# Notes of Cryptography

Squirrel

March 23, 2025

# Preface

## Course

密碼學設計與分析 Cryptography Design and Analysis (11320IIS500900) in NTHU

# 1 L1

## 1.1 Merkle 的故事

Merkle 在大學部修了一個課，然後要交一個 project。他在交這個作業的時候，提到了 Public Key Cryptography 的想法。當時的導師並不看好這個東西，所以 reject 了，最後他也退掉了這門課。之後他找到另一個很欣賞他的老師，覺得應該要「Publish it, win fame and fortune」，所以他將這篇文章那個投到了 CACM（Communications of the ACM）。第一次投期刊就因為「這個想法不是當今的主流想法」而被拒絕。在 Merkle 的某些堅持之下，過了快三年終於讓 CACM 接受了這篇文章。

這邊的故事及當時的論文，可以在 https://ralphmerkle.com/1974/ 找到。

另外影片中的 link 有誤，應該改成 https://ralphmerkle.com，不然你只會找到一間搞 CRM 和賣資料的公司。

## 1.2 Conventions

- 離散且有限的時間 (discrete and finite world)
  $\Rightarrow$ 因為我們正在討論 computer science
- Data v.s. Information
- Machine (function/algorithm) 需要在 polynomial time 下執行
  $\Rightarrow$ 因為我們需要能在一定時間內看到結果，不想要等到天荒地老
  $\Rightarrow$ 不一定**強制**要求 polynomial time，但這堂課大部分會是這樣
- Alice and Bob：就是 sender 和 receiver，通常是 Alice 要傳訊息給 Bob
  $\Rightarrow$ 還有其他角色，可以參見 Wikipedia：
  https://en.wikipedia.org/wiki/Alice_and_Bob
- 計算（computation）: 任何遵循 well-defined model（例如 algorithm、protocol）的 calculation。
- Efficiency
  Input size: $|x| = n$ bits
  其他的就是拿 complexity 概念來作為 efficiency 的概念
- Crypto 像是信仰 (Faith)？
  密碼學不一定總是對的，但我們需要相信某些東西才能繼續在密碼學上前進
  這些東西包含：
  $\Rightarrow$ 某些數學問題很難被解決
  $\Rightarrow$ 某些假設無法被打破（通常指在 poly-time 底下）
  $\Rightarrow$ 某些底層的密碼工具 (underlying crypto primitives) 是安全的
  $\Rightarrow$ P $\neq$ NP
  $\Rightarrow$ 亂數/隨機 (randomness)，因為我們不知道真的亂數長什麼樣，所以無法驗證

## 1.3 Overview

**§ 什麼是密碼學？**

如果我們不在意安全，那麼我們不需要密碼學。
(If do not care security, we won’t need crypto.)

安全 (security) 可以由以下兩點來定義：

- 目的 (purposes)：我們需要達到什麼效果
- 需求 (requirements)：為了達到目的，我們需要達成哪些目標

一些密碼學相關的內容：

- 加密 (entryption)
- 數位簽章 (signature)
- 零知識 (zero knowledge)
- 安全計算 (secure computation)

## 1.4 Notations

**Private key encryption (or “secret key encryption”)**

就是對稱式加密，加密和解密皆使用同一個 key

**Public key encryption**

公鑰系統。一個公鑰會對應一個私鑰。公鑰會公開，私鑰不公開。
若 Alice 要傳訊息給 Bob，則 Alice 會使用自己的公鑰加密，並且讓 Bob 使用「與 Alice 的公鑰相對應的」私鑰進行解密。

**Zero knowledge**

A 想向 B 證明某件事情，但不想透漏任何其他的額外資訊。
Ex1：我想向你證明我有 100 萬，但不想真的放 100 萬現金在你眼前（以免被你搶走），所以我可以要求銀行開立證明來達到這個目的。Ex2: 我想向你證明我真的知道「威利在哪裡」。我可以用一張比原圖更大張的紙，並且在上面挖一個威利形狀的洞，以此來達到目的。

## 1.5 Story of solving impossibility

（這邊的例子經過一點點調整）
你的上司要求你解決一個問題 $Q$，並且告知你如果無法解決問題就會被炒魷魚，並被另一個比你聰明的傢伙取代。你雖然不知道怎麼解決 $Q$，但你知道另一個**相關的**知名問題 $\widetilde{Q}$ (Q tilde) 在現今根本就沒人會解。最後你告訴你的上司，由於「現在根本沒人知道如何解 $\widetilde{Q}$」，所以「也沒人會解 $Q$」，因此這問題解不了，而另一個自稱聰明的傢伙其實是騙子。

重點就是
If there’s a good algorithm for $Q$, then there exists a good one for another well-known problem $\widetilde{Q}$.
這句話的逆否命題就是
If there’s no algorithm for $\widetilde{Q}$, then there’s no algorithm for $Q$ either.

這背後的概念就是 reduction（就演算法的那個 reduction）。

## 1.6 Principle of modern crypto

**Kerckhoff’s principle**

「加密方法不能被要求是保密的，就算它落入敵人手中也不應該造成麻煩」
意即，整套加密方法的安全性只仰賴金鑰的保密。

（原文：It should not require secrecy, and it should not be a problem if it falls into enemy hands.）

**Principle of modern crypto**

1. Formal definition
    - System framework (model)：系統長什麼樣子
    - Security definition：如何定義安全

2. Precise assumption $\Pi'$
   通常會是已知難題
   從上一節的重點可以知道，我們通常會將加密法與某個已經被研究過的難題 (well-studied hardness) 做連結。若難題不是 well-studied，一來無法說服別人這個加密法安全，二來代表可能有人知道這個問題如何解決。

3. Construction $\Pi$
   加密法的步驟是什麼

4. Security proof
   基本上就是上一節的 reduction
   如果假象的攻擊者可以在 definition（即第一個要素）底下破解 $\Pi$，那麼我可以構造另一個攻擊者，使其破解已知難題 $\Pi'$。
   上面逆否命題的推論可以寫成：如果 $\Pi'$ 是安全的（意即不被破解），那麼 $\Pi$ 就是安全的。

加密系統 = 產生 key (key generation) + 加密 (encryption) + 解密 (decryption)

## 1.7 History of cryptography

### § Shift cipher

使用 private key encryption。
Key 是每個字母需要做 shift 的次數。

Key generation：選擇一個 $key \in \{0, 1, \ldots, 25\}$
Encryption：將每個字母對應的數字 shift $key$ 位
Decryption：將每個字母對應的數字**反方向** shift $key$ 位

破解：最多嘗試 26 次就可以找到答案

### § Substitution cipher

使用 private key encryption。

Key generation：將每個字母逐一對應到另一個字母，以此這個 mapping 作為 key

Encryption：將明文中的字母按照 key 逐一對應過去
Decryption：將密文中的字母按照 key 逐一對應回來

破解：字典攻擊（常用詞）+ 頻率分析（「E」在英文中出現的次數比較多）

加強：明文中不使用頻率較高的字母

## § Stronger cipher?

Vigenère cipher：設定偏移量為字母在明文中所在的位置。

DES (first published in 1975, and standardized in 1977)

AES

## § History about PKC

1974: Merkle proposed the notion
1976: Diffie-Hellman proposed the key exchange solution (Turing Awad 2015)
1977: Rivest-Shamir-Adleman proposed the first PKE (Turing Award 2002)
UK claimed their Government Communications Headquarters proposed such PKC idea before them.

