SECURITY

Assignment 5, Saturday, October 14, 2017

S1013793 Carlo Jessurun

S1013792 Tony Lopar

Radboud University

Teaching assistant: Joost Rijneveld

Assignment 1

1. (30 points) Alice (A) and Bob (B) are both trying to authenticate each other using a shared secret key (K_{AB}) only they know. Eve is trying to impersonate either Alice or Bob.

In which of the following four authentication protocols can Eve impersonate Alice or Bob by using a replay attack? Recall that in a replay attack, Eve records a message sent by Alice or Bob (while possibly preventing that message from reaching the addressee) and at any later point in time retransmits this recorded message.

For the vulnerable protocols write down the attack, using the 'E(A) → B : message' notation (for E impersonating A, by sending message to B). Clearly say which message an attacker stores and replays. If not, explain why a replay attack would fail.
Note that a replay attack is not the same as a man-in-the-middle attack!

$$
\text{(a)}\quad
\begin{array}{lcll}
1. & A \longrightarrow B & : & hello \\
2. & B \longrightarrow A & : & B, K_{AB}\{B\} \\
3. & A \longrightarrow B & : & A, K_{AB}\{A\}
\end{array}
$$

$$
\text{(b)}\quad
\begin{array}{lcll}
1. & A \longrightarrow B & : & A, K_{AB}\{N_A\} \\
2. & B \longrightarrow A & : & B, N_A, K_{AB}\{N_B\} \\
3. & A \longrightarrow B & : & A, B, N_A, N_B, K_{AB}\{N_A, N_B\}
\end{array}
$$

$$
\text{(c)}\quad
\begin{array}{lcll}
1. & A \longrightarrow B & : & A, N_A, K_{AB}\{A, N_A\} \\
2. & B \longrightarrow A & : & B, N_B, K_{AB}\{B, N_A, N_B\} \\
3. & A \longrightarrow B & : & K_{AB}\{A, B, N_A\}
\end{array}
$$

$$
\text{(d)}\quad
\begin{array}{lcll}
1. & A \longrightarrow B & : & A, N_A \\
2. & B \longrightarrow A & : & B, N_B, K_{AB}\{B, N_A - 1\} \\
3. & A \longrightarrow B & : & K_{AB}\{A, B, N_B + 1\}
\end{array}
$$

Solutions 1

A. This first protocol is highly vulnerable since there is no identify verification.
   Eve could impersonate as alice by replaying the hello message to Bob:
   $A \rightarrow B$: hello
   $E(A) \rightarrow B$: hello
   $B \rightarrow E(A)$: B, $K_{AB}\{B\}$, which contains the full response + keys back
   $E(A) \rightarrow B$: E(A), $K_{AB}\{E(A)\}$
B. Also here Eve could impersonate Bob by sending the same message back:
   E(B), NA, KAB{NB} would result into
   $A \rightarrow B$: A, $K_{AB}\{N_A\}$
   $B \rightarrow A$: B, N_A, $K_{AB}\{N_B\}$
   $E(B) \rightarrow A$: B, N_A, $K_{AB}\{N_B\}$
   $A \rightarrow E(B)$: A, B, N_A, N_B, $K_{AB}\{N_A, N_B\}$
C. $A \rightarrow B$: A, N_A, $K_{AB}\{A, N_A\}$
   $B \rightarrow A$: B, N_B, $K_{AB}\{B, N_A, N_B\}$
   $A \rightarrow B$: $K_{AB}\{A, B, N_A\}$
   Eve can replay the message from Bob to Alice, but then she will only get the encrypted identities. These are necessary for the connection, which means she can't pretend to be Alice.
D. Alice does not send a key here, only the nonce so Eve could simply:
   $E(A) \rightarrow B$: A, N_A and get the full response + key back from Bob:
   $B \rightarrow A$: B, N_B, $K_{AB}\{B, N_A - 1\}$

Assignment 2

2. (30 points) Consider the following two flawed mutual authentication protocols.

$$(i)\begin{cases} A \longrightarrow B : A, N_A \\ B \longrightarrow A : N_B, K_{AB}\{N_A+3\} \\ A \longrightarrow B : K_{AB}\{N_B+6\} \end{cases} \qquad (ii)\begin{cases} A \longrightarrow B : A, K_{AB}\{N_A-1\} \\ B \longrightarrow A : N_A, K_{AB}\{N_B-1\} \\ A \longrightarrow B : K_{AB}\{A, B, N_A\} \end{cases}$$

In this exercise we are *not* interested in man-in-the-middle attacks, only reflection or replay Attacks.

(a) Show that protocol (i) is flawed in the sense that an attacker Eve (E) can pretend to be Alice (A). Use the protocol attack notation E(A) −→ B : m.

(b) Fix protocol (i) by modifying only one message.

(c) Show that also protocol (ii) is flawed – in the sense that an attacker Eve (E) can pretend to be Alice (A).

(d) Fix protocol (ii) by, once again, only modifying one message.

