T-79.159 Cryptography and Data Security **Spring 2003**
Tutorial 3: Solutions
4.–5.2.2003

1. Recall that CBC-mode works as follows. To encrypt $(x_1, \ldots, x_n)$, select a random IV y_0 and compute $y_i = E_K(x_i \oplus y_{i-1})$ for $i = 1, \ldots, n$. The ciphertext is $(y_0, y_1, \ldots, y_n)$. To decrypt the message, compute $x_i = E_K^{-1}(y_i) \oplus y_{i-1}$ for $i = 1, \ldots, n$. The plaintext is $(x_1, \ldots, x_n)$.

   The modified version works as follows. To encrypt a message $(x_1, \ldots, x_n)$, pick a random IV and denote it $x_0 = y_0$ (in CBC decryption, the IV is part of the ciphertext, but here we have to choose it) and compute $y_i = E_K^{-1}(x_i) \oplus x_{i-1}$ for $i = 1, \ldots, n$. The ciphertext is $(y_0, y_1, \ldots, y_n)$. To decrypt the message, set $x'_0 = y_0$ and compute $x'_i = E_K(y_i \oplus x'_{i-1})$ for $i = 1, \ldots, n$. Since $x'_0 = x_0$, we see by induction that $x'_i = E_K((E_K^{-1}(x_i) \oplus x_{i-1}) \oplus x'_{i-1}) = E_K(E_K^{-1}(x_i)) = x_i$. Thus, this modified CBC mode works in the sense that decryption is the inverse of encryption.

   The security is drastically decreased by this modification. Suppose that the plaintext is $(x_1, x_2) = (0, 0)$. Then the ciphertext is (y_0, y_1, y_2), where y_0 is the IV, $y_1 = E_K^{-1}(0) \oplus y_0$ and $y_2 = E_K^{-1}(0) \oplus 0 = E_K^{-1}(0)$. Thus, $y_2 = y_1 \oplus y_0$ for all choices of the IV. This means that we can recognise encryptions of $(0, 0)$ with very high probability by checking if $y_2 = y_1 \oplus y_0$. This is not the case for CBC mode.

   Roughly speaking, swapping the roles of encryption and decryption means that chosen-plaintext attacks against the modified scheme corresponds to chosen-ciphertext attacks against the original scheme. Thus, these kinds of modifications are generally not advisable.

2. The hint given in the problem statement was perhaps a bit misleading. We will instead consider the case where the attacker knows that a given ciphertext corresponds to a small set of possible plaintexts (e.g. "attack now" or "wait").

   The attacker is given a ciphertext $y = (y_0, y_1, \ldots, y_n)$. Suppose that the attacker wants to check if the first plaintext block is x_1. That is, he wants to check if $y_1 = E_K(x_1 \oplus y_0)$. If the attacker can predict the next IV y'_0, he can set $x'_1 = x_1 \oplus y_0 \oplus y'_0$ and get the encryption (y'_0, y'_1) of x'_1 (one chosen plaintext). If the first plaintext block of y indeed is x_1, we see that $y'_1 = E_K(x'_1 \oplus y'_0) = E_K(x_1 \oplus y_0) = y_1$. This can easily be checked by the attacker.

   If the IV is unpredictable, this attack is not possible.

3. Let (x_{i-1}, x_i, x_{i+1}) be the plaintext blocks containing the message, target account number and amount, respectively. Let (y_{i-1}, y_i, y_{i+1}) be the corresponding ciphertext blocks. Let α be your account number. Replace the three ciphertext blocks with the blocks $(y'_{i-1}, y'_i, y'_{i+1}) = (y_{i-1} \oplus x_i \oplus \alpha, y_i, y_{i+1})$. This does not affect any plaintext block before the $(i-1)$th. We see that $x'_{i-1} = E_K^{-1}(y_{i-1} \oplus x_i \oplus \alpha) \oplus y_{i-2}$ is garbage, $x'_i = (E_K^{-1}(y_i) \oplus y_{i-1}) \oplus x_i \oplus \alpha = x_i \oplus x_i \oplus \alpha = \alpha$ is your account number, and $x'_{i+1} = E_K^{-1}(y_{i+1}) \oplus y_i = x_{i+1}$ is unchanged. The remaining plaintext

blocks are also unchanged. That is, this turns the message into garbage, changes the target account number to your account number and leaves the remaining plaintext blocks unchanged. If the transaction is processed by machine, the garbage message is probably not detected.

