CSCI 390 Cryptography- Course Announcement
Fall 2024

Instructor: Michael Levet; levetm (at) cofc (dot) edu.
Meetings: TR 3:35-4:50

Prerequisites: CSCI 221 (C- or better), and Math 207.

Grading: There will be regular homework and quizzes, as well as two midterms. In place of a traditional final exam, there will be a take-home final reflection assignment.

Course Description: Cryptography deals with securely communicating information so that only authorized parties can read the intended message. In this course, we will investigate cryptosystems such as transposition ciphers, block ciphers, the Enigma machine from World War II, AES, and RSA. In the process, we will carefully develop the mathematical tools necessary to implement these cryptosystems, including for instance, the extended Euclidean algorithm, the cycle decomposition of permutations, Euler's totient function, and finite fields. The goal will be to thoroughly understand *how* and *why* the cryptosystems work, including being able to work through small examples by hand. While there may be occasional programming assignments, we will not discuss secure implementation (cryptographic engineering).

After our discussions of cryptography, we will introduce information theory, which aims to quantify the amount of information present in a given channel. We will focus on one such key measure: entropy. One interpretation of entropy is that it provides a precise measure of how much a given piece of data can be compressed. It is of interest to compress our data as much as possible *without* sacrificing information (lossless compression). The Huffman coding scheme provides a surprisingly simple and elegant way to achieve optimal and maximal lossless compression, and we will discuss this scheme in detail.

Lastly, we will discuss error-correcting codes. In practice, data signals can become damaged or distorted during transmission (e.g., solar winds damaging signals to the International Space Station). It is important to identify and correct these errors *efficiently*. We will investigate error-correcting codes such as the Hamming and Hadamard codes.

While this course is not proofs-intensive, a secondary goal will be for you to develop your mathematical maturity, including your ability to read and formulate mathematical proofs. Many of our tools will be number-theoretic in nature, and number theory serves as an accessible vehicle to learn how to formulate mathematical proofs. I will rigorously prove theorems in class, and there will be a small number of proofs-based problems throughout the course. A tertiary goal of this course will be to introduce important results from number theory, such as the infinitude of the primes, unique factorization of the natural numbers (the Fundamental Theorem of Arithmetic), and Fermat's Little Theorem. These results and their proofs are beautiful in their own right, and also worthwhile to know as computer scientists.