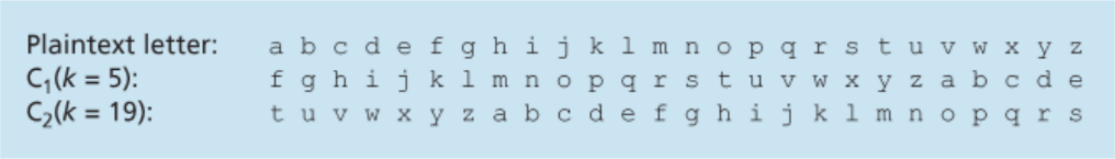
COSC 4P14
Assignment 4

Due date: December 6th, 2020 at 23:55 (11:55pm)
Delivery method: the student needs to delivery the assignment only through Sakai.
Delivery contents: document with answers and [Java, C, C++] codes if applicable (see Submission instructions).
Attention: check the Late Assignment Policy.

Network Security [70]

1.1. Consider the polyalphabetic system shown in the figure below. Will a chosen-plaintext attack that is able to get the plaintext encoding of the message "The quick brown fox jumps over the lazy dog." be sufficient to decode all messages? Why or why not? **[10]**



Plaintext letter:	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
$C_1(k = 5)$:	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
$C_2(k = 19)$:	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s

Figure 1: A polyalphabetic cipher using two Caesar ciphers.

1.2. Let's practice asymmetric encryption with RSA using very short messages. **[20]**

- **a.** Using RSA, choose p=3 and q=11 , and encode the word "dot" by encrypting each letter separately. Apply the decryption algorithm to the encrypted version to recover the original plaintext message.
- **b.** Repeat part (a) but now encrypt "dot" as one message m.

1.3. In the BitTorrent P2P file distribution protocol, the seed breaks the file into blocks, and the peers redistribute the blocks to each other. Without any protection, an attacker can easily wreak havoc in a torrent by masquerading as a benevolent peer and sending bogus blocks to a small subset of peers in the torrent. These unsuspecting peers then redistribute the bogus blocks to other peers, which in turn redistribute the bogus blocks to even more peers. Thus, it is critical for BitTorrent to have a mechanism that allows a peer to verify the integrity of a block, so that it doesn't redistribute bogus blocks. Assume that when a peer joins a torrent, it initially gets a .torrent file from a fully trusted source. Describe a simple scheme that allows peers to verify the integrity of blocks. **[10]**

1.4. Suppose Alice wants to send an e-mail to Bob. Bob has a public-private key pair(KB+,KB-), and Alice has Bob's certificate. But Alice does not have a public, private key pair. Alice and Bob (and the entire world) share the same hash function $H(\cdot)$. **[10]**

- **a.** In this situation, is it possible to design a scheme so that Bob can verify that Alice created the message? If so, show how with a block diagram for Alice and Bob.
- **b.** Is it possible to design a scheme that provides confidentiality for sending the message from Alice to Bob? If so, show how with a block diagram for Alice and Bob.

1.5. In a man-in-the-middle attack scenario, it is known that without sequence numbers, Trudy (a woman-in-the-middle) can wreak havoc in an SSL session by interchanging TCP segments. Can Trudy do something similar by deleting a TCP segment? What does she need to do to succeed at the deletion attack? What effect will it have? **[10]**

1.6. Consider the example in the figure. Suppose Trudy is a woman-in-the-middle, who can insert datagrams into the stream of datagrams going from R1 and R2. As part of a replay attack, Trudy sends a duplicate copy of one of the datagrams sent from R1 to R2. Will R2 decrypt the duplicate datagram and forward it into the branch-office network? If not, describe in detail how R2 detects the duplicate datagram. **[10]**

