計算機安全 HW1 Writeup

- Real name: 張瀚文
- Nickname on course website: Hwww
- Student ID: b07505027.

POA

解法

這題與講義上CBC Mode / Padding Oracle Attack的差別只差在padding的方法，講義上是利用PKCS#7，而這題則是採用「第一個padding放0x80，後面全部都放0x00」因此只要從尾開始讀到0x80就知道是padding的結尾。

這題參考了講義中GitHub上的範例程式碼（請見 `POA_solve.py` ），將原本padding的部分改成了0x80跟0x00：回推當前明文的地方需要改成0x80， `oracle()` 中已知明文部分需要改成與相同長度0x00做xor。其餘架構大致不變。

這題的另外一個問題點在於第二個block。當原本的密文與暴搜的密文相同時，一定會讓padding正確並通過，但卻不是我們想要的解，所以需要跳過他。

然而如果跳過了前面的情況，造成沒有解讓padding正確通過的話，前面情況的解就會是我們想要的，要將他取回來。正確來說是當明文本來就是0x80時，只有在原本的密文與暴搜的密文相同時，才會讓padding正確，如果跳過的話後面就搜不到解，所以需要特別注意，暴搜完如果都沒有解，要將解補回去。

心得

這題在第二個block遇到問題，想了滿久才搞懂為什麼會出問題，解出來後感覺對POA有比較深入的了解。

COR

解法

這題跟講義上的LFSR correlation attack範例差不多，都是三個LFSR組合而成的Mixed LFSR，回饋係數已知，寫在題目generate.py的裡面。這題的初始值是把FLAG大括號裡的字串分成三份，

先利用「其中一個LFSR的輸出與總輸出有大約75%的相關率」這點，分別暴搜LFSR3跟LFSR2的初始值（程式碼參考 `solve_32.py` ），因為不確定相關率會是多少，所以從70%開始往上更新當前最大值，同時輸出當前最大值對應的初始值，

這題的LFSR的初始值只有16bits，因此暴搜一個LFSR的初始值只要 $2^{16}=65536\approx10^4$，每次驗證約10^2，因此不會太久。最後找到最有可能的字串分別為hj跟ui。

而LFSR1沒有辦法利用相關性暴搜，不過我們已經知道了LFSR2跟LFSR3的初始值，因此可以利用MYLFSR暴搜LFSR1的初始值（程式碼參考 `solve_all.py` ），只要暴搜的某次初始值與題目輸出比對的正確率是100%，就找到正確答案了，暴搜複雜度跟暴搜LFSR3大致相同。最後找到的結果是df。根據題目程式碼，將1, 2, 3的結果合併在一起加上前後的FLAG{}就是答案了。

心得

因為這題跟講義上的範例幾乎一樣，而觀念又沒有特別難懂，所以很快就可以寫出暴搜程式了～