SENG2250 System and Network Security
School of Electrical Engineering and Computing
Semester 2, 2020

Assignment 3 (25 marks, 25%) - Due: 13 November, 23:59

Aims

This assignment aims to establish a basic familiarity with network security topics via analysing, designing, and implementing solutions.

Questions

1. Network Security (10 marks)

A bank system, including the internal and external sub-systems, is used by different users. Based on the security requirements, these accesses should be protected in different ways depending on access methods. We will focus on network security for internal and external access to the bank system in this task.

- There are two types of users: bank **customers** and bank **employees**.
- The bank system provides a range of services, such as personal savings, bank statements, money transfer, internal message management, and account management.
- As a customer, it is allowed to use web browsers to access the bank website and make transactions.
- A customer can also use the mobile app to access the services. In this case, The customer is likely to use a mobile network or WiFi connection.
- As a bank employee, it is allowed to access the bank system via the website or desktop application.
- When an employee is travelling for business, it may need to connect the bank servers via a secure connection.

Your task.

a. Consider the security of the above system, discuss **three** potential security issues and provide countermeasures.
   For each of the issues, specify the related security service(s), attack(s) and mechanism(s). The demonstrated issues must not relate to the same security service(s). (**3 marks**)

b. Consider that a bank **employee** requests to modify a bank **customer**'s daily cash transfer limit. Briefly describe the essential security-related step(s) that demonstrate the security checks for the operation. For each step, specify the aimed security service(s). (**3 marks**)

c. An employee accesses the internal system with proper authentication and authorisation. Consider Kerberos, SAML, and OAuth, which one is better for internal system authentication and authorisation? Justify your answer. (**2 marks**)
d. To provide secure connection services for the travelling employees, which of IPSec, SSL/TLS, and SSH, would be a better option? Justify your answer. (**2 marks**)

2. Programming Task (15 marks)

A client and a server are planning to do data exchange. They decide to use a simplified SSL handshake (see Figure 1) to establish a secure channel (session key) then exchange data. The simplified SSL handshake removes the messages for alert, change cipher spec, certificate, etc.

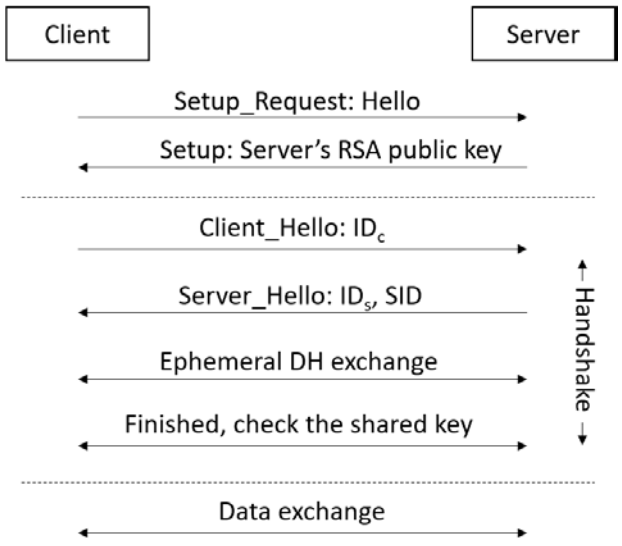

Figure 1. Secure data exchange.

ID_C: client ID; ID_S: server ID; SID: session ID;

Your task: implement the above mechanism in Java. (You are also allowed to use C++/Python.) The following components are mandatory for implementation.

- Fast modular exponentiation (**2 marks**)
- RSA signature scheme. (**3 marks**)
  - RSA key generation: randomly generate two primes p, q (for 2048-bit RSA). Set the public key as the fixed $e = 65537$. Server's RSA public key will be sent to the client in the Steup message. Assume this message can be securely delivered, no security protection is needed. Note that the client DOES NOT have client RSA keys.
  - RSA signature generation: using SHA256 for message digest computation.
  - RSA signature verification: using SHA256 for message digest computation.
  - The underlying hash function is SHA256. You can use it from the Java library.
  - Key generation needs to be implemented using (Java) BigInteger.
  - RSA signature generation and verification need to be implemented using your own fast modular exponentiation method.