Other impovements: ID-based encryption from Weil Pairing
使用了不同的 assumption，所以概念上較簡單，執行起來也較有效率（關於 ID-based 的概念，之後如果有時間，可能會提到）

# 2 L2: Perfect Secrecy

## 2.1 Encryption definition

三個 space：

- $\mathcal{M}$: message space
- $\mathcal{C}$: ciphertext space
- $\mathcal{K}$: key space

三種動作：

- Gen (key generation): probabilistic algorithm。
  $\mathrm{Gen}(1^\lambda) \to k \in \mathcal{K}$, where $\lambda$ is security parameter, or a symbol length (usually related to enc/dec execution time).

- Enc (encryption): probabilistic algorithm。
  For $m \in \mathcal{M}$, $\mathrm{Enc}_k(m) \to c \in \mathcal{C}$

- Dec (decryption): deterministic algorithm。
  For $c \in \mathcal{C}$, $\mathrm{Dec}_k(c) \coloneqq m \in \mathcal{M}$

注意上述使用 $\to$ 表示 probabilistic algorithm；使用 $\coloneqq$ 表示 deterministic algorithm。Probabilistic algorithm 就是每次執行都有可能產生不同結果，而 deterministic algorthm 則代表每次執行必定產生出相同結果。

正確性 (Correctness) 定義：

$$\Pr[\mathrm{Dec}_k(c) \coloneqq m : c \leftarrow \mathrm{Enc}_k(m), k \leftarrow \mathrm{Gen}(1^\lambda)] = 1$$

即由正確的金鑰一定可以成功進行解密。
對於某些系統，我們不一定會要求其機率是 1，可能會是接近 1 （即 $\approx 1$）

## 2.2 Notations

Distribution over $\mathcal{K}$：denoted as $\mathrm{dist}(\mathcal{K})$, which is defined by running $\mathrm{Gen}$, and taking the output key。
一個好的 key generation algorithm 應該要均勻地 (uniformly) 選擇 key（即選擇 key space 中的每個 key 的機率都是相等的）。因為如果我們有意地提高某些 key 的選擇機率，那麼攻擊者便可以藉由頻率分析知道我們的偏好，進而增加破解的機率。

$K$：a random variable, denoting the value of key generated by $Gen$.

$\Pr[K = k]$：for all $k \in \mathcal{K}$, it denotes the probability that the key generated by $\mathrm{Gen}$ is equal to $k$.

上面三項皆可以套用至明文（$\mathrm{dist}(\mathcal{M})$、$M$、$\Pr[M = m]$）和密文（$\mathrm{dist}(\mathcal{C})$、$C$、$\Pr[C = c]$）。

當我們固定一個 encryption scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ 且 dist over $\mathcal{M}$，這就可以根據所給定的 $k \in \mathcal{K}$ 和 $m \in \mathcal{M}$，確定 $\mathrm{dist}(\mathcal{C})$。

## 2.3 Examples of notations

### § Example 1

一個 adversary A 知道訊息是「attack today」的機率是 70%、「not attack」的機率是 30%，所以

$$\Pr[M = \text{A.T.}] = 0.7, \quad \Pr[M = \text{N.A.}] = 0.3$$

Random variables $K$ 和 $M$ 會假設沒有關係 (independent)。因為 $\text{dist}(\mathcal{K})$ 由 $\text{Gen}$ 決定，而 $\text{dist}(M)$ 由我們想要加密的 context 決定。

### § Example 2 - Shift cipher

$K = \{0, 1, 2, \ldots, 25\}$ with $\Pr[K = k] = \dfrac{1}{26}$ (aka uniformly distributed).

Let distribution of $\mathcal{M}$

$$\text{dist}(\mathcal{M}) = \begin{cases} \Pr[M = \text{'a'}] = 0.7 \\ \Pr[M = \text{'z'}] = 0.3 \end{cases}$$

Then

$$\begin{aligned} \Pr[C = \text{'b'}] &= \Pr[M = \text{'a'} \wedge K = 1] + \Pr[M = \text{'z'} \wedge K = 2] \\ &= \Pr[M = \text{'a'}] \cdot \Pr[K = 1] + \Pr[M = \text{'z'}] \cdot \Pr[K = 2] \quad \text{(By independence)} \\ &= 0.7 \cdot \frac{1}{26} + 0.3 \cdot \frac{1}{26} \\ &= \frac{1}{26} \end{aligned}$$

Condition probability

$$\begin{aligned} \Pr[M = \text{'a'} \mid C = \text{'b'}] &= \frac{\Pr[C = \text{'b'} \mid M = \text{'a'}] \cdot \Pr[M = \text{'a'}]}{\Pr[C = \text{'b'}]} \\ &= \frac{\frac{1}{26} \cdot 0.7}{\frac{1}{26}} \\ &= 0.7 \end{aligned}$$

where $\Pr[C = \text{'b'} \mid M = \text{'a'}]$ iff. $K = 1$, and $\Pr[K = 1] = \dfrac{1}{26}$

[Bayes' theorem]

$$\Pr[A \mid B] = \frac{\Pr[B \mid A] \cdot \Pr[A]}{\Pr[B]} \qquad \text{if} \quad \Pr[B] \neq 0$$

## 2.4 Intuition for security

Adversary 通常在收發兩端的中間進行竊聽 (eavesdrop)。
Adversary 知道 $\text{dist}(\mathcal{M})$ 和 encryption scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$，而不知道 key。

A scheme $\Pi$ meets **perfect secrecy** means observation (usually from adversary) on ciphertext $c$ should give no additional infomation.
意即密文 $c$ 不能給攻擊者有更多的資訊可以更準確地進行猜測，也可以說 $c$ 不會洩漏更多的資訊。

