CS 406 Project
Linear Cryptanalysis

Krishna Narasimhan Agaram, Ameya Vikrama Singh

September 12, 2023

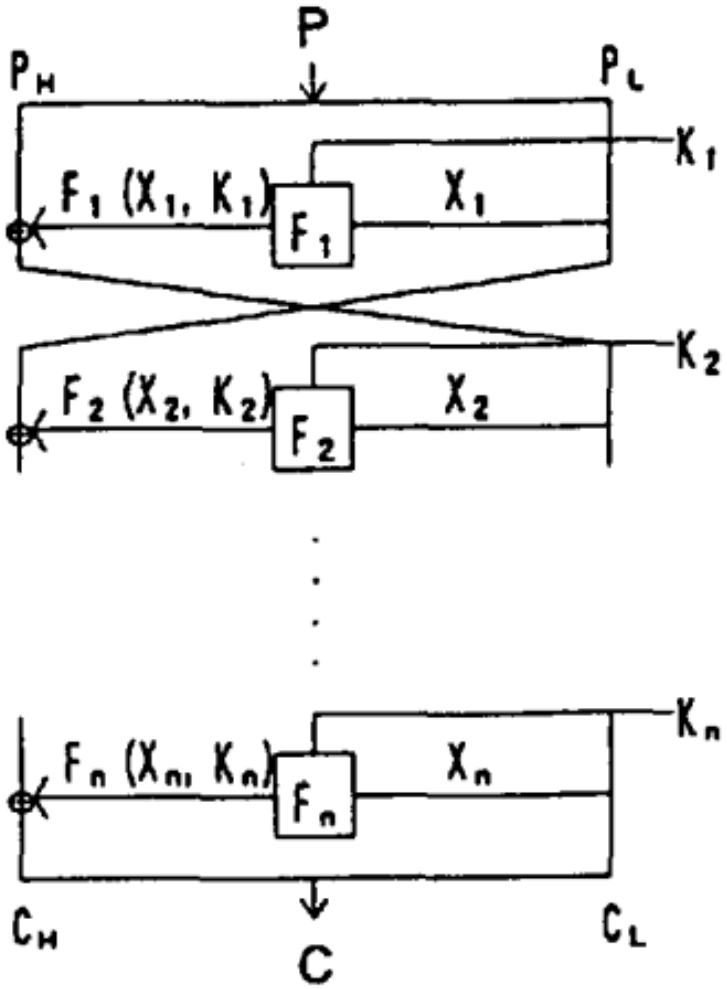
The DES Cipher

1. DES is a Feistel network based cipher.
2. As we know, Feistel networks are secure when the Feistel function is pseudorandom (with rounds at least 3)
3. The Feistel function in DES is a heuristic construction for a pseudorandom function.
4. The DES cipher was designed to encrypt plaintexts with block size of 64 bits, and a 56 bit key.
5. It uses 16 Feistel rounds. The subkey for each round is generated by splitting the key into two halves, then rotating the halves.

The Feistel function

1. The Feistel function uses an **expander** function to convert 32 bits into 48 bits, and then XORs the result with the subkey.
2. **So far, the cipher has been linear**
3. The next step is the **only non-linear step** in the Feistel function - The substitution boxes!
4. The DES standard specifies 8 substitution boxes which convert 6 bit inputs to 4 bits.
5. Finally 32 bits are output from the F-function.

DES - Image



Linear Cryptanalysis - Motivation

- Can the operation of the nonlinear parts be approximately represented by linear equations?
- We try to measure the accuracy of equations like

$$P[\mathbf{p}] \oplus C[\mathbf{c}] = K[\mathbf{k}]$$

  where $\mathbf{p}, \mathbf{c}, \mathbf{k}$ are bitmasks, and the notation $P[\mathbf{p}]$ means the XOR of all bits $P[i]$ for $i \in \mathbf{p}$.
- An arbitrary expression like this is likely to be true for only about half the plaintexts. However, **there might exist such linear approximations that have larger biases**.
- In the presence of a large number of plaintext-ciphertext pairs, such approximations can be used to recover the key bits with high probability! This reduces our search space among keys and thus greatly reduces the search-time.

S box linear approximations?

- Start with the small non-linear parts - the S boxes! Can the outputs be linearly correlated?
- Since the S box has just 64×16 input-output points, we can brute force through all possible bitmasks $\mathbf{a}, \mathbf{b}$ and consider the expression

$$X[\mathbf{a}] = S_i(X)[\mathbf{b}]$$

  Let us call the number of pairs satisfying this equation $N_i(\mathbf{a}, \mathbf{b})$
- We plotted all the S box tables and obtained each one's bias value. In the DES standard, it turns out that $N_5(16, 15) = -20$, a very large bias! This gives us a linear relationship for the input and output of the 5^{th} S-Box. But how do we use this relationship?

