Problem 1 (8 points), due by 17:00 March 29

You need to submit your **programs** and the **executable codes** to the directory `/Assignment/4/yourid` in the gateway `cs-vnl-01.csil.sfu.ca`, `cs-vnl-02.csil.sfu.ca`, `cs-vnl-03.csil.sfu.ca`, or `cs-vnl-04.csil.sfu.ca` by the due time. Please have clear comments in your programs to explain how they work. In addition to your programs, please also submit a text file to explain how did you conduct your experiments and how to view your results.

(1) (4 points) Substitution and transposition are basic techniques for encryption schemes.

- Below is a substitution encryption scheme example defined by permutation π:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
V	M	L	C	B	G	K	Y	Q	U	S	O	P	A	X	E	J	T	W	F	R	W	H	P	Z	I

  To encrypt a plaintext, the scheme substitutes each character α in the plaintext by the character $\pi(\alpha)$ (e.g., A is replaced by $\pi(A) = V$) to create the ciphertext. To decrypt the ciphertext, the scheme substitutes each character β in the ciphertext by the character $\pi^{-1}(\beta)$ (e.g., V is replaced by $\pi^{-1}(V) = A$).

- Below is a transposition encryption scheme example.

  To encrypt a plaintext, the scheme (1) cuts the plaintext into segments, each segment is a string of $r \times n$ (3×3 in this example) characters $c_1c_2c_3c_4c_5c_6c_7c_8c_9$; (2) puts the string of characters into an $r \times n$ (3×3) array A as shown below (the plaintext is read from left to right on each row and from top to bottom by rows in A); and (3) reads the elements of A from top to bottom on each column and columns in the order of (1,2,3) (from left to right) to create the ciphertext $c_1c_4c_7c_2c_5c_8c_3c_6c_9$. The decryption is the reverse process of the encryption.

  An alternative description of Step (3) of the encryption scheme above is to change the positions of characters in A by the permutation Π on the positions of A to get array $\Pi(A)$ as shown below and read the characters in array $\Pi(A)$ from left to right on each row and from top to bottom by rows to get the ciphertext. The transposition encryption scheme is then defined by the permutation Π on the positions of array A.

  Plaintext $c_1c_2c_3c_4c_5c_6c_7c_8c_9$,

  $$A = \begin{array}{ccc} c_1 & c_2 & c_3 \\ c_4 & c_5 & c_6 \\ c_7 & c_8 & c_9 \end{array} \quad \Pi(A) = \begin{array}{ccc} c_1 & c_4 & c_7 \\ c_2 & c_5 & c_8 \\ c_3 & c_6 & c_9 \end{array} \quad \Pi^{-1}(A) = \begin{array}{ccc} c_1 & c_4 & c_7 \\ c_2 & c_5 & c_8 \\ c_3 & c_6 & c_9 \end{array}$$

  ciphertext $c_1c_4c_7c_2c_5c_8c_3c_6c_9$.

  To decrypt the ciphertext, the scheme (1) puts each ciphertext segment (a string of 9 characters) into a 3×3 array $\Pi(A)$ (the string is read from left to right on each row and from top to bottom by rows); (2) move the characters at position (i, j) of $\Pi(A)$ to position (x, y) defined by the inverse permutation Π^{-1} of Π (which defines the transposition for the encryption above) to get array A ($\Pi^{-1}(\Pi(A)) = A$, in this example, $\Pi^{-1} = \Pi$); and (3) reads the characters in array A from left to right on each row and from top to bottom by rows to get the plaintext.

Below is a secret key encryption scheme framework based on the substitution and transposition techniques shown above for secure communications between a client and a server.

- Both the server and client have a list of N security schemes. $S_0, ..., S_{N-1}$.
- Each security scheme S_i, $0 \leq i \leq N-1$, specifies a substitution encryption scheme and a transposition encryption scheme.

  The substitution encryption scheme is defined by a permutation π_i on the set Σ of m characters to be used for communication.

  The transposition encryption scheme is defined by a permutation Π_i on the positions of the $r \times n$ array.

  S_i specifies a two (or multiple) stage encryption scheme, one stage uses the substitution scheme and the other stage uses and transposition scheme defined above for encryption. Decryption is the reverse process of encryption.
- The client and server use the Diffie-Hellman algorithm (W. Diffie and M.E. Hellman, ”New Directions in Cryptography”, IEEE Trans. on Information Theory, Vol. IT-22, Nov. pp. 644-654, 1976, or refer to the lecture notes) to deliver a secret integer k (key) between them.
- Let $i = k \bmod N$. The client and server use the security scheme S_i to realize a secure communication between them: the sender uses the encryption scheme specified by S_i to encrypt the plaintext and the receiver uses the reverse process of the encryption to decrypt the ciphertext.

