ECE424: Introduction to Information and Network Security

Final Project
Due: 12/15/2024

Project guidelines. This is a group-based project. Each group may consist of up to two people. Please ensure that you form your group and register it on Canvas by Tuesday, December 3.

You are also welcome to complete the project individually if you prefer. However, please note that the expectations and grading criteria remain the same whether you work alone or in a group. If you choose to work individually, please also register a group on Canvas.

If you have not registered by December 3, a partner will be automatically assigned to you using the Canvas group assignment tool.

Collaboration Policy. Groups are expected to work independently. While discussing general concepts with other groups is allowed, sharing specific ideas, codes, or solutions is strictly prohibited. All submissions must reflect the collective effort of the group members and comply with academic integrity policies.

Submission format. Each group must submit:

1. One report in PDF format: Use 11pt Arial font, single-spaced, with 1-inch margins on all sides. The report must not exceed 5 pages. Include the names of all group members on the first page.

2. A ZIP file containing all code: The ZIP file must include all relevant code files and documentation necessary to understand and run the code. Include clear instructions in a README file for running the code, if applicable.

Grading. The project will be graded primarily on the quality and clarity of the report, not the advance of the implementation. Specifically:

- Clarity of Explanation: The report should clearly describe the problem, methodology, and results. All statements need to be paired with sufficient evidence. All interpretations need to follow logic. The best way to present your work is through tables and figures (histogram, etc). If you cannot justify some statements, they should not be included in the report. **The quality of your content is more important than the quantity of your content. Do proofread before submission.**

- Quality of Documentation: The report should be well-structured, self-contained, and easy to follow. Assume the audience is not familiar with the topic.

- Citations: Properly cite any external resources or references used in your work.

(Q1) Message Preprocessing.
In various algorithms that we reviewed in class, a concatenation of messages is sent from User 1 to User 2. In this problem, we design a protocol that enables two users concatenation of k messages, $m_1, \ldots, m_k$, which has the following properties:

1. The receiver does not know a priori the number of messages concatenated and the length of each message.

2. The receiver is able to uniquely identify the k messages from the received bit stream.

Here is the algorithm: Assume that User 1 wants to send concatenations of $m_1, \ldots, m_k$ to User 2. Assume that m_i, the message i, $i = 1, \ldots, k$, is a bit string of length n_i. Assume that $k \leq 32$. To encode the concatenation of the messages, User 1 proceeds as follows:

- It allocates the first 5 bits to the number of messages that are to be communicated (k). For example, if it intends to send $k = 3$ messages, the first 5 bits are 00011.
- Then for $i = 1, \ldots, k$,
  - it allocates 8 bits (1 byte) to describe n_i (the length of m_i)
  - it allocates n_i bits to describe the content of message m_i

1. What is the length of the final bit string, as a function of $n_1, \ldots, n_k$?

2. In the described method, what is the maximum length of each message m_i that the algorithm can handle?

3. Argue that after receiving the message, the receiver can identify all the messages $m_1, \ldots, m_k$ with no error.

4. Implement the encoding and decoding of the described algorithm.

5. Test your codes by first randomly choosing a number k in $\{0, \ldots, 31\}$ and then randomly choosing $n_i \in \{1, \ldots, 5\}$, $i = 1, \ldots, k$. Then, randomly generate the messages, $m_1, \ldots, m_k$, such that each message m_i is a random element in $\{0, 1\}^{n_i}$.

(Q2) Needham and Schroeder protocol with symmetric encryption.
In this question, you will implement the Needham and Schroeder key distribution protocol using symmetric encryption.

Here are the notations used in the description of the algorithm and the rest of the problem.

- ID_A and ID_B are the identities of Alice and Bob, respectively.
- N_A and N_B are the Nonces generated by Alice and Bob, respectively.
- K_{AB} is the shared secret symmetric key between Alice and Bob generated by the TTP.
- K_A and K_B are the secret symmetric keys of Alice and Bob, respectively. Alice and Bob only know their own secret keys. The TTP knows all the secret keys.

