Exercise Set 6

**Ground Rules.** You may choose to work with one other student if you wish. Only one submission is required per group, please ensure that both group members names are on the submitted copy. Work must be submitted in hard copy by the start of class on December 3, 2019. Questions marked as COSC 583 are required for those enrolled in the 583 section. If you are in the 483 section you are more than welcome to attempt any of them, extra credit will be awarded for correct answers.

### 1. X.509 Break

In the early days of SSL (TLS’s predecessor) the following protocol was used to share a secret key, \( X \) between two parties, say Alice (A) and Bob (B). We can assume that Alice, Bob (and Eve) correctly know each other’s public keys (certificate).

First Alice Computes and sends to Bob:
\[
M_A = (\text{TimeStamp}, \text{SessionID}, B, \text{Enc}_B(X))
\]
\[
S_A = \text{Sign}_A(M_A)
\]

Bob verifies Alice’s signature on \( M_A \) and then decrypts \( \text{Enc}_B(X) \). He then computes and sends to Alice:
\[
\text{Enc}_A(X)
\]

No big shockers, this protocol is broken. If Eve can intercept messages from Alice heading to Bob, Eve can connect to Bob (as Eve), and as a result convince Alice that she is talking with Bob when Alice is actually talking with Eve. In other words, she, along with Bob, can learn \( X \). Build this atack.

### 2. Broken Key Exchange

Consider the following scenario, Alice and Bob want to compute a shared key, but their devices are weak, so they must rely on a trusted third party, Steve the Server. Steve computes an RSA key, and distributes the public key, \( n \) and \( e \). Alice and Bob each generate a random number, \( r_a \) and \( r_b \) respectively, and will use both of these numbers to generate their shared key. In order to secretly exchange nonces, Alice and Bob execute the following protocol:

\[
A \rightarrow S : r_a^e \mod N
\]
\[
B \rightarrow S : r_b^e \mod N
\]
\[
S \rightarrow A, B : X = r_a \oplus r_b
\]

Alice knows \( r_a \) and XORs it with \( X \) to recover \( r_b \). Bob does the opposite. Show how, if Eve and Edward conspire, after watching Alice and Bob’s interaction with the server, they can recover Alice and Bob’s shared key by doing their own key exchange.
In Section 12.4.1 it is shown how to attack the plain RSA signature scheme, forging a signature on an arbitrary message using two signing queries. Show how an attacker can forge a signature on an arbitrary message using a single signing query.

Assuming the RSA problem is hard, show that the plain RSA signature scheme is weakly secure. That is, demonstrate that given a message \( m \in \mathbb{Z}_n^* \), an adversary can not forge a signature without access to a signing oracle.