Quiz 4

110550108 施柏江

Problem 1

LFSR is simply an arrangement of n stages in a row with the last stage, plus any other stages, modulo-two added together and returned to the first stage. An algebraic expression can symbolize this arrangement of stages and tap points called the characteristic polynomial. One kind of characteristic polynomial called primitive polynomials over GF(2), the field with two elements 0, 1, can be used for pseudorandom bit generation to let linear feedback shift register (LFSR) with maximum cycle length.

(a) Is x^8 + x^4 + x^3 + x^2 + 1 a primitive polynomial?

Ans: Yes.

By the definition, an irreducible polynomial F(x) of degree m over GF(p), where p is prime, is a primitive polynomial if the smallest positive integer n such that F(x) divides x^n − 1 is n = p^m − 1. In this case, the polynomial x^8 + x^4 + x^3 + x^2 + 1 is of degree 8, so we need to check if it divides x^255 −1. It can be checked by the code below.

```python
import sympy as sp

x = sp.symbols('x')
polynomial = x**8 + x**4 + x**3 + x**2 + 1
dividend = x**255 - 1
remainder = sp.rem(dividend, polynomial, domain=sp.GF(2))
if remainder == 0:
    print("The polynomial is primitive.")
else:
    print("The polynomial is not primitive.")
```

```
PS C:\Users\brian\OneDrive\桌面\hw\三下\密碼\hw4> python problem1.py
The polynomial is primitive.
```

(b) What is the maximum cycle length generated by x^8 + x^4 + x^3 + x^2 + 1?

Ans: 255.

This length is achieved when the polynomial is primitive, meaning it generates a maximal-length sequence of 2^8 - 1 unique states before repeating. Each state in the sequence is distinct until it returns to the initial state after 255 clock cycles.

(c) Are all irreducible polynomials primitive polynomials?

Ans: No

An irreducible polynomial is one that cannot be factored into the product of two non-constant polynomials over the same field. In the context of LFSRs, irreducible polynomials are used to ensure that the generated sequence has a maximal cycle length. However, not all irreducible polynomials generate sequences with the maximum possible cycle length. So, while all primitive polynomials are irreducible, not all irreducible polynomials are primitive. Some irreducible polynomials may generate shorter sequences in an LFSR compared to the sequences generated by primitive polynomials.

Problem 2

(a) Please use x^8 + x^4 + x^3 + x^2 + 1 as a characteristic polynomial to write a Python program to encrypt the following plaintext message with the initial key 00000001, then decrypt it to see if your encryption is correct.

Ans:

In each iteration, first calculates the feedback based on the polynomial, takes the last bit as the output bit, shifts the state left by one bit, and appends the feedback to the rightmost end of the state. For each character in the message, it performs XOR operation with the corresponding position of the pseudo-random bit and converts the result to a character, thus encrypting the message.

```
PS C:\Users\brian\OneDrive\桌面\hw\三下\密碼\hw4> python problem2.py
Encrypted message:
AUOYCUVD@SERTRIVHOFUNCD@GRDATUNHVDRSHUXUIAUTSAO
RBDNDSDIRBIPLHN@RXEIVHEESUORNMWDUHDHOBRE@RIOFMYBN
MPMEYPSOBMELSTH@TTIEVNRMDFABDSWDVHLLBNOUINUDU
NCEGUIEEDBXTIDHDDAUI@UWDBAO@BIHEVESNLEUHIOFMT
BIGSDAUESTOGETIERUI@NVDCAOINEIVHEUALLYAGUDR@LLU
HAUV@STHDIDEATHAULDETOTIDBSE@TIONNGNTSTOHVESSI
TXIOTHDGHSRTQL@CD

Decrypted message:
ATNYCUWEARESTRIVINGTOBEAGREATUNIVERSITYTHATTRAN
SCENDSDISCIPLINARYDIVIDESTOSOLVETHEINCREASINGLYCO
MPLEXPROBLEMSTHATTHEWORLDFACESWEWILLCONTINUET
OBEGUIDEDBYTHEIDEATHATWECANACHIEVESOMETHINGMU
CHGREATERTOGETHERTHANWECANINDIVIDUALLYAFTERALLT
HATWASTHEIDEATHATLEDTOTHECREATIONOFOURUNIVERSI
TYINTHEFIRSTPLACE
```

How to run the code: python problem2.py

(b) Due to the property of ASCII coding the ASCII A to Z, the MSB of each byte will be zero (left most bit); therefore, every 8 bits will reveal 1 bit of random number (i.e. keystream); if it is possible to find out the characteristic polynomial of a system by solving of linear equations?