## 2.5 Perfect secrecy

**Formal definition of perfect secrecy (Definition 1)**

An encrytion scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ with message space $\mathcal{M}$ is perfect secrecy if for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$ and every chiphertext $c \in \mathcal{C}$ for $\Pr[C = c] > 0$

$$\Pr[M = m \mid C = c] = \Pr[M = m]$$

簡單來說，就是在觀察 $c$ 之後，所得知的 $\mathrm{dist}(\mathcal{M})$ 與在觀察 $c$ 之前相等。
若 $c$ 洩漏了某些資訊，則上式中的等號 (=) 應該改成大於符號 (>)。

**Example: shift cipher**

這邊用和前面一樣的例子：

$$\begin{aligned}
\Pr[C = \text{'b'}] &= \Pr[M = \text{'a'} \land K = 1] + \Pr[M = \text{'z'} \land K = 2] \\
&= \Pr[M = \text{'a'}] \cdot \Pr[K = 1] + \Pr[M = \text{'z'}] \cdot \Pr[K = 2] \quad \text{(By independence)} \\
&= 0.7 \cdot \frac{1}{26} + 0.3 \cdot \frac{1}{26} \\
&= \frac{1}{26}
\end{aligned}$$

$$\begin{aligned}
\Pr[M = \text{'a'} \mid C = \text{'b'}] &= \frac{\Pr[C = \text{'b'} \mid M = \text{'a'}] \cdot \Pr[M = \text{'a'}]}{\Pr[C = \text{'b'}]} \\
&= \frac{\frac{1}{26} \cdot 0.7}{\frac{1}{26}} \\
&= 0.7 \\
&= \Pr[M = \text{'a'}]
\end{aligned}$$

由此可知，shift cipher 是 prefect secrecy。

# 3 L3

## 3.1 Perfect secrecy II

### Formal definition of perfect secrecy (Definition 2)

For every $m, m' \in \mathcal{M}$ and every $c \in \mathcal{C}$,

$$\Pr[\mathrm{Enc}_K(m) = c] = \Pr[\mathrm{Enc}_K(m') = c]$$

### Example: shift cipher

$\Pr[M = \text{'a'}] = 0.7$
$\Pr[M = \text{'z'}] = 0.3$

Let $m = \text{'a'}$, and $m' = \text{'z'}$.
Then

$$\Pr[\mathrm{Enc}_K(\text{'a'}) = \text{'b'}] = \frac{1}{26} = \Pr[\mathrm{Enc}_K(\text{'z'}) = \text{'b'}]$$

(For further explanantion, if $\mathrm{Enc}_K(\text{'a'}) = \text{'b'}$, $K$ must be $1$, where probability is $\frac{1}{26}$; similarly, if $\mathrm{Enc}_K(\text{'z'}) = \text{'b'}$, $K$ must be $2$. That's why their probabilities are same.)

### Lemma

An encryption scheme $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ with message space is perfectly secret (which means $\Pi$ satisfies Def. 1), the above equation (which is Def. 2) holds for every $m, m' \in \mathcal{M}$ and every $c \in \mathcal{C}$.

意即 Def. 1 等價 (equivalent) 於 Def. 2.

***Proof*** (Proof from Def. 2 to Def. 1)

Fix a $\mathrm{dist}(\mathcal{M})$, a message $m$ and a ciphertext $c$ for which $\Pr[C = c] > 0$.
If $\Pr[M = m] = 0$, then $\Pr[M = m \mid C = c] = \Pr[M = m]$. It always holds.
If $\Pr[M = m] > 0$:

(i) $\Pr[C = c \mid M = m] = \Pr[\mathrm{Enc}_K(M) = c \mid M = m] = \Pr[\mathrm{Enc}_K(m) = c] = \alpha$

(ii) For every $m' \in \mathcal{M}$,

$$\Pr[C = c \mid M = m'] = \Pr[\mathrm{Enc}_K(M) = c \mid M = m'] = \Pr[\mathrm{Enc}_K(m') = c] = \alpha$$

(iii) By Bayes' Theorem,

$$\begin{aligned}
\Pr[M = m \mid C = c] &= \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{\Pr[C = c]} \\
&= \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{\sum_{m' \in \mathcal{M}} \Pr[C = c \mid M = m'] \cdot \Pr[M = m']} && \text{(by (i) and (ii))} \\
&= \frac{\alpha \cdot \Pr[M = m]}{\sum_{m' \in \mathcal{M}} \ \alpha \cdot \Pr[M = m']} \\
&= \frac{\alpha \cdot \Pr[M = m]}{\alpha \cdot \sum_{m' \in \mathcal{M}} \ \Pr[M = m']} \\
&= \frac{\cancel{\alpha} \cdot \Pr[M = m]}{\cancel{\alpha} \cdot \sum_{m' \in \mathcal{M}} \ \Pr[M = m']} \\
&= \Pr[M = m]
\end{aligned}$$

□

***Proof*** (Proof from Def. 1 to Def. 2 (Quiz))

Fix a $\mathrm{dist}(\mathcal{M})$, a message $m$ and a ciphertext $c$ for which $\Pr[C = c] > 0$.
If $\Pr[C = c] = 0$, then $\Pr[C = c \mid M = m] = \Pr[C = c \mid M = m'] = 0$. It always holds.
If $\Pr[C = c] > 0$:
(i) For $\Pr[\mathrm{Enc}_K(m) = c]$,

$$\begin{aligned}
\Pr[\mathrm{Enc}_K(m) = c] &= Pr[C = c \mid M = m] \\
&= \frac{\Pr[M = m \mid C = c] \cdot \Pr[C = c]}{\Pr[M = m]} \\
&= \frac{\Pr[M = m] \cdot \Pr[C = c]}{\Pr[M = m]} && \text{(by Def. 1)} \\
&= \frac{\cancel{\Pr[M = m]} \cdot \Pr[C = c]}{\cancel{\Pr[M = m]}} \\
&= \Pr[C = c]
\end{aligned}$$

(ii) For $\Pr[\mathrm{Enc}_K(m') = c]$,

$$\begin{aligned}
\Pr[\mathrm{Enc}_K(m) = c] &= Pr[C = c \mid M = m'] \\
&= \frac{\Pr[M = m' \mid C = c] \cdot \Pr[C = c]}{\Pr[M = m']} \\
&= \frac{\Pr[M = m'] \cdot \Pr[C = c]}{\Pr[M = m']} && \text{(by Def. 1)} \\
&= \frac{\cancel{\Pr[M = m']} \cdot \Pr[C = c]}{\cancel{\Pr[M = m']}} \\
&= \Pr[C = c]
\end{aligned}$$

