Security
Assignment 8, Monday, November 14, 2011

Handing in your answers: There are two options:

1. Put your solutions in Pim's post box (in the white cabinet, lower door, near the printer at the station-side-end of the corridor on the second floor of the Huygens building). Before handing in, make sure:
   - your names and student numbers are written on the document.
2. Send your solutions by e-mail to `pim@cs.ru.nl` with subject '*assignment 8*'. This e-mail should only contain a single PDF document as attachment. Before handing in, make sure:
   - your names and student numbers are included in the document (since it will be printed),
   - your names are part of the filename (example: PimVullers_FlavioGarcia-8.pdf), and
   - the file is a PDF document.

Deadline: Monday, November 21, 8:30 sharp!

Note: Do not forget to perform the modulo operations.

1. **(2 points)** In question 2 of assignment 3 we noticed that a deterministic symmetric-/secret-key algorithm is not secure when used more than once. This time we want to use a deterministic public key algorithm (for instance RSA) in the same situation, i.e. without using nonces to randomize the message.

   Discuss why the protocol from question 2 of assignment 3, in which Alice asked Bob out on a date and Bob answered with either encrypted "yes" or encrypted "no", cannot be used even once if RSA is used.

2. **(2 points)** Public key cryptography often involves exponentiation with large numbers. To do this quickly one can use *exponentiating by squaring* which is also known as the square-and-multiply algorithm or binary exponentiation (see Wikipedia[1][2]).

   Compute *by hand* (that means you are not allowed to use a computer or calculator), using this method, $7^{1063} \mod 13$. Give all intermediate steps.

3. **(2 points)** Compute, using the Extended Euclidean Algorithm[3], the multiplicative inverse of $312 \mod 601$. Give the intermediate steps.

4. **(4 points)** Consider the RSA cryptosystem[4]. Suppose $p = 7, q = 11$ and $e = 37$.
   (a) Determine the corresponding d (using the Extended Euclidean Algorithm).
   (b) Give the public key is in terms of p, q, e, d.
   (c) Give the private key is in terms of p, q, e, d.
   (d) We are going to encrypt the message "banana" with the public key in Electronic Code Book (ECB) mode, that is block-by-block.
      i. Take blocks of length 1 letter, so we have 6 blocks: "b", "a", ..., "a".
      ii. Translate each block into a number: "a" $\mapsto$ 1, "b" $\mapsto$ 2, ..., "A" $\mapsto$ 27, "B" $\mapsto$ 28, ...
      iii. Complete Table 1 with these mappings from letter-blocks to integer-blocks.

[1] `http://simple.wikipedia.org/wiki/Exponentiation_by_squaring`
[2] `http://en.wikipedia.org/wiki/Exponentiation_by_squaring`
[3] `http://en.wikipedia.org/wiki/Extended_Euclidean_algorithm`
[4] `http://en.wikipedia.org/wiki/RSA`

(e) Encrypt each integer-block with the public key. Give intermediate steps and fill out the second row of Table 1.

(f) Decrypt it with the private key filling out the third row of Table 1. Again, give the intermediate steps as well.

	b	**a**	**n**	**a**	**n**	**a**
Mapping (Step d)	.	1	.	.	.	.
Ciphertext (Step e)	.	.	.	.	.	.
Recovered plaintext (Step f)	.	.	.	.	.	.

Table 1: Computation steps RSA