Linear Cryptanlaysis using Matsui Algorithm

Linear Cryptanalysis is a known-plaintext attack introduced by Mitsuru Matsui in the early 1990s. It's one of the most powerful and general attacks against block ciphers, especially Feistel networks like DES. Linear cryptanalysis exploits statistical biases in linear combinations of plaintext, ciphertext, and sometimes key bits. The attacker looks for a linear approximation of the form

$$P \cdot \alpha \oplus C \cdot \beta = K \cdot \gamma$$

where (P, C, K) refers to plaintext, ciphertext and key respectively. THe input mask, output mask and the key mask are denoted by (α, β, γ). A successful linear approximation is characterized by a bias ϵ, defined as

$$\epsilon = Pr[P \cdot \alpha \oplus C \cdot \beta = K \cdot \gamma] - \frac{1}{2}$$

The attacker's goal is to identify such linear approximations with significant bias. Matsui proposed two algorithms: distinguishing attack and Key recovery attack. Distinguishing attack tests whether a particular linear approximation holds with the expected bias over many plaintext-ciphertext pairs and compares the empirical bias to the theoretical bias derived from the cipher structure. It serves primarily to distinguish the cipher from a random permutation. Key recovery attack recovers partial key bits by focusing on linear approximations. It selects a linear approximation involving plaintext bits, bits at the input to the last round, and partial key bits. For each key candidate for the last round, partially decrypt the ciphertexts and evaluate the approximation. Identify the key guess that yields the highest empirical bias.

This project implements two Matsui attacks on reduced rounds DES. The first attack on 3-round DES is a distinguishing attack that is used to distinguish a cipher from a random permutation using a known linear approximation. The second attack is on 4 round DES is used to recover the partial key bits of the last round of the cipher is partially implemented. It includes the following files:

- test.cpp: Defines all the keys, s-boxes, permutations, Implements attack 1 on 3 round DES with 75 plaintext-ciphertext pairs. It return 0 if the attack is success . Attack on Des on 4 rounds with 50 plaintext-ciphertext pairs. Attack1() and attack2() are defined in attack.cpp.

- attack.h and attack.cpp: contains two functions attack_1 and attack_2. In attack_1(), the bias is calculated for certain random plaintext-ciphertext pairs. It check with the bias obtained from the piling up lemma. If it matches, the attack returns 0, otherwise, it returns 1. The attack_2() is attempted to recover the partial key of the last round. It identifies the active S-boxes based on the masks. For each possible guess of the key bits corresponding to these active S-boxes, it builds a candidate round key, processes 50 plaintext-ciphertext pairs, counts how many pairs satisfy the linear relation involving the input, output masks, and the effect of the guessed round key bits. After evaluating all guesses, it selects the guess with the strongest statistical bias (highest deviation from random). Using that guess, it recovers a partial key corresponding to those active S-box bits.

- Placement.h and placement.cpp: Given the input, output, permutation and an input vector. It returns the permutation of the input vector and inverse permutation of the input vector if it is invertible.

- Bitstring.hand bitstring.cpp: This program breaks the bitstream into chunks, sets the specific bit or range of bits and performs all the bitwise operations like concatination, bitwise AND, XOR etc.

- Fiestel.h and fiestel.cpp: Claculate the encryption and decryption function of the fiestel cipher. It also computes the best bias for each round using Lienar Approximation Table (LAT).

- s_box.cpp: Create a LAT for a s-box and sort the LAT based on the bias. It returns the entries with top bias, Top LAT entry for specific input maskm Top LAT entry for specific output mask and All LAT entries for a specific output mask. It includes s_box.h which declares all the functions and defines the LAT entry structure.

Compile and run:
$make
$./test
Output: Implements attack 1: output Key bit 1: 51 = 1, Key bit 2: 48 = 1, RHS: 0 (distinguisehs as DES)

Future Work: Full implementation of Matsui key recovery attack on DES with key Scheduling algorithm.