

**FACULTY OF INFORMATICS**

**COURSEWORK COVERSHEET**

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBJECT’S INFORMATION:** | | | |
| Subject: | CSCI368 Network Security | | |
| Session: | February 2020 | | |
| Programme / Section: | BCS | | |
| Lecturer: | **Mohamad Faizal Alias** | | |
| Coursework Type  *(tick appropriate box)* | ❑ Individual Assessment | | |
| Coursework Title: | Assessment 1 | Coursework Percentage: | 10% |
| Hand-out Date: | Week 3 | Received By :  (signature) |  |
| Due Date: | Week 6 | Received Date : |  |
| **STUDENT’S INFORMATION:** | | | |
| Student’s Name & ID: |  |  |  |
| Contact Number / Email: |  |  |  |
| **STUDENT’S DECLARATION** | | | |
| By signing this, I / We declare that:   1. This assignment meets all the requirements for the subject as detailed in the relevant Subject Outline, which I/ we have read. 2. It is my / our own work and I / we did not collaborate with or copy from others. 3. I / we have read and understand my responsibilities under the University of Wollongong’s policy on plagiarism. 4. I / we have not plagiarised from published work (including the internet). Where I have used the work from others, I / we have referenced it in the text and provided a reference list at the end of the assignment.   I am / we are aware that late submission without an authorised extension from the subject co-ordinator may incur a penalty. *(See your subject outline for further information).* | | | |
| Name & Signature: |  |  |  |

------------✂ -------------------✂ ------------------✂ -------------------✂ --------------------✂ ---------------------✂ --

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Subject: | CSCI368 Network Security | | Session: | February 2020 |
| Programme / Section: | BCS | | Lecturer: | **Mohamad Faizal Alias** |
| Coursework Type:  (Tick appropriate box) | ❑ Individual Assessment | | | |
| Coursework Title: | Assessment 1 | Coursework Percentage: | | 10% |
| Hand-out Date: | Week 3 | Received By:  (Signature) | |  |
| Due date: | Week 6 | Received Date: | |  |
| **STUDENT’S INFORMATION:** | | | | |
| Student’s Name & ID: |  |  | |  |
| Contact Number / Email: |  |  | |  |

**COURSEWORK SUBMISSION RECEIPT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Assessment Criteria** | | **Total Marks** | **Given Marks** |
|  | **Part 1:** |  |  |
|  | Quiz 1 | 2 |  |
|  |  |  |  |
|  | **Part 2:** |  |  |
|  | Modified DH key exchange protocol and the shared key computations | 3 |  |
|  | AES Encryption and RSA PKC | 3 |  |
|  | CFB - Mode of Block Cipher | 1 |  |
|  | Presentation of work product | 1 |  |
|  | | **10** |  |
|  | | **Penalty** |  |
| Marked by: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_ | | **Final Mark (10 %)** |  |
| **Lecturer’s Comments** | | | |
|  | | | |
| **Penalty for late submission:** | | | |
| 1 day – minus 20% of total mark awarded  2 days – minus 50% of total mark awarded  3 days – 0 mark for this piece of coursework | | | |

**University of Wollongong**

**CSCI368 NETWORK SECURITY**

**February 2020**

**Individual Assessment 1 (10 %)**

**Aims**

This assignment consists of two parts. Part 1 is a Quiz 1 that focuses on revision aspects of Computer Network concepts learnt in the earlier semester. This is to build-up a good foundation before further topics on Network Security are covered.

Part 2 of this assessment 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 and networking
* Understand key transport and secure communication.
* Understand network programming.
* Applying AES and Key Exchange concepts.

**Part 1 – Quiz 1**

**Online Quiz 1 – Computer Networks Revisited**

* The quiz consists of 10 MCQs each. You are given only 20 minutes to complete the quiz. Refer to Blended Learning Plan on Moodle.

**Part 2 – Key Exchange Program Development**:

**Specifications**

Write C++ UDP or TCP programs allowing two parties to establish a secure communication channel. For simplicity, let us call the 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 encryption** in which operates on **128-bit blocks** using a **192-bit key**.

The **key establishment** is done by using modified Diffie-Hellman Exchange scheme (modified for this assignment). Alice has a pair of private/public keys (x1, y1) and Bob has a pair of private/public keys (x2, y2), generated by your program via **KeyGen**. By key exchange, they obtain a share secret key, which is a **192 bit AES session key.** The PKC that handles both public key and private key should be **RSA**.

