CS4286: Internet Security and E-commerce Protocols Autumn, 2024
Exercise 5, Week 6 (Oct 15)

Questions:

1. **RSA** Assume that we use RSA with the prime numbers $p = 3$ and $q = 11$.
   (a) Calculate n and $\phi(n)$.
   (b) Given the public exponent $e = 7$, calculate d.
   (c) Encrypt the message $M = 9$
   (d) Sign the message $M = 15$.
   (e) Verify the Signature $S = 9$.

2. **Hash Function 1**
   (a) Does the hash function $h(x) = g^x \mod p$ satisfy one-wayness? x in this case is the data being hashed.
   (b) From Fermat's little theorem we can see that if $y = x + k \cdot (p - 1)$, and k is an integer/ p is prime and is not divisor of a, then we should have $g^x \mod p = g^y \mod p$.

   Why? $g^{x+k\cdot(p-1)} \mod p \equiv g^x \cdot g^{k\cdot(p-1)} \mod p \equiv (g^x \mod p) \cdot (g^{k\cdot(p-1)} \mod p) \equiv g^x \mod p \cdot (1 \mod p) \equiv g^x \mod p$

   Using this fact can you find another message that will result in the same hash as for message $x = 1$ when $p = 19$ and $g = 7$?
   (c) Do you think $h(x) = g^x \mod p$ is a good hash function?

3. **Hash Function 2** Suppose we use a hash function $H(x)$ of output length 128 bits, which accepts input in blocks of size 16 characters (16 bytes/128 bits), meaning that a message is always split into blocks of 16 characters and input into the hash function.
   (a) We want to hash the message `TheMessage` (10 characters). Because we can only hash 16 characters at a time, we pad the input message with `0`, so the input becomes `TheMessage000000`. As a rule, padding always has to be applied, meaning that if a message is already 16 characters long, we add a whole block of 16 characters of padding. Show that using this padding scheme, it is trivial to come up with a different message which will produce the same hash value.
   (b) Show a minor modification of the padding scheme which will resist the attack above.
   (c) Assuming that padding scheme is secure, what is the estimated computational effort needed to find a collision on this hash?

4. **MAC**

Suppose you are using a MAC based on a block cipher in CBC mode ($C_i = E(K; P_i\, XOR\, C_{i-1}),\ IV = 0\ for\ C_0$), and you know the following two messages:
$M` = M_0||M_1$
$M`` = M_2||M_3||M_4$
together with their corresponding MAC tags T` and T``. Show that you can create a new message $M``` = M_0||M_1||X||M_3||M_4$ and the correct MAC tag T``` without knowing the key K. You can choose any value for X to make this work. For the purposes of calculating the MAC IV is always 0.