**COMP 5370/6370/6376**

**Computer and Network Security**

**Fall 2018**

Project

Due on Canvas:

11:55pm December 4, 2018

*Only graduate students and honors undergraduate students are required to complete this project. In this project, you will work in groups of 2 or 3 students. Outreach students will complete this project individually.*

*Any copying of other group’s work, or misrepresentation of other work as your own, will be grounds for failing this project or the course.*

You will demonstrate your project on December 5, 2018 to show that your implementation works correctly.

Penalty for late work is 20 points per day late.

**Objectives**

The main goal of this project is to design and implement the Blowfish encryption and decryption and use it to encipher and decipher messages sent and received between two processes (Alice and Bob) in a TCP/IP network. The processes must communicate with each other through UDP sockets. You may use UDP client-server programming model to implement the messages communication.

**Overview**

This project consists of two parts. In the first part, you will implement simple UDP client-server programs for Alice and Bob to exchange some contract documents. In the first part, the documents are sent and received in plain text.

In the second part, the documents between exchanged between Alice and Bob are encrypted and decrypted using the Blowfish encryption/decryption algorithm.

**Implementation**

All programs must be implemented in C, C++ or Java.

You may implement and execute your programs on the Auburn University network of Linux computers called the tux computers, which you can access remotely using ssh or secureCRT. Use winSCP to transfer files to/from the tux computers and your personal computers. When running on the tux computers, you can run each process (e.g. Alice and Bob) on different tux computers and the documents will be sent and received to/from each other through UDP sockets. In order to execute socket programs in the tux computers you must use port numbers that I assign to you (Ask me for the port numbers if you are using tux computers.).

Alternatively, you can use your own laptop, in which case all processes will execute on your laptop, but all exchange of messages (documents) must be sent and received through UDP socket interfaces.

**Description**

This project consists of two parts as follows. In the first part, you will gain experience writing a simple UDP client server program. In the second part, you will implement the Blowfish encryption algorithm and use it to send encrypted messages in the UDP client server program.

1. Write UDP client-server socket programs for Alice and Bob to exchange some contract documents. In the first part, the documents are sent and received in plain text.
   1. Alice will initiate the communication by sending to Bob through a UDP socket interface a signed contract that reads: “Contract: This is an agreement between Alice (Buyer) and Bob (Seller) to purchase Property XYZ at Lake Martin, Alabama, for the price of $900,000. Signed: Alice (Buyer) Date: November 27, 2018.” Your program must first read this contract from an input file.
   2. When Bob receive the UDP message containing the contract, he will then sign it and send it back to Alice through his UDP socket interface. The signed contract will read as follows: “Contract: This is an agreement between Alice (Buyer) and Bob (Seller) to purchase Property XYZ at Lake Martin, Alabama, for the price of $900,000. Signed: Alice (Buyer) Date: November 27, 2018. Signed: Bob (Seller) Date: November 29, 2018.” Your program must write this message into an output file.
2. Implement the Blowfish encryption and decryption algorithms and use it for encrypting and decrypting documents that are being exchanged between Alice and Bob based on UDP client-server socket programming model.
   1. Before Alice and Bob can exchange documents securely, the first step is for them to exchange a shared secret key that is used by Alice and Bob with your Blowfish program. In this project, we will simplify the secret shared key exchange by just having Alice create the secret shared key and then Alice will send the shared key to Bob through the UDP socket interface. Bob will reply with “OK” to complete the client-server interaction.
   2. Right before Alice sends the document to Bob, she will first use your Blowfish encryption program and the shared key to encrypt the following contract: “Contract: This is an agreement between Alice (Buyer) and Bob (Seller) to purchase Property XYZ at Lake Martin, Alabama, for the price of $900,000. Signed: Alice (Buyer) Date: November 27, 2018.” Your program must first read this contract from an input file. Alice will then send this encrypted document to Bob through the UDP socket interface.
   3. When Bob receives the encrypted message, he must use your Blowfish decryption algorithm and the shared key to decrypt Alice’s ciphertext. He will then sign the contract as follows: “Contract: This is an agreement between Alice (Buyer) and Bob (Seller) to purchase Property XYZ at Lake Martin, Alabama, for the price of $900,000. Signed: Alice (Buyer) Date: November 27, 2018. Signed: Bob (Seller) Date: November 29, 2018.” He will then encrypt the signed contract document using your Blowfish encryption program and the shared key. Then he sends the encrypted document back to Alice through his UDP socket interface.
   4. Alice will receive Bob’s encrypted document through her UDP socket interface. She will then use your Blowfish decryption program and the shared key to decrypt Bob’s document into plaintext. She verifies Bob’s signature and the sale is complete and legal. Your program must write this plaintext final contract into an output file.

To simplify your programs, you may use a simplified contract as follows: “Purchase Contract. Signed: Alice Date: Nov. 27, 2018 Signed: Bob Date: Nov. 29, 2018.”

**Submission**

Submit in Canvas on or before the due date the following.

1. Sources codes of your programs.
2. Makefile and all other build information.
3. Execution traces of your programs in execution (use script to capture the execution traces).
4. The input and output files for both parts.
5. Short 6-page document describing your design, implementation issues and results.

All the above must be submitted in a zipped file in Canvas.

**References**

Here is a useful link for the Blowfish algorithm: <https://www.schneier.com/academic/blowfish/>

The original Blowfish paper is here: <https://link.springer.com/content/pdf/10.1007%2F3-540-58108-1_24.pdf>