Cryptography: Algorithms and History

Syllabus, updated Dec. 21, 2021, with Daily Schedules

Flabbergastingly, it is possible for two people to talk to each other while another listens, and for those two people to have a conversation that the third person cannot understand or even hope to understand with all the computer power in the world, and — now we get to the flabbergasting part — *to do this with no prearranged code or secrets.* The method is called public-key cryptography. It is at the heart of every modern messaging system that has (or claims to have) privacy as one of its features. Interestingly, the method is defeatable, but only if the third person not only can listen in, but can actually intercept and substitute their own messages into the communications channel. This is known as the man-in-the-middle attack. Public key cryptography will be one the final topics in the course, but only after we have progressed through many much-less-advanced schemes. To bring the historical and political importance of the material to life, we will also work through Snowden's *Permanent Record*. We will close with some 21st century topics: blockchain, quantum cryptanalysis, and quantum cryptography.

In the age of ubiquitous computers encryption and decryption is all done in software. Therefore we will proceed in our study of cryptography in three ways:

- By learning about the long and steadily advancing history of cryptography
- By studying cryptographic algorithms
- By writing Python code for encryption and decryption

Class activities and homework assignments will include: studying the entire history of cryptography using a well-regarded popular history, *The Code Book* by Simon Singh. Because early algorithms were very simple, there will be an extremely natural progression in difficulty in the algorithms we study and this progression will be mirrored as we progress from writing simple programs to more complex ones. *This course will require downloading software (PyCharm) onto a laptop.*

A large number of supplementary readings are listed in the detailed daily schedules.

Overall, class time and studying time will be split about three ways between cryptographic history, cryptographic algorithms, and learning to implement these algorithms in the Python language. There will be an end-of-term-2 midterm to assess progress in programming ability.

Course credits: four (standard semester-length course)

Cryptography Daily Schedule Term 2

Course home page

See also Daily Schedule Term 3

Week 1 — Transposition Ciphers

- Thursday, Sept. 2 — Reading: Intro and pp. 1-14 of Singh — Steganogrpahy, Ciphers, Codes, Transposition Ciphers, Substitution Ciphers — Rail Fence Notebook

Week 2 — Substitution Ciphers

- Monday, Sept. 6 — Reading: pp. 14-32 of Singh — Substitution Cipher Breaking using Frequency Analysis — Problem Set 1: 3-line Rail Fence Notebook — Rail Fence with Arbitrary Key
- Bonus: Rail Fence with Slicing Tricks
- Thursday, Sept. 9 — Reading: pp. 32-44 of Singh — Substitution Cipher — Problem Set 2: Frequency Counting

Week 3 — Cracking Substitution Ciphers and Introducing the Vigenere Cipher

- Monday, Sept. 13 — Reading: pp. 45-63 of Singh — The Vigenere Cipher — Problem Set 3: Crack Stage 1 and Stage 2 of Singh's Cipher Challenge — Stage 1 we completed by hand after using your frequency counting program to get an indication of which letters are E, T, and A; Stage 2 do by brute force, (e.g., by writing a program that tries all 25 keys) — Introducing Functions and Unit Tests
- Thursday, Sept. 16 — Reading: pp. 63-78 of Singh — Babbage Breaks the Vigenere Cipher — Problem Set 4 for Thursday: Encipher with the Vigenere Cipher (this notebook has all the steps for setting up a new project with the important code in functions and with unit tests to test those functions) — Introduction to the Python Debugger

Week 4 — Cracking the Vigenere Cipher and the Playfair Cipher and Introducing the ADFGVX Cipher

- Monday, Sept. 20 — Reading: Finish Chapter 2 of Singh — Problem Set 5 for Monday: You are going to crack Stage 4 of Singh's Challenge by starting with Notebook 2.2 which finds repetitions and separations of words (usually of length 4); once the key length is known, apply frequency analysis to the Caesar ciphers

- Thursday, Sept. 23 — Reading 1: Start Chapter 3 of Singh pp. 101-124 — Reading 2: Introducing the ADFGX and ADFGVX ciphers (Singh Appendix F) — Problem Set 6: Upgrade your Vigenere-cracking notebook to use matplotlib.pyplot for bar charts of frequencies, and to use list comprehensions — Notebook with Singh Challenge 4 Cracked

