Homework Assignment # 4

Due Date: Monday, October 4

0. Read the document "Status Report on the First Round of the Development of the Advanced Encryption Standard", which can be found on the NIST web pages at `csrc.nist.gov/encryption/aes/round1/r1report.htm`

1. In this problem we want to study the clock frequency requirements for a hardware implementation of DES in real-world applications. The speed of a DES implementation is mainly determined by the time required to do one core iteration. This hardware kernel is then used 16 consecutive times in order to generate the encrypted output. (An alternative approach would be to build a hardware pipeline with 16 stages, resulting in 16-fold increased hardware costs.)

   (a) Let's assume that one core iteration can be performed in one clock cycle. Develop an expression for the required clock frequency for encrypting a stream of data with a data rate r [bits/sec]. Ignore the time needed for the initial and final permutation.

   (b) Which clock frequency is required for encrypting a network link running at an ATM speed of 155 Mb/sec? What is the clock frequency if we want to support ATM at 622 Mb/sec? Are these clock rates realistic? (no points on the last question)

   (c) How many DES chips do we have to use in parallel if we want to encrypt a 1 Gb/sec data stream and the chips can only operate at a clock frequency of up to 22 MHz?

2. A popular approach for finding an upper security threshold of a private-key cipher against a brute force attack is estimating the cost of key search machine. We will study this approach in this problem.

   Assume a DES chip with pipelined hardware so that one encryption (or key test) can be done in one clock cycle. If we clock the system at 50 MHz, how many chips must run in parallel for an average search time of 24 hours (assume that a worst case run searches through 2^{56} keys). What is the cost for such a machine if one chip costs \$10 and we calculate a 100% overhead on this price for connection the chips and building the machine?

   Why does any design of a key-search machince constitute only an upper security threshold? By *upper security threshold* we mean a (complexity) measure which describes the maxium security that is provided by a given cryptographic algorithm.

3. (a) Assume another block cipher with a key length of 128 bits such as IDEA. Again, assume that we have a special purpose chip which searches $5 \cdot 10^7$ keys per second. We parallelize 100,000 of the chips. How long does an average key search take? Relate this time to the age of the universe (about 10^{10} years).

   (b) We try now to take the advances in computer technology into account. Prediciting the future tends to be tricky, but the estimate usually applied is Moore's Law, according to which computer power doubles every 18 months. How many years do we have to wait until a key search machine can be build for a private-key algorithm with 128 bits with an average search time of 24 hours? We assume that 100,000 of these chips are used in parallel in our machine.

4. Consider the storage of data in encrypted form in a large database using DES. One record has a size of 64 bits. Assume that the records are not related to oneanother. Which mode would be best suited? Why? (Short answers are sufficient.)

5. We consider known-plaintext attacks on block ciphers by means of an an exhaustive key search, where the key is k bits long. Any ECB mode cipher can be broken in a straightforward manner in 2^k steps using one pair (x, y) (for simplicity assume that the block length is much longer than the key length.)

   (a) Let's assume we do not know the vector IV in the CBC mode. This seems to impose another difficulty. Describe how many pieces of (i) plaintext and (ii) ciphertext are at least required in order to break a CBC cipher by an exhaustive key search. How many search steps are required in a worst-case scenario?

   (b) Is breaking a block cipher in CBC mode by means of an exhaustive key search considerably more difficult than breaking an ECB mode block cipher?

6. One important issue when chosing a mode of operation in practice is error propagation.

   (a) Assume an error occurs during transmission in one block of ciphertext, let's say y_i. Which cleartext blocks are affected on Bob's side when using the ECB mode?

   (b) Again, assume block y_i contains an error introduced during transmission. Which cleartext blocks are affected on Bob's side when using the CBC mode?

   (c) Suppose there is an error in the cleartext x_i on *Alice's* side. Which cleartext blocks are affected on Bob's side when using the CBC mode?

   (d) Assume a single bit error occurs in the transmission of a ciphertext character in 8-bit CFB mode. How far does the error propagate? Describe exactly *how* each block is affected.

7. We want to attack data which has been double encrypted with the CAST block cipher. CAST has a key length of 64 bits and an I/O block size of 64 bits. Using the meet-in-the-middle attack, how many pairs (x, y) should be available so that the probability to determine an incorrect key pair (k_1, k_2) is suffienctly low for a practical attack?

8. Triple-DES with three different keys can be broken with about 2^{2k} encryptions and 2^k memory cells, $k = 56$. Design the corresponding attack. How many pairs (x, y) should be available so that the probability to determine an incorrect key triple (k_1, k_2, k_3) is suffienctly low?

9. (**Optional problem, 20 extra points**)

   This is your chance to break a cryptosystems. As we know by now, cryptography is a tricky business. The following problem illustrates how easy it is to turn a strong scheme into a weak one with minor modifications.

   We saw in the lecture that key whitening is a good technique for strengthening block ciphers against brute-force attacks. We now look at the following variant of key whitening against DES, which we'll call DESA:

   $$DESA_{k,k_1}(x) = DES_k(x) \bigoplus k_1$$

   Even though the method looks similar to key whitening, it hardly adds to the security. Your task is to show that breaking the scheme is roughly as difficult as a brute force attack against single DES. Assume you have a few pairs of plaintext/ciphertext.