**ΕΠΛ 674 Ασφάλεια Συστημάτων και Δικτύων** **Spring 2011**

# Assignment 1

**Assigned: 7 February 2011** **Due: 28 February 2011**

**Instructions**

- All assignments should be submitted typed neatly using a document processing application of your choice. Please make sure to include your name and student number for proper recording of grades.
- The assignment is due at the beginning of the lecture at the due date. Late assignments will incur a five-point penalty. Assignments late by more than one day will not be accepted.

---

**The goal of this homework is (a) to give you a deeper understanding of RSA encryption and decryption and its underlying principles and (b) to gain further insights into hashing.**

**Problem Statement**

**PART A**

Write a C/C++ program that implements RSA encryption and decryption algorithm. You are free to choose the integers *n* and *e* used in the RSA algorithm that you need to hardcode it in your source code. Do NOT use the ready-made key-generator functions already implemented in the language libraries to find *d* (Use your own implementation of the Extended Euclidean Algorithm to find the multiplicative inverse).

**PARTB**

You are required to implement in C/C++a very simple hash function (that is meant more for play than for any serious production work). Write a function that creates a 32-bit hash of a file through the following steps:

- Initialize the hash to all zeros.
- Scan the file one byte at a time.
- Before a new byte is read from the file, *circularly shift* the bit pattern in the hash to the left by four positions.
- XOR the new byte read from the file with the least significant byte of the hash.
- Scan your directory and compute the hash of all your files.
- Dump the hash values in some output file.

By using a couple of files (containing random text) created especially for this demonstration, show how you can make their hash codes to come out to be the same if you alter one of the files by appending to it a stream of bytes that would be the XOR of the original hash values for the files (after you have circularly rotated the hash value for the first file by 4 bits to the left).

**Notes:**

You must turn in only two files, one for each part. Don't turn in files other than those listed above.

Kindly include comments in your code. The 2 files must be send via email to paul.antoniou@cs.ucy.ac.cy. Also you must provide a hardcopy including the 2 files. The 2 programs will be demostrated in the lab right after the end of the deadline (lab 28/2/2011).

**References**

[1] RSA Algorithm: http://www.di-mgt.com.au/rsa_alg.html
[2] Extended Euclidean Algorithm: http://www.di-mgt.com.au/euclidean.html#extendedeuclidean