CSCI-UA.0480-63: Intro to Security & Cryptography Fall 2025

Homework 2

due: Thursday 2025-10-09 22:00 EST via brightspace.nyu.edu

1. **Public key crypto at toy security levels [note: the numbers here are small, but you will likely need to do a little bit of programming, or use an online calculator]**

    a. Suppose Alice and Bob agree to use p=10007 and g=3 as public parameters

        i. What is the order of g modulo p?

        ii. For Shamir's Three-Pass protocol, Alice wants to send the message m=1337. She uses a temporary key a=2461 and Bob uses a temporary key b=4319. What are a^{-1} and b^{-1} mod $p-1$? What are the three values transmitted to send m from Alice to Bob?

        iii. For a Diffie-Hellman key exchange, Alice chooses a=2461 and Bob chooses b=4319. What will their shared secret be?

    b. Alice chooses an RSA key with p=383 and q=401, giving a public key of N = 153583.

        i. What is φ(N)?

        ii. If she chooses e=11, what will d be?

        iii. If Bob wants to encrypt the message 1337 to Alice, what will the ciphertext be?

        iv. If Alice wants to sign the message 1337 for Bob, what will the signature be?

2. **Digital signatures:** Most signature schemes have a limit on the size of messages they can be used to sign. For example, with RSA signatures the message must be an integer mod N for a modulus N that is typically 1024-4096 bits long. With DSA messages typically must be an integer mod p for a prime that is 1024-2048 bits long.

    Typically, the message is first hashed and then the message hash is what is actually signed. Alice decides that hash functions are a bore and instead she'll just break her message up into 1024-bit blocks m_1, m_2 ..., sign each block using 1024-bit DSA and concatenate the results:

    $$\text{ConcatSign}(m) = \text{DSA.Sign}(m_1) \,\|\, \text{DSA.Sign}(m_2) \,\|\, \ldots$$

    Explain to Alice why this is insecure. In particular, explain how to create a forgery.

3. **Signatures with related secret values:** Consider the algorithm for of El Gamal signatures:
    a. Choose random integer $1 < y < p-1$
    b. Let $r = g^y \pmod p$
    c. Let $s = (m - rx) \cdot y^{-1} \pmod{p-1}$
    d. $\sigma = (r, s)$

    Suppose that Eve receives signatures from Bob on two distinct messages m_1 and m_2. Now, Bob was paying attention in class, so he knows he should never repeat the value y in step (a). But, he also knows generating random numbers is hard. So Bob decides that he'll use his favorite constant value k=7, and multiply this by y after each signature to get a new value y' to use for the next signature.

    Assume Eve knows that Bob implemented his signing code this way (in particular, assume she knows k). Further assume she receives two signatures: $[m_1, \sigma_1=(r_1,s_1)]$ and $[m_2, \sigma_2=(r_2,s_2)]$. Eve doesn't know the y values used, but she knows $y_2 = k \cdot y_1$. Show how she can recover Bob's private key (give an exact algebraic derivation).

4. **Reductions:** In class we saw the discrete-log assumption and two variations, the Computational Diffie-Hellman (CDH) assumption and the Decisional Diffie-Hellman (DDH) assumption:

    CDH problem: Given $(g, p, g^a \bmod p, g^b \bmod p)$, Output $g^{ab} \bmod p$

    DDH problem: Given $(g, p, g^a \bmod p, g^b \bmod p, h \bmod p)$, Output whether $h = g^{ab} \pmod p$

    The corresponding assumptions state that these problems are probabilistic polynomial-time algorithm (PPT) algorithms.

    You want to show that the CDH assumption implies the DDH assumption and therefore that CDH is a stronger assumption. To do so, show that given an algorithm A_{CDH} which can solve the CDH problem with probability P for non-negligible P, you can create an algorithm A_{DDH} which uses A_{CDH} as a subroutine to solve the DDH problem with probability non-negligibly greater than 1/2.

    Assume that in the problem instances given to A_{DDH}, the value h is chosen to be g^{ab} with probability 1/2 and randomly (mod p) otherwise.