INF 385T: Applied Encryption (27615)

Fall 2019
UTA 1.504
Thursday 9:00 am – 12 pm

Instructor: Walker Riley
walker.riley@utexas.edu

Office Hours: 12pm-3pm Thursday in the Makerspace

I. Course Description

Encryption is at the heart of almost everything we do online. It keeps our traffic safe and makes e-commerce possible by allowing us to trust that our financial information cannot be intercepted. Encryption is also key to confidentiality and privacy online, which is a fundamental value of the ALA.
This course aims to give students a thorough understanding of how encryption works by examining and implementing the most foundational and widely used forms of encryption. This knowledge will arm students with the technical security literacy that is vital to understanding and combating threats to confidentiality and privacy.

II. Pre-Requisites

None. Some experience with programming will be helpful, but it is by no means required. Every skill you need to do well in this course will be built, from the ground up, in this course.

III. Course Aims and Objectives

After taking this course, students will be able to:

- Discuss cyber-security with a grounding in both theory and application
- Write efficient and effective code using the scripting language Python
- Read and write in Binary
- Implement their own custom unbreakable and nearly-unbreakable crypto-systems
- Understand and utilize number theory without a background in mathematics

IV. Disclaimer

This syllabus is subject to change at any time.

V. Tentative Course Schedule

Week	***Discussion Topic***	***In Class Exercise***	***Assignment Due***
Jan 23	Syllabus Overview	Get Familiar with Python	
Jan 30	History of encryption	More Python	
Feb 6	Start on Caesar Cipher	One-time pad	Python script: Basics
Feb 13	Binary	ASCII	Python script: caesar cipher decryption
Feb 20	Modular Math	Fermat and Fast Power	Python script: one time pad
Feb 27	Backdoors		Python script: Binary Juggling
Mar 5	Diffie Hellman Key Exchange	ElGamal cryptosystem	Python script: modulo calculator
Mar 12	Security Overview		Python script: Diffie Hellman and Elgamal

Mar 19	Spring Break!		
Mar 26	Units and Roots; PQ	RSA	Paper proposal
Apr 2	Primality Testing	Miller Rabin	Python script: RSA
Apr 9	Elliptic Curves	Calculations in EC space	-Python script: Miller Rabin -Paper rough draft
Apr 16	Application of EC math	Send a message	Python script: EC Calculator
Apr 23	Quantum Computers	Shor's Algorithm	Python script: ECC
Apr 30	Ethical Hacking	Wargames	
May 7	Final Projects	Presentations	Final Paper

VI. Course Requirements

- **Python Scripts**

  Python assignments will make up the majority of your homework. You are welcome to work with other students, but do let me know who you are working with. All code should be documented. This documentation should be done in commented out lines in your code and should explain what each piece of code is doing.

  While working together on code is fine, all documentation is expected to be unique. I need to see that you are able to explain your code in your own words.

  - **Scripting platforms**

    You are welcome to write your code using any interface you choose, however, I cannot recommend enough that you run your code through command line or through idle before submitting as many "Smart" interfaces will fix errors in your code without telling you. This is a problem because when I am testing your code in command line, those fixes don't get done and your code might not run.

  - **Submission details**

    When submitting please upload all relevant documents separately. It may seem more efficient to zip all of your files, but I strongly prefer that you do not.

    For python scripts specifically, I will only accept ".py" files. So please do not submit URLs for your cloud-based script editor, and please do not submit any proprietary variant of a python file.

- **Final Paper**

  Your final paper can come in many forms. I encourage students to find (or create) a cryptosystem not covered in class. Your paper should explain, in an accessible way, what your cryptosystem does. I encourage the use of Alice and Bob illustrations.

  If you are feeling particularly ambitious, I will accept code in place of a paper. If you can create a cryptosystem, encryption method, or other cryptographic algorithm in Python with sufficiently detailed documentation, then I will know that you understand the material.

- **Participation**

  All students are expected to attend every class. Attendance will be taken. What you get out of this course will be significantly impacted by your participation, and as such, I expect everyone to not just show up, but to be engaged and contribute.

  Having that been said, I understand that life gets in the way some times. If you know you will be missing class, try to let me know as soon as possible. If you miss class and I never hear from you about why, your participation grade will be lowered accordingly.

  This policy has been abused in previous semesters, so I have put a cap on number of absences I will excuse. As long as you do not try to abuse my leniency with attendance this will not affect you.

  Note: The participation grade is penalty based. You start the semester with full credit, and lose points as you skip class or don't engage in class. Even though participation is 10% of the final grade, if you skip enough classes, your participation grade can go negative and thereby reduce your final grade by more than 10 points.

VII. Lab Hours (Thursday 12pm - 3pm, Makerspace)

Every Thursday after class I will be in the Makerspace. You can find me there if you need to talk about anything, or if you would like to work on python. I encourage all students to come in even if they do not need assistance. The Makerspace is an excellent collaborative learning environment to work on homework, and it allows you to reach out to me in the off-chance you get stuck on a problem.

VIII. Grading Procedure

Grades will be determined as follows:

Attendance and Participation: (10%)
Python Scripts: (70%)
Final Paper: (20%)