MCS 425: Codes and Cryptography (Spring 2020)
Project Guidelines

Objectives.

1. The project itself (a report, a webpage, code with documentation, etc.).
2. A 12-minute presentation (1-min transition + 9-min talk + 2-min Q&A).

Teams. You may work alone or with one other student. The grading standards for contribution and depth will be more stringent for teams of 2 than for teams of 1.

Submission instructions.

- Email the content of your project as a single `.zip` or `.pdf` file to `yucheng2@uic.edu`.
- Name your file `<Name>.ext` or `<Name1>_<Name2>.ext`, where `Name = <Lastname><Firstname>`. For example, `ChengYu.zip` or `ChengYu_DoeJohn.pdf`.

Timeline.

- **March 25:** Decide upon teams and meet with me to go over your project idea.
- **April 17 – April 29:** Class presentations.
- **May 4:** Project due (at 11:59pm).

Grading. The projects will be graded out of 20 points based on the following rubric:

- *Contribution and Originality* (10 points):
  - Will the project be useful to researchers or students in cryptography?
  - Does the project shed new light on any aspect of cryptography?
  - How creative is the idea and the execution of the project?
  - Does the project offer something that is not available in textbooks or on the Internet?
- *Depth* (4 points):
  - How thorough is the project?
  - Does the project show a deep understanding of the subject?
- *Presentation* (6 points):
  - Did the presentation inspire other students to learn more about the topic?
  - Did the presentation give a good overview of the project?
  - Was the presentation well-organized?

Examples of project ideas. The topics below are just examples. You are highly encouraged to pick a topic that aligns with your interests!

- Survey the history of some classical cryptosystems and/or where they are still used (e.g., one-time pads, Enigma).
- Compare letter frequencies in different languages. For these languages, write code for ciphertext-only attacks on Vigenère ciphers.
- Implement your favorite factoring algorithm and compare it with available factoring packages/modules.
- Investigate the effectiveness of different primality tests. Survey more sophisticated primality tests (e.g., AKS).
- Build an interactive webpage to explain how the RSA cryptosystem works.
- Survey quantum factoring algorithms and quantum-resistant cryptography.
- Compare empirically the collisions of hash functions to the birthday problem approximation.
- Explore basic cryptographic protocols for secure multi-party computation (e.g., oblivious transfer, garbled circuit).
- Gather a list of interesting and neat cryptographic problems and their solutions (e.g., coin flipping over the telephone, Yao's Millionaires' problem, private set intersection).
- Explore elliptic curves and their applications in cryptography.
- Explore lattice-based cryptography and/or homomorphic encryption.
- Interview researchers in cryptography.