4. Recall that 3EDE-DES encryption is defined by

$$E_K(x) = \mathrm{DES}_{K_3}(\mathrm{DES}^{-1}_{K_2}(\mathrm{DES}_{K_1}(x))) \ .$$

The block and key sizes of DES are $n = 64$ and $k = 56$. Thus, the total key length is $3 \cdot 56 = 168$ bits.

Suppose we are given a few known plaintext-ciphertext pairs (P_i, C_i). The attack has two stages.

- For all the 2^{56} values of K_1, compute $\mathrm{DES}_{K_1}(P_1)$. Store each value of K_1 in a table T indexed by $\mathrm{DES}_{K_1}(P_1)$. That is, $T[\mathrm{DES}_{K_1}(P_1)] = K_1$. This requires 2^{56} blocks of memory.
- For all values of (K_2, K_3), compute $M = \mathrm{DES}_{K_2}(\mathrm{DES}^{-1}_{K_3}(C_1))$ (that is, decrypt C_1 trough two of the stages). Check if M is found from the table T. If it is found, the table gives the guess $K_1 = T[M]$. Using a few other known plaintext-ciphertext pairs (P_i, C_i), check that P_i is mapped to C_i under (K_1, K_2, K_3). If the check succeeds, the key has been found. Otherwise, try the next value of (K_2, K_3). This part takes 2^{112} steps and uses a small amount of memory.

Since most of the time is spend in the second stage, this finds the key (K_1, K_2, K_3) using about 2^{112} time and 2^{56} memory.

5. There are several popular was to combine a MAC and an encryption scheme to get an authenticated encryption scheme. In encrypt-then-MAC, the plaintext is first encrypted and then the MAC is applied to the ciphertext (used e.g. in IPSec). In MAC-then-encrypt, the MAC is first applied to the plaintext and then the plaintext and the output of the MAC are encrypted (used e.g. in TLS). In encrypt-and-MAC the MAC is applied to the plaintext, only the plaintext is encrypted, and both the encrypted plaintext and the unencrypted MAC are send (used in SSH2). Think about how you would decrypt the ciphertext and verify the MAC in each case. For simplicity, we only consider encrypt-then-MAC and assume that the encryption scheme is CBC-mode in this exercise. That is, we consider the case where the plaintext is encrypted in CBC-mode and then the ciphertext is authenticated with CBC MAC using the *same key*.

Let the plaintext be $(x_1, \ldots, x_n)$. The ciphertext is $(y_0, y_1, \ldots, y_n)$, where y_0 is the IV and $y_i = E_K(x_i \oplus y_{i-1})$ for $i = 1, \ldots, n$. The ciphertext is then authenticated by computing $z_{-1} = 0$ and $z_i = E_K(y_i \oplus z_{i-1})$ for $i = 0, \ldots, n$. The authenticated and encrypted message is $(y_0, y_1, \ldots, y_n, z_n)$. Suppose that we know $(x_1, \ldots, x_n)$.

Note that $(x_1 \oplus y_0, x_2, y_2)$ is a valid authenticated encryption. To check the authentication, we set $z'_{-1} = 0$, $z'_0 = E_K((x_1 \oplus y_0) \oplus 0) = y_1$ and $z'_1 = E_K(x_2 \oplus z'_0) = E_K(x_2 \oplus y_1) = y_2$ and check if $z'_1 = y_2$. That is, we can easily do blind modifications of the message without being detected.