CSCI 476: Computer Security—**Assignment 5 (6 marks)**

This assignment is to enhance your learning on DNS cache poisoning and also on pseudo-random number generation (**read Chapter 8**!). It is recommended that you try to solve this problem using both programming and non-programming skills. It is highly recommended that you find some fellow students to form a 2- or 3-person group.

Background. In DNS cache poisoning, the attacker Eve tries to trick the ISP DNS server to store a wrong IP address in the DNS cache for, say, `www.example.com`. This can be done as follows:

1. The attacker Eve sends a query: *Where is www.example.com*?
2. The ISP DNS server try to query some name server to get the answer: *www.example.com is at 222.22.2.2, Query ID=x(=12345).*
3. The attacker Eve, using a spoofed IP address, sends a reply to the original query by a guessed ID: *www.example.com is at 111.11.1.1, Query ID=y.* If $y = 12345$, then the wrong IP address 111.11.1.1 for `www.example.com` would be saved in the DNS cache by the ISP DNS server. And the attacker has had a successful attack. After this all internet accesses to `www.example.com` would be directed to the address 111.11.1.1.

Assume that the ID is a 20-bit number. To increase the chance of success, at Step (3) the attacker Eve now tries to randomly guess the query ID by sending n (instead of 1) replies immediately following the initial query.

(I) What is the probability that Eve would succeed?

(II) If Eve wants this probability to be at least 0.5, what should be the value of n?

(III) If you use `rand()` function in C or Java to generate the query IDs, it is possible that you might need to try more than 2^{20}=1,048,576 times to have a success. Can you design a pseudo-random number generator so that it will guarantee a success within 2^{20} tries? What is it?

In reality, Eve must act on Step 3 fast — before the reply from a genuine name server arrives at the ISP DNS server at Step 2. (So if Eve tries a brute-force method she might fail

completely, especially when the IDs are much longer. This is why the so-called **birthday attack** kicks in.) Now, to obtain a success in a much much faster way, Eve changes Step 1 by sending m queries with the same question (but with different query IDs, using spoofed IP addresses), forcing the ISP DNS server for m replies at Step 2 with m random query ID x_i's. At Step 3, the attacker Eve still tries to guess one query ID x_i by sending m replies with random ID y_j's immediately following the initial query at Step 1. In this case, if one x_i matches with one y_j, then Eve has a success. (This is very much the algorithm explained on P. 281.)

(IV) What is the smallest m such that Eve succeeds with a probability at least 0.5?

Date Due: **11:30pm on Wednesday, April 17, 2019 (on or before 11:30pm, April 17, 2019).** Load your output as a separate file on D2L in the folder **Assignment 5**, preferably in the form of **family_name.output**. If you form a group with some fellow students (with at most 3 students in each group), only one of you needs to submit your solution and all of you will get the same mark. But don't forget to put all the names in the group!