CSE539 Fall 2017
Project Guidelines

September 24, 2017

> *A carelessly planned project takes three times longer to complete than expected; a carefully planned project takes only twice as long.* – Golub's Law

1 Goals

The goal of the project is for you to gain experience and knowledge beyond the material covered in class. Your project *emphasis will be practical*. In your project, you should clearly show that you have gained significant knowledge and experience beyond the topics covered in class. You should show in your project and your report that you have learned best practices for: (1) secure coding in general, (2) coding for crypto in particular (security and performance).

While you cannot be expected to become experienced at these practices from a semester project, your implementation should highlight particular practices that you are following. The grade you will get will depend on how involved your work is and how polished it is.

There are many resources online that you can use for acquiring the new knowledge and experience. I suggest some below, but you can use other resources. The resources you should use must be high quality.

You can define your own project and I will give you feedback on the appropriateness of the project. For any project you propose, you should explain how you will satisfy the goals outlined above. Example projects include:

- Implementing hash functions.
- Implementing symmetric key encryption.
- Implementing public-key encryption
- Implementing a PRG and subjecting it to randomness tests

You should implement some cryptographic function for your project. This need not be a single function (like AES or SHA-256), but I will refer to it as a function.

Under no circumstances should you refer to any existing source code in developing your solutions. You will be asked to sign a statement to that effect as part of your report.

C and or C++ are the acceptable programming languages.

2 Groups

Groups of 2 or 3 students are allowed. All students in a group will get the same grade on the project report and submission but not necessarily on the demo.

3 Proposal

The proposal should list the following:

- Group members
- Project topic
- What you expect to learn from the project

The proposal is due Monday October 2nd in class, but I encourage you to talk to me about your proposal before then.

4 Demo

I will schedule demos after Thanksgiving. I will meet with each group who will demo the project and explain how the project meets the goals outlined above. You can think of the demo as an oral exam for the project. Projects reports and code will be due before the demo.

Groups should come very well prepared for the demo . The demo time will be limited and there will be other groups waiting so we cannot afford using the demo time inefficiently.

5 Resources

- Standards and guidelines
  - NIST guidelines. There are a number of guidelines by NIST on implementing crypto in the federal government, key management.
    AES standard: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
  - NIST example algorithms. This includes example input/output for various algorithms.
    http://csrc.nist.gov/groups/ST/toolkit/examples.html
  - NIST secure hash standard. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf
- Secure coding. The Software Engineering Institute has a number of guidelines on security coding. You can check https://www.securecoding.cert.org for specifics.

6 Report

The report should not be too long and should document what you learned. I am not interested in detailed description of the cryptographic function you implemented; there are detailed references for those. I am interested in what you learned. Remember that the report's goal is to document the learning you have done while doing the project. **It should not read like it is copied and pasted from NIST or other standard documents. It should read like you are describing what you have done and what you have learned**. Here is an outline of the report format. **You should follow this format very closely**.

1. Introduction. This should be a summary of the report and of what you learned. It should not be longer than 1.5 pages. It should contain a brief description of what you have done, what the implementation involves and a high level description of what you learned.

2. Implementation description. This part emphasizes the functional implementation as specific in the documentation you used (typically a standard document or something equivalent like an RFC or other standard documentation for your function). This section should show that you have indeed read the documentation by highlighting important parts or parts that you found particularly interesting.

3. Crypto learning. In this section you should describe what you learned relating to coding cryptographic functions. You should show that you have done a serious effort at implementing the protocol. This can be achieved by

   - Showing that you are aware of attacks on implementations and describing what you have done in that regard or describing what you could have done.
   - Identifying particular parts of your code that are designed to handle attacks or common implementation pitfalls or identifying particularly weak parts of your code.

   In brief, you should show that you have done a serious effort to have an implementation that takes into account potential implementation pitfalls, that you are aware of attacks on the function that you are implementing, and that you are aware of the weaknesses of your implementation.

4. Secure coding. You should show that you have studied SEI secure coding practices as they relate to your choice of language and that you have followed such practice in your code. I realize that you might not have used everything you have learned and you should document some of what you learned and did not use. In brief, you should show that you have done a serious effort at learning some secure coding practices.

5. Summary.

6. References.

7. Appendix A. should contain a printout of your documented code. The documentation should make it easy to identify secure coding practices and crypto coding practices that you follows

8. Appendix B. A list of crypto coding practices that you have learned. This can be something as simple as using the right crypto library for random number generation for example.

9. Appendix C. A list of secure coding practices that you have learned and used and those that you have learned and not used