Exercise Sheet 7

Hellman Tables

Obligatory homework this week: Exercise 3. Hand-in by November 7, 23:59.

Exercise 1: Success Probability
In general, the success probability of the Hellman Table method, using ℓ tables with m rows and t columns, is given by the formula:

$$P_H = 1 - \exp\left(-\sqrt{\frac{2m\ell^2}{2^l}} \cdot \frac{\exp\left(\sqrt{2mt^2/2^l}\right) - 1}{\exp\left(\sqrt{2mt^2/2^l}\right) + 1}\right) \tag{1}$$

Here, l is the size the key in bits.

Consider a version of AES-128 where 104 bits of the input are fixed to zero, i.e. the effective key-length is 24 bits. Fix the number of tables $\ell = 2^8$ and let $mt\ell = 2^{24}$. Plot the success probability as a function of the number of rows m. What parameters of m and t seem reasonable to you? How does this relate to the online and memory complexity?

Exercise 2: Low Memory TMTO
Consider a time-memory trade-off against AES-128 using Hellman Tables. What is the lowest amount of memory you can use, and still achieve a faster online phase than brute force, with a reasonable success probability?

Exercise 3: Coverage of Hellman Tables
Consider the 24-bit key version of AES-128 described in Exercise 1. Create a function

$$f(k) : [0, 2^{24} - 1] \rightarrow [0, 2^{24} - 1]$$

which encrypts a fixed plaintext with AES-128 using the 24-bit key k, and then restricts the output to 24 bits (e.g. by throwing away the last 104 bits). Then define $f_i(k) = (f(k)+i) \mod 2^{24}$.

Your task is now to calculate the coverage of a set of Hellman Tables for this version of AES. Fix the number of tables $\ell = 2^8$. The i'th table should use $f_i(k)$ as its reduction function.

Using the results of Exercise 1, choose a reasonable value for m. Keep track of how many points in $[0, 2^{24} - 1]$ the tables cover, for each iteration of $f_i(k)$ (i.e. each step of the $m \cdot \ell$ chains). Make a graph over how the number of covered points develops over time. Does the graph match the predictions made by Equation 1? (Remember that the coverage is just $P_H \cdot 2^l$)

Your hand-in (via CampusNet) should include:

1. source code of your Hellman Table implementation,
2. a short report containing your results and discussion of these.

Group work is strongly encouraged, but please make sure to hand in individually.

NOTE: Exercise sheet 8 will contain the second part of this homework!