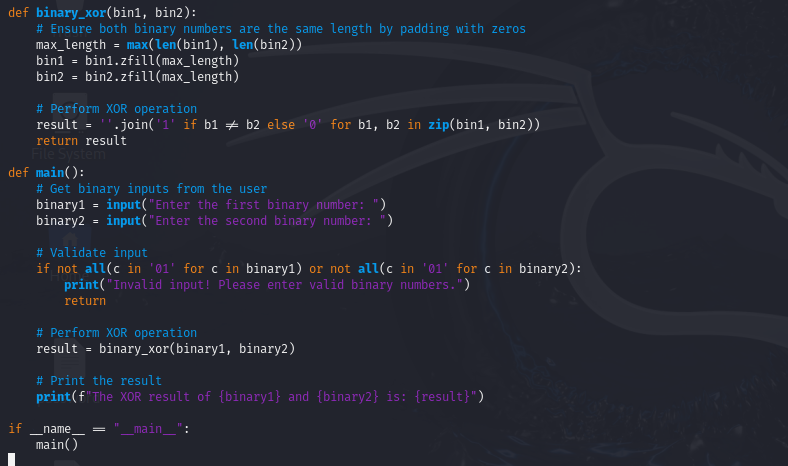
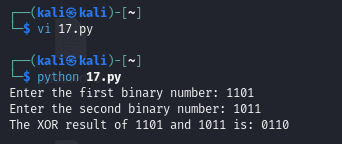
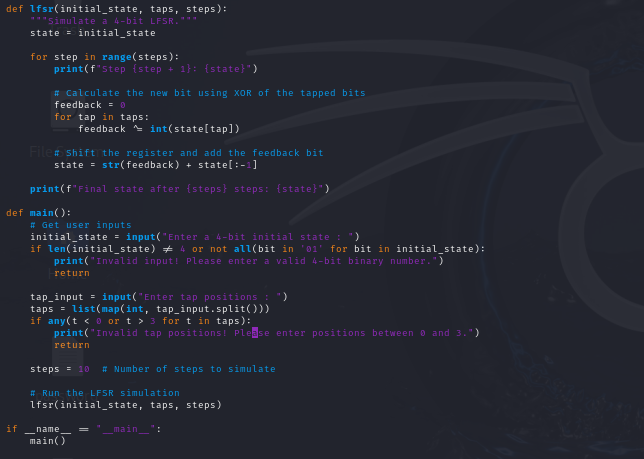
1. Write a python script to get the binary values from the user and perform XOR operation.

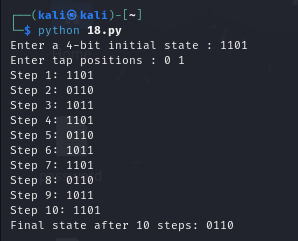




2. Write a Python script that implements a simple 4-bit LFSR. The initial state of the register and the tap positions should be user inputs.

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





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

Linear Feedback Shift Registers (LFSRs) are widely used in cryptographic applications, particularly in stream ciphers and pseudo-random number generation. Their simplicity and efficiency make them attractive for hardware implementations. However, LFSRs are not without vulnerabilities.

Types of Attacks on LFSRs

1. Brute Force Attack:

This approach involves attempting every possible key until the correct one is found. The complexity increases exponentially with the length of the LFSR, making it impractical for long registers.

2. Known Plaintext Attack:

In this scenario, an attacker has access to both plaintext and corresponding ciphertext. By analyzing the relationship between them, the attacker can derive the LFSR’s initial state or coefficients.

3. Chosen Plaintext Attack:

Here, the attacker can choose specific plaintexts to encrypt and observe the resulting ciphertexts. This can reveal patterns in the output and help reconstruct the LFSR.

4. Side-Channel Attacks:

These attacks exploit information leaked during the computation process, such as timing information, power consumption, or electromagnetic leaks. By analyzing these signals, an attacker can infer the internal state of the LFSR.

