2024 암호분석경진대회

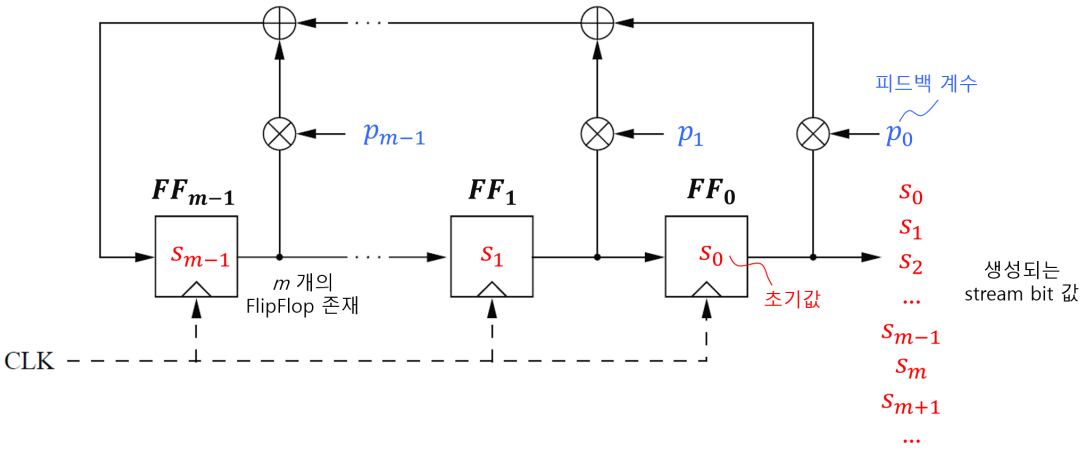
1번 문제

<문제 개요>

다음은 LFSR(Linear Feedback Shift Register)을 보여주고 있다. LFSR에 비선형적 요소를 추가하여, Stream cipher로도 많이 사용된다. 아래 그림 1은 LFSR 구조에 대한 설명이다. 아래 그림 1을 보면, 총 m개의 FF(플립플롭)을 가지며, 이는 Clock의 rising edge에서 동작한다. 또한, 그림 상에는 3개의 tap-out이 보인다. (한편, p_2에서 p_{m-2}까지의 존재 여부는 그림1 에서는 생략함)

한 CLK이 들어가면(rising edge), $s_m \leftarrow s_{m-1}p_{m-1} + \ldots + s_1p_1 + s_0p_0$ 값이 결정된다.
그 다음 CLK에서는 s_{m+1} 이 결정되며, 또한, 그 다음 CLK에서는 s_{m+2} 값이 결정된다. 이때, "+"은 XOR이다.

LFSR에서 generate 되는 값(s_i)은 tap-out되는 방식에 따라 결정되는데, 이를 polynomial로 표시할 수 있으며, 이를 feedback polynomial이라고도 한다.



[그림 1] 일반적 LFSR 구조 (feedback 계수 p_i와 초기값 s_{m-1}, ..., s_0 가짐)

아래 그림 2는 feedback polynomial $x^8+x^4+x^3+x+1$를 가지는 경우의 LFSR 그림이다. 즉, 위 그림 1에서 p_0, p_1, p_3, p_4 가 1인 경우이다(tap-out 존재함). 이때 해당 polynomial을 (0,1,3,4,8)로 표현한다.

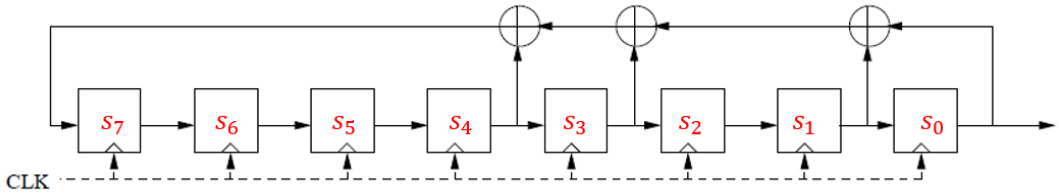


그림 2 특정 feedback polynomial을 갖는 LFSR 사례

이를 참고하여, 다음 문제에 대한 답을 제시하라.

< 문제 >

(1) 세부 문제 1: feedback polynomial이 (0,1,2,7,128)인 LFSR을 구현하라. 임의의 초기값이 설정될 경우, 해당 되는 초기값에 대한 stream을 생성하는 코드를 구현하라.

(2) 세부 문제 2: CNN 등, DNN(Deep Neural Network)을 사용하여, LFSR의 출력되는 주기값을 알 수 있는 방법이 있을 수 있다. 이를 구현(학습)하여, 주어진 stream에서 주기값을 파악할 수 있음을 보여라.

< 주의 사항 및 평가 방법>

1) 세부 문제 1을 푸는데 있어서의 참고 사항:
- Python, SageMath, Java, C 등 프로그래밍 언어를 사용하여 구현해야 함.
- 본 문제의 의도는 범용 programming language를 사용하여, LFSR을 실제 구현할 수 있는지에 대한 내용을 파악하기 위함임.
- 이 때문에, Python이나 SageMath, Java, C, C++, C#, Javascript 등을 사용하는 것은 허용되지만, Maple이나 Mathematica, Matlab, R 등 수식 혹은 graphical block을 기반으로 하는 package 툴을 사용한 최종 결과물은 인정하지 않음 (물론 결과물 검증 등에 있어서, 이를 사용하는 것은 무방함)

2) 세부 문제 2를 푸는데 있어서의 참고 사항:
- DNN을 설계하는 것이 가장 중요함(어떤 DNN 알고리즘을 사용할 것인지 등도 자유임)
- 또한, 필요할 경우, 시계열 형태의 bitstream 데이터를 어떤 특별한 형태로 전처리 할 필요도 있음
- DNN을 설계하고 학습, 추론에 있어서, Tensorflow나 PyTorch를 선호함. Graphical한 툴을 사용할 경우에는 세부 문제 2를 해결하는 방법론의 우수성 측면에서 패널티가 있을 수 있음

3) 최종 결과물은 다음 2종을 포함해야 함
- 세부 문제 1 해결을 위한 소스 코드와 결과물 bit stream 값, 그리고 이를 설명하는 문서
- 세부 문제 2 해결을 위해 사용한 DNN의 구조와 구조 설명서, 구현 결과물, 추론 결과, 그리고 이를 설명하는 문서

4) 평가 방법은 다음과 같음
- 세부 문제 1에 대한 정상 동작 여부 확인 및 생성 bit stream 값과 소스 코드의 정확성 검증 (30점)
- 세부 문제 2를 해결하기 위해 제시된 방법론과 제시된 DNN 구조의 정확성 및 우수성 확인(30점)
- 세부 문제 2에서 제시한 LFSR의 stream 값 뿐만 아니라, 이 보다 작은 주기를 갖는 bit stream을 임의로 넣었을 때, 해당 주기를 정확히 추론하는지를 측정함 (20점)
- 보고서의 완성도 및 설명 내용의 정확성 확인(20점)
- 총 100점

● 참고 #1: C. Paar의 "Understanding Cryptography" p.44와 solution(#2.7 등을 참고하여 문제를 냄)
● 참고 #2: esslinger learning and experiencing cryptography with cryptool, p.443 (Section 9.3의 Stream ciphers 쪽)