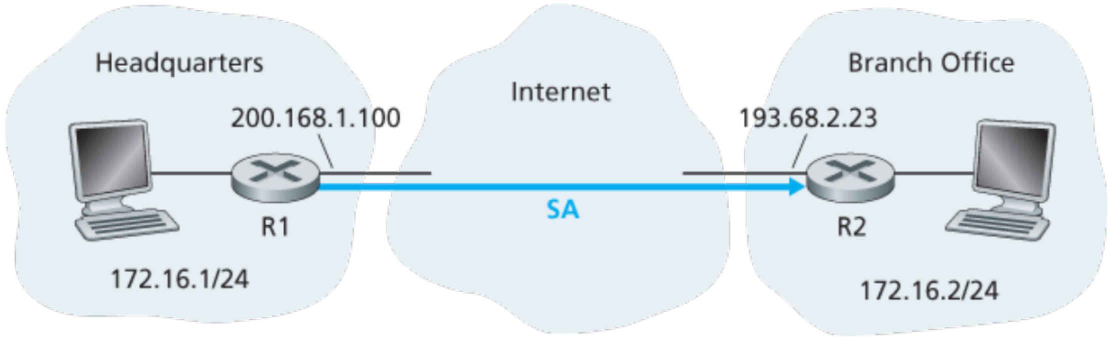


Figure 2: Security association (SA) from R1 to R2.

Secured Java Application [30]

2.1. Create a Java network-based (TCP or UDP sockets) client-server application where the communication channel is encrypted. You may choose AES or RSA encryption for ensuring communication confidentiality. The completion and submission of this item must strictly follow the rules below: **[30]**

a. The student must explicitly tell which encryption method was utilized.

b. The answer for this assignment item must have two parts: a description file and the Java code.

c. The description file must be descriptive, succinct, and objective - reasoning the design decisions.

- External libraries can be used to support the encryption. Students are not expected to implement RSA or AES methods from scratch.
- It must explain the whole process/code, relevant calls/procedures, and [external] libraries.
- It must contain the Java code in an appendix (we are not expecting a long program - it is a simple network-based client-server with encryption).

d. The Java code must compile and run from command line.

- The student must explicitly provide the commands to compile and run the submitted Java application.
- The student must provide the code through a download link. The code must be in zip file. The link must be explicitly indicated in the description document.
- The Java code must match perfectly with the code in the appendix.

e. A performance analysis should be conducted, comparing the non-encrypted and encrypted communications.

- It will measure application-layer end-end delay for each of the scenarios (encrypted and not encrypted).

- The analysis must be included in the description document.
- It is highly suggested, but not needed:
  - To parameterize the analysis (plain text message sizes, encryption arguments - key size)
  - To observed the communication channel using Wireshark.

Marking Scheme

Marks will be awarded for completeness and demonstration of understanding of the material. It is important that you fully show your knowledge when providing solutions in a concise manner. Quality and conciseness of solutions are considered when awarding marks. Every code added to the originals should be well commented and explicitly indicated in the Java files; lack of clarity may lead you to loose marks, so keep it simple and clear.

Submission

The submission is expected to contain one part: a description document. The document can only be in PDF format; it should be single column, at least single spaced, and at least in font size 11. All the submission should be performed electronically through Sakai.

It is highly recommended that students generate their description document following a standard layout. Please utilize the Latex files enclosed with this assignment for generating your document. You can use any Latex-enabled text editor or even OverLeaf to generate your PDF file.

Late Assignment Policy

A penalty of 25% will be applied on late assignments. Late assignments are accepted until the Late Assignment Date, three days after the Assignment Due Date. No excuses are accepted for missing deadlines. However, deadline extensions may be granted under extenuating circumstances, such as medical or physical conditions; please note that granting the extension is under the instructor's discretion. However, deadline extensions may be granted under extenuating circumstances, such as medical or physical conditions; please note that granting the extension is under the instructor's discretion.

Plagiarism

Students are expected respect academic integrity and deliver evaluation materials that are only produced by themselves. Any copy of content, text or code, from other students, books, web, or any other source is not tolerated. The similarity check tool, Turnitin, will be employed to identify plagiarism in the submitted document. If there is any indication that an activity contains any part copied from any source, a case will be open and brought to a plagiarism committee's attention. In case plagiarism is determined, the activity will be cancelled, and the author(s) will be subject to the university regulations.

For further information on this sensitive subject, please refer to the document below:

`https://brocku.ca/academic-integrity/`