6 Algebraic Attacks for Few Rounds

Formulas for Few Rounds

We write the recursion formula for a FEISTEL cipher as

$$(L_i, R_i) = (R_{i-1}, L_{i-1} + f(R_{i-1}, k_i))$$

where $k_i = \alpha_i(k)$ is the round key.

Proposition 1 *The results* (L_r, R_r) *of a* FEISTEL *cipher after* $r = 2$, 3, *or* 4 *rounds satisfy the equations*

$$\begin{aligned} L_2 - L_0 &= f(R_0, k_1), \\ R_2 - R_0 &= f(L_2, k_2); \end{aligned}$$

$$\begin{aligned} L_3 - R_0 &= f(L_0 + f(R_0, k_1), k_2), \\ R_3 - L_0 &= f(L_3, k_3) + f(R_0, k_1); \end{aligned}$$

$$\begin{aligned} L_4 - L_0 &= f(R_0, k_1) + f(R_4 - f(L_4, k_4), k_3), \\ R_4 - R_0 &= f(L_4, k_4) + f(L_0 + f(R_0, k_1), k_2). \end{aligned}$$

We used minus signs in order to make the formulas valid also for a generalization to abelian groups. In the (present) binary case plus and minus coincide. The purpose of the formulas is that beside the round keys k_i they involve only the plaintext (L_0, R_0) and the ciphertext (L_r, R_r), data that are assumed as known for algebraic cryptanalysis.

Proof. In the case of two rounds the equations are

$$\begin{aligned} L_1 &= R_0, \\ R_1 &= L_0 + f(R_0, k_1), \\ L_2 &= R_1 = L_0 + f(R_0, k_1), \\ R_2 &= L_1 + f(R_1, k_2) = R_0 + f(L_2, k_2); \end{aligned}$$

the assertion follows immediately.

In the case of three rounds we have

$$\begin{aligned} L_1 &= R_0, \\ R_1 &= L_0 + f(R_0, k_1), \\ L_2 &= R_1 = L_0 + f(R_0, k_1), \\ R_2 &= L_1 + f(R_1, k_2) = R_0 + f(L_2, k_2), \\ L_3 &= R_2 = R_0 + f(L_0 + f(R_0, k_1), k_2), \\ R_3 &= L_2 + f(R_2, k_3) = L_0 + f(R_0, k_1) + f(L_3, k_3). \end{aligned}$$

The case of four rounds is left to the reader. $\diamond$

Two-Round Ciphers

For a known plaintext attack assume that L_0, R_0, L_2, R_2 are given. We have to solve the equations

$$\begin{aligned} L_2 - L_0 &= f(R_0, k_1) \\ R_2 - R_0 &= f(L_2, k_2) \end{aligned}$$

for k_1 and k_2. Thus the security of the cipher only depends on the difficulty of inverting the kernel function f. Since usually q, the bitlength of the partial keys, is much smaller than the total key length l the 2^{q+1} evaluations of f for an exhaustion could be feasible. Note that this consideration doesn't depend on the key schedule α—the attacker simply determines the actually used keybits (k_1, k_2).

Example: We equip $\mathbb{F}_2^s$ with the multiplication "$\cdot$" of the field $\mathbb{F}_t$, $t = 2^s$, [see Appendix A] and take

$$f(x, y) = x \cdot y.$$

(Note that f is non-linear as a whole, but linear in the key bits.) Assume the key schedule is defined by $l = 2q$ and $k_i =$ left or right half of k, depending on whether i is odd or even. Then the equations become

$$\begin{aligned} L_2 - L_0 &= R_0 \cdot k_1, \\ R_2 - R_0 &= L_2 \cdot k_2, \end{aligned}$$

hence are easily solved. (If one of the factors R_0 or L_2 vanishes, we need another known plaintext block.)

Of course chosing a kernel map f that is linear in the key bits was a bad idea anyway. But we could solve also slightly more complicated equations, say quadratic, cubic, or quartic.

Three-Round Ciphers

In the case of three rounds the equations are considerably more complex because f is iterated. However the attacker can mount a Meet-in-the-Middle attack with a single known plaintext, if the bit length q of the partial keys is not too large: She calculates the intermediate results (L_1, R_1) of the first round for all possible partial keys k_1, and stores them in a table. Then she performs an exhaustion over the last two rounds as described for two-round ciphers above. The total expenses are $3 \cdot 2^q$ evaluations of f, and 2^q memory cells.

These considerations suggest that FEISTEL *ciphers should have at least four rounds* and support the above mentioned result by LUBY and RACKOFF. We see how the resistance of the scheme against an algebraic attack increases with the number of rounds, at least if the kernel map f is sufficiently complex.

For the example above with kernel map = multiplication of $\mathbb{F}_{2^s}$ the equations become:

$$\begin{aligned} L_3 - R_0 &= [L_0 + R_0 \cdot k_1)] \cdot k_2, \\ R_3 - L_0 &= [R_0 + R_3] \cdot k_1. \end{aligned}$$

They are nonlinear in the key bits but easily solved in the field $\mathbb{F}_{2^s}$.

Four-Round Ciphers

The equations are much more complex. Even in the example they are quadratic in two unknowns:

$$\begin{aligned} L_4 - L_0 &= [R_0 + R_4 + L_4 \cdot k_2] \cdot k_1, \\ R_4 - R_0 &= [L_4 + L_0 + R_0 \cdot k_1] \cdot k_2. \end{aligned}$$

However in this trivial example they can be solved: eliminating k_1 yields a quadratic equation for k_2 [**Exercise**].