Security

Assignment 13, Wednesday, December 16, 2015

Handing in your answers: For 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, January 7, 23:59 (midnight) sharp!

Goals: This assignment repeats and reinforces concepts and techniques from this course. After completing these exercises successfully, you should

- be able to perform necessary computations of a Diffie–Hellman key exchange;
- recognise the main weakness of the Diffie–Hellman key exchange;
- be able to perform necessary computations for ElGamal encryption/decryption.

Marks: You can score a total of 100 points.

1. **(40 points)** Diffie-Hellman (DH) key exchange is used to agree on a secret key between Alice and Bob. The prime $p = 1021$ determines the group $\mathbb{Z}_p^* = \{1, \ldots, p-1\}$ in which all operations are performed (i.e. all computations are performed modulo 1021).
   The following messages are exchanged:

$$\begin{array}{lll} 1. & A \longrightarrow B & : & p = 1021, g = 10, g^a = 93 \\ 2. & B \longrightarrow A & : & g^b = 491 \end{array}$$

   (a) Given Alice's secret $a = 317$, compute the shared secret key. (You can use a computer, but do explain what exactly you compute.)

   (b) Since the modulus is very small, one can compute the secret values. Derive Bob's secret from the exchanged messages. You have two options:
   - Use a script/program. Give the source code, and explain what it does.
   - Do it by hand[1]. Explain how you did it.

   (c) Check that Bob has the same (shared) key as Alice using the private key from (b) (by doing the DH-computation for Bob's side).

   (d) We describe a modified communication when there is a middle-man. Assume that message 1. $A \rightarrow B$ is as showed above, but Eve captures the message and picks two random values: $r_A = 37, r_B = 404$. She uses these random values for the communication with Alice and Bob, respectively.

   i. Show the *four messages*: $A \rightarrow E(B), E(A) \rightarrow B, B \rightarrow E(A), E(B) \rightarrow A$.

   ii. Compute the *established keys* K_{AE}, K_{BE} between Alice and Eve, and between Eve and Bob, respectively.

[1] Feel free to use a calculator on your PC (like `bc` for linux) or online (like the Magma Calculator `http://magma.maths.usyd.edu.au/calc/`, syntax: `x^y mod z`), to help you with the calculations.

2. **(60 points)** Consider the ElGamal public-key encryption system, see Wikipedia[2]. For $p = 31$, $G = \mathbb{Z}_p^*$ is a multiplicative cyclic group with generator $g = 3$. Suppose that the secret number in the system is $x = 17$. You will encrypt messages and decrypt ciphertexts in this group.

   Please make sure that you describe how the computations are carried out; you do not need to write down all steps here (unless otherwise required).

   (a) Determine the corresponding value $y = g^x \in G$.

   (b) We are going to encrypt the message "remember" (in ECB mode) using ElGamal. To map letters to integers we use the mapping $a \mapsto 1$, $b \mapsto 2, \ldots, z \mapsto 26$. For the following steps, fill in each row in Table 1 and explain the required computations:

      i. For each integer block calculate a separate session key $s = y^r$ with a temporary value r: 3, 6, 9, 12, 15, 18, 21 and 24.

      ii. For each integer block calculate the first component $c_1 = g^r$ of the ciphertext using that same sequence.

      iii. Finally, for each integer block calculate the second component $c_2 = m \cdot s$ of the ciphertext.

   (c) Let's now decrypt the ciphertext; complete Table 1.

      i. For each integer block calculate the inverse session key $s^{-1} = {c_1}^{-x}$, where ${c_1}^{-x}$ can be calculated as ${c_1}^{p-1-x}$. (Remark: This is true because $\varphi(p) = p - 1$ and, by Euler's theorem, $a^{\varphi(n)} \equiv 1 \pmod{n}$ for any integer n. So, ${c_1}^{p-1-x} \equiv {c_1}^{p-1} \cdot {c_1}^{-x} \equiv 1 \cdot {c_1}^{-x} \equiv {c_1}^{-x} \pmod{p}$.)

      ii. For each integer block, use the inverse s^{-1} to cancel out s in c_2 and thus retrieve $m = c_2 \cdot s^{-1}$. Show intermediate steps for the first three blocks.

	r	**e**	**m**	**e**	**m**	**b**	**e**	**r**
Encryption								
Mapping	18	5	.	.	.	.	.	.
r	3	6	9	12	15	18	21	24
$s = y^r$	.	.	.	.	.	.	.	.
$c_1 = g^r$	.	.	.	.	.	.	.	.
$c_2 = m \cdot s$	.	.	.	.	.	.	.	.
Decryption of ciphertext (c_1, c_2)								
$s^{-1} = {c_1}^{-x}$	.	.	.	.	.	.	.	.
$m = c_2 \cdot s^{-1}$	.	.	.	.	.	.	.	.

Table 1: Encryption and decryption with ElGamal

[2] http://en.wikipedia.org/wiki/ElGamal_encryption