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:

1. Alice runs **KeyGen** to generate a pair of her private and public keys including all required parameters based on **RSA** requirements. These keys and parameters are stored in directory Alice.
2. Bob runs **KeyGen** to generate a pair of his private and public keys including all required parameters based on **RSA** requirements. These keys and parameters are stored in directory Bob.
3. Alice executes Host.
   1. Host is running and listening to the opened port (you need select a port for your code).
   2. Preferable that the moment the user execute Host program, the program will ask the user to setup the IP and a port number.
4. Bob executes Client.
   1. Preferable that the moment the user execute Client program, the program will ask the user identify the IP and a port number to connect to (which is the Host executed earlier)
   2. Client (Bob) sends his **public key y2** to Host (Alice) together with Hash of this key in SHA-1 format.
   3. Client is ready and listens to the port for next response.
5. Upon receiving the public key and SHA-1 hash from Bob, Alice first verify the integrity of the Public key. Alice then send a “Verified” message to Bob.
6. Alice then computes/generates the **192 bit AES session key and SHA-1 hash of this session key**. This session key is unique for each runs of the program.
7. Alice sends **her public key y1 + SHA-1 hash** to Bob (Client) – expose. Then she send **encrypted 192 bit AES session key + SHA-1 hash** with **Bob Public Key y2**.
8. Bob received the above components. He verify Alice public key y1 with the SHA-1 hash.
9. Then Bob decrypt the encrypted components of 192 bit AES session key + SHA-1 hash using **his Private Key x2**.
10. As for confirmation of receiving the session key, Bob encrypt back a message of “Acknowledge” with the **192 bit AES session key** using **Alice Public key y1** and send back to Alice. (note: only when step 8 and 9 above are verified with their respective Hash).
11. Upon receiving the encrypted message, Alice Decrypt it (using **her local copy of 192 bit AES session key).** If the message represent **“Acknowledge”** she knows that the secure communication now can take place. If match, Alice send a signal message “Ready” encrypted using the AES session key to Bob.
12. Once the message “Ready” sent to Bob, **Alice reverse the 192 bit AES key** to be use for the next secured communication.
13. Bob, upon receiving this Encrypted message “Ready”, he decrypts it and identifies this signal. With this, Bob now, **reverse the 192 bit AES key**.
14. Now, the secure channel is established.
    1. Either Alice or Bob can send a message encrypted with the **reverse** **192 bit AES session key** (from the original key) now. They type the message on their own terminal. The message is encrypted by the program (Host or Client) and sent out.
    2. All message must use **Cipher Feed Back Mode (CFB) mode of block Cipher**.
    3. The received encrypted message and decrypted message are displayed on the screen.
    4. Either one Host or Client can quit the program by using “exit”.

**Questions:**

Attempt these questions either by answering it in your report or directly implement in your code.

What happen if acknowledgment messages such as “verified”, acknowledge” and “ready” never arrives or corrupted during transmission?

What happen if connection is terminated half-way?

How to ensure the program continue running and “catch” this error?

Suggest the best way to tackle the above problem. (**Note**: there are multiple strategies or combination of strategies for the above questions)

**Note:**

For presentation/demo purposes, each steps of generating Private, Public key by RSA, the 192 bit AES session key, The matching process for acknowledgement and the reverse of 192 bit AES session key should be displayed on your terminal. This is to verify that your client and server program conform with all the requirements stated above.

**Coding requirement:**

You need to write (or re-use from library) four functions:

1. KeyGen for RSA Public/Private keys
2. **192 bit AES session key** generation.
3. **Reversing process of 192 bit session key.**
4. **AES** encryption/Decryption using **reverse of 192 bit AES session key** for secure communication.
5. **CFB mode**
6. Host.
7. Client.

**How to run during presentation?**

Your programs should run according to the protocol mentioned above. Host and Client should be executed on different computers. You need a pair of computers. I suggested that you execute the Host program, and the Client program on your friend’s computer (or computer available in the lab – if required). For simplicity, there is no GUI required in this assignment. That is, messages are simply typed on the terminal and printed on the receiver’s terminal.

Remember, during secure communication between Alice and Bob, displayed text message should be as a pair, Encrypted message and the decrypted message. The looping should continue until the moment the user chooses to exit with some sort of termination command.

Presentation day and slot will be announced by your lecturer.

**Submission:**

* Part 1 – Both Quiz 1 is on-the-spot quizzes during Blended Hours.
* Part 2 – Submission is on or before the end of Week 7 via Submission link on Moodle

**Part 2 Submission requirements:**

You are required to prepare a report on your program development (part 2), the execution (with screen captures) and simple testing done. The softcopy of the report together with the program (source files) are required to be in a Zipped folder.

**Plagiarism**

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