Design an encryption scheme for a secure TCP connection using the above framework with $N \geq 5$, $r, n \geq 4$ and $m \geq 40$ characters, including at least the 26 letters, numbers $0 \sim 9$, a symbol for space, and punctuation marks of comma, full stop and question mark. Write programs (in JAVA or C++, please get the approval from your TA if other languages are used) which realize a secure communication by TCP between a client and a server using your encryption scheme.

Submit a document which specifies your design for the encryption scheme, the programs which realize the secure communication, and data samples (before the encryption, after the encryption, captured from the TCP connection in the network, and after the decryption) in the secure communication.

Your programs must work in the virtual networks.

(2) (2 points) Design a stream encryption scheme (Vernam Cipher) for a secure TCP connection. You may cut the plaintext into segments, each segment is one character (or two or four consecutive characters) and viewed as a segment of binary bits, and use a pseudo-random number generator at the client and server to create a sequence of integers as keys for the encryption and decryption. The client and server can use the Diffie-Hellman algorithm to share a same seed for the pseudo-random generators so that the sequence of keys created at the client is identical to that created at the server.

Submit a document which specifies your design for the encryption scheme, the programs which realize the secure communication, and data samples (before the encryption, after the encryption, captured from the TCP connection in the network, and after the decryption) in the secure communication.

Your programs must work in the virtual networks.

(3) (2 points) A simple firewall is a filter which blocks the packets based on the pre-defined filter rules. There are two general strategies to set-up a filter. One strategy is `restrictive firewall` which blocks all packets except those specified. The other is `connectivity-based firewall` which allows all packets to pass through but blocks those specified. The packets to be allowed to pass through the filter in a restrictive firewall and the packets to blocked in a connectivity-based firewall can be defined case-by-case by the filter rules. The filter rules can be defined by the protocol type, host/network IP-address/name, TCP/UDP port number, interface name, etc. In the Linux the filter rules are set-up and maintained by `ipchains` or `iptables` commands (`iptables` is more powerful than `ipchains`).

Set-up and test a simple restrictive firewall and a simple connectivity-based firewall in a host for data packets from/to `eth1`. The restrictive firewall should allow the input packets of a (or a few) specific protocol type(s) from/to `eth1` to pass through the filter and block all other packets from/to `eth1`. The connectivity-based firewall should block the input packets of a (or a few) specific protocol type(s) from/to `eth1` but allow all other packets from/to `eth1` to pass through. Log the allowed packets in the restrictive firewall and the blocked packets in the connectivity-based firewall. Submit a document to explain the design of your firewall (filter rules), the scripts for setting-up and testing the firewall, and a brief summary of each type of the logged packets (protocol type, source and destination IP addresses and host names, etc).

Some notes

- Consult the `man` page to find the details of `iptables` command. Further information is available at `www.netfilter.org`.
- You may need to clean out any existing filter rules created by others before testing your firewall and you **should clean out** what you have set-up after you finish your test. You may include the following sample at the beginning and end of your scripts for the clean out.

  ```
  sudo iptables -F INPUT
  sudo iptables -P INPUT ACCEPT
  sudo iptables -F FORWARD
  sudo iptables -P FORWARD ACCEPT
  sudo iptables -F OUTPUT
  sudo iptables -P OUTPUT ACCEPT
  ```
- You may need to set-up the firewall on one machine and use a different machine to send packets to test the firewall.

Absolutely do not use any of the routers **December, January, February, or March** for setting-up the firewall.

Make sure that your firewall works only on `eth1`.

Schedule your time accordingly since the Lab. may become crowed as the due time approaches and networks will become unstable when firewalls are tested.

Your programs must work in the virtual networks.

Problem 2 (0 points)

1. IP must always check the destination addresses on incoming multicast datagrams and discard datagrams if the host is not in the specified multicast group. Explain how the host might receive a multicast destined for a group to which that host does not belong.

2. The IGMP general query message is used to monitor the memberships in multicast groups. This message may cause a large number of membership report messages. Explain briefly the main techniques used in the IGMP implementation to minimize the overhead of traffic caused by IGMP messages.

3. Assume that host H sends a leave report to router R, expressing H leaves multicast group G. What is the destination address in the IPv4 datagram which carries the leave report? What is the address in the address field in the IGMP leave report? Router R sends a group specific query to confirm the leave. What is the destination address in the IPv4 datagram which carries the query? What is the address in the multicast group address field in the query?

4. For the AS in Figure 1, assume that the routing tables for unicast are computed by OSPF. Assume that G is a multicast group with members in $N_2, N_4, N_5, N_6.N_7$ and TRPF (Truncated Reverse Path Forwarding) is used by routers to deliver multicast messages. When Router R_5 receives a multicast message P with source N_2 and destination G from interface `eth0`, which interfaces R_5 forwards P to? If P receives P from `eth1`, which interfaces R_5 forwards P to?

