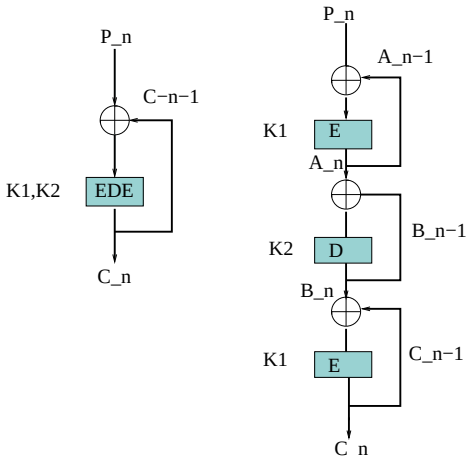
Introduction to Computer Security, exercise 5, April 26-30, 2010

1. Examine what the standard
   NIST SP-800-88 (http://csrc.nist.gov/publications/nistpubs/800-88/NISTSP800-88_rev1.pdf)
   says about cleaning harddisks and other memories. What kind of techniques are in use? Is it always necessary to overwrite many times in order to quarantee that the memory has been cleaned?

2. An older cipher, as for example DES, can still be used, if the encryption is done three times with two or three keys. A typical method is to encrypt with one key, then to decrypt with another key, and finally still to encrypt with the first key. If the encryption is done this way, the method is backwards compatible with a single encryption. Why?

   Let us use now the CBC mode. The methods in figure **??** show two possible ways to do this. Which of the two would you choose a) for securitry, b) for performance? Can you suggest a security improvement to either option, using only three DES chips and some number of XOR functions? Assume you are still limited to two keys.



Kuva 1: 3DES ja CBC

3. Suppose that there is a bit error in the source version of the plaintext. Through how many ciphertext blocks is this error propagated? What is the effect at the receiver? Consider both CBC abd CF.

4. One approach to defeating the tiny fragment attack against firewalls is to enforce a minimum length of the transport header that must be contained in the first fragment of an IP packet. If the first fragment is rejected, all susequent fragments can be rejected. However, the nature of IP is such that fragments may arrive out of order. Thus, an intermediate fragment may pass through the filter before the initial fragment is rejected. How can this situation be handled?

5. Consider the following simple protocol, where A sends an encrypted message M to B using public key cryptography.

   (a) $A \longrightarrow B$: (A, $E(PU_B, [M, A])$, B)

   (b) $B \longrightarrow A$: (B, $E(PU_A, [M, B])$, A)

   Thus A sends a message to B and the message contains sender and receiver in plaintext plus message M and sender A encrypted by B's public key. B acknowledges in the same way. Simplify the protocol as folow an attackerlows:

   (a) $A \longrightarrow B$: (A, $E(PU_B, M)$, B)

   (b) $B \longrightarrow A$: (B, $E(PU_A, M)$, A)

   Thus a sender is no more in the encrypted part. Now an attacker can decrypt M, if he can listen to the data traffic and he also has a public key. How?