Isogeny-based cryptography

Table of content

1 Introduction

Post-quantum cryptography (PQC) is a young field anticipating the availability of large-scale quantum computers in the context of digital security. Seen by many as the next logical step to supersede classical computers, the quantum paradigm offers algorithms that could endanger classical public key cryptography. Possibly the most famous one, Shor's algorithm [] brings integer factorization down to a polynomial problem. This has obvious dire consequences for classical RSA and Diffie-Hellman (DH) protocols. The performances of current quantum computers are still far from reaching usability in attack scenarios. But cautiousness calls for quantum-resistant algorithms that could secure classical computers. In that regard, the NIST (National Institute of Standards of Technology) has organized competitions to foster developpment in PQC.

Different usages require different cryptosystems and as such there are multiple classes of post-quantum protocols investigated. Of interest in this paper is public-key cryptography (PKC) also known as asymmetric cryptography. The benefit of PKC stems from the fact that no pre-sharing of information is needed to establish a secure channel, hence making it very convinient. To achieve this, a Diffie-Hellman key-exchange protocol is usually used. In our case we use elliptic curve isogenies and associated hard homogeneous spaces (HHS) to build such a protocol.

More precisely, we investigate the Couveignes-Rostovtsev-Stolbunov protocol (CRS). This protocol works with isogenies bewteen ordinary curves with fixed complex multiplication over a finite field of prime order $\sim 2^{256}$. Key-exchanges using this scheme are comparatively slower than their supersingular counterpart such as SIDH protocols. However, CRS heuristically gives less room for potential attacks than SIDH. This is due to the fact that supersingular endomorphism rings embed as orders in quaternion algebras whereas ordinary ones embed in imaginary quadratic fields. As such, more structure is available for exploitation in an attack of SIDH protocols. This kind of structural attack has been demonstrated against the NTRU protocol and led to Prime-NTRU. Another advantage of CRS is that it is truely symmetric compared to SIDH where parties have to agree on specific primes to use.

The main focus of this project was to implement as efficiently as possible the CRS protocol following the 2018 article of Luca De Feo, Jean Kieffer and Benjamin Smith []. Recent publications on the subject of isogeny computation allow for very efficient walking algorithm in the isogeny graphs. On the one hand, *radical isogenies* [] are used to take k steps in the l-isogeny graph in time complexity $\mathcal{O}(k)$ for primes 3, 5 and 7. On the other hand, the $\sqrt{}$-*Velu* algorithm [] is used to take walk in the remaining graphs with time complexity $\mathcal{O}(k\sqrt{l})$. The reason this subject was choosen specifically is after a previous unoptimized JULIA implementation of these techniques. It gave promising results (around 35% faster than the 2018 implementation) but was neither optimized nor using the expected quadratic-twist symmetry. The second issue was completely resolved by our C implementation while drastic optimizations were found.

This paper is divided in three sections. The first one mainly accounts for the minimal mathematical background surrounding isogeny-graphs and complex multiplication. An exposition on the CRS protocol is also given. The second part tackles the architecture of the program. The complete project is around 4000 lines long. Its source code is available at []

along with a documentation. Finally, the thirs part reports prime-by-prime timing results, optimized bound for the secret keys as well as some inner data observed at runtime.