Exercise Sheet Week 10

—Side Channel Analysis on Elliptic Curve Cryptography—

Submission deadline: November 28, 2024, 11:30 via MyCourses or Aplus (for coding tasks)

Each exercise gives up to 2 points. The exercise sheet gives at most 4 points. **This exercise sheet is still work in progress.**

For this exercise you need **Python3**

For your solutions you should provide your code and additional documentation elaborating on your solutions. Remember to add comments to your code

Files. In the folder ECC_ex_assignment you will find a simple ECC implementation based on the standardised **curve P-256**. The main file performs a key generation, an encryption and decryption (all based on the El Gamal scheme). The program checks for correctness by verifying that the input points are valid and checking whether the original plaintext (i.e. original point that we want to encrypt) matches with the recovered plaintext after decryption.

Exercise 1 (DPA countermeasure 1: Key Randomisation)**.** *2 points, via Aplus