1 Project Overview

In the term project, your team has two options that they may explore. First, your team may attempt a computer-oriented project that will involve programming in Java and becoming familiar with Java Security. Or, second, your team may attempt a theory-oriented project that will involve proofs and derivations.

Both projects require a written report, either explaining the results of the computer project, or providing the derivations associated with the theory project. The project reports are due May 5th at 5pm (turn in either during class or to Core 523).

2 Computer-Oriented Project

In this project, you are to generate a random key using the Diffie-Hellman scheme, which was described in lecture (see the Powerpoint slides or Chapter 13.1 for reference). The generation of the key should use a random 512 bit prime $p$.

After you have generated the Diffie-Hellman Key, you are to use it as the encryption key for DES, and encrypt a text message (which is available on the course website). Now, there is a slight problem here! The DH key you generated is 512 bits, while DES only needs a 56 bit key. You will have to come up with an appropriate way to cut down the key size to 56 bits. After performing encryption, you should write the output to disk, then read the ciphertext in and decrypt it using the key you generated (you do not have to save the key to disk). Warning: A simple approach, such as extracting the last 56 bits of the DH key as the key to use for DES might seem easy, but is considered cryptographically weak, so try to come up with something different!

You should measure the time it takes to perform the Diffie-Hellman key establishment, the time it takes to perform encryption, and the time it takes to perform decryption. Note: In order to get accurate and stable measurements of these numbers, you will actually have to perform the operation more than once! Finally, your measurements should be done on two different machines in order to quantify the performance difference between different platforms.

In your report, make certain to discuss the observations you have regarding the time to perform Diffie-Hellman on the two machines you chose, and compare the time difference between DH and encryption/decryption. What do you observe? Why would one machine be faster than the other?

Final Note: I am providing a sample of code from our optional textbook that provides an example of how to perform Diffie-Hellman key establishment. This code is meant to get you started, the rest is up to you...

3 Theory Project

The theory project is concerned with a powerful tool that combines number theory and DSP, known as number-theoretic transforms. In this project, we will work our way up to number-theoretic transforms and derive several results along the way. You are suggested to refer to Chapter 3 for relevant tools.

1. **Result 1:** Suppose that $\gamma$ is an $N$th root of unity mod $p$, that is $\gamma^N = 1 \pmod{p}$, and $p$ is a prime. Show that there are a total of $\phi(N)$ $N$th roots of unity mod $p$ where $\phi$ is the Euler phi function. Show that $N$ must divide $\phi(p) = p - 1$.

2. **Result 2:** Now comes the transform. Let $p$ be a prime. Let $\gamma$ be an $N$th root of unity mod $p$, that is $\gamma^N = 1 \pmod{p}$. 


Suppose we have \( N \) data points \( x(n) \in \{0, 1, \cdots, p - 1\} \), we define the forward transform:

\[
x(n) \rightarrow X(k) : X(k) = \sum_{n=0}^{N-1} x(n)\gamma^{kn} \pmod{p}
\]

while the reverse transform is

\[
X(k) \rightarrow x(n) : x(n) = \sum_{k=0}^{N-1} X(k)\beta^{nk} \pmod{p}
\]

where \( \beta \) is an element that is derived from \( \gamma \), and which we shall derive shortly. In order for the forward and reverse transform pair to exist, we must have that \( x(n) \rightarrow X(k) \rightarrow x(n) \) by applying the forward transform followed by the reverse transform.

(a) Show that the transform pair only exists if

\[
\sum_{n=0}^{N-1} \beta^{ni}\gamma^{kn} \pmod{p} = 1 \quad \text{for } i = k
\]

\(= 0 \quad \text{otherwise.} \) (3)

(b) Applying the above result, show that \( \beta = N^{-1}\omega^{-1} \pmod{p} \).

3. Result 3: Now let us look at an example. Consider \( p = 127 \) (\( p \) is called a Mersenne prime since it is of the form \( p = 2^q - 1 \) for \( q \) a prime). Let \( \gamma = 2 \). Show that 2 is a 7th root of unity (there are two ways: first, one can just do the calculation directly, or second one can observe that \( q = 7 \) argue from there). Suppose

\[
\{x(0), x(1), x(2), x(3), x(4), x(5), x(6)\} = \{1, 2, 3, 4, 3, 2, 1\}.
\]

Calculate the forward transform \( X(k) \) for \( k = 0, 1, \cdots, 6 \).

4. Result 4: Finally, one of the nice properties of NTT transforms is that the transform of a convolution of \( x \) and \( y \) is just the product of the transforms \( X \) and \( Y \). In this part, you are to show that the NTT transform of the circular convolution \( x(n) \ast y(n) \) is just \( X(k)Y(k) \). (Recall that the circular convolution \( z(n) = x(n) \ast y(n) = \sum_{m=0}^{N-1} x(n-m)y(m) \) where the indexing is done modulo \( N \), i.e. there is wrap around).