From (i) and (ii), we know that

$$\Pr[\mathrm{Enc}_K(m) = c] = \Pr[\mathrm{Enc}_K(m') = c]$$

□

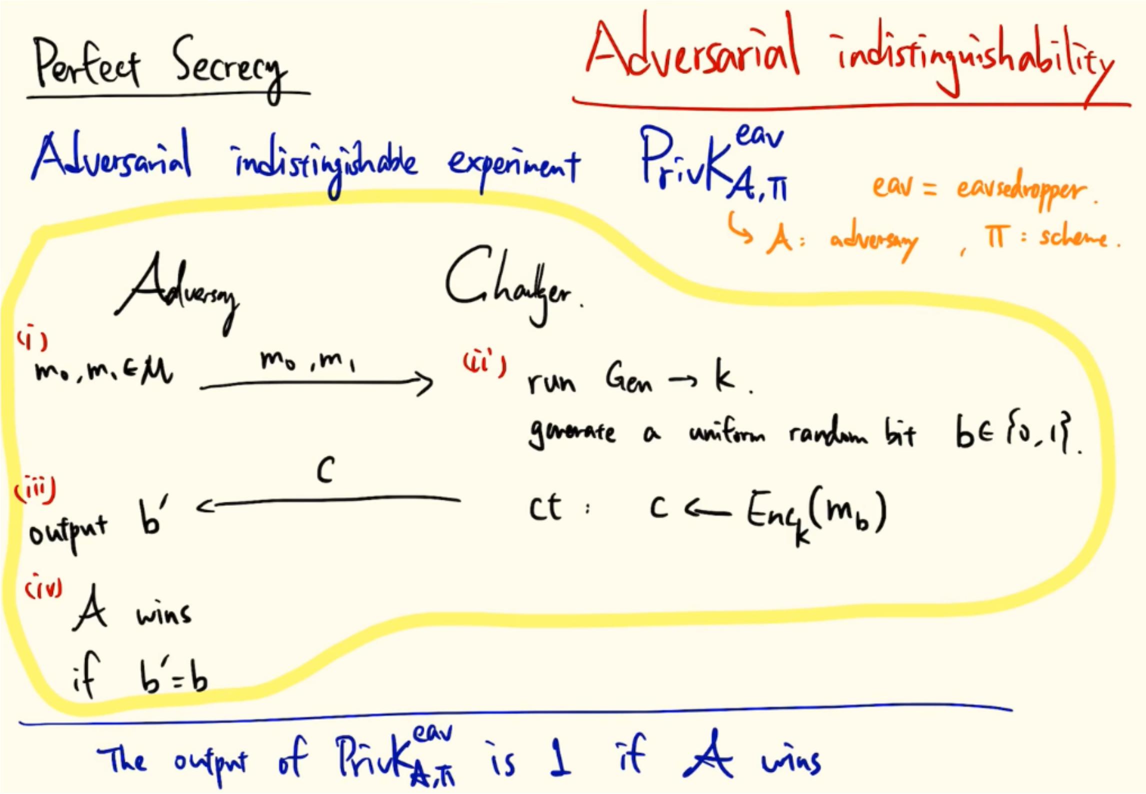
## 3.2 Perfect secrecy III

### Adversarial indistinguishability

Adversarial indistinguishable experiment

$$PrivK^{eav}_{A,\Pi}$$

其中 $A$ 代表 adversary, $\Pi$ 代表 scheme, and $eav$ 代表 eavesdropper.




這個 experiment 有兩個人：adversary 和 Challenger。

- Step 1： Adversary 會從 message space 中選出兩份訊息 $m_0$ 和 $m_1$，並這兩份訊息發送給 Challenger。
- Step 2： Challenger 會執行 key generation algorithm $Gen$ 來產生 key $k$，並 generate 一個 uniform random bit $b \in \{0, 1\}$。最後產生出 ciphertext $c \leftarrow \mathrm{Enc}_k(m_b)$，再將 $c$ 回傳給 adversary。
- Step 3： Adversary 會 output 一個 $b'$ 來代表它猜測 $b$ 的結果。
- Step 4： 若 $b' = b$，則 adversary 成功猜對了。

這個 experiment $PrivK^{eav}_{\mathrm{A},\Pi}$ 的 output 就是 adversary 是否猜對；也可以說，當 $PrivK^{eav}_{\mathrm{A},\Pi} = 1$，則 $b' = b$。

**Formal definition of perfect secrecy (Definition 3, defined by perfect indistinguishability)**

$\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ with message space $\mathcal{M}$ is perfectly indistinguishable if for every adversary $A$, it holds

$$\Pr[PrivK^{eav}_{\mathrm{A},\Pi} = 1] = \frac{1}{2}$$

意思：猜中的機率為 $\frac{1}{2}$，和沒有 $c$ 的前提下，隨便亂猜的機率（即 $\Pr[(\text{randomly output } b') \wedge (b' = b)] = \frac{1}{2}$）是一樣的。代表 $c$ 並沒有洩漏任何額外資訊。

這個命題和 $\Pr[PrivK^{eav}_{\mathrm{A},\Pi} = 0] = \frac{1}{2}$ 是等價的。

注意：若 $\Pr[PrivK^{eav}_{\mathrm{A},\Pi} = 1] < \frac{1}{2}$ 並不代表攻擊者更不會猜。因為 $\Pr[PrivK^{eav}_{\mathrm{A},\Pi} = 1] + \Pr[PrivK^{eav}_{\mathrm{A},\Pi} = 0] = 1$，所以 $\Pr[PrivK^{eav}_{\mathrm{A},\Pi} = 0] > \frac{1}{2}$。因此猜另一種情況的正確機率會更高。

**Lemma**

$\Pi$ is perfectly secret if and only if it is perfectly indistinguishable.