The crack in the wall

63	4	-4	0	-4	-8	0	4	-4	0	-8	-4	0	-4	-4	0

Substitution Box 5:e

alpha|beta	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	-2	2	-2	2	-4	0	4	0	2	-2	2	-2	0	-4
3	0	-2	6	-2	-2	4	-4	0	0	-2	6	-2	-2	4	-4
4	2	-2	0	0	2	-2	0	0	2	2	4	-4	-2	-2	0
5	2	2	-4	0	10	-6	-4	0	2	-10	0	4	-2	2	4
6	-2	-4	-6	-2	-4	2	0	0	-2	0	-2	-6	-8	2	0
7	2	0	2	-2	8	6	0	-4	6	0	-6	-2	0	-6	-4
8	0	2	6	0	0	-2	-6	-2	2	4	-12	2	6	-4	4
9	-4	6	-2	0	-4	-6	-6	6	-2	0	-4	2	-6	-8	-4
10	4	0	0	-2	-6	2	2	2	2	-2	2	4	-4	-4	0
11	4	4	4	6	2	-2	-2	-2	-2	-2	2	0	-8	-4	0
12	2	0	-2	0	2	4	10	-2	4	-2	-8	-2	4	-6	-4
13	6	0	2	0	-2	4	-10	-2	0	-2	4	-2	8	-6	0
14	-2	-2	0	-2	4	0	2	-2	0	4	2	-4	6	-2	-4
15	-2	-2	8	6	4	0	2	2	4	8	-2	8	-6	2	0
16	2	-2	0	0	-2	-6	-8	0	-2	-2	-4	0	2	10	-20
17	2	-2	0	4	2	-2	-4	4	2	2	0	-8	-6	2	4
18	-2	0	-2	2	-4	2	-8	4	6	4	6	-2	4	-6	0
19	-6	0	2	-2	4	2	0	4	-4	4	2	-6	4	-2	0
20	4	-4	0	0	0	0	0	4	-4	4	4	0	4	-4	0

Extending the Approximation, Propagation Probability

- The relation on the S-Box automatically turns into a relation over a single round. But does this affect the overall cipher?

$$X[15] \oplus F(X, K)[7, 18, 24, 29] = K[22]$$

- Let X_i $0 \leq i \leq n+1$ be the input to the i^{th} round. A state X_i is called **hidden** if it neither plaintext nor ciphertext, i.e it is an intermediate input.
- Any linear expression we get, appears in the following form:

$$\bigoplus_{i=0}^{n+1} X_i[\mathbf{x_i}] = \bigoplus_{i=1}^{n} K_i[\mathbf{k_i}]$$

- **Since the intermediate inputs are hidden from the attacker, we would like approximations that relate only plaintext, ciphertext and key bits, i.e do not contain any hidden states!**

The Piling Up Lemma

- Given two such linear approximations, we can XOR the two equations in order to get a third linear approximation! This approximation holds if both the former hold or neither one holds. Thus, it's probability is $p_1 p_2 + (1 - p_1)(1 - p_2)$
- It can be shown by induction easily that 'piling up' n such approximations gives a success probability of

$$\frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} \left(p_i - \frac{1}{2} \right)$$

- Piling up approximations reduces the bias! So we cannot reach out too far with the piling, we must 'rate' each combination of linear approximations with the piling up probability.

Linear Approximations for n Rounds - Matsui

- Matsui in 1993 suggested a brilliant way to compute the linear approximations that offer the highest bias, and involve no hidden states. The following elegant explanation is due to Federico Lebron:
- Consider each possible linear approximation as a node in a huge graph. We consider a 'small and simple' linear approximation, and combine it with this node, to get another node in the graph. This is a directed edge in the graph.
- We assign a weight to this edge in terms of how it affects the probability of the new node approximation holding. Now, **paths** in the graph correspond to sequences of **successive approximations**, and a distance metric can be used to quantify the effect on the bias.

Linear Approximations for n Rounds - Matsui

- Let a node be 'good' if it doesn't contain any hidden states. To obtain paths that lead to good nodes, we must eliminate the X_is. So, we substitute the i^{th} round with a single round approximation, and XOR the two. These are the 'small and simple' approximations.
- Now we can use any path optimization and graph search algorithms like Dijkstra/Bellman-Ford to compute the best possible paths to good approximations!
- Matsui lists the best possible approximation along with the bias for each value of n from 3 to 20. We can see how the bias values reduce as the number of rounds are increased.
- This shows that larger number of rounds imply less susceptibility to a linear cryptanalytic attack, since they leak much lesser information!

Key Recovery Attacks

- Given a good approximation of the form for $n-1$ rounds,

$$P[\mathbf{p}] \oplus C[\mathbf{c}] = K[\mathbf{k}]$$

- We essentially 'reverse' the last round, regarding it as having been deciphered using K_n, and get an approximation of the form, which holds for n rounds with probability p_{n-1}, *iff* K_n is the right key:

$$P[\mathbf{p}] \oplus C[\mathbf{c}] \oplus F_n(C_L, K_n)[\mathbf{f}] = K[\mathbf{k}] \tag{1}$$

- Now, we look back at the n^{th} round, and locate those bits of K_n which affect the bitmask $\mathbf{f}$. Call these bits effective.

Key Recovery Attacks

- We do an exhaustive search over effective key bits as follows:
- Iterate over all possibilities of effective bits on the RHS of (1). The key set's most likely value which has the most bias in equation (1). This gives us the most likely guess for those bits of K_n (and hence K)
- When this process is done over multiple such linear equations, we can recover enough K_n bits, until the total number of remaining bits is reduced to a tiny number which can be exhausted.
- After recovering K_n, we have reduced the problem to $n-1$ round DES, by reversing the last round using the found K_n
- The probability of success of this guessing method depends on the biases of each of the approximations available and the number of known plaintext pairs available.

Some probabilities

- Due the 5^{th} S-Box's linear approximation, the following linear expression for one Feistel round holds with probability $12/64$:

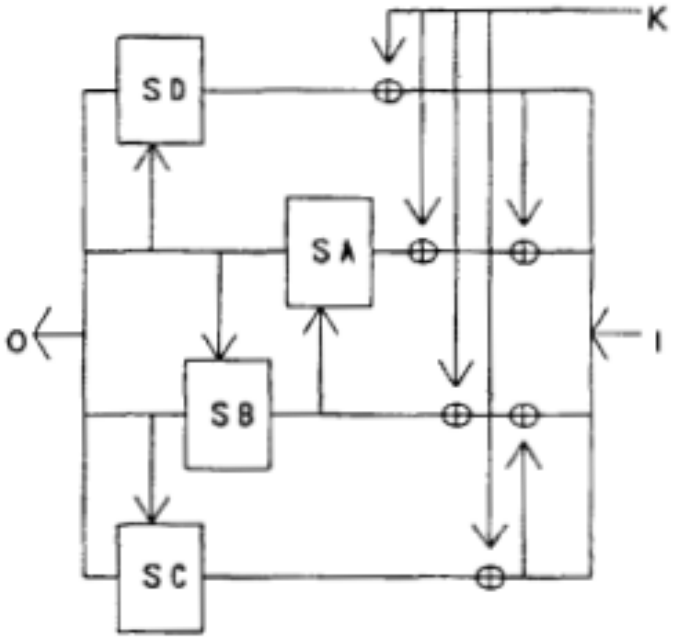
$$X[15] \oplus F(X, K)[7, 18, 24, 29] = K[22]$$

- By applying this approximation to rounds 1 and 3 of 3-round DES, we get the 3 round approximation with probability $(12/64)^2 + (1 - 12/64)^2 = 0.7$. Remember, the actual values of probability aren't important, the biases are!

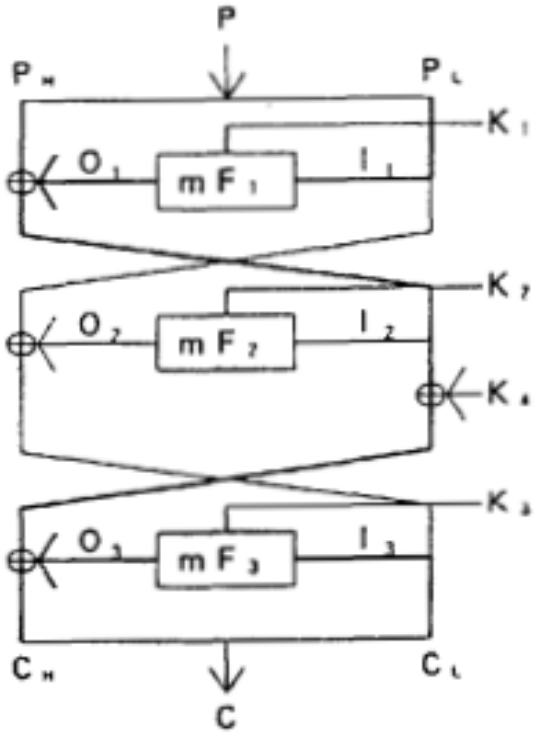
$$P[15, 39, 50, 56, 61] \oplus C[15, 39, 50, 56, 61] = K_1[22] \oplus K_3[22]$$

FEAL

- FEAL (Fast Encryption ALgorithm) is yet another Feistel network algorithm which has non-linear substitution boxes.
- FEAL was shown to be highly insecure against Linear Cryptanalysis by Matsui et al. 4 round FEAL can be broken in under 2 seconds with 10 known plaintexts!



$$SA(x, y) = SC(x, y) = ROL2(x+y+1 \pmod{256})$$



Attacking other Feistel Networks

- No step in Matsui's algorithm is specific to DES, except for the derivation of the single round approximation from the S-Box.
- All we need is a method to find one round approximations!
- For ciphers which involve all non-linearity only in the (small) S-Boxes, exhaustive search through the S-Boxes often reveals vulnerabilities. This is not specific to Feistel networks, it is also true for **Substitution-permutation networks**.
- The challenge is to design S-Boxes which are robust against such attacks.
- The Advanced Encryption Standard (AES) was constructed keeping in mind the possibility of such an attack, so it's S-Boxes show strong security properties against Linear Cryptanalysis.