- Diffie-Hellman key exchange (**2 marks**)
  - Use the parameters p, g from the System Parameters section.
- HMAC (**2 marks**)
  - Use SHA256 as the underlying hash function.
  - Use the DH key (e.g., $k = g^{xy}$) to generate the authentication key k', such that $k' = H(k)$, where $H()$ is the SHA256 hash function.
  - HMAC is calculated as (refer to lecture 2)

$$H(k', m) = H((k' \oplus opad)||H(k' \oplus ipad)||m)$$

- CBC mode (**2 marks**)
  - Assume a message is always a multiple of 16-byte, i.e. no padding needed.
- The simplified SSL handshake message flow. (**2 marks**)
- Data exchange (**2 marks**)
  - When a shared session key is created, they use 256-bit AES encryption with **CBC** and **HMAC** to protect data confidentiality and integrity, respectively.
  - Demonstrate at least **two** message exchanges, where each message is exactly 64 bytes.

You may refer to the FAQ for more information.

Compilation

- Please provide a readme.txt file for compilation and execution instructions.
- Uncompilable or unexecutable program may receive zero marks.

Input/Output

- Print (to standard input/output) all messages exchanged between the client and server.
- Use a proper output format to demonstrate the message exchange.

System Parameters

Hash function: you should use SHA256 whenever a hash function is needed.

Diffie-Hellman Key Exchange parameters (p, g)

p =
178011905478542266528237562450159990145232156369120674273274450314442865788737020770612695252123463079567156784778466449970650770920727857050009668388144034129745221171818506047231150039301079959358067395348717066319802262019714966524135060945913707594956514672855690606794135837542707371727429551343320695239

g =
174068207532402095185811980123523436538604490794561350978495831040599953488455823147851597408940950725307797094915759492368300574252438761037084473467180148876118103083043754985190983472601550494691329488083395492313850000361646482644608492304078721818959999056496097769368017749273708962006689187956744210730

Notes

- Your implementation MUST be able to handle large numbers. Otherwise, **3 marks** will be deducted.
  - Java
    https://docs.oracle.com/javase/7/docs/api/java/math/BigInteger.html
  - C++ users should use NTL library.
    https://www.shoup.net/ntl/doc/tour-examples.html
- Your implementation MUST use socket programming. Otherwise, **2 marks** will be deducted.
  - Java tutorial

    https://docs.oracle.com/javase/tutorial/networking/sockets/
  - C manual (This can be used with C++ with a few modifications)

    http://man7.org/linux/man-pages/man2/socket.2.html
  - C++ tutorial (uses boost, you would want build tool to manage that, such as https://cmake.org/)

    https://theboostcpplibraries.com/boost.asio-network-programming
  - Python example and documentation

    https://docs.python.org/3/library/socket.html#example

FAQ

1. What is about the "Setup_Request: Hello" message?
   It is just the text "Hello" that initiates the setup phase.
2. Can I use modpow() (or some function like that from the library) for modular exponentiation computation?
   No. You need to implement the function based on the pseudocode of the Lab 2 task.
3. What should I do to create 2048-bit RSA keys?
   You need to choose two 1024-bit prime numbers for p and q, respectively, then follow the RSA scheme.
4. What are the identities like IDs?
   They are random character/number string of your choice.
5. Which is the shared session key for (CBC-AES256) encryption and HMAC?
   It is k'.
6. Can I use the "CBC" encryption mode from the library?
   No. You need to implement CBC encryption and decryption processes.
7. What should I send for the data exchange demonstration?
   Anything, as long as 64 bytes of each message.
8. Can I use the external cryptography library?
   Yes, but you have to implement the required components.
9. Can I reuse the code from the labs?
   Yes.

Submission

All assignments must be submitted via Blackboard (Assessment tab for SENG2250). If you submit more than once, then only the latest will be graded. Your submission should be one ZIP file containing:

- Assessment item cover sheet.
- A PDF file which contains answers to Question 1 and screenshot(s) of Question 2 program execution.
- All source code files of the program and the readme.txt.

The mark for an assessment item submitted after the designated time on the due date, without an approved extension of time, will be reduced by 10% of the possible maximum mark for that assessment item for each day or part day that the assessment item is late. Note: this applies equally to week and weekend days.

Plagiarism

A plagiarised assignment will receive ZERO marks (and be penalised according to the university rules).