***Proof*** (Proof from Def. 2 to Def. )

由 Def. 2 可知

$$\Pr[\text{Enc}_K(m_0) = c] = \Pr[\text{Enc}_K(m_1) = c]$$

又因為 $c \leftarrow \text{Enc}_k(m_b)$，所以

$$\begin{aligned}\Pr[\text{Enc}_K(m_0) = c] = \Pr[b = 0] \\ \Pr[\text{Enc}_K(m_1) = c] = \Pr[b = 1]\end{aligned}$$

因此 $\Pr[b = 0] = \Pr[b = 1] = \frac{1}{2}$ ( 因為在本例中 $\Pr[b = 0] + \Pr[b = 1] = 1$ )。

$$\begin{aligned}
\Pr[PrivK^{eav}_{A,\Pi}] &= \Pr[b' = b] \\
&= \Pr[b' = b \wedge b = 0] + \Pr[b' = b \wedge b = 1] && \text{(rewrite)} \\
&= \Pr[b' = b \mid b = 0] \times \Pr[b = 0] + \Pr[b' = b \mid b = 1] \times \Pr[b = 1] && \text{(rewrite)} \\
&= \Pr[b' = 0] \times \Pr[b = 0] + \Pr[b' = 1] \times \Pr[b = 1] && \text{(rewrite)} \\
&= \Pr[b' = 0] \times \frac{1}{2} + \Pr[b' = 1] \times \frac{1}{2} && \text{(by Def. 2 denoted above)} \\
&= \frac{1}{2}(\Pr[b' = 0] + \Pr[b' = 1]) \\
&= \frac{1}{2} && (\because \Pr[b' = 0] + \Pr[b' = 1] = 1)
\end{aligned}$$

□

**Proof of Def. 3 to Def. 2 (Bonus)**

欲證 Def. 3 ( $\Pr[PrivK^{eav}_{A,\Pi} = 1] = \frac{1}{2}$ ) ⇒ Def. 2 ( $\Pr[\text{Enc}_K(m) = c] = \Pr[\text{Enc}_K(m') = c]$ )

***Proof*** (Prove by contraposition)

□

## 3.3 One-Time Pad (OTP)

**Construction of OTP**

Fix an integer $l > 0$, and let $|\mathcal{M}| = |\mathcal{C}| = |\mathcal{K}| = l$.
(whch means all are binary strings of length $l$, i.e., $\{0,1\}^l$)

Key generation algorithm $\text{Gen}$: uniformly randomly chooses a key $k \in \mathcal{K}$, $k$ is $l$-bit key.

Encryption algorithm $\text{Enc}$: given $k \in \{0,1\}^l$ and a message $m \in \{0,1\}^l$, $\text{Enc}$ outputs a ciphertext $c = m \oplus k$.

Decryption algorithm $\text{Dec}$: given $k, c$, $\text{Dec}$ outputs message $m = c \oplus k$.

**Prove that OTP is perfectly secret**

***Proof*** (Proved by Def. 1)

(i) For an arbitrary $c \in \mathcal{C}$ and $m \in \mathcal{M}$

$$\Pr[C = c \mid M = m] = \Pr[\mathrm{Enc}_K(m) = c] = \Pr[m \oplus K = c] = \Pr[K = m \oplus c] = \frac{1}{2^l}$$

(ii) Fix any $\mathrm{dist}(\mathcal{M})$, for any $c \in \mathcal{C}$

$$\begin{aligned}
\Pr[C = c] &= \sum_{m' \in \mathcal{M}} \Pr[C = c \mid M = m'] \cdot \Pr[M = m'] \\
&= \sum_{m' \in \mathcal{M}} \frac{1}{2^l} \cdot \Pr[M = m'] \\
&= 2^{-l} (\sum_{m' \in \mathcal{M}} \Pr[M = m']) \\
&= 2^{-l}
\end{aligned}$$

(iii)

$$\begin{aligned}
\Pr[M = m \mid C = c] &= \frac{\Pr[C = c \mid M = m] \cdot \Pr[M = m]}{Pr[C = c]} \\
&= \frac{2^{-l} \cdot \Pr[M = m]}{2^{-l}} \\
&= \Pr[M = m]
\end{aligned}$$

$\square$

# 4 L4

## 4.1 Limitation of Perfect Secrecy

**Theorem 1** (Limitation of perfect secrecy)
If $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$ and key space $\mathcal{K}$, then

$$|\mathcal{M}| \leq |\mathcal{K}|$$

***Proof***

Suppose $|\mathcal{K}| < |\mathcal{M}|$, $\Pi$ cannot be perfectly secret.

Consider the uniform $\mathrm{dist}(\mathcal{M})$ and fix $c \in \mathcal{C}$, $\Pr[C = c] = 0$.

Let $\mathcal{M}(c)$ be the set of possible message which contains all possible messages decrypted by $c$. That is,

$$\mathcal{M}(c) \stackrel{\text{def}}{=} \{m \mid m = \mathrm{Dec}_K(c) \text{ for some } k \in \mathcal{K}\}$$

$\mathrm{Dec}$ is deterministic function, so $|\mathcal{M}(c)| \leq |\mathcal{K}|$.
(We know $\mathrm{Dec}_k(c) := m$, and different values of $k$ may map to the same $m$. If all $m$ are distinct for different $k$, then equation holds; otherwise, $|\mathcal{M}(c)| < |\mathcal{K}|$.)

If $|\mathcal{K}| < |\mathcal{M}|$ and $\mathcal{M}(c) \leq |\mathcal{K}|$, there exist some $m' \in \mathcal{M}$ but $m' \notin \mathcal{M}(c)$.
$\Rightarrow \Pr[M = m' \mid C = c] = 0 \neq \Pr[M = m']$, which is not perfect secrecy. $\square$

**Quiz**

We know that it's impossible to achieve pefect secrecy with shorter key size. So, what can we do or modify some factors to achieve shorter key? Any tradeoff (factor)?

**§ Shannon's Theorem**

**Theorem 2** (Shannon's theorem)
Let $\Pi = (\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$ be an encryption scheme with message space $M$ for which $|\mathcal{M}| = |\mathcal{K}| = |\mathcal{C}|$.
The scheme is perfectly secret if and only if:

1. Every key $k \in \mathcal{K}$ is chosen with probability $\dfrac{1}{|\mathcal{K}|}$ by $\mathrm{Gen}$
2. For every $m \in \mathcal{M}$ and every $c \in \mathcal{C}$, there exists a unique key $k \in \mathcal{K}$ such that $\mathrm{Enc}_k(m)$.

