Selected Areas in Cryptology - Part 1: Post-quantum cryptography

Exercise sheet 6, 20 April 2024

1. Slides have been added to the handout that show some isogeny computations in SageMath. Use these scripts as examples (all functions you might need for this exercise are used at least once in the script) to write the functions that perform a positive ℓ-step and a negative ℓ-step in the CSIDH setting. Specifically, these functions should take as input (1) a supersingular elliptic curve defined over $\mathbb{F}_p$ and (2) an integer ℓ and then, output the image curve under an isogeny of degree ℓ. If you are doing this exercise algorithmically, you can assume the functions used in the scripts on the slides exist as a blackbox. **Hint:** For computing a negative step, use the quadratic twist. You can use the example figure on slide 47 to check that your functions compute correctly.

2. Take a look at the *fundamental theorem of cyclic groups* stated on slide 41 and think about how you can use the very-last statement (*namely,...*) to improve (computationally) the scripts that you have written in Exercise 1. To find a generator of the group $E(\mathbb{F}_p)$, in SageMath use

   ```
   P=E.gens()[0]
   ```

   Hint: As part of the solution of the previous exercise, you probably have a loop that picks random points on the curve until it finds a point of a specific order. Replace this loop.

3. Let $\mathbb{F}_p$ be the prime field with $p = 2^e 3^f - 1$ (for some integers e and f) and let E_0, E_1 be supersingular elliptic curves defined over $\mathbb{F}_{p^2}$. We are considering the problem of finding a 2^e-isogeny from E_0 to E_1. The Meet-In-The-Middle (MitM) algorithm for tackling this problem can be summarized as follows. We first compute and store all $2^{e/2}$-isogenous curves to E_0 in a table. Then we proceed by computing each $2^{e/2}$-isogenous curve to E_1 and check if it is present in the table. Any matching pair then allows to recover the secret isogeny.

   - What is the time and memory complexity of this attack?

- The MITM attack has huge memory requirements. Propose a variation of this attack that can be used when the available memory of an attacker is limited to storing W curves.
- What is the time complexity of your proposed attack?

4. **Bonus:** Parallel Collision Search

   (a) Describe in detail the parallel collision search algorithm by van Oorshoot and Wiener.

   (b) Suppose that the distinguished points are stored in a hash table. What are the exact parameters (elements) that need to be stored for one distinguished point and what is the reference parameter for deriving the hash table key?

   (c) Calculate the expected memory requirements for running this attack on a 115-bit curve (an elliptic curve over defined $\mathbb{F}_p$ where p is a 115-bit prime - this just means that all elements that you store take up 115 bits, or 15 bytes). **Hint:** Use *Hasse's theorem* for the approximate number of points of an elliptic curve over $\mathbb{F}_p$. A good choice of a distinguishing property for this attack would be to distinguish points whose x-coordinate has 28 trailing zero bits.

5. Describe a basic Sigma protocol (recall: this is the construction we learned in Lecture 3. See the *triangle* figure) based on the isogeny path finding problem. This would be, for instance, a protocol where the Prover needs to prove that they know a secret 2^e-isogeny (or an isogeny of any fixed smooth degree) from curve E_0 to curve E_1. Discuss whether this protocol satisfies the

   - Completeness
   - Special soundness
   - Honest Verifier ZK

   properties, or not.