Security

Assignment 4, Friday, October 6, 2017

Handing in your answers:

- Include your name and student number **in** the document (they will be printed!), as well as the name of your teaching assistant (Bart or Joost). When working together, include **both** your names and student numbers.
- Submit one single **pdf** file – when working together, only hand in **once**.
- Hand in via Blackboard, before the deadline.

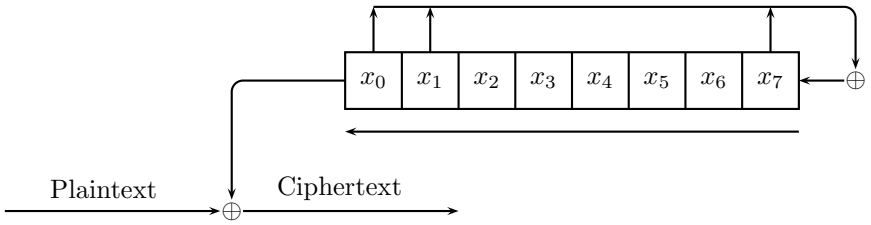
Deadline: Monday, October 16, 09:00 sharp!

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

- encrypt and decrypt using an LFSR;
- reason about key sizes and attack complexity;
- understand and use meet-in-the-middle attacks.

Marks: You can score a total of 100 points.

1. **(30 points)** Consider the following simple Linear Feedback Shift Register (LFSR). The plaintext is bitwise XOR-ed with the output bits of the LFSR which **first computes** $x_0 \oplus x_1 \oplus x_7$ and **then shifts** such that x_0 falls out.



Example:

	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
The initial state	**0**	**1**	1	1	0	1	0	**1**
is followed by	**1**	**1**	1	0	1	0	1	**0**
and outputs	**0**							

   (a) Describe the next five states of the LFSR if it is initialized according to the box above. The first successor state is already given as illustration, so you have to give the four subsequent ones.

   (b) Also do a "rollback" and compute the *previous* four states, starting from the initial state.

   (c) Assume you know that the LFSR is in the initial state given above. After four shifts you intercept 0110 as resulting ciphertext. Reconstruct the 4 bits of plaintext that were encrypted to this ciphertext.

2. **(30 points)** Alice wants to send Bob a confidential message. In the arrow protocol notation introduced in the lecture, this would be expressed as $A \longrightarrow B : K_{AB}\{m\}$ (i.e. using a block cipher). However, this assumes that Alice and Bob already share a (symmetric) key K_{AB}. This is usually not the case for the pair of any two random people.

   Instead, we assume that a trusted third party, Trudy, shares keys with everyone (i.e. a large group of users can rely on the same Trudy). Using these keys, Alice can send m to Trudy and ask her to confidentially forward it to Bob.

   For this exercise, we assume a passive attacker: Eve can only eavesdrop messages, but not insert, delete or modify them.

   (a) Write the above description of the two-step protocol down in arrow notation.

   The crucial downside of the protocol described in (a) is the fact that Trudy learns the content of every message. Ideally, Alice and Bob would set up a shared symmetric key K_{AB} without trusting a third party. Consider the following protocol. Note that K_A is a key only Alice knows.

   Step 1. Alice picks 1 million different key candidates K^i_{AB} and 20-bit *puzzle keys* K^i_{puz} (i.e. $i = 0, 1, 2, \ldots$).

   Step 2. Alice computes identifiers $ID_i = K_A\{i\}$ and cipher texts $c_i = K^i_{puz}\{ID_i, K^i_{AB}\}$.

   Step 3. Alice sends all cipher texts c_i to Bob, in a random order.

   Step 4. Bob picks a random c_i and decrypts it by trying all possible puzzle keys. (You can assume one can tell apart valid and malformed decryptions of such c_i.)

   Step 5. Upon success, Bob sends Alice ID_i.

   Step 6. Alice uses K_A to decrypt $ID_i = K_A\{i\}$ to find i.

   Step 7. Alice and Bob now use K^i_{AB} as their shared key.

   (b) How many encryptions does Alice have to perform?

   (c) How many keys (and thus: decryptions) does Bob have to try to find K^i_{AB}?

   (d) Describe what Eve would have to do to find the key K_{AB}. How many encryption or decryption operations does this include?

   Say that Alice picks n key candidates, and uses puzzle keys of $\log_2(n)$ bits.

   (e) Express the number of operations Eve has to perform (and thus: the security of the scheme) in terms of n.

   The solution to the 'key exchange problem' described in this exercise dates back to a paper written in 1974 by Ralph Merkle, and is often referred to as Merkle's Puzzles[1]*. Later in this course, we will revisit this problem by looking a much more secure solution posed by Diffie and Hellman*[2]*, only two years later.*

3. **(40 points)** Recall the DES block cipher, which has a block length of 64 bits and operates with 56-bit keys. In the lecture, you learned that by the end of the 1980's, the key length of DES was considered too short, and DES was eschewed in favor of triple-DES. In this exercise,

[1] `http://www.merkle.com/1974/PuzzlesAsPublished.pdf`
[2] `https://ee.stanford.edu/%7Ehellman/publications/24.pdf`

you will learn the real reason why so-called "double-DES" was omitted. In consistency with the lectures' notation, we define the three functions as follows:

$$\begin{aligned}
\text{DES} &= \left(\cdot \xrightarrow[\text{Encrypt}]{K} \cdot \right). \\
\text{2DES} &= \left(\cdot \xrightarrow[\text{Encrypt}]{K_1} \cdot \xrightarrow[\text{Decrypt}]{K_2} \cdot \right). \\
\text{3DES} &= \left(\cdot \xrightarrow[\text{Encrypt}]{K_1} \cdot \xrightarrow[\text{Decrypt}]{K_2} \cdot \xrightarrow[\text{Encrypt}]{K_3} \cdot \right).
\end{aligned}$$

In this exercise, you are the adversary.

(a) Suppose that you are given a single plaintext-ciphertext tuple (P, C) of DES_K for secret key K. Explain the exhaustive key search algorithm on DES to recover key K.

(b) Suppose that you are given a single plaintext-ciphertext tuple (P, C) of $\text{2DES}_{K_1,K_2}$ for secret key (K_1, K_2). How many evaluations of 2DES would exhaustive key search take? How many *unique* evaluations of the Encrypt functionality of DES would this attack take? Explain the difference.

(c) Explain how you can recover the *entire key* (K_1, K_2) in $2 \cdot 2^{56}$ evaluations of DES. Hint: only perform encryptions with DES, and no decryptions. Feel free to describe the steps procedurally, or even in pseudo-code.

(d) Explain how the attack extends to 3DES to recover the entire 168-bit key in $2 \cdot 2^{112} + 2^{56}$ evaluations of DES.