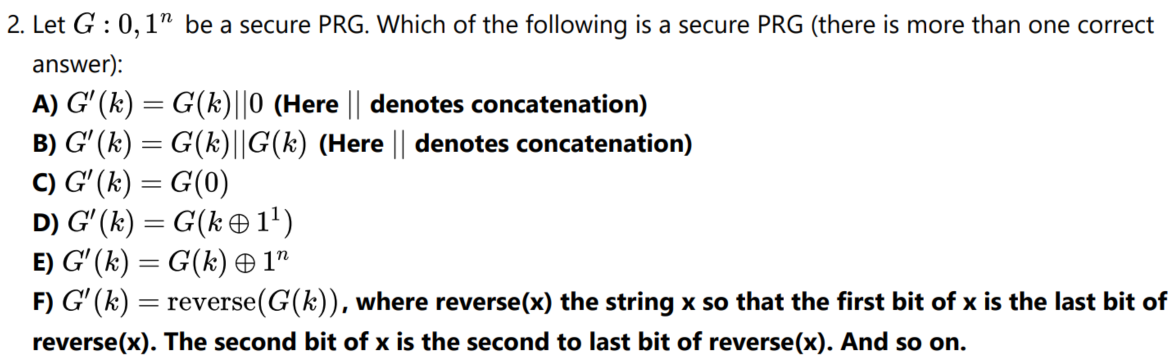
Quiz 4

Status	Done
Course	🖥️ 密碼工程
Due date	@March 30, 2023
Paper	
Type	

1. Data compression is often used in data storage and transmission. Suppose you want to use data compression in conjunction with encryption. Does it make more sense to:
   A) Compress then encrypt
   B) The order does not matter -- either one is fine
   C) The order does not matter -- neither one will compress the data
   D) Encrypt then compress

Ans. A

如果先將明文加密，這段密文會非常接近random distribution，這會讓壓縮沒辦法有非常好的效果。因為大部分的壓縮演算法會利用到類似省略重複字串，變成random distribution就幾乎沒有重複字串了，所以壓縮效果會變很差，壓縮後的體積還是會很大。如果先壓縮後再做加密，可以讓壓縮後體積變小，達到真的壓縮的效果。



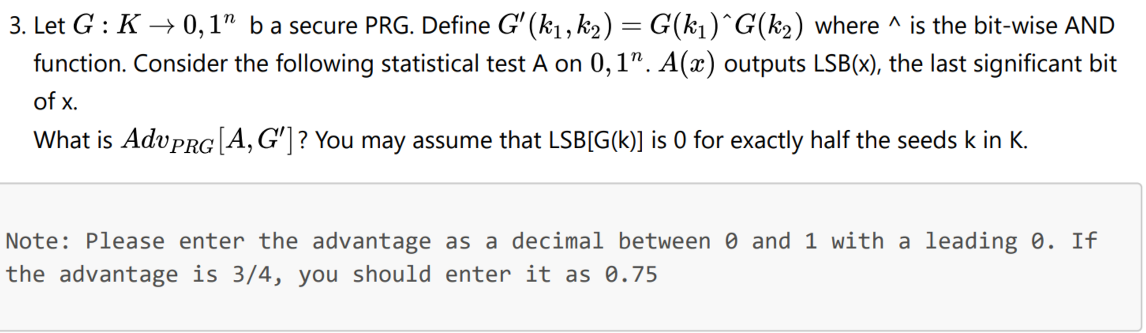
2. Let $G : 0, 1^n$ be a secure PRG. Which of the following is a secure PRG (there is more than one correct answer):
A) $G'(k) = G(k)||0$ **(Here** $||$ **denotes concatenation)**
B) $G'(k) = G(k)||G(k)$ **(Here** $||$ **denotes concatenation)**
C) $G'(k) = G(0)$
D) $G'(k) = G(k \oplus 1^1)$
E) $G'(k) = G(k) \oplus 1^n$
F) $G'(k) = \text{reverse}(G(k))$**, where reverse(x) the string x so that the first bit of x is the last bit of reverse(x). The second bit of x is the second to last bit of reverse(x). And so on.**

Ans. DEF

A錯的原因是多加了一個0, 會讓原本1/2出現1, 1/2出現0改變，變成出現0的機率比較多，所以A會讓PRG變的效果更差。
B錯的原因是直接接上會讓這段密文重複出現，會讓人發現有規律，所以效果變差。