- $\mathcal{E}(K, m)$ denotes the encryption of message m using key K. You can choose to use either the DES or AES protocol with the ECB mode here.

Here is the protocol:

Step 1) Alice $\rightarrow$ TTP: $\mathcal{E}(K_A, \mathrm{ID}_A \| \mathrm{ID}_B \| N_A)$

Step 2) TTP $\rightarrow$ Alice: $\mathcal{E}(K_A, \mathrm{ID}_B \| N_A \| K_{AB} \| \mathcal{E}(K_B, K_{AB} \| \mathrm{ID}_A))$

Step 3) Alice $\rightarrow$ Bob: $\mathcal{E}(K_B, K_{AB} \| \mathrm{ID}_A)$

Step 4) Bob $\rightarrow$ Alice: $\mathcal{E}(K_{AB}, N_B)$

Step 5) Alice $\rightarrow$ Bob: $\mathcal{E}(K_{AB}, N_B - 1)$

1. Implement the described protocol. In your implementation, the TTP should keep a table of users, their identities, and their corresponding keys. You can decide how you to represent the identities of the users. The number of users in the systems should be a flexible free parameter. Show the successful key exchange between Alice and Bob.

2. If Eve acquires $\mathcal{E}(K_B, K_{AB} \| \mathrm{ID}_A)$ from Step 3, she can replay this message, and Bob can not tell the freshness of K_{AB}. To address this issue, read the solution provided in the provided **Paper 1** and implement it.

3. As we studied in class, the ECB mode of operation is not secure. Argue the vulnerability of this approach with evidence. Implement the CTR mode to enhance security. What extra information should the third party store? Show that CTR is more secure than the ECB.

4. **(Optional)** Implement some other mode of operation and compare its performance with the ECB and CTR modes operation.

(Q3) Needham and Schroeder protocol with public-key encryption.
As we discussed in class, the primary Needham and Schroeder protocol is typically coupled with symmetric encryption methods. In this question, we will implement a variation of the Needham and Schroeder protocol that employs public encryption.

The following notations are used:

- S represents the identity of the TTP.

- $K_{e,A}$, $K_{e,B}$, and $K_{e,S}$, the public keys of Alice, Bob, and the TTP, respectively. $K_{e,S}$ is known and stored by all users. $K_{e,A}$ and $K_{e,B}$ are known to the public but not stored by Alice and Bob.

- $K_{d,A}$, $K_{d,B}$, and $K_{d,S}$ the description (private) keys of Alice, Bob, and the TTP, respectively.

- The encryption $\mathcal{E}$ refers to the RSA encryption $\mathcal{E}_{\mathrm{RSA}}$.

The protocol is as follows:

Step 1) Alice $\rightarrow$ TTP: $\mathcal{E}(K_{e,A}, \mathrm{ID}_A, \mathrm{ID}_B)$

Step 2) TTP $\rightarrow$ Alice: $\mathcal{E}(K_{d,S}, \mathrm{ID}_B \| K_{e,B}\})$

Step 3) Alice $\rightarrow$ Bob: $\mathcal{E}(K_{e,B}, N_A, \mathrm{ID}_A)$

Step 4) Bob $\rightarrow$ TTP: $\mathcal{E}(k_{e,B}, \mathrm{ID}_B \| \mathrm{ID}_A)$

Step 5) TTP $\rightarrow$ Bob: $\mathcal{E}(K_{d,S}, \mathrm{ID}_A \| K_{e,A})$

Step 6) Bob $\rightarrow$ Alice: $\mathcal{E}(K_{e,A}, N_A \| N_B)$

Step 7) Alice $\rightarrow$ Bob: $\mathcal{E}(K_{e,B}, N_B)$

1. Implement the described protocol. Show the successful key exchange between Alice and Bob and a successful digital signature.

2. This protocol has a potential vulnerability. Read the attached **Paper** 2 on how this issue can be addressed. Implement the proposed modification.