**Quiz**

Design a tricky scheme $\Pi$ that $k \in \mathcal{K}$ is **NOT** uniformly chosen. Show $\Pi$ is **NOT** perfectly secret by using Definition 1, 2 or 3.
(Hint: modify shift cipher or one-time pad)

## 4.2 Private Key Encryption

### § Computational Security

Perfect secrecy 的缺點 (weakness)：

- 只能用一次 (one time use)
- key 的長度一定要大於訊息的長度 ($|\mathcal{K}| \geq |\mathcal{M}|$)

Computational security 是從計算上保證安全的一種安全性。它不像 pefect secrecy 那樣地完美，但可以更靈活地建立 scheme（如減少 key 的長度）。

從 adversary 的觀點來看：

| Adversary's power | time/space | success probability |
|---|---|---|
| Perfect secrecy | unbounded | = random guess |
| Computational security | polynomial time | = random guess + small probability |

目的：減少安全性，來換取更好的效率 (by weakening the security, to achieve better efficiency)。

### § Concrete Definition

**Definition 1** (Concrete definition)
A scheme $\Pi$ is $(t, \epsilon)$-secure if any adversary $A$ running for time at most $t$, succeeds in breaking $\Pi$ with probability at most $\epsilon$.

Ex: $t = 2^{10}$, $\epsilon = \dfrac{1}{2^{100}}$

### § Asymptotic Definition

在這裡的 adversary $A$ 的能力 (power) 是以漸進式術語來定義的 (asymptotic setting)：

- Efficient adversary：這種 adversary 會執行可以在 polynomial time 內跑完的演算法。這種演算法的執行時間是 $p(n)$，其中 $p$ 為多項式集合，而 $n$ 為安全參數 (security parameter)。
- Small probability of success: 成功機率小於任何 polynomial 的倒數。也就是

$$\Pr[\text{success}] < \frac{1}{p(n)}, \text{ where } p \text{ is arbitary polynomial}$$

PPT = Probabilistic Polynomial Time

**Definition 2** (Asymptotic definition)
A scheme is secure if for any PPT adversary succeeds in breaking the scheme with at most **negligible** probability.

### § Negligible Probability

Negligible function 是漸進小於 (asymptotic smaller) 任何 polynomial function 的函數。

**Definition 3**
A function $f$ is negligible if
for every positive polynomial $p$, there exists a number $N$ such that $f(n) < \frac{1}{p(n)}$ where $n > N$.

Example:
Let $g(x) = \frac{1}{2^x}$.
There exists $N$ such that $g(n) < \frac{1}{p(n)}$.

$$
\begin{aligned}
& g(n) < \frac{1}{p(n)} \\
\Rightarrow \quad & \frac{1}{2^n} < \frac{1}{n^k} && \text{(k is positive constant)} \\
\Rightarrow \quad & 2^n > n^k \\
\Rightarrow \quad & n > k \cdot log_2(n) \\
\Rightarrow \quad & \frac{n}{log_2(n)} > k
\end{aligned}
$$

If $n > k^2$, this inequality holds.

**Quiz**
Let $\mathrm{negl}(x), \mathrm{negl}'(x)$ be negligible functions.

1. A function $f_1$, defined by $f_1(x) = \mathrm{negl}(x) + \mathrm{negl}'(x)$
2. A function $f_2$, defined by $f_2(x) = p(x) \cdot \mathrm{negl}(x)$, where $p(x)$ is positive polynomial.

Are $f_1$ and $f_2$ are still negligible functions? **Yes**

**Summary**
任何關於 computational security 的 security definition 都由下列組成：
1. 破解 scheme 的定義（也就是怎麼樣才叫 scheme 被破解了）
2. 關於 adversary 的能力

我們通常將 adversary 塑造 (model) 成有效率（有計算能力）的演算法，且只考慮 adversary 可以在 polynomial time 之內執行的 probabilistic stratigies。

**Definition 4**
A scheme is secure if for every PPT adversary $A$ carrying out an attack of some formally specified attack type, and the probability that $A$ succeeds is negligible.

**§ Private Key Encryption**

**Definition 5** (Private key encryption)
A private key encryption is a tuple of PPT algorithm $(\mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec})$

- Key generation: $\mathrm{Gen}(1^k) \to k$. 這裡 $n$ 的意義是 $|\mathcal{K}| \geq n$ 或 $|\mathcal{K}| = \mathrm{poly}(n)$。
- Encryption: $\mathrm{Enc}_k(m) \to c$, where key $k$ and $m \in \{0,1\}*$ are inputs. 若 $m \in \{0,1\}^{l(n)}$，我們會稱這個等式為 fixed-length private key encryption with message length $l(n)$。
- Decryption: $\mathrm{Dec}_k(c) \coloneqq m$. If $c$ cannot be decrypted, then outupt $\perp$ (error).

**Basic definition of security**

Eavesdropping ( 竊聽 ): adversary 的策略或能力

這裡和之前的 $PrivK_{A,\Pi}^{eav}$ 大致一樣，參見 3.2 Perfect secrecy III。

差異：

- Perfect secrecy：沒有 security parameter，因為不在意 adversary 有多少的能力

$$\Pr[PrivK_{A,\Pi}^{eav} = 1] = \frac{1}{2}$$

- Computational security：有 security parameter $n$

$$\Pr[PrivK_{A,\Pi}^{eav} = 1] \leq \frac{1}{2} + \mathrm{negl}(n)$$

# 5 L5

## 5.1 Basics

### § Scenario

Sender S 和 receiver R 彼此有有一把相同的 key $k$，且 S 想要發送訊息給 R。
在發送訊息前，S 會先使用 $k$ 將明文 $m$ 加密為密文 $c$（$c \leftarrow \mathrm{Enc}_k(m)$），之後 S 將 $c$ 傳送給 R。
R 在收到 $c$ 後，使用同一把 key $k$ 將 $c$ 解密（$m := \mathrm{Dec}_k(c)$）來得到 $m$。

關於這個 scenario 的正式的定義可以參見 Definition 5 Private key encryption。

### § 安全性定義

使用前面提到的 $PrivK^{eav}_{A,\Pi}$，參見 3.2 Perfect secrecy III。

## 5.2 EAV-security

EAV = eavesdropping

**Definition 6** (EAV-secruity of private key encryption)
A private key encryption scheme $\Pi$ is **EAV-secure** if for all PPT adversary $A$, there is a negligible function $\mathrm{negl}$ such that for all $n$,

$$\Pr[PrivK^{eav}_{A,\Pi}(n) = 1] \leq \frac{1}{2} + \mathrm{negl}(n)$$

(The probability is taken over randomness used by adversary and used in experiment.)