Week 5 — Cracking and Coding the ADFGVX Cipher

- Monday, Sept. 27 — Reading 1: Military Cryptanalysis, Part IV Excerpt — Reading 2: Konheim 1985 Abstract
- Thursday, Sept. 30 — Continue working on the two readings from Monday — The goal is to understand Friedman in a completely detailed way and to generally understand Konheim — Complete the notebook that introduces tuples and implements the ADFGVX Cipher

Week 6 — The Mechanization of Secrecy

- Monday, Oct. 4 — In Singh, read the remainder of Chapter 3, "The Mechanization of Secrecy," which is pp. 124-142 — For understanding the Enigma operator's procedure, it is helpful to see an actual Monthly Key Sheet — Notebook with my ADFGVX Cipher Solution — As the next assignment, apply the method that Friedman describes on pp. 65-69 to crack the following two messages:

```
Cipher 1:FORRFNBAUDCLEMRSTLLGCSLNTEYOPLNINEEUUODOREY
Cipher 2:RENAETLSEEDIEYIXIUCCLDCICEUUDTIVHSLNERDLRFNEOSIOPLN
```

- Thursday, Oct. 7 — Instead of using Singh, Chapter 4, for the ingenious tale of how the Enigma was cracked, we will use Gordon Welchman, The Hut 6 Story (2016 re-issue) — Read pp. 31-71 of Welchman — Python notebook: Implementing the Reflector — The reflector notebook introduces Object-Oriented Programming (your assignment is to to start familiarizing yourself with objects and then to finish implementing Reflector) — Photo of General Heinz Guderian with an Enigma Operator

Week 7 — Breaking of Enigma

- Monday, Oct. 11 — Read pp. 71-115 of Welchman — The Enigma Notebook goes much further into objects (your assignment is to finish implementing ScramblerWheel at which point all the tests of ScramblerWheel and ReversedScramblerWheel should pass, and the encryption of `TOTHEPRESIDENTOFTHEUNITEDSTATES` should agree with the emulator)

- Thursday, Oct. 14 — Read pp. 119-169 of Welchman — Reproduction of the Oct 1944 Wehrmacht Keys — Assignment: Generating the Enigma Sheets — Corrected Wheel Order 1 3 2 Sheet A for Type(2, 5) Females (reload for corrected version)

Cryptography Daily Schedule Term 3

Course home page

See also Daily Schedule Term 2

Week 8 — Defeating Enigma with the Bombes

- Thursday, Oct. 28 — Finish the Welchman Material on the Development of the Bombes, including Appendix A on the Wiring of the Bombes — Reconcile our various Jeffreys' sheets implementations — I found and fixed a mistake(!) in my implementation — Corrected Wheel Order 1 3 2 Sheet A for Type(2, 5) Females (reload for corrected version) — A photo of a bombe in operation — A handout for practicing and appreciating bombe wiring

Week 9 — Midterm Exam — Start Snowden — Start DES

- Monday, Nov. 1 — Midterm — Read the Preface and Chapter 1 of Snowden — Optional and on your own, Read Singh's account of cracking the Enigma (Singh Chapter 4) which is quite simplified, but which has some material that is complementary to Welchman (especially Rejewski's early work on cracking Enigma before Germany invaded Poland — The Polish work is well-summarized in a Sky History article) — Assignment for Thursday: Generating DES Sub-Keys
- Thursday, Nov. 4 — The first part of Singh Chapter 6, pp. 243-252 up to DES — Casually read the DES standard, but carefully study Figure 3 which you will be using in the assignment — Chapters 2, 3, and 4 of Snowden — Assignment for Monday: complete DES Sub-Key generation — Here is an improved version of my Notebook 6.1 for you to leverage: Notebook 6.2 Improved Generating DES Sub-Keys

Week 10 — Finish DES — Modular Arithmetic — Diffie-Hellman-Merkle Key Exchange

- Monday, Nov. 8 — Continue Singh, Chapter 6, pp. 252-267, Modular Arithmetic and Diffie-Hellman-Merkle Key Exchange — Chapters 5, 6, 7 of Snowden — Complete DES sub-key generation notebook Solution — It would also be very educational for you to look at my Midterm Exam Solution and modify it to make it tail-recursive
- Thursday, Nov. 11 — Understand the Proof and Complete the Investigation in the Modular

