CSCI368
Session 2, 2012
Assignment 1: Secure communication
(20 marks, 20%)
Due: Tutorial 5

Aims

This assignment aims to establish a basic familiarity with the cryptographic methods and provides an exercise of key establishment and secure communication in a networked environment.

Objectives

On completion of this assignment you should be able to:

- Understand some basic concepts in cryptography.
- Understand key exchange and secure communication.
- Understand network programming.

Specifications:

Write (Java or C/C++) UDP programs allowing two parties to establish a secure communication channel. For simplicity, let us call programs "Host" and "Client"; each can be used by a user. Again, for simplicity, let us assume that Alice uses Host and Bob uses Client.

Alice and Bob want to establish a secure communication channel where messages are encrypted with AES. They need to carry out the following tasks. (1) Establish a share AES session key so that they can use it to encrypt messages. (2) Use the shared key to secure the communication.

The key establishment is done by the Diffie-Hellman Exchange scheme. Assume that Alice has a pair of private/public keys (x1, y1) and Bob has a pair of private/public keys (x2, y2). By key exchange, they obtain a share secret key, which is then converted to a 128 bit string as AES session key.

Place Host and Client in two separate directories: Alice and Bob. Alice's keys are stored in a file located at her directory (Alice) and Bob's keys are stored in a file located at his directory (Bob).

The protocol is described as follows:

- Alice runs KeyGen to generate a pair of her private and public keys including all required parameters. These keys and parameters are stored in directory Alice.
- Bob runs KeyGen to generate a pair of his private and public keys including all required parameters. These keys and parameters are stored in directory Bob.
- Alice executes Host.
  - Host is running and listening to the opened port (you need select a port for your code).
- Bob executes Client.
  - Client (Bob) sends his public key y2 to Host (Alice).
  - Client is ready and listens to the port.
- Upon receiving the public key from Bob, Alice computes the shared secret key, which is then converted to 128 bit AES key.
- Alice sends her public key y1 to Bob (Client).
- Upon receiving the public key from Alice, Bob computes the shared secret key, which is then converted to 128 bit AES key.
- Now, the secure channel is established.
  - Either Alice or Bob can send a message encrypted with the shared AES key. They type the message on their own terminal. The message is encrypted by their code (Host or Client) and sent out.
  - The received message is printed on the screen.
  - To quit the program, type "exit".

  (Note: you should use a proper encryption mode (CBC, CRT, CFB, etc.) for AES encryption and a proper padding method.)

Coding requirement:
You need to write three programs:

1. KeyGen.
2. Host.
3. Client.

You do not have to handle big integers, but the keys you select should be reasonably large (say, 16 bits). If you want to handle big integers:

> Java programmers should use `BigInteger` class (check Java API).
> C++/C programmers should use NTL (http://www.shoup.net/ntl/) or other big integer library. NTL is available on our Unix machines (e.g., banshee). It is located at /packages/ntl.

You can choose to either write the AES code yourself, or download a free one from other sources. You should cite the source if you use a downloaded code.

You should name your programs: keygen.cpp or keygen.java, host.cpp or host.java and client.cpp or client.java.

How to run?

Your programs should run according to the protocol. Host and Client should be executed on different windows. Of course, they can be executed on different computers (mind the IP and port). For convenience of marking, please use the local IP: 127.0.0.1 for the submitted version. For simplicity, there is no GUI required in this assignment. That is, messages are simply typed on the window and printed on the receiver's window.

Java programmers might want to create a GUI window allowing users to type on the window. It is entirely up to you to decide.

Files to be submitted:

> All source codes (Do not submit any executable).
> A readme file (text/ACSII only): instructions about how to compile and run your code.
> A Makefile: for C++ programmers. Alternatively, you provide the compilation instruction in the readme.

Submission

Burn your assignment into a CD and submit it to your tutor in Tutor 5.

Comments in code files should be concise. A header should give your information (including name, student ID, and Unix login name) and briefly summarize the contents of the file - identifying purpose of program, listing classes etc. Individual functions should only require comment if they are complicated or result in non-obvious side effects etc.

Marking

Mark distribution:

1. UDP codes: 4 marks
2. KeyGen: 4 marks
3. DH key exchange and the shared key computations: 6 marks
4. AES Encryption and encryption mode code: 6 marks

The code that does not compile will receive a zero. Each other error will receive a deduction of 0.5.

Late Submission: 25% deduction per day.

Plagiarism

A plagiarised assignment will receive a zero mark and be penalised according to the university rules. Plagiarism detection software will be used.