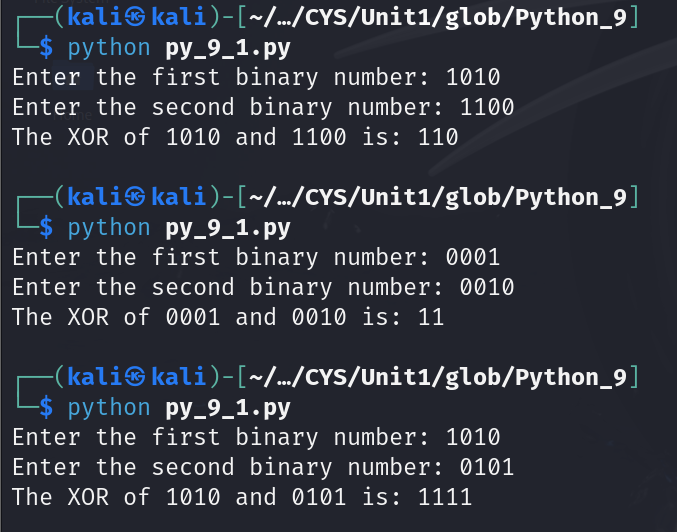
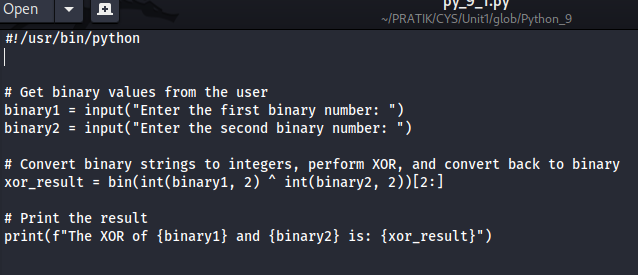
Crypto lab 9

1. Write a python script to get the binary values from the user and perform XOR operation.

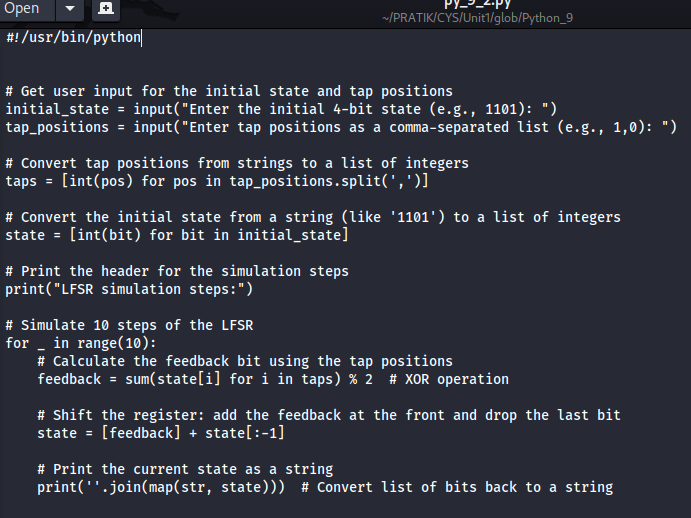




2. Write a Python script that implements a simple 4-bit LFSR. The initial state of the register and the tap positions should be user inputs.

Simulate 10 steps of the LFSR, displaying the state of the register at each step.





3. Write a report on attacks on LFSR. Explain any one attack in detail.

**Report on Attacks on Linear Feedback Shift Registers (LFSRs)**

**Introduction**

Linear Feedback Shift Registers (LFSRs) are widely used in cryptography, telecommunications, and error detection systems due to their efficiency and ability to generate pseudorandom sequences. However, their predictable nature and structure make them susceptible to various attacks. This report outlines common attacks on LFSRs, with a detailed examination of one specific attack: the Berlekamp-Massey algorithm.

**Common Attacks on LFSRs**

1. **Berlekamp-Massey Algorithm**:
   * A method for finding the shortest LFSR that can produce a given sequence.
   * It efficiently reconstructs the polynomial used in the LFSR based on the output bits.
2. **Time-Memory Tradeoff Attacks**:
   * Attacks that precompute LFSR states for various initial conditions to quickly recover the secret state during runtime.
   * These involve storing known sequences in a lookup table for efficient retrieval.
3. **Cryptanalysis using Known Output**:
   * An attacker observes a certain number of output bits generated by an LFSR.
   * Using this output, they can analyze and deduce the structure of the LFSR.
4. **Exploiting Linear Properties**:
   * LFSRs exhibit linear properties which can be exploited to predict future bits based on previous output.

**Detailed Examination: Berlekamp-Massey Algorithm**

**Overview**

The Berlekamp-Massey algorithm is a polynomial-time algorithm used to determine the coefficients of an LFSR given a sequence of output bits. It reconstructs the minimal polynomial that describes the LFSR's feedback function. This allows attackers to recover the internal state of the LFSR and predict future outputs.

**Steps of the Berlekamp-Massey Algorithm**

1. **Initialization**:
   * Set the initial state with a known polynomial and a length of the LFSR.
   * Maintain a "connection" vector that stores the last feedback coefficients.
2. **Iterate through the Output Sequence**:
   * For each output bit in the sequence, update the connection polynomial using the previously calculated coefficients.
   * Calculate the discrepancy to determine if the current polynomial is sufficient.
3. **Update the Polynomial**:
   * If a discrepancy is found (i.e., the current output does not match what is expected from the polynomial), adjust the polynomial degree and coefficients.
   * Use previously calculated states to minimize the degree of the polynomial.
4. **Return the Polynomial**:
   * Once the algorithm processes the entire sequence, it returns the coefficients of the LFSR polynomial.

**Example**

Suppose an attacker observes the output sequence: 1, 0, 1, 1, 0, 1. They can apply the Berlekamp-Massey algorithm to reconstruct the LFSR:

1. Initialize with the first few bits, determining the initial polynomial and degree.
2. As each bit is processed, discrepancies are calculated to adjust the polynomial.
3. After iterating through the entire sequence, the final polynomial coefficients describe the LFSR used to generate the observed output.

**Implications**

The effectiveness of the Berlekamp-Massey algorithm poses significant risks for systems relying on LFSRs for security. Once the LFSR's polynomial and initial state are reconstructed, an attacker can predict future outputs and potentially decrypt secured messages.

**Conclusion**

LFSRs are fundamental components in many digital systems, but their vulnerabilities are critical for the design of secure systems. The Berlekamp-Massey algorithm exemplifies the type of attacks that can be executed against LFSRs. Understanding these vulnerabilities is essential for enhancing cryptographic designs and improving the overall security of systems using LFSRs. To mitigate these risks, alternative cryptographic methods and more complex state machines should be considered.

BONUS POINT:

4. write a python script to break hill cipher (2X2) using known plain text attack.

Known Plaintext: "MEET"

Corresponding Ciphertext: "URRG"