C錯的原因是他直接拿0去生，生出來的東西很明顯不會是隨機的

D、E、F對的原因都是因為生出來的東西都會是隨機的。

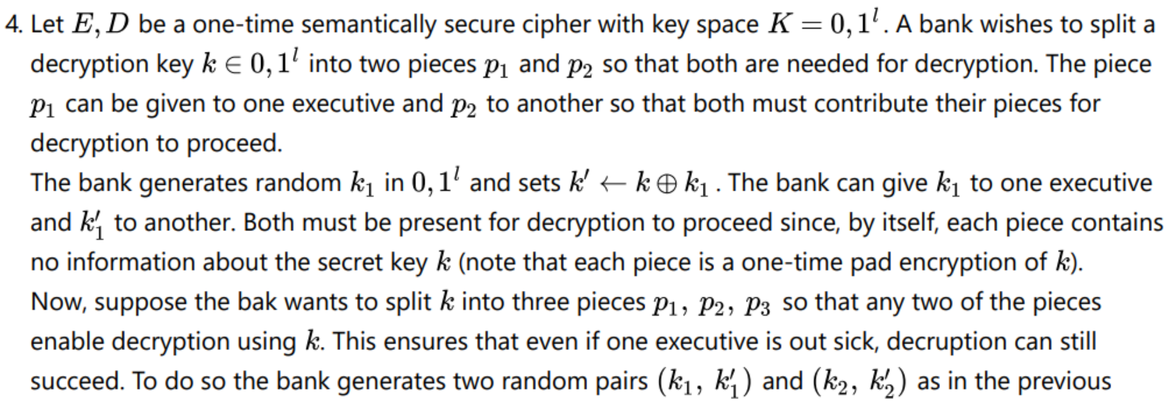


3. Let $G : K \to 0,1^n$ b a secure PRG. Define $G'(k_1,k_2) = G(k_1)\hat{}G(k_2)$ where ^ is the bit-wise AND function. Consider the following statistical test A on $0,1^n$. $A(x)$ outputs LSB(x), the last significant bit of x.
What is $Adv_{PRG}[A,G']$? You may assume that LSB[G(k)] is 0 for exactly half the seeds k in K.

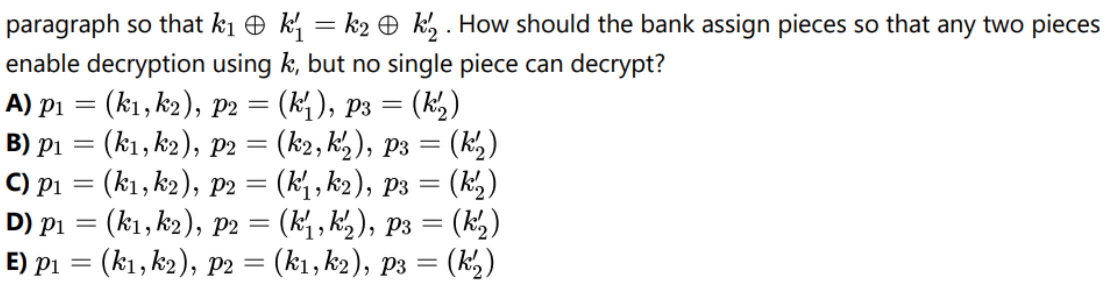
```
Note: Please enter the advantage as a decimal between 0 and 1 with a leading 0. If
the advantage is 3/4, you should enter it as 0.75
```

Ans. 0.25

因為G(k)的LSG有1/2機率出現0，所以代表LSB(G(k1))出現1和LSB(G(k2))出現1的機率為1/2 * 1/2 = 1/4 = 0.25，也就是代表G’(k1, k2)出現1的機率為 0.25。



4. Let E, D be a one-time semantically secure cipher with key space $K = 0,1^l$. A bank wishes to split a decryption key $k \in 0,1^l$ into two pieces p_1 and p_2 so that both are needed for decryption. The piece p_1 can be given to one executive and p_2 to another so that both must contribute their pieces for decryption to proceed.
The bank generates random k_1 in $0,1^l$ and sets $k' \leftarrow k \oplus k_1$. The bank can give k_1 to one executive and k'_1 to another. Both must be present for decryption to proceed since, by itself, each piece contains no information about the secret key k (note that each piece is a one-time pad encryption of k).
Now, suppose the bak wants to split k into three pieces p_1, p_2, p_3 so that any two of the pieces enable decryption using k. This ensures that even if one executive is out sick, decruption can still succeed. To do so the bank generates two random pairs $(k_1,\ k'_1)$ and $(k_2,\ k'_2)$ as in the previous



paragraph so that $k_1 \oplus\ k'_1 = k_2 \oplus\ k'_2$. How should the bank assign pieces so that any two pieces enable decryption using k, but no single piece can decrypt?

A) $p_1 = (k_1,k_2),\ p_2 = (k'_1),\ p_3 = (k'_2)$

B) $p_1 = (k_1,k_2),\ p_2 = (k_2,k'_2),\ p_3 = (k'_2)$

C) $p_1 = (k_1,k_2),\ p_2 = (k'_1,k_2),\ p_3 = (k'_2)$

D) $p_1 = (k_1,k_2),\ p_2 = (k'_1,k'_2),\ p_3 = (k'_2)$

E) $p_1 = (k_1,k_2),\ p_2 = (k_1,k_2),\ p_3 = (k'_2)$

Ans. C

因為只有C的分法可以讓任意兩人都可以湊出 k，但又不會讓任何人可以自己湊出 k。

A不行的原因: p2, p3不能湊出k

B不行的原因: p2自己能凑出k

D不行的原因: p2, p3不能凑出k

E不行的原因: p1, p2不能凑出k