5.8 Systematic Search for Linear Relations

The search for useful linear relations over several rounds has no general elegant solution. The published examples often use linear paths that more or less appear from nowhere, and it is not evident that they are the best ones.

Let n be the block length of the cipher, and r, the number of rounds. Then for each round the choice is between 2^n linear formes, making a total of $2^{n(r+1)}$ choices. This number also specifies the cost of determining the best relation by complete search. There are some simplifications that however don't reduce the order of magnitude of the cost:

- In the first round consider only linear forms that activate only one S-box.
- Then choose the next linear form such that it activates the least possible number of S-boxes of the next round (with high, but not necessarily maximum potential).
- If one of the relations in a linear path has probability $\frac{1}{2}$, or I/O-correlation 0, then the total I/O-correlation is 0 by multiplicativity, and this path may be neglected. The same is true componentwise if the linear forms split among the S-boxes of the respective round. However this negligence could introduce errors since we deal with average probabilities not knowing the key-dependent ones.

For our 2-round example with Mini-Lucifer the systematic search is feasible by pencil and paper or by a Sage or Python script. Our example has the following characteristics:

- $\alpha = (\alpha_1, \alpha_2)$ with $\alpha_1 \mathrel{\hat{=}} 1 \mathrel{\hat{=}} \texttt{0001}$ and $\alpha_2 \mathrel{\hat{=}} 0 \mathrel{\hat{=}} \texttt{0000}$ (α_1 was formerly denoted α. Now for uniformity we make both components of all linear forms explicit and index them by 1 and 2.)
- $\beta = (\beta_1, \beta_2)$ with $\beta_1 \mathrel{\hat{=}} 13 \mathrel{\hat{=}} \texttt{1101}$ and $\beta_2 \mathrel{\hat{=}} 0 \mathrel{\hat{=}} \texttt{0000}$
- $\beta' = (\beta'_1, \beta'_2)$ with $\beta'_1 \mathrel{\hat{=}} 1 \mathrel{\hat{=}} \texttt{0001}$, $\beta'_2 \mathrel{\hat{=}} 12 \mathrel{\hat{=}} \texttt{1100}$
- $\gamma = (\gamma_1, \gamma_2)$ with $\gamma_1 \mathrel{\hat{=}} 13 \mathrel{\hat{=}} \texttt{1101}$, $\gamma_2 \mathrel{\hat{=}} 6 \mathrel{\hat{=}} \texttt{0110}$
- $\tau_1 = \frac{3}{4}$, $\tau'_2 = \frac{3}{4}$, $\tau''_2 = \frac{1}{2}$, $\tau_2 = \frac{3}{8}$, $\tau = \frac{9}{32}$, $p = \frac{41}{64} = 0,640625$
- $c_0 + c_1 + c_3 + c_5 + c_6 \overset{p}{\approx} a_3 + k_0 + k_1 + k_5 + k_6 + k_{11} + k_{12} + k_{13}$

An alternative choice of γ_2 is $\gamma_2 \mathrel{\hat{=}} 14 \mathrel{\hat{=}} \texttt{1110}$; this yields a linear path with the characteristics

- $\alpha \mathrel{\hat{=}} (1, 0)$, $\beta \mathrel{\hat{=}} (13, 0)$, $\beta' \mathrel{\hat{=}} (1, 12)$, $\gamma \mathrel{\hat{=}} (13, 14)$

- $\tau = -\frac{9}{32}$, $p = \frac{23}{64} = 0,359375$
- $c_0+c_1+c_3+c_4+c_5+c_6 \overset{p}{\approx} a_3+k_0+k_1+k_4+k_5+k_6+k_{11}+k_{12}+k_{13}$

The systematic search finds two even "better" linear paths, characterized by

- $\alpha \mathrel{\hat{=}} (8,0)$, $\beta \mathrel{\hat{=}} (8,0)$, $\beta' \mathrel{\hat{=}} (1,0)$, $\gamma \mathrel{\hat{=}} (13,0)$
  - $\tau = -\frac{3}{8}$, $p = \frac{5}{16} = 0,3125$
  - $c_0 + c_1 + c_3 \overset{p}{\approx} a_0 + k_1 + k_3 + k_{11}$
- $\alpha \mathrel{\hat{=}} (15,0)$, $\beta \mathrel{\hat{=}} (8,0)$, $\beta' \mathrel{\hat{=}} (1,0)$, $\gamma \mathrel{\hat{=}} (13,0)$
  - $\tau = -\frac{3}{8}$, $p = \frac{5}{16} = 0,3125$
  - $c_0 + c_1 + c_3 \overset{p}{\approx} a_0 + a_1 + a_2 + a_3 + k_2 + k_{11}$

that do not completely exhaust the potential of the single S-boxes but on the other hand activate only one S-box of the second round, and thereby show the larger potential $\lambda = \frac{9}{64}$. Thus we get a 95% success probability with only

$$N = \frac{3}{\lambda} = \frac{64}{3} \approx 21$$

known plaintexts for determining one bit.

