Security

Assignment 5, Wednesday, October 7, 2015

Handing in your answers: the full story, see

`http://www.sos.cs.ru.nl/applications/courses/security2015/exercises.html`

Briefly,

- submission via Blackboard (`http://blackboard.ru.nl`);
- one single pdf file;
- make sure to write all names and student numbers and the name of your teaching assistant (Brinda or Joost).

Deadline: Thursday, October 15, 24:00 (midnight) sharp!

Goals: After completing these exercises successfully you should be able to

- analyse simple authentication protocols with regard to vulnerabilities against replay and impersonation attacks;
- encrypt block ciphers using any modes of operation.

Marks: You can score a total of 100 points.

1. **(40 points)** In the previous lecture, diversified keys were discussed. A similar scheme is used in many of the smart card systems that are around today, such as the Dutch OV-chipkaart and the (now defunct) Chipknip.

   (a) Why is it necessary to use diversified keys in scenarios such as the one discussed in the lecture, from a practical point of view? Think about how the same security could be achieved without this technique. Be as specific as possible.

   (b) In the slides, the following protocol was shown. Assume that each card stores an identification number Id_c and a key $K_c = K_m\{Id_c\}$ that is generated when the card is issued (here, K_m is the master key). Furthermore, assume that the terminals all have this master key K_m. What form of authentication is achieved here (*i.e.* what party has been authenticated)? Explain why.

$$\begin{array}{lcll} C & \longrightarrow & T & : \quad Id_c \\ & & \multicolumn{2}{l}{\text{Terminal computes } K_c = K_m\{Id_c\}} \\ T & \longrightarrow & C & : \quad K_c\{N\} \\ C & \longrightarrow & T & : \quad N \end{array}$$

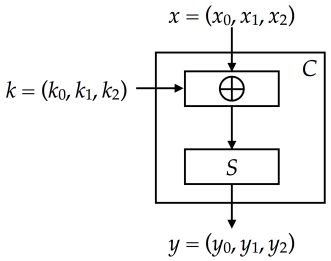
   (c) We now change the protocol to the one described below. What form of authentication is achieved here? Explain why.

$$\begin{array}{lcll} C & \longrightarrow & T & : \quad Id_c, N_c \\ & & \multicolumn{2}{l}{\text{Terminal computes } K_c = K_m\{Id_c\}} \\ T & \longrightarrow & C & : \quad K_c\{N_t, N_c\} \\ C & \longrightarrow & T & : \quad N_t \end{array}$$

   (d) We make one more change to the protocol. Now, each unique terminal sends along its unique identifier T_i, as follows. Can the card now securely distinguish between terminals? Why (not)?

$$\begin{array}{lcll} C & \longrightarrow & T_i & : \quad Id_c, N_c \\ & & \multicolumn{2}{l}{\text{Terminal computes } K_c = K_m\{Id_c\}} \\ T_i & \longrightarrow & C & : \quad K_c\{N_t, N_c, T_i\} \\ C & \longrightarrow & T_i & : \quad N_t \end{array}$$

2. **(60 points)** A block cipher C encrypts a plaintext block with a key in two steps.



In particular, C maps a 3-bit input block $x = (x_0, x_1, x_2)$ to a 3-bit output block $y = (y_0, y_1, y_2)$ using a 3-bit key $k = (k_0, k_1, k_2)$ and a function S as follows:

$$y = C(x, k) = S(x_0 \oplus k_0, x_1 \oplus k_1, x_2 \oplus k_2),$$

where S is the following substitution:

Plaintext	**Ciphertext**
000	001
001	000
010	011
011	110
100	010
101	111
110	100
111	101

So, for instance encrypting `001` with key `101` becomes $C(001, 101) = S(100) = 010$ and decrypting `100` with key `110` becomes $C^{-1}(100, 110) = S^{-1}(100) \oplus 110 = 110 \oplus 110 = 000$. Compute the ciphertext belonging to plaintext `011 111 101 001` (using a block size of 3 bits) with key $k = 101$ for the following different modes.

(a) In Electronic Code Book (ECB) mode. Show intermediate steps.

(b) In Cipher Block Chaining (CBC) mode where the Initialisation Vector (IV) is `111`. Show intermediate steps.

(c) In Cipher Feedback (CFB) mode where the IV is `101`. Show intermediate steps.

(d) In Output Feedback (OFB) mode where the IV is `011`. Show intermediate steps.

(e) Give at least one reason why CBC mode is preferred over the ECB mode.

(f) CBC can also be used to produce a code, for message integrity. This is called a 'CBC-MAC' (message authentication code), which is effectively the last block of a CBC encryption. The recipient can perform the same computation, and compare codes. The CBC-MAC only provides integrity for fixed-length messages. Which block (so three bits) can you add *in front* of the plaintext message, to come to the same CBC-MAC (again with Initialisation Vector is `111`). Why does this work?

(g) The previous question was a concrete example of a generic problem. Let's call the CBC-MAC of message m: T (for tag). If an attacker knows two messages and their CBC-MAC's, so (m, T) and (m', T') and the IV (which can always be considered public information), show that the attacker can always create a message $m'' \neq m'$, so that CBC-MAC(m'') $= T'$.

(h) Suppose Alice sends a message to Bob using this cipher in CFB mode. The ciphertext is `111 100 101`, the key is `100` and the IV is `010`. Unfortunately, the channel is noisy and the third bit of the ciphertext flips, so Bob receives `110 100 101`. How much difference will there be between the plaintext messages that were sent and received in terms of bits (so-called *Hamming distance*)? Show intermediate steps.