Ans: Yes

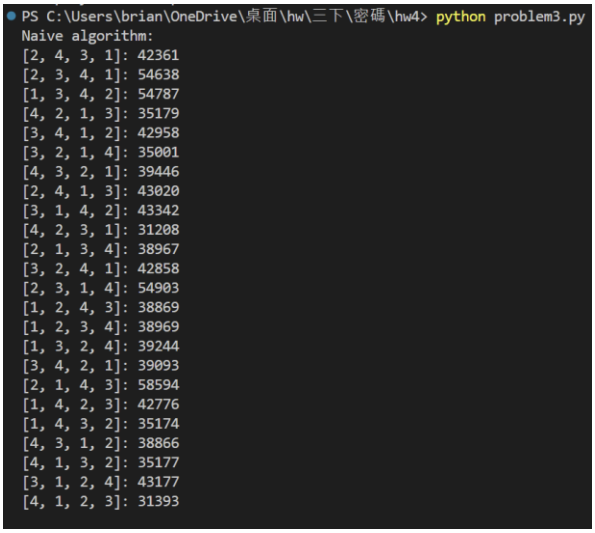
The characteristic polynomial represents the feedback structure of the LFSR, which determines its output sequence. By XORing known plaintext with ciphertext to obtain the keystream, and then setting up equations representing the relationship between the LFSR state and the keystream bits, we can solve for the characteristic polynomial.

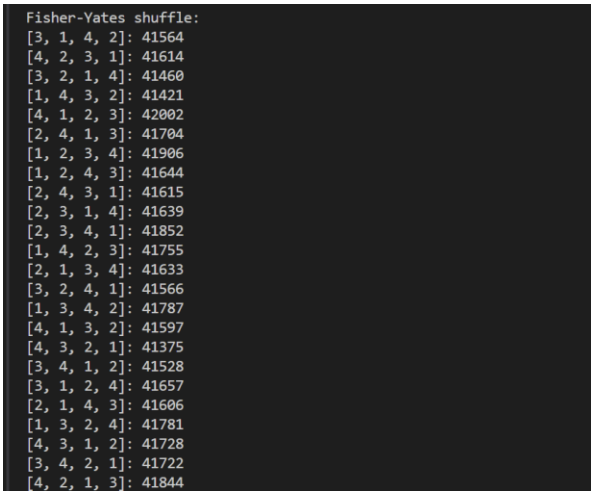
Problem 3

RC4's vulnerability mainly arises from its inadequate randomization of inputs, particularly the initialization vector (IV) and key integration, due to its reliance on the initial setup by its Key Scheduling Algorithm (KSA). The cipher operates through two phases: KSA, which shuffles a 256-byte state vector based on the key to ensure dependency and randomization, and the Pseudo-Random Generation Algorithm (PRGA), where it further manipulates this state to produce a seemingly random output stream. To help you understand the importance of randomization algorithms, here we provide the pseudocode for two slightly different shuffle algorithms.

(a) Please write a Python program to simulate two algorithms with a set of 4 cards, shuffling each a million times. Collect the count of all combinations and output.

Ans:





How to run the code: python problem3.py

(b) Based on your analysis, which one is better, why?

Ans: Fisher-Yates

Based on the analysis provided by the simulation, the Fisher–Yates shuffle algorithm is better than the naive shuffle algorithm. The Fisher–Yates shuffle algorithm ensures a more uniform distribution of permutations. This is because in the Fisher–Yates shuffle, each element has an equal probability of being swapped with any other remaining element, ensuring that all possible permutations are equally likely.

(c) What are the drawbacks of the other one, and what causes these drawbacks?

Ans:

The naive shuffle algorithm may not produce uniformly distributed permutations. Certain permutations may occur more frequently than others, leading to biased results. Due to its simplistic swapping process, the naive shuffle algorithm may be more predictable or susceptible to exploitation by attackers in cryptographic contexts.