Arithmetic and Key Generation notebook — Continue Singh, Chapter 6, pp. 268-279, Factorization into Primes and RSA Cryptography — Chapters 8, 9, and 10 of Snowden

Week 11 — RSA Public-Key Cryptography — More Developments in Public-Key Cryptography: GCHQ and PGP

- Monday, Nov. 15 — My Key Generation investigation — Read Appendix J of Singh — Finish Singh, Chapter 6, pp. 279-292, Secret Developments in Britain's GCHQ — Chapters 11, 12, 13, and 14 of Snowden — Start Hellman, The Mathematics of Public-Key Cryptography — Implement the algorithms in the Prime Number Generation notebook
- Thursday, Nov. 18 — All of Singh, Chapter 7, pp. 293-313, Phil Zimmermann and PGP ("weapons-grade" cryptography for everbody) — Chapters 15, 16, and 17 of Snowden — Continue reading Hellman — My notebook investigation for prime number generation — A Tail-Recursive Version of GCD (or how I learned that Python does not have tail-call optimization!) — Great summary of primality tests that Hellman only alludes to.

Week 12 — Continue Public-Key Cryptography — Start Blockchain

- Monday, Nov. 22 — Do examples of the computations in Hellman, for example the sample knapsack problem I put on the board — Read Chapters 18, 19, and 20 of Snowden — Read Sections 1-3 (just two pages!) of the original Bitcoin white paper — Return to Hellman, and understand how public-key cryptography can be used to create cryptographic signatures — Read up on md5 as an example of a hash function (invented by Ronald Rivest, and still used a hash function, but no longer as a cryptographic hash, because it has been defeated) and also find out what it means for a hash algorithm to have been defeated by reading Some Wikipedia Excerpts on MD5 and Cryptographic Hashes

Week 13 — Finish Public Key Cryptography — Continue Blockchain — The `conda` Package Manager

- Monday, Nov. 29 — Last detail to understand from Hellman: Why is ϕ(n) the relevant modulus in the exponent? That is the Fermat-Euler Theorem (aka "Euler's Totient Theorem") and is apparently pretty hard! How about we all just understand the proof of the much simpler case known as Fermat's Little Theorem — Read Sections 4-7 (just two more pages) of the Bitcoin white paper — Start thinking about how you would implement the timestamp server described in sections 3 and 4 — Read Chapters 21, 22, and 23 of Snowden
- Thursday, Dec. 2 — Read Sections 8-10 of the Bitcoin white paper — Read Chapters 24,

25, and 26 of Snowden — Assignment for Monday, Dec. 6: complete the Bitcoin toy in this notebook — To even start the assignment, you will need to use conda to install pycryptodome — Perhaps knowing how I did that on my system will help

Week 14 — Finish Snowden — Finish Blockchain — Exceptions in Python — The Cracking of Linear B

- Monday, Dec. 6 — Read Sections 8, 9, and 10 of the Bitcoin white paper — Finish Snowden (Chapters 27, 28, and 29) — To complete the assignment made on Dec. 2, you will need to learn about "handling exceptions" in Python
- Thursday, Dec. 9 — Finish Bitcoin White Paper (Sections 11 and 12) — My Solution to the Bitcoin Assignment — Python Calculation of Bitcoin Attacker Success Probability — Chapter 5 of Singh, pp, 191-217 — Champollion Cracks Egyptian Hieroglyphics and Kober, Chadwick, and Ventris Crack Linear B — Read Chapter 2 of John Chadwick, The Mycenaeans, 1976

Week 15 — Quantum Cryptanalysis and Quantum Encryption

- Monday, Dec. 13 — Quantum Code-Breaking — Start Singh Chapter 8, pp. 317-331 — Supplementary Articles on Current Status of Quantum Computing from Physics Today and Scientific American — IBM's Introduction to Shor's Algorithm — Optionally watch John Preskill Y Combinator Interview — Big-O Notation — Even- and Odd-Parity Error Correction
- Thursday, Dec. 16 — Quantum Cryptography — Finish Singh Chapter 8, pp. 331-350 — Quantum Mechanics Readings from Richard Feynman and Albert Stetz — Quantum Interference — The Schrodinger's Cat Paradox