Project Description

Form a team of 1 – 3 people to complete the final project.
The project has 100pts + 20pts (extra credit)

Your choice: assigned project, or your proposed project, or survey

Assigned project:
- See the description of the assigned project on the next page.
- Report requirement: at least 5 pages, at most 1.5 line spacing, using Time New Roman, 11pt.
- Deliverables for grading:
  • Submit your report along with your software code.
  • Demo your system to the class (if time allows)

Your own project:
- You can propose your own project, as long as it is related to security.
  • It does not necessary to be cryptography based. But the project must use cryptographic primitives.
  • The workload should be comparable to the assigned project. Otherwise, there will be points deduction.
- Report requirement: at least 5 pages, at most 1.5 line spacing, using Time New Roman, 11pt.
- Deliverables for grading:
  • Submit your report along with your software code.
  • Demo your system to the class (if time allows)

A survey on existing studies:
- You can write a comprehensive survey on a security topic. Examples: Survey on attacks against modern hash functions; Security vulnerability and analysis of RSA; Survey on Denial-of-Service Attacks and Defense
- Page requirement: at least 15 pages, at most 1.5 line spacing, using Time New Roman, 11pt.
- Format requirement:
  • The survey must include an abstract and the main report.
  • Feel free to present the content of the survey using as many sections as you can. But the report must have an introduction section, a future research direction section and a conclusion section.
- Deliverables for grading:
  • Submit your survey.
  • Present the survey to the class (if time allows)
Assigned Project: Secure Instant Point-to-Point (P2P) Messaging

In this project, you need to design a secure instant messaging tool for Alice and Bob (like gtalk, skype or icq chat). The system supports the following functions

- Alice and Bob can use the tool to send messages to each other.
- Alice and Bob share the same password (or passphrase), they must use the password to set up the tool to correctly encrypt and decrypt messages shared between each other.
- Each message during Internet transmission must be encrypted using a 56-bit key.

You can use any computer language (Java, C++, Python, …) and leverage any existing open-source software, tools, or commands (e.g., md5sum, sha1sum) to design the system.

Some design issues/requirements you need to consider:

- With a 56-bit key, what cipher you should use?
- DONOT directly use the password as the key, how can you generate the same key between Alice and Bob to encrypt messages?
- What will be used for padding?
- A graphical user interface (GUI) is preferred. When send a message, display the sent ciphertext. When receive a message, display the received ciphertext and decrypted plaintext.
- How should Alice and Bob set up an initial connection and also maintain the connection with each other on the Internet? (You may refer to socket/network programming in a particular computer language)
- If Alice or Bob sends the same message multiple times (e.g., they may say “ok” many times), it is desirable to generate different ciphertext each time. How to implement this?
- Have a good plan to show all major functions in the demo.

Justify all your design.

Extra Credit (20 pts):

- Design a key management mechanism to periodically update the key used between Alice and Bob. Justify why the design can enhance security. (10 pts)
- Think about this scenario: if you can hide the detailed procedure of your encryption algorithm, how would you improve the security by designing a new algorithm? For example, you may do two encryptions using different standard ciphers, then XOR the two outputs together. Please give your new design and justify its security and efficiency. (10 pts)