Brandon LaPointe

CSC332 Programming Component Proposal

With the coding component of my MAT/CSC332 project, I hope to be able to first provide a simple working example of a standard XOR cipher making use of the SecureRandom library, a cryptographically secure pseudorandom number generator provided by Java, as the base for a post quantum safe encryption and decryption functions. Second, I hope to provide a modified version of the XOR cipher code to show a simplified example of the McEliece cryptosystem. The program will encrypt a string message and then decrypt the encrypted message and display the results on a scaled down level to be time and memory-friendly on a modern-day computer. The program will be written in Java and it will make use of the BouncyCastle library, “a Java library that complements the default Java Cryptographic Extension (JCE)” (Baeldung.com), which contains methods used to implement the McEliece cryptosystem as well as methods for many other post quantum cryptography systems such as Frodo, Rainbow, and Khyber.

I believe that this is important to the future of cryptography as the McEliece cryptosystem was the first code-based public-key asymmetric cryptosystem developed in 1978 by Robert J. McEliece and over it’s 40+ years of existence, the cryptosystem has persistently stood up to dozens of deliberate attacks intended to test the strength of the system in hopes of breaking it, proving its ability to be secure even with the drastic advances in computers we have seen come into existence since its release. Originally, the McEliece cryptosystem was only designed for 264 security parameters but has the ability to be upscaled to “overkill” parameters, making the McEliece cryptosystem able to provide security for computers even when quantum computing becomes more regularly achievable. (classic.mceliece.org)

Sources:

Brief history on Classic McEliece

<https://classic.mceliece.org/>

Introduction to Bouncy Castle With Java

<https://www.baeldung.com/java-bouncy-castle#:~:text=BouncyCastle%20is%20a%20Java%20library,such%20as%20encryption%20and%20signature>.

Package org.bouncycastle.pqc.crypto.mceliece JavaDoc

<https://javadoc.io/static/org.bouncycastle/bcprov-jdk15on/1.54/org/bouncycastle/pqc/crypto/mceliece/package-summary.html>

Goppa Codes and Their Use in the McEliece Cryptosystems (I am still researching Goppa codes)  
<https://surface.syr.edu/cgi/viewcontent.cgi?article=1846&context=honors_capstone#:~:text=The%20McEliece%20cryptosystem%20is%20a,and%20send%20it%20to%20Bob>.