CS 198 Codebreaking at Cal
Spring 2023 Final Project

Open-Ended Report

This is the **open-ended** version of the final project. You can also complete the coding-based challenges version instead (you only need to complete one or the other).

To pass, you will need to get **22/30 points** ($\approx 75\%$) on the project. We won't grade super harshly, but make sure not to neglect the report entirely. This assignment is due **Sunday, April 30th, 11:59PM**.

Outline

The open-ended final project consists of writing a short report on a cryptosystem we did not cover in class, providing an implementation as a Python script, and a proof of security OR example attack. The scheme does not need to be considered secure – you can choose a scheme that is insecure, and provide an example attack + explanation instead of a security proof.

Report (10 points)

The explanation should be 1-2 pages long, **with technical writing**. Do not write paragraphs of words explaining how it works – use technical writing with math symbols to make it easy to read. You should write this as if it were a specification for a cryptographic system, to be read by security engineers / cryptographers.

The report will be graded on technical depth, quality of explanations, and correctness.

Implementation (10 points)

Provide a Python script with functions to **generate keys, encrypt, and decrypt** your scheme. If you do a digital signature scheme, replace encrypt and decrypt with sign and verify respectively. For key exchange protocols, generate keys + the functions each party executes to communicate/derive the shared key is sufficient.

The implementation does not have to be perfect (i.e. no need to worry about side channel attacks), just implement the expected behavior correctly with no glaring security issues.

The implementation will be graded by testing for common cases and looking for any glaring security weaknesses.

Security Proof / Attack (10 points)

If your scheme is secure, provide a proof sketch for its security. For symmetric schemes, outline how each component of the scheme provides security.

If your scheme is insecure, outline an attack on the scheme with an example.

This section will be graded on the correctness of your proof / attack.

Example Ideas for New Cryptosystems

Here's a list of cryptosystems not covered in class that might be of some interest for your report. Note that you do not need to choose from this list if you don't want to, as long as your scheme has not been directly covered in class.

1. Hill cipher
2. DES
3. RC4
4. ElGamal
5. BLS signatures
6. Tripartite Diffie-Hellman using the Weil pairings
7. NTRU
8. NewHope key exchange protocol

Contributors:

- Ryan Cottone