Solutions 2

A. (a) E → B: E(A), N_A
   (a) B → E: N_B, $K_{AB}\{N_A + 3\}$
   (b) E → B: E(A), $(N_B + 3)$
   (b) B → E: N_{B2}, $K_{AB}\{N_B + 3\}$
   (a) E → B, $K_{AB}\{N_B + 6\}$

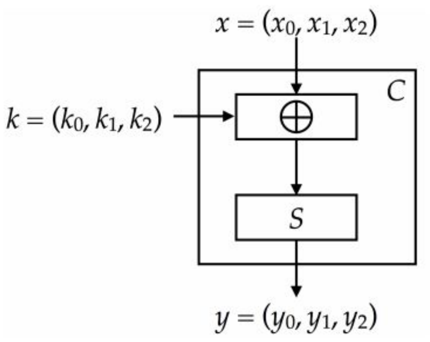
B. Using domain separation on the second message, we could fix this with:
   B → A: N_B, $F(K_{AB}, N_A||1)$. We therefore would not have the predictable counter with numbers anymore.
   By sending N_B encrypted we can prevent that Eve can easily send this number(modified) is another session, to obtain $N_B + 6$.
   B → A: $K_{AB}\{N_A + 3, N_B\}$

C. Eve could replay the first message to retrieve N_A by replaying the first message to Bob. Since, Alice does not send N_B back to Bob, Eve doesn't need to know what this nonce is.

D. In the encryption of the last messag N_B could be added to fix the protocol. In this case Alice would also need to compute what N_B should be, which makes an attack more difficult. The last message would be as follows then:
   A → B: $K_{AB}\{A, B, N_A, N_B\}$

Assignment 3

3. (20 points) Assume a block cipher C that encrypts a plaintext block x using a key k.



S:

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

In particular, C maps a 3-bit input block x = (x0, x1, x2) to a 3-bit output block y = (y0, y1, y2) using a 3-bit key k = (k0, k1, k2) and a function S as follows: y = C(x, k) = S(x0 ⊕ k0, x1 ⊕ k1, x2 ⊕ k2), where S is the substitution described above.

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) ⊕ 110 = 110 ⊕ 110 = 000.

A. Compute the ciphertext belonging to plaintext 011 111 101 001 (so, using blocks of three bits) with key k = 101 using Electronic Code Book (ECB) mode. Show intermediate steps.
B. Do the same for Cipher Block Chaining (CBC) mode, where the Initialisation Vector (IV) is 111. Show intermediate steps.
C. Give at least one reason why CBC mode is preferred over the ECB mode.

Solutions 3

A. First we split the plaintext into blocks of three. These blocks can be found in the header of the table below. Then we performed an XOR on the plaintext bits and key bits to generate the input for function S. With this input we could recover the bits of the cipher using the table in the exercise.

Plaintext	011	111	101	001
Key	101	101	101	101
XOR(Input for S)	110	010	000	100
Ciphertext	**100**	**011**	**001**	**010**

B. In CBC we first perform an XOR on the input before the block encryption. The key of the XOR is the ciphertext from the previous block. The first block uses the IV 111 as key.

Plaintext	011	111	101	001
XOR key	111 (IV)	000	011	110
Block input	100	111	110	111
Key	101	101	101	101
XOR(Input for S)	001	010	011	010
Ciphertext	**000**	**011**	**110**	**011**

C. The main limitation of equal plaintext blocks is that the resulting ciphertext blocks will be equal if the text is equal leading to patterns in the ciphertext.
ECB mode can also make protocols without integrity protection even more susceptible to replay attacks, since each block gets decrypted in exactly the same way.

Assignment 4

4. (20 points) In this exercise, we will take a look at the counter mode (CTR). We use the same block cipher C that was introduced in the previous exercise.

A. Assume that the key k = 101, and IV = 100. Compute the first 9 bits of keystream. Show intermediate computations! Hint: interpret the IV as a 3-bit binary number.
B. Assume that the plaintext is 001 110 111. Compute the matching ciphertext.
C. While CTR is generally a great choice, there is one pitfall: an IV should never be repeated. Assume you have a plaintext p1 = 010 110 110 and a corresponding ciphertext c1 = 110 001 101, for a certain unknown key and IV, as well as a different ciphertext c2 = 101 011 111 that was obtained by encrypting p2 with the same key and IV. Compute the matching plaintext p2. Show your computations!

Solutions 4

A. The counter mode increments the IV with one every block. The IV is 100, so the next IV will be 101

Counter	100(IV)	101	110
Key	101	101	101
Keystream	**001**	**000**	**011**

B. Using the keystream from exercise A, we got the following result:

Plaintext	001	110	111
Keystream	001	000	011
Ciphertext	**000**	**110**	**100**

C. Since we know a pair of plaintext and ciphertext, we can discover the keystream. Since the same key and IV are used for cipher 2, we may use the same keystream to decrypt c_2. For the decryption we will XOR the ciphertext with the keystream. These steps are shown in the table below:

P_1	010	110	110
S input	110	101	010
Keystream	100	111	011
C_1	110	001	101
C_2	101	011	111
Keystream	100	111	011
P_2	**001**	**100**	**100**