ENEE/CMSC/MATH 456 Cryptography: Homework 4

Due by 2pm on 4/10/2024.

1. In our attack on a one-round SPN, we considered a block length of 64 bits and 8 S-boxes, each taking an 8-bit input. Repeat the analysis for the case of 16 S-boxes, each taking a 4-bit input. What is the complexity of the attack now? Repeat the analysis again with a 128-bit block length and 16 S-boxes that each take an 8-bit input.

2. In this question we assume a three-round SPN with 64-bit block length. Assume independent 64-bit sub-keys are used in each round, so the master key is 256 bits long. Show a key-recovery attack using approximately $2 \cdot 128 \cdot 2^{128}$ time. The number and input size of the S-boxes is not needed to answer the question.
   Hint: Use a meet-in-the-middle attack.

3. What is the output of an r-round Feistel network when the input is (L_0, R_0) in each of the following two cases:

   (a) Each round function outputs all 0s, regardless of the input.

   (b) Each round function is the identity function.

4. In this question, you are asked to recover the key for a 1-round SPN with 6-bit input, 6-bit output and 6-bit key, given a single input-output pair. It is possible that there is more than one consistent key-pair based on the given information. If this is the case then all consistent key-pairs will be marked correct. Make sure to show all work.

The SPN has the following structure:

To compute the permutation $F_k(x)$ on input x (6 bits) with key k (12 bits):

- Parse $k = k^1 || k^2$, where k^1 and k^2 are the round keys and each have length 6 bits.
- Compute the intermediate value $z = x \oplus k^1$.
- Parse $z = z_1 || z_2$, where z_1 and z_2 each have length 3 bits.
- For each $i \in [2]$, input z_i to the corresponding S-box S_i defined below, obtaining outputs w_1, w_2. Let $w = w_1 || w_2$ (length 6 bits) be the combined output.
- Permute the bits of w to obtain w' as described in the chart below.
- Output $y = w' \oplus k^2$.

S-box S_1:

000	100
001	111
010	010
011	000
100	011
101	101
110	001
111	110

S-box S_2:

000	110
001	111
010	011
011	101
100	000
101	010
110	100
111	001

The following chart shows how the 6 bits of w are permuted to obtain w'.

1	2	3	4	5	6
3	5	2	6	1	4

Namely, on input $w := w_1, w_2, w_3, w_4, w_5, w_6$, we permute the bits to obtain output $w' := w_3, w_5, w_2, w_6, w_1, w_4$. Assume you are given that $F_k(111010) = 010011$ and $F_k(011100) = 101010$. Additionally, you are given that the first bit of k^1 is equal to 1 and the fourth bit of k^1 is equal to 0. So k^1 has the form $1 * * 0 * *$. Find $k = k^1 || k^2$.

Given the above information, there is an attack that requires you to evaluate the SPN at most 16 times. Solutions that recover the correct key but take longer, may not receive full credit.

5. A student has 3,500 songs on her phone, and chooses songs to play at random. How many songs should the student expect to play before hearing some song twice (with probability at least 50%)? Use the upper and lower bounds on the probability of collision given in Lemma A.15.

6. Sample uniform $y_1, \ldots, y_q \in \{0,1\}^\ell$ and $y'_1, \ldots, y'_q \in \{0,1\}^\ell$. What is the expected number of collisions across the two lists. I.e. the expected number of pairs $(i,j) \in [q] \times [q]$ such that $y_i = y'_j$?

7. Let $H : \{0,1\}^n \to \{0,1\}^{2n}$ be a random function, and define the keyed function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{2n}$ by $F_k(x) = H(k \oplus x)$. Show that an attacker given oracle access to $F_k(\cdot)$ can expect to recover the n-bit key after making $\approx 2^{n/2}$ oracle queries.

8. Prove formally that the hardness of the CDH problem relative to G implies the hardness of the discrete logarithm problem relative to G.

9. Determine the points on the elliptic curve $E : y^2 = x^3 + 2x + 1$ over Z_{11}. How many points are on this curve?

10. Can the following problem be solved in polynomial time? Given a prime p, a value $x \in Z_{p-1}^*$ and $y := g^x \bmod p$ (where g is a uniform value in Z_p^*), find g, i.e., compute $y^{1/x} \bmod p$. If your answer is "yes," give a polynomial-time algorithm. If your answer is "no," show a reduction to one of the assumptions introduced in this chapter.