5. Algebraic Attacks:

These attacks exploit the algebraic structure of LFSRs to form equations that describe the output sequences. By solving these equations, attackers can recover the LFSR’s parameters.

Correlation Attack on LFSR

The correlation attack is another effective method for breaking Linear Feedback Shift Registers (LFSRs) used in cryptographic systems. This attack exploits the predictable nature of the sequences generated by LFSRs, allowing attackers to recover the internal state of the LFSR through statistical analysis.

Correlation attacks focus on identifying relationships between the generated bits of an LFSR and the known properties of the linear function used for feedback. By analyzing the correlation between the output sequence and known patterns, attackers can infer information about the internal state of the LFSR.

Methodology of Correlation Attack

1. Understanding the Output Sequence:

An LFSR generates output bits based on its internal state and feedback polynomial. If an attacker has access to a significant number of output bits (e.g., `s\_0, s\_1, s\_2, ...`), they can begin to analyze patterns within this sequence.

2. Constructing the Correlation Function:

The attacker constructs a correlation function to assess how likely it is that a specific guess for the LFSR state matches the observed output. This involves defining a set of potential linear combinations of the LFSR’s output and computing the correlation for these combinations.

3. Statistical Analysis:

- Using statistical techniques, the attacker calculates the correlation coefficients for various potential states and the output sequence. High correlation values indicate that a guessed state is likely correct.

4. Identifying Candidate Polynomials:

By systematically analyzing different polynomial configurations and observing the correlations, the attacker can narrow down the potential feedback polynomials that the LFSR could be using.

5. Recovering the Key/State:

Once sufficient correlations have been established, the attacker can reconstruct the state of the LFSR and the feedback polynomial, allowing them to predict future outputs or even decrypt messages.

Example of a Correlation Attack

Let’s consider an LFSR that outputs a sequence of bits: `011010011001`. Here’s how an attacker might execute a correlation attack on this sequence:

1. Gather Output Sequence: The attacker has the output `011010011001`.

2. Analyze Possible States:

Suppose the attacker suspects that the LFSR is of degree 3. They generate potential polynomial candidates:

- \( C\_1(x) = 1 + x + x^2 \)

- \( C\_2(x) = 1 + x^2 + x^3 \)

- \( C\_3(x) = 1 + x + x^3 \)

3. Construct Correlation Function:

For each polynomial candidate, the attacker calculates the expected output based on the polynomial and compares it to the actual output sequence to compute correlation coefficients.

4. Compute Correlations:

- By summing the products of the observed output and expected output for each polynomial, the attacker identifies which polynomial produces the highest correlation.

5. Identify Likely State:

- Suppose \( C\_2(x) \) yields a high correlation. The attacker deduces that this polynomial is likely the feedback polynomial used by the LFSR.

6. Recover the State:

- Using the identified polynomial, the attacker reconstructs the state of the LFSR, allowing them to predict future outputs or decrypt messages encrypted with the LFSR.

Complexity and Effectiveness

Correlation attacks can be quite effective, particularly against shorter LFSRs. The computational complexity depends on the number of output bits analyzed and the degree of the LFSR, but it is generally lower than brute force methods.

The strength of the attack lies in its statistical nature, which can reveal information even when the attacker has limited knowledge of the LFSR configuration.

Conclusion

Correlation attacks present a significant threat to cryptographic systems utilizing LFSRs by exploiting their linearity and predictability. Through careful analysis of output sequences and statistical correlation, attackers can effectively deduce the internal state and polynomial of the LFSR. This highlights the necessity for robust countermeasures, such as using non-linear feedback mechanisms or additional layers of encryption to mitigate the risk posed by such attacks. Understanding these vulnerabilities is crucial for anyone involved in the design and implementation of cryptographic systems.

BONUS POINT:

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

Known Plaintext: "MEET"

Corresponding Ciphertext: "URRG"