### § Equivalent Formulation of EAV-security

前一節 EAV-security 的定義等價於下面這句話：
「無論 PPT adversary $A$ 看到由 $m_0$ 或 $m_1$ 加密過後的密文，其表現都相同。」
( Every PPT adversary behaves the same whether it sees ciphertext of $m_0$ or $m_1$. )

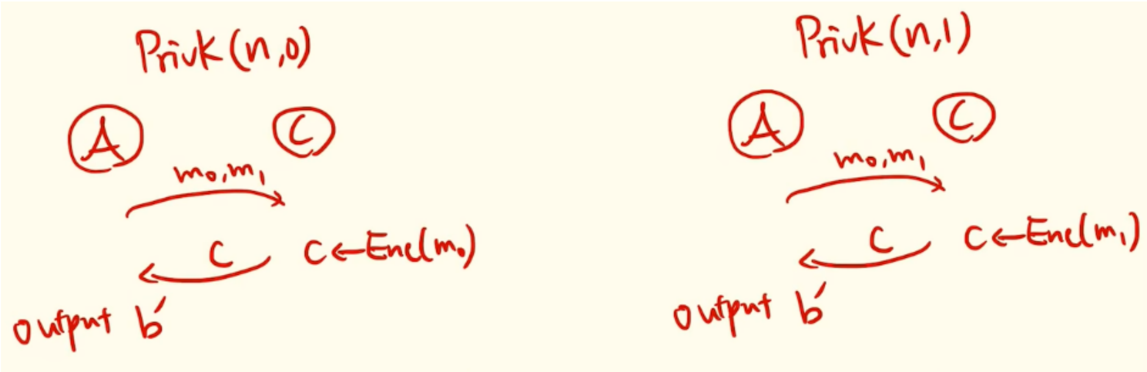
更精確的定義是：

- 修改之前的定義為 $PrivK^{eav}_{A,\Pi}(n, b)$，其定義都和之前一樣，除了 $b$ 是固定的，而不是隨機選擇的。
- 定義 $out_A(PrivK^{eav}_{A,\Pi}(n, b)) = b'$，其中 $b'$ 是 $A$ 的 output。
- 沒有 PPT adversary $A$ 可以知道現在是 experiment $PrivK^{eav}_{A,\Pi}(n, b)$ 或 $PrivK^{eav}_{A,\Pi}(n, b)$。

正式定義如下：

**Definition 7** (Equivalent formulation of EAV-security)
$\Pi$ is EAV-secure if for all PPT adversary $A$, there is a negligible function $\mathrm{negl}$ such that

$$|\Pr[out_A(PrivK^{eav}_{A,\Pi}(n, 0)) = 1] - \Pr[out_A(PrivK^{eav}_{A,\Pi}(n, 1)) = 1] \leq \mathrm{negl}(n)$$




**Quiz**

In $PrivK$, we define $A$ to choose two messages with the same length.
Please write your thought for the impossibility to support arbitrary-length messages.

## 5.3 Private Key Encryption

### § Pseudorandom Generator

**Definition 8** (pseduorandom generator, PRG)
Let $l$ be a polynomial and $G$ is a deterministic polynomial-time algorithm. For any $n$ and input $s \in \{0,1\}^n$, the output of $G(s)$ is $l(n)$-length.
We say $G$ is a PRG if:

- Exapnsion: for every $n$, it holds $l(n) > n$. $l$ is a so-called expansion factor of $G$.
- Pseudorandomness: for any PPT algorithm $D$ (aka distinguisher), there is a negligible function $\mathrm{negl}$ such that
$$|\Pr[D(G(s)) = 1] - \Pr[D(r) = 1]| \ \leq \ \mathrm{negl}(n)$$
where $s \in \{0,1\}^n$ and $r \in \{0,1\}^{l(n)}$ is a turly random variable.

### § PRG-based Construction of Fixed-length Private Key Encryption

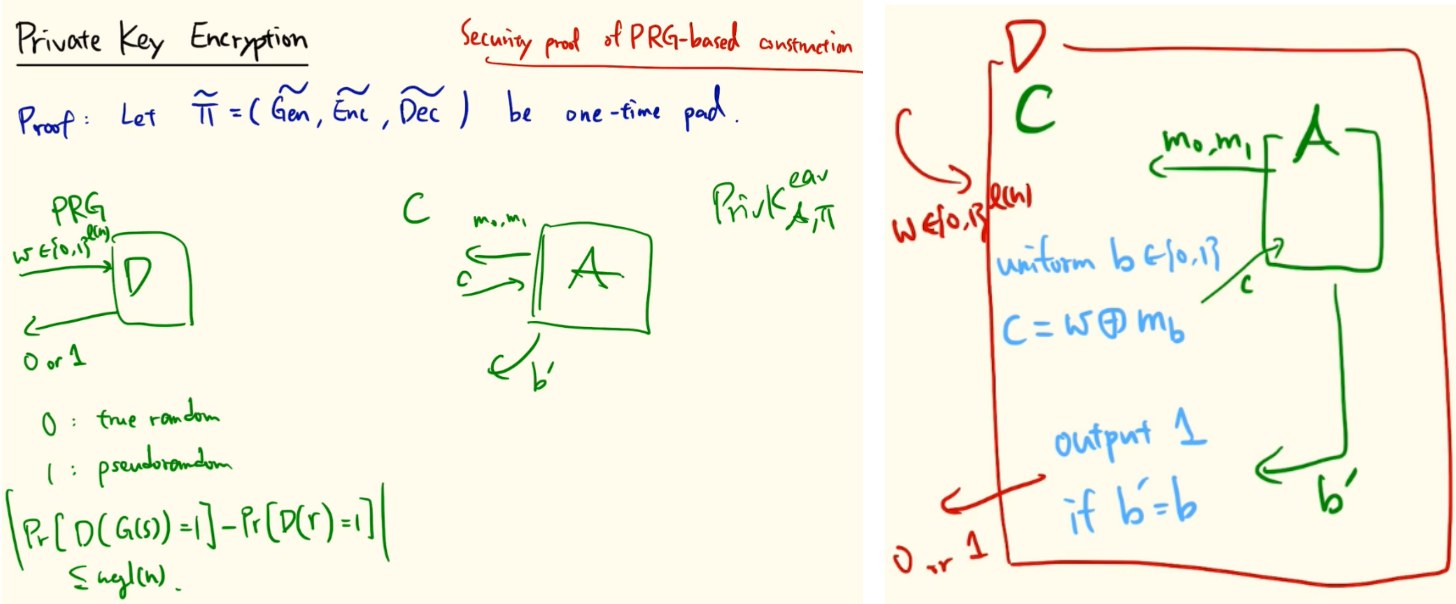
Let $G$ be a PRG with expansion factor $l$.
Scheme $\Pi = \mathrm{Gen}, \mathrm{Enc}, \mathrm{Dec}$.

