1. 請問本次DPA on AES實驗中，我們選擇的攻擊點(Point of Interest) 在AES-128整個流程的哪個地方(第幾個round的哪個步驟)?為什麼?

攻擊點是在第一個round的sub byte出來的步驟，我推測原因有三個:

第一，因為我們的目標是要破解key、且已知plaintext，所以挑algo中plaintext與key運算的步驟攻擊最能蒐集到有用的資料，而在sub byte前便是add round key。

第二，挑最一開始”add round key”的原因是此時用到的key還是original key，並非像是其他round所用的round key，也就是說，我們破解出來的結果即是原始密鑰。

最後，取S-box output作為攻擊點的原因是s-box的功能就是要讓plaintext增加非線性的轉換，目的是要打散資料，使其他想要正面破解AES的攻擊方式更難找到規律性；但對side channel attack來說，反而是好事，因為好的s-box能幫我們把資料弄亂，增加正確密鑰和錯誤密鑰相關係數的差異，讓破解更順利。

。

請問AES的S-Box內容是否只能是標準的S-box一種?為什麼? \*

不對，因為作業就測不只一種S-box了(誤。如果只要實作s-box功能的話，即使讓0~127隨機排列在一個16\*16矩陣中，都可以推出相對應的inverse s-box，能夠順利執行AES演算法，可見S-box有非常多種可能。但是，在設計的時候若遵循一些數學準則，例如: No output bits of any S box should be close to a linear function of input bits.，才能增加破解的困難度。

1. 請問是否能用(2( Input-Byte ) + 1 ) mod 256 做Byte substitution?為什麼? \*

不行，反例: input1 = 010, (2(010)+1)mod 256 = 1

input2 = 12810，(2(12810)+1)mod 256 = 1

如果不同的input值對應到相同的output值，會造成inverse時不會唯一。

1. 在本實驗中，為何我們能將破解的複雜度大幅下降?

讓複雜度大幅下降的重點是我們針對128bits中的每8bits去分別破解，讓原本高達2^128種可能的密鑰組合，變成只剩下2^8種，最後再分別破解16個box就可以完成破解，也就是總共測試16\*2^8的可能性即可。

不過要能執行此破解有幾個前提。首先，要能對系統不斷出入自訂的plaintext去蒐集很多條power trace當作分析的dataset；此外，還要事先知道系統使用的S-box為何，不然光靠這次作業規格的DPA是無法破解出密鑰的。最後，也要知道硬體是在哪裡做每個Box的第一次add round key，才知道觀察哪裡的Power變化。

1. 在本實驗中，我們採用何種統計方法做為判斷猜測的 key 正確與否?

Correlation coefficient。

1. 請比較 不同 S-Box 版本的AES 經過DPA攻擊的結果，並詳述分析 。

我發現助教的s-box數值分布非常規律，在TA\_sbox中的數字幾乎是從小排到大，所以我在測試前便推測說TA版本應該會很容易破解出正確密鑰。(不然幹嘛還要大費周章去創造出一個好的s-box來用(?))

但結果卻出乎我意料!因為原始sbox和TA\_sbox都成功破解出大部分的key，甚至原始sbox的正確率還略勝一籌。不過，我觀察到兩者最大的區別是其他錯誤key的相關係數，在原始sbox的Results table中，相關係數最高的key和第二高的key，數值差異很大；但在TA\_sbox的Result table中，相關係數最高的key和第二高的key，兩者數值卻很接近。我認為是因為原始的sbox能順利將input做non-linear的轉換，所以對於side-channel attack來說，其s-box output 的hamming weight分布較廣，帶入correlation的公式後，只有正確key才能得出高數值；相反地，因為TA\_sbox output比較像是linear轉換，所以hamming weight分布較平均，導致有不少錯誤的key結果的correlation都非常高。總結來說，正常s-box能讓針對ALGO的攻擊更加困難，但對於DPA來說反而會有幫助。