CATIE ACITELLI

⋄ Research Statement ⋄

The field of Cryptography is rapidly evolving, and there exists a promise for quantum computers in the near future. It is the responsibility of institutions to prepare students for related careers. I assert that undergraduate students in mathematics, computer science, and other related fields can – and should – study lattice-based cryptography. I also assert that no prior computer programming or abstract algebra knowledge is necessary to do so. My research focuses on the undergraduate accessibility of Post-Quantum Mathematical Cryptography with Python. My dissertation

1. developed a self-contained undergraduate mathematical Cryptography course that depends only on a first semester Linear Algebra course, and
2. created a written curriculum for undergraduates in post-quantum mathematical Cryptography with Python.

1 Background

Public Key Cryptography is an asymmetric scheme that uses two different but mathematically related keys - the public key and the private key - to exchange information. A plaintext message is encrypted using the *one-way function*, or the *public key*. The function can only be inverted to recover the plaintext message from the ciphertext in a reasonable amount of time if the message recipient has some *trapdoor information*, or the *private key*. The more difficult the inverse of the encryption function is to find, the more difficult it is to carry out an adversarial attack. While true one-way functions are not known to exist, the security of Public Key Cryptosystems relies on the assumed hardness of their underlying problems. Hardness refers to, roughly, the running time of the algorithm. Breaking a cryptosystem amounts to inverting the encryption function or solving the hard problem in a reasonable amount of time.

For instance, the RSA Cryptosystem of Rivest, Shamir, and Adelmam - the most widely used encryption scheme - relies on the hardness of integer factorization. There is currently not a way for a classical computer to find the prime factors of a composite number in less than exponential time. However, in 1994, Peter Shor developed an algorithm that solves the integer factorization problem in polynomial time on a quantum computer, effectively breaking RSA.

Many large companies have teams developing quantum computers. In September 2020, IBM announced that they will have a 1000 quantum bit (qubit) computer by the year 2023. This gives rise to a need for quantum-resistant cryptosystems, and constructions based on lattices are proving promising. My dissertation research has prepared me to leverage students' prior linear algebra knowledge throughout the course to introduce lattices, their properties, and their known hard problems before diving deep into lattice-based cryptographic schemes and lattice reduction algorithms.

2 The Course and The Written Curriculum

Table 1 illustrates the standard $15-$week pacing guide for this course. I wove the use of Python, the most widely used computer language, throughout the course to complement the mathematical theory. Introducing students to programming with Python in the first week allows them to acquaint with a new language while reviewing familiar content. Additionally, this allows them to develop introductory code upon which they can build later in the course. Students continue this trend throughout the semester. Structuring the course in this way allowed for the design of projects for which students will work collaboratively. One of the projects that I have designed guides student teams through using their Gram-Schmidt Orthogonalization Python code to build the LLL Lattice Reduction Algorithm of Lenstra, Lenstra, and Lovász.

Content is introduced as it becomes relevant. For instance, aspects of Abstract Algebra are presented in the beginning of the course; however, polynomial rings are saved for the week prior to the Cryptosystem that uses them - the NTRUEncrypt of Hoffstein, Pipher, and Silverman.

Week	**Topic**
1	Linear Algebra Review & Introduction to Python
2	Introduction to Number Theory and Abstract Algebra
$3, 4, 5$	Pre-Quantum Cryptosystems and Their Hard Problems
6	Introduction to Lattices: Definitions and Properties, Hard Lattice Problems
$7, 8$	Introduction to Public Key Cryptosystems Based on Hard Lattice Problems
9	Polynomial Rings
$10, 11$	NTRU Cryptosystem
$12, 13$	Lattice Reduction Algorithms
$14, 15$	Applications of the LLL Lattice Reduction Algorithm

Table 1: Syllabus for a Standard 15-Week Semester

2.1 Course Outcomes

By expanding upon decades of cryptography research and taking a novel application-based approach, completion of this course accomplishes five main outcomes for students:

1. It enhances their algebraic thinking. Students gain exposure to important topics in abstract algebra, which they may otherwise never encounter. They also explore practical applications of both linear and abstract algebra content.

2. It allows them to learn the Python computing language. Students benefit from the practical experience of becoming proficient in the most widely-used computer programming language.

3. It gives them the opportunity to explore multiple facets of algorithm design. Students are able to see the design and implementation of algorithms from both the mathematical lens and the programming lens.

4. It permits them to see the fluidity of mathematics. Students use a mixture of pure and applied mathematics and therefore understand that the two are not separate fields of study. Furthermore, they become more mindful of the interconnectedness of mathematics and other disciplines.

5. It prepares them well for a variety of different career paths. Students who are well-acquainted with both the theory and the application of post-quantum cryptography are competitive in the pool of applicants for any related next steps.

2.2 Written Curriculum

In building this course, I realized a gap in current literature. Cryptography textbooks are extremely advanced, strictly from the computer science perspective, or lacking implementable computer code. As a result, my dissertation research has led to the development of a *Post-Quantum Mathematical Cryptography with Python* textbook.

The written curriculum mirrors the course syllabus in Table 1, with appendices on Factoring Polynomials and Getting Started with Python. In each of the completed text sections, I have included student learning objectives, an introduction to each Cryptosystem, the mathematical theory behind the systems, worked examples, algorithms with pseudocode, implementable Python code, exercises, collaborative exercises, and computer exercises.

3 Future Work

The textbook is in its early stages and is approximately halfway complete. I aim to publish this text as a resource for students and instructors everywhere, so that more undergraduate students have access to this real-world content.

Each time I teach the accompanying course, revisions to the text will be made as a result of student learning and student feedback. Additional revisions will occur as advancements in lattice-based Cryptography do, via mathematics or other related fields.

I also plan to engage both undergraduate and graduate students in projects and research opportunities. For example, undergraduate students can extend encryption algorithms to signature schemes by way of examples. In another project, undergraduate or masters students can use what they know about the RSA Cryptosystem to develop an Additive RSA Scheme. Graduate students who are interested in Graph Theory can explore those connections to Cryptography. Both undergraduate and graduate students can make modifications to known cryptographic schemes and develop all of the standard protocols of the systems (i.e. encryption algorithm, decryption algorithm, adversarial attack, signature scheme, etc.).