- $\mathrm{Gen}(1^n)$: on input $1^n$, choose uniform $k \in \{0,1\}^n$.
- $\mathrm{Enc}(k,m)$: with input of a message $m \in \{0,1\}^{l(n)}$ and outputs a ciphertext $c = G(k) \oplus m$
- $\mathrm{Dec}(k,c)$: with input of a ciphertext $c \in \{0,1\}^{l(n)}$ and outputs a message $m = G(k) \oplus c$

這種構造法和 OTP（見 3.3 One-Time Pad (OTP)）很像。那時候的 OTP 會遇到 pefect secrecy 的限制，也就是 key 的長度至少要和 message 一樣長（$|\mathcal{K}| \geq |\mathcal{M}|$）。在這裡，我們通過 PRG 來將原本的 key 長度 $n$ 擴展成 $l(n)$，藉此來降低 key 的長度。而其代價就是，這種使用 PRG 的方法一定不是 perfect secrecy。

P.S. 由於 private key encryption 要求雙方要事先使用安全通道交換同一把 key。若在這種情景下使用和 message 一樣長的 key，那我們就可以直接使用這個安全通道交換訊息本身了，而無需進行加密。

### § PRG-based construction is EAV-secure




(a) Distinguisher $D$ and adversary $A$

(b) Reduction in proof

**Theorem 3**
If $G$ is a pseudorandom generator, then the construction $\Pi$ is a EAV-secure.

其逆否命題為「如果 $\Pi$ 不是 EAV-secure，則 $G$ 也不是 PRG」。

**證明思路**
由 $D$ 扮演 challenger。
在 reduction 時是 $D$ 包在 $A$ 的外面。

Let $\tilde{\Pi} = (\widetilde{\mathrm{Gen}}, \widetilde{\mathrm{Enc}}, \widetilde{\mathrm{Dec}})$ be one-time pad.

1. If $w$ is uniform chosen form $\{0,1\}^{l(n)}$,
$$\Pr[D(w)=1] = \Pr[PrivK^{eav}_{A,\tilde{\Pi}}(n)=1] = \frac{1}{2}$$
這種情況是 one-time pad 的情況，也就是使用 true randomness。
2. If $w = G(k)$ by choosing uniform $k \in \{0,1\}^n$,
$$\Pr[D(G(k))=1] = \Pr[PrivK^{eav}_{A,\Pi}(n)=1]$$
這種情況是使用 pseudorandomness。
這個機率是我們所要證明的，可以透過第三點來反推其機率為 $\leq \frac{1}{2} + \mathrm{negl}(n)$
3. If $G$ is PRG,
$$|\Pr[D(G(k))=1] - \Pr[D(w)=1]| \ \leq \ \mathrm{negl}(n)$$

**Proof details**
Let $A$ be a PPT adversary. Our goal is to contract a distinguisher $D$ (which is going to break PRG) that takes a string $w$ as input.

Goal of $D$: determine whether
(i) $w$ was chosen uniformly (where $w \in \{0,1\}^{l(n)}$)
(ii) $w$ was generated by choosing uniform $k \in \{0,1\}^n$ and computing $w = G(k)$ (where $w \in \{0,1\}^{l(n)}$ and $l(n) > n$)

Output of $D$: outputs 1 if case (i) mentioned above; otherwise, outputs 0

Theorem used:

$$|\Pr[D(r)=1]-\Pr[D(G(k))=1]| \ \leq \ \mathrm{negl}(n)$$

where $r \leftarrow \{0,1\}^{l(n)}$, and $k \leftarrow \{0,1\}^n$.

Activites of $D$: (connect $A$ and $D$)
Emulate the eav experiment $PrivK_{A,\Pi}^{eav}$ for $A$

- If $A$ wins, $D$ thinks $w := G(k)$.
- If $A$ fails, $D$ thinks $w$ is uniform chosen.

***Proof***

(Refer to figure Reduction in proof)
Distinguisher $D$ get an input of a string $w \in \{0,1\}^{l(n)}$.

Step 1 : Run $A$ to obtain a pair of messages $m_0, m_1 \in \{0,1\}^{l(n)}$

Step 2 : Choose a uniform bit $b \in 0, 1$. Set $c = w \oplus m_b$

Step 3 : Send $c$ to $A$

Step 4 : Later, $A$ returns $b'$

$D$ outputs

- 1, if $b' = b$
- 0, if $b' \neq b$

Note that probability of output of $D$ is related to $\Pr[PrivK_{A,\Pi}^{eav}]$.
If $\Pr[PrivK_{A,\Pi}^{eav}] > \dfrac{1}{2} + \mathrm{negl}$,

$$\Pr[out_D = 1] > \frac{1}{2} + \mathrm{negl}$$
$$\Pr[out_D = 0] \leq \frac{1}{2} - \mathrm{negl}$$

□

## 5.4 Chosen Plaintext attack & CPA-security

CPA = Chosen Plaintext Attack

### § CPA security

在這個情景下的 adversary $A$ 可以存取 encryption oracle。

Encryption oracle：是一個黑盒子，我們不知道其運作原理，但給它輸入和取得它的輸出。$A$ 可以將明文 $m$ 給 oracle，之後 oracle 會將明文加密為密文 $c \leftarrow \mathrm{Enc}_k(m)$ 回傳給 $A$。

**Experiment** $PrivK_{A,\Pi}^{cpa}$

Step 1： $A$ 可以選擇明文 $m_i$ 給 $C$

Step 2： $C$ 建立密鑰 $k \leftarrow \mathrm{Gen}(1^n)$，並將明文加密為密文 $c_i \leftarrow \mathrm{Enc}\ \ m_i)$ 回傳給 $A$。

Step 3： $A$ 此時可以將這些收集到明文-密文對（plaintext-ciphertext pair）儲存起來。由於 $A$ 是 PPT adversary，所以 $A$ 可以收集的 pair 數為 poly-many。

Step 4： $A$ 選擇 $m_0$ 和 $m_1$ 傳給 $C$ 進行 chanllenge。之後的事情都和之前的 EAV-secure 的 experiment 一樣。

Step 5： 若 $A$ 贏了，則 $PrivK^{cpa}_{A,\Pi}(1^n) = 1$。

P.S. 前三步稱為 encryption oracle query。而 challenge 之後一樣可以進行 eneryption oracle query，直到 $A$ output $b'$。

**Quiz**

Show PRG-based construction $\Pi$ is not CPA-secure.
(Hint: give $A$ in $PrivK^{cpa}_{A,\Pi}$ to break $\Pi$)