DAT 510: Assignment 2

Submission Deadline: 23:59, Friday, Oct. 13, 2022

Implementation of secure communication

In this assignment, you will explore the application of RSA encryption in ensuring secure communication. RSA is a widely used public-key cryptosystem known for its robust security. You will have a practical use case to implement RSA encryption and decryption, highlighting its role in safeguarding sensitive information.

Part I. Understanding RSA Encryption

• Research and provide a brief overview of the RSA encryption algorithm, including its principles and mathematical underpinnings.
• Explain the roles of public and private keys in RSA encryption.
• Explain the applications where RSA is applicable and not applicable.
• Write and reference some modified/modern/recent RSA-based encryption.

Part II. Use Case Scenario

• Present a real-world use case scenario where secure communication is crucial and RSA is applicable. For example, this could involve transmitting sensitive financial data, medical records, or confidential business documents.
• Explain why RSA encryption is a suitable choice for securing communication in this scenario.

Part III. Generating RSA Key Pairs

• Using Python as a programming language, write your own script to generate an RSA key pair consisting of a public key and a private key for the use case scenario discuss in Part 1.
• Demonstrate selecting suitable values for the key length and encryption exponent (e).

Part IV. Encryption Process

• Develop a program that mimics the use case scenario discussed above which takes a plaintext message as input and encrypts it using the RSA public key generated in Part III.
• Provide a step-by-step explanation of the encryption process, including the modular exponentiation.

• Provide GUI for the encryption process to mimic the use case scenario discussed above.

Part V. Decryption Process

• Create a program that mimics the use case scenario discussed above to decrypt the ciphertext generated in Part 4 using the RSA private key.
• Describe how the decryption process works, emphasizing the modular exponentiation with the private key.
• Provide GUI for the decryption process to mimic the use case scenario discussed above.

Part VI. Security Considerations

• Discuss the security strengths of RSA encryption, such as its resistance to known attacks.
• Highlight any limitations or vulnerabilities that should be considered when implementing RSA.

Part VII. Conclusion and Recommendations

• Summarize the key findings from your research and practical implementation.
• Offer recommendations on when and how RSA encryption should be employed for secure communication based on the strengths and limitations identified.

Assignment Approval (by TA and SA)

Assignment approval will have a weight on your grade for the assignment. If you are not going to get the approval before the deadline, your assignment will not be evaluated and you will fail the assignment.

What needs to be done to get the approval for the assignment:

1. Show all parts of the assignment are working i.e. show the code with proper comments, and results.
2. Code should have a proper README file that describes the contents of the directory and any special instructions needed to run your programs (i.e. if it requires any packages, commands to install the package. describe any command line arguments with the required parameters).
3. Source code submitted for the assignment should be your own code. If you have used sources from the internet everything should be added to the references. If you used someone's code without reference, that will also be treated as plagiarism.
4. Provide the references in Code and Report, show these parts for TA's and Student Assistants.
5. You should **NOT** use available libraries/packages/classes for implementing the core functionality of the assignment.

You may use any "reasonable" programming language for part one of the assignment. Reasonable languages include: Java, C, C++, Python, MatLab, R and others with permission of Jayachander Surbiryala (Email: `jayachander.surbiryala@uis.no`) or Chunming Rong (Email: `chunming.rong@uis.no`).

Assignment Submission

Deadline: 23:59, Friday, Oct. 13, 2022 (submit your assignment through canvas)
Final submission:

1. Source Code
   - Source code submitted for the assignment should be your own code. If you have used sources from the internet everything should be added to the references. If you used someone's code without reference, that will also be treated as plagarism.
   - Source code should be single, compressed directory in .tar.gz or .zip format.
   - Directory should contain a file called README that describes the contents of the directory and any special instructions needed to run your programs (i.e. if it requires and packages, commands to install the package. describe any command-line arguments with the required parameters).
   - You should **NOT** use available libraries/packages/classes for implementing the core functionality of the assignment.
2. A **separate** report with PDF format
   - Texts in the report should be readable by human, and recognizable by machine;
   - Other formats will **NOT** be opened, read, and will be considered missing;
   - Report should follow the formal report style guide in next page.
   - Each student should write an individual report. Each report will be checked for plagiarism. If it is copied from some where else, you will fail the assignment.

NOTE: Please upload the archive file in *.zip, *.tar only and report in *.pdf format only to the website https://stavanger.instructure.com/.

Note: The assignment is individual and can **NOT** be solved in groups.

Project Title

Abstract

Write an abstract of your complete assignment.

1. Introduction

This section will start will Part I of the assignment where key points are given. In the subsection, write the use case scenario clearly and thoroughly.

2. Design and Implementation

A detailed description of your implementation part, which consists of generating RSA key pairs, encryption process, and decryption process.

3. Test Results

Results of testing the software, as you observed/recorded them. Note that this section is only for observations you make during testing. Your analysis belongs in the Discussion section.

4. Discussion

Discuss security considerations of Part VI.

5. Conclusion

A paragraph that restates the objective from your introduction, relates it to your results and discussion, and describes any future improvements you would recommend.

References

A bibliography of all of the sources you got information from in your report.