}$$

KW19和KMT19两篇论文主要思想都是基于这3个步骤

HKW20与这两个方案的对比：

- HKW20还是基于”Signal技术“——仅恢复部分随机数进行重加密再检验（另外一部分用承诺性质保证），不过是将KW19、KMT19的**窗口结构一般化为集合结构**，将相应的承诺也抽象了两个关键性质（setup的替换保证摸棱两可性质 + ABO类型的上界保证）得到了一个叫做集合承诺的新原语
- 关键进步点：在还是在如何使用s的随机性上，下一篇短文将介绍Matsuda的最新文章，将一并分析；我们会看到应该如何去使用s的随机性，在**保证s辅助验证Signal-1加密**的同时，也可以独立地、干净地加密消息m

---

一个思考：

- 我们注意到：所有和s_i^*直接相关的位置（即Signal-1区域）都需要通过重加密进行检验，而Signal-0区域则无需在s恢复之后进行重加密检验（这是成功的关键，因为归约的时候始终**只持有一部分的密钥**）。能否有一种加密模式/承诺设计方法使得Signal-0.5这样的东西出现，首先其Signal了s，同时其无需利用重加密检测这种随机数使用机制？
- 为了达到更好的结果，我们也许需要设计新的Dec检验的方式！