Popular Science Summary

There is a current challenge to develop new cryptographic standards that are secure against quantum algorithms. With the knowledge that a sufficiently powerful quantum computer can break current asymmetric cryptographic schemes, it is important to redesign such protocols. This thesis explores the practical implementation of a threshold-signature scheme that is secure even with access to a quantum computer.

Digital signatures are schemes utilizing asymmetric cryptography, allowing an entity to create a public/private key pair. The public key can then be distributed freely, while the private key can be used to sign messages, documents, software and more. Anyone with access to the public key can verify that the signature is legitimate, i.e. the signature was created by the holder of the private key. This scheme can be extended to a threshold scheme, where multiple participants are required to create a valid signature. Threshold-signature schemes are typically called t-out-of-n signature schemes, where t denotes the number of participants required to create a valid signature, and n denotes the total number of participants.

This is achieved in our scheme with two underlying homomorphic schemes for commitments and encryption. The homomorphic property allows for mathematical operations such as addition of encrypted data, e.g. ciphertexts, where adding multiple ciphertexts is equivalent to the sum of the encrypted corresponding plaintext.

Lattices are a popular candidate going forward for constructing quantum-safe algorithms. Additionally, homomorphic schemes were first realized in lattice-based cryptography. A lattice is a set of points in a multi-dimensional space, created by taking the linear combinations of a set of basis vectors.

There exists various theoretical implementations of lattice-based t-out-of-n schemes, but publicly available practical implementations are lacking. Our implementation serves as a first prototype to demonstrate the feasibility of implementing these schemes. We evaluate execution time, sizes of messages sent, and number of mathematical operations in the protocol for different values of (t, n). We found that our system scaled exponentially, with Shamir's Secret Sharing method requiring the most time out of all algorithms. We also discovered that key generation, which was the most expensive algorithm, was the most expensive for each participant when t was equal to $\frac{n+1}{2}$.