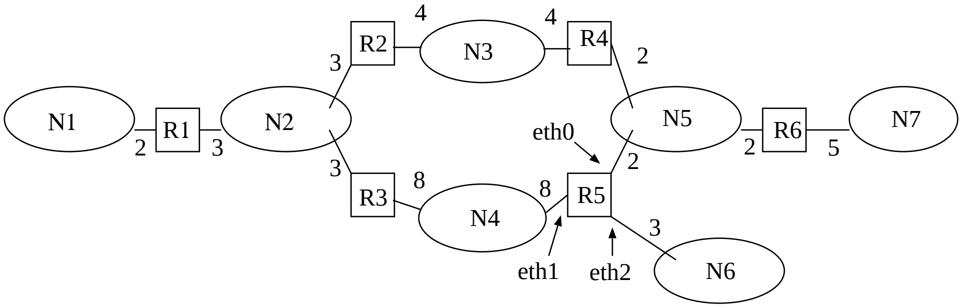

Figure 1: An autonomous system.

5. For the AS in Figure 1, assume that G is a multicast group with 1,3,3,4,4 members in $N_2, N_4, N_5, N_6.N_7$, respectively; the routing tables for unicast are computed by OSPF; and a multicast tree with the member in N_2 as the root is formed by the shortest paths for unicast from the root to all other members in the AS. Assume that forwarding one datagram over one network uses one unit of bandwidth. How many units of bandwidth are used if the root sends one datagram to all other members in G by multiple unicast? How many units of bandwidth are used if the root sends one datagram to all other members by multicast?

6. Main items in an NAT translation table include internal IP addresses, internal port numbers, external IP addresses, external port numbers, NAT port numbers, and payload type. Which items are used to identify a communication session between an NAT box and an internal host? Which items are used to identify a communication session between an external host and an NAT box? Why is the payload type included as a main item in the table?

7. Select an English document (plaintext) of a reasonable length and find the top 5 letters in terms of the high relative frequency in the plaintext. Create the ciphertext of the plaintext by your encryption scheme designed in (1) of Problem 1 and find the relative frequencies of the top 5 letters in the ciphertext. Compare the relative frequencies of the 5 letters in the plaintext and ciphertext.

   Create the ciphertext by the stream encryption scheme designed in (2) of Problem 1, view the ciphertext as a sequence of "letters" and find the relative frequencies of the top 5 "letters" in the ciphertext. Compare the relative frequencies of the 5 "letters" in the plaintext and ciphertext.

8. Vernam cipher is a stream encryption scheme works as follows: A plaintext M of n bits is partitioned into segments $S_1, .., S_r$, each of m bits; a stream of keys $K_1, .., K_r$, each of m bits, are created; in encryption, the ciphertext is computed by $C_i = S_i \oplus K_i$, $1 \leq i \leq r$, $\oplus$ is the bit-wise exclusive-or; and in decryption, the plaintext is computed by $S_i = C_i \oplus K_i = (S_i \oplus K_i) \oplus K_i$.

   Assume that every key $K_i, 1 \leq i \leq r$, is uniformly selected from $\{0,1\}^m$ at random. Let $C_i = c^i_1..c^i_m$ be the bit string of the ciphertext C_i. Prove that $\Pr[c^i_j = 1] = 1/2$ for $1 \leq j \leq m$, where $\Pr[c^i_j = 1]$ is the probability that $c^i_j = 1$.

9. A transposition encryption scheme works as follows.

   To encrypt a plaintext, the scheme (1) cuts the plaintext into segments, each segment is a string of 4 × 4 characters $c_1c_2...c_{15}c_{16}$; (2) puts the string of characters into a 4 × 4 array A as shown below; and (3) reads the elements of A from top to bottom in each column and from left to right on columns to get the ciphertext. The order of reading columns is $(1, 2, 3, 4)$. The decryption is the reverse process of the encryption. Give the permutation Π on the positions of A that defines the encryption. Given the inverse permutation Π^{-1} of

Π that defines the decryption.

$$A = \begin{array}{cccc} c_1 & c_2 & c_3 & c_4 \\ c_5 & c_6 & c_7 & c_8 \\ c_9 & c_{10} & c_{11} & c_{12} \\ c_{13} & c_{14} & c_{15} & c_{16} \end{array}$$

Assume that in the encryption, the ciphertext is obtained by reading the elements of A from top to bottom in each column and reading the columns in the order of $(2, 4, 1, 3)$. Give the permutation Π on the positions of A that defines the encryption. Given the inverse permutation Π^{-1} of Π that defines the decryption.

10. DES (Data Encryption Standard) has been widely used as secret key encryption scheme for many years. However, DES has been broken and is no longer considered as very secure. What weakness do you think DES has and how would you improve the security of DES if you are asked to modify it?

11. Explain briefly how public-key and secret-key encryption schemes are combined to make a more efficient encryption scheme.

    Explain briefly how a public-key encryption scheme is used in digital signature.

12. Explain how the public key and private key are computed in RSA and the encryption and decryption processes of RSA.

    The PGP protocols use several encryption/decryption keys. Explain the purpose of each key.