The designer of a cipher should take care that in each round the active bits fan out over as many S-boxes as possible. The inventors of AES, Daemen and Rijmen call this design approach "wide-trail strategy". The design of AES strengthens this effect by involving a linear map instead of a mere permutation, thereby replacing the "P" of an SP-network by an "L".

Figure 5.11 shows an example of a linear path with all its ramifications.

Example (Continued)

For an illustration of the procedure we generate 25 pairs of known plaintexts and corresponding ciphertexts using the key $k \mathrel{\hat{=}}$ `1001011000101110`. The target key bits are

$$\begin{aligned} b_0 &= k_0 + k_1 + k_5 + k_6 + k_{11} + k_{12} + k_{13} \\ b_1 &= k_0 + k_1 + k_4 + k_5 + k_6 + k_{11} + k_{12} + k_{13} \\ b_2 &= k_1 + k_3 + k_{11} \\ b_3 &= k_2 + k_{11} \end{aligned}$$

that we know in cheat mode are $b_0 = 1$, $b_1 = 1$, $b_2 = 1$, $b_3 = 0$. We use all four good relations at the same time without fearing the possible reduction

of the success probability. All of these relations assert the probable equality of the bits

$$\begin{aligned} b_0 &\overset{p}{\approx} c_0 + c_1 + c_3 + c_5 + c_6 + a_3 \\ b_1 &\overset{p}{\approx} 1 + c_0 + c_1 + c_3 + c_4 + c_5 + c_6 + a_3 \\ b_2 &\overset{p}{\approx} 1 + c_0 + c_1 + c_3 + a_0 \\ b_3 &\overset{p}{\approx} 1 + c_0 + c_1 + c_3 + a_0 + a_1 + a_2 + a_3 \end{aligned}$$

each with its individual corresponding probability p. For the last three of these sums we have to take the complementary bits since the corresponding I/O-correlations are negative (the probabilities are $< \frac{1}{2}$). This is done by adding the bit `1`.

Table 5.13 shows the results for these plaintext-ciphertext pairs. As we see our guess is correct for all four bits.

As a consequence of our analysis we get a system of four linear equations for the 16 unknown key bits:

$$\begin{aligned} 1 &= k_0 + k_1 + k_5 + k_6 + k_{11} + k_{12} + k_{13} \\ 1 &= k_0 + k_1 + k_4 + k_5 + k_6 + k_{11} + k_{12} + k_{13} \\ 1 &= k_1 + k_3 + k_{11} \\ 0 &= k_2 + k_{11} \end{aligned}$$

that allow us to reduce the number of keys for an exhaustion from $2^{16} = 65536$ to $2^{12} = 4096$. Note the immediate simplifications of the system: $k_{11} = k_2$ from the last equation, and $k_4 = 0$ from the first two.

As a cross-check we run some more simulations. The next four yield

- $15, 16, 19, 16$
- $15, 16, 13, 17$
- $15, 20, 19, 17$
- $19, 19, 20, 18$

correct guesses, and so on. Only run number 10 produced a wrong bit (the second one):

- $17, 12, 14, 17$

then again run number 25. Thus empirical evidence suggests a success probability of at least 90% in this scenario.

nr	plaintext	ciphertext	b_0	b_1	b_2	b_3
1	00001111	00001010	1	1	1	1
2	00010001	11001110	1	1	1	0
3	00010110	11001001	1	1	1	0
4	00111101	10110010	0	1	1	1
5	01000000	11100111	0	1	1	0
6	01001000	01010111	0	1	1	0
7	01001100	11101010	1	1	1	0
8	01001101	01011100	1	1	1	0
9	01001111	01111010	1	1	1	0
10	01100111	00110011	0	1	0	0
11	10000011	11110100	0	1	1	1
12	10010011	01101011	1	1	1	0
13	10011000	01100111	0	1	1	0
14	10101011	11011001	1	1	1	0
15	10110001	11001000	1	1	0	0
16	10110010	10100100	1	0	1	1
17	10110110	11000100	0	1	0	0
18	10111001	11000001	1	0	0	0
19	10111101	10111111	1	1	0	0
20	11000100	01001111	1	1	1	0
21	11000111	00111111	1	1	1	0
22	11011111	11011010	1	1	1	1
23	11100000	11101110	0	0	0	0
24	11100100	01110011	1	0	0	0
25	11110101	11110101	1	0	1	0
		true bit:	1	1	1	0
		correct guesses:	17	20	18	20

Table 5.13: Plaintext/ciphertext pairs for Mini-Lucifer

Analysis over Four Rounds

Now let's explore how an increasing number of rounds impedes linear cryptanalysis.

Consider the toy cipher Mini-Lucifer over four rounds. Searching an optimal linear path over four rounds is somewhat expensive, so we content ourselves with extending the best example from the two round case, the third one, over two additional rounds. Slightly adapting the notation we get:

- for the first round $\beta_0 = \alpha \mathrel{\hat{=}} (8, 0)$ and $\beta_1 \mathrel{\hat{=}} (8, 0)$ (the "old" β) with $\tau_1 = -\frac{1}{2}$,
- for the second round (applying the permutation P to β_1) $\beta_1' \mathrel{\hat{=}} (1, 0)$ and $\beta_2 \mathrel{\hat{=}} (13, 0)$ (the "old" γ) with $\tau_2 = \frac{3}{4}$,
- for the third round $\beta_2' \mathrel{\hat{=}} (1, 12)$ and $\beta_3 \mathrel{\hat{=}} (13, 6)$ with $\tau_3 = \frac{3}{8}$,
- for the fourth round $\beta_3' \mathrel{\hat{=}} (5, 13)$ and $\beta = \beta_4 \mathrel{\hat{=}} (3, 12)$ (the "new" β) with $\tau_4 = -\frac{1}{4}$.

Figure 5.11 shows this linear path with its ramifications.

The repeated round keys we used are not independent. Therefore multiplicativity of I/O-correlations is justified by the rule of thumb only yielding an approximate value for the I/O-correlation of the linear relation (α, β) over all of the four rounds:

$$\tau \approx \frac{1}{2} \cdot \frac{3}{4} \cdot \frac{3}{8} \cdot \frac{1}{4} = \frac{9}{256} \approx 0,035.$$

The other characteristics are

$$p \approx \frac{265}{512} \approx 0,518, \quad \lambda \approx \frac{81}{65536} \approx 0,0012, \quad N \approx \frac{65536}{27} \approx 2427,$$

the last one being the number of needed known plaintexts for a 95% success probability.

Comparing this with the cost of exhaustion over all 65536 possible keys we seem to have gained an advantage. However there are only 256 different possible plaintexts all together. So linear cryptanalysis completely lost its sense by the increased number of rounds.

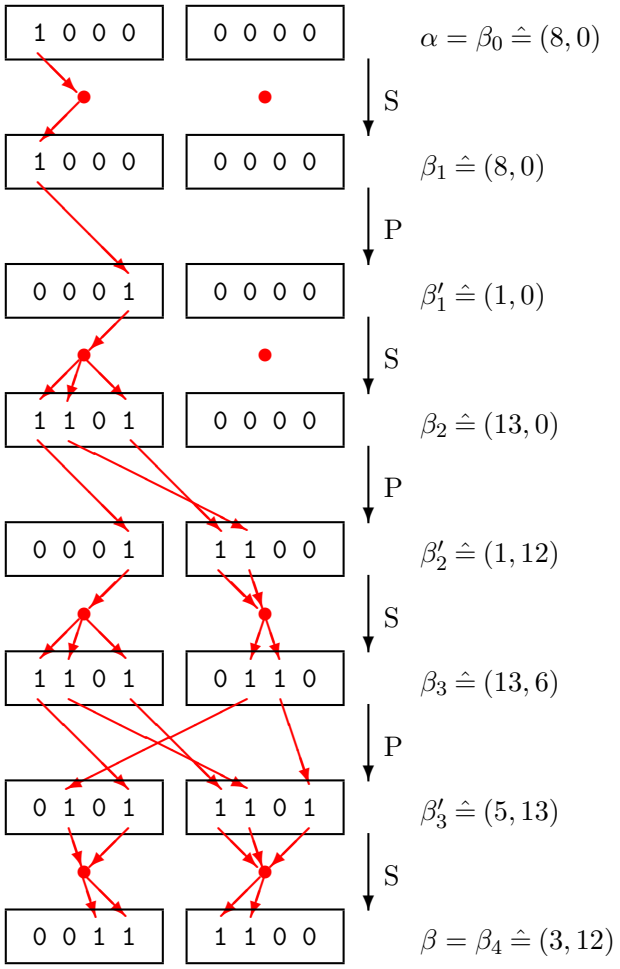

Figure 5.11: A linear path with ramifications ("trail"). For S the linear form in the range is *chosen* (for high potential), indicated by a red dot. For P the linear form in the range results by applying the permutation.