Differential-Linear Cryptanalysis

See:

Langford and Hellman,
Linear-Differential Cryptanalysis, Crypto '94.

Introduction

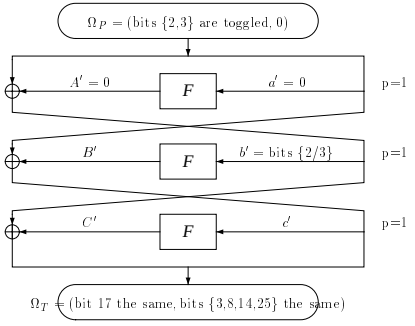
We have witnessed several cases, where we know that some differential occur with probability 1.

We can use such differentials in attacks, like, for example, impossible differential.

We first present an attack which exploits a truncated differential with probability 1.

The Differential Part

Consider the following 3-round truncated differential of DES:



Note that there are more bits with difference 0, but we will be interested only in the marked ones.

The Differential Part (cont.)

Hence, if we toggle bits 2 or 3 (or both) in the left half of input, then in the second round only the input for S1 is affected.

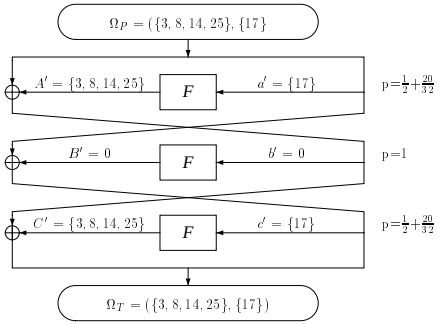
Thus, the output bits of S1 do not affect the inputs to S1 and S7 in the next round.

Therefore, the output differences of S1 and S7 in round 3 are 0.

After the third round, bit 17 (and 7 others) in the left half is equal in both encryptions, as well as bits 3,8,14 and 25 (and 24 others) in the right half.

Linear Approximation

The best linear approximation for 3 rounds of DES found by Matsui is



with probability $p = 0.695$

Observation: The input bits are **a subset of** those which are not affected during the 3-round truncated differential with probability 1.

Question

Can we exploit the above to find an attack?

Concatenating Differential and Linear

If we take a pair of plaintexts with the given input difference, then after the third round both intermediate values has no difference in the bits of the truncated differentials.

Thus, they both have the same value for $\Omega_P \cdot P$.

As both plaintexts are encrypted under the same key, then $\Omega_K \cdot K$ is also the same.

We also know that $\Omega_P \cdot P \oplus \Omega_C \cdot C \oplus \Omega_K \cdot K = 0$ with probability $p = 0.695$.

Thus, the probability that the pair has the same $\Omega_C \cdot C$ is $p^2 + (1-p)^2 = 0.576$.

What's next?

We now describe an attack which retrieves 10 key bits of 8-round DES using about 1,400 pairs. We take pairs which differ in the right half in bit 2/3 or both, and in the left half with some possible difference value for the output of S1.

For each pair we check for each possible subkey whether the input difference of the characteristic is satisfied. Once the input difference is satisfied, we check the output parity of the linear characteristic (by decrypting S1 one round) for each possible value of the subkey of S1 in the 8th round.

We count on those 10 key bits (2 bits are used both in $K_{1,1}$ and in $K_{8,1}$) which satisfy the equality in the parity of the pairs with the highest probability (0.576).

Using the right subkeys, about 0.576 of the pairs should agree on the parity. In the rest of the subkeys, the probability should be lower lower.

Improvements

Using structures can reduce the number of pairs needed. By trying all possible values of output of S1 together, we can reduce the number of plaintexts.

As we play with 6 bits (2 in the left half, and 4 in the right half), we retrieve 96 pairs out of each structure, in the cost of 64 encryptions. Thus we would need only 900 plaintext-ciphertext pairs.

Using 512 chosen plaintexts the attack has 80% success rate. Using 768 chosen plaintexts the attack has 95% success rate.

Also one can try the 10 most frequent values, and increase the success rate dramatically.

We retrieved 10 key bits with cost of 512 encryptions, with work of $2^{10} \cdot 512/32 = 2^{14}$ 8-round DES encryptions.

Retrieving the Full Key

We can continue using exhaustive search over the rest of the key bits. This requires 2^{46} work, and can be reduced by using another differential-linear approximation.

Using the same differential property, one can use the following linear approximation:

$$(L_4[45, 15, 21, 27] \oplus R_4[31]) \oplus (R_7[5, 15, 21, 27] \oplus L_7[29]) = 0$$

with probability p=0.527 (for the pair). This approximation can retrieve another 6 key bits of $K_{8,6}$, as the attack counts $K_{1,1}$ which is already known and $K_{8,6}$.

Using 768 encryptions (in the above structures) can retrieve the additional 6 key bits with success ratio of 85%. The time complexity for those 6 bits is $768 \cdot 2^6/64 = 2^{9.6}$.

Additional bits can be found using other linear approximations (or other differential-linear characteristics).