CS255: Cryptography and Computer Security **Winter 2000**

Assignment #1

Due: Wednesday, January 26th, 2000.

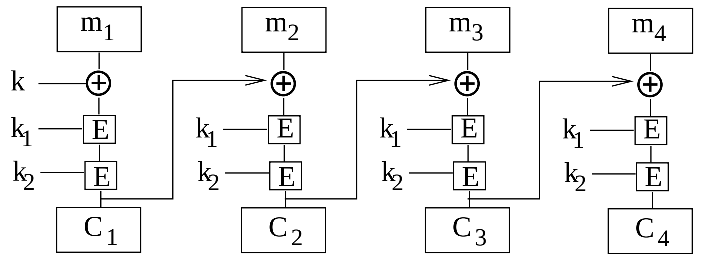
Problem 1 Data compression is often used in data storage or transmission. Suppose you want to use data compression in conjunction with encryption. Does it make more sense to

A. Compress the data and then encrypt the result, or

B. Encrypt the data and then compress the result.

Justify your answer. Try to give at least two reasons.

Problem 2 Let E, D be the encryption/decryption algorithms of a certain block cipher. Consider the following chaining method for double DES like encryption:



The secret key is a triple (k, k_1, k_2) where k is as long as E's block size (64 bits for DES) and k_1, k_2 are as long as E's key size (56 bits for DES). For example, when E is DES the total key size is 64+56+56 = 176 bits.

a. Describe the decryption circuit for this system.

b. Suppose an attacker is given one *random* (plaintext,ciphertext) pair $\langle M, C\rangle$ where M is n blocks long (say, 100 blocks, $M = m_1 \dots m_{100}$). Show that the adversary can recover the full key (k, k_1, k_2) using approximately $2n \times 2^\ell$ runs of algorithm D where ℓ is the length of the block cipher's key (56 bits for DES). Your attack algorithm may use as much space as you like. Your attack shows that this system can be broken much faster than exhaustive search.

Problem 3 Before DESX was invented, the researchers at RSA Labs came up with DESV and DESW, defined by

$$\begin{aligned} DESV_{kk_1}(M) &= DES_k(M) \oplus k_1 \text{ and} \\ DESW_{kk_1}(M) &= DES_k(M \oplus k_1) \end{aligned}$$

As with DESX, $|k| = 56$ and $|k_1| = 64$. Show that both these proposals do not increase the work needed to break the cryptosystem using brute-force key search. That is, show how to break these schemes using on the order of 2^{56} DES encryptions/decryptions. You may assume that you have a moderate number of plaintext-ciphertext pairs, $C_i = DES\{V/W\}_{kk_1}(M_i)$.

Problem 4 Given a cryptosystem E_k, define the randomized cryptosystem F_k by

$$F_k(M) = (E_k(R), R \oplus M),$$

where R is a random bit string of the same size as the message. That is, the output of $F_k(M)$ is the encryption of a random one-time pad along with the original message XORed with the random pad. A new independent random pad R is chosen for every encryption.

We consider two attack models. The goal of both models is to reconstruct the actual secret key k.[1]

- In the key-reconstruction chosen plaintext attack (KR-CPA), the adversary is allowed to generate strings $M_1, M_2, \ldots$ and for each M_i learn a corresponding ciphertext.
- In the key-reconstruction random plaintext attack (KR-RPA), the adversary is given random plaintext/ciphertext pairs.

Note that for the case of F_k the opponent has no control over the random pad R used in the creation of the given plaintext/ciphertext pairs. Clearly a KR-CPA attack gives the attacker more power than a KR-RPA attack. Consequently, it is harder to build cryptosystems that are secure against KR-CPA.

Prove that if E_k is secure against KR-RPA attacks then F_k is secure against $\mathsf{KR}-\mathsf{CPA}$ attacks.

Hint: It is easiest to show the contrapositive. Given an algorithm A that executes a successful $\mathsf{KR}-\mathsf{CPA}$ attack against F_k, construct an algorithm B (using A as a "subroutine") that executes a successful $\mathsf{KR}-\mathsf{RPA}$ attack against E_k. First, define precisely what algorithm A takes as input, what queries it makes, and what it produced as output. Do the same for B. Then construct an algorithm B that runs A on a certain input and properly answers all of A's queries. Show that the output produced by A enables B to complete the $\mathsf{KR}-\mathsf{RPA}$ attack against E_k.

Extra credit Show that if all the S-boxes in the DES cipher are replaced by an identify function (i.e. given a six-bit input the function outputs the four least significant bits) then the resulting cipher is easily broken. In other words, show that given a few plaintext/ciphertext pairs an attacker can quickly recover the secret key.

[1] This is a very strong goal - one might be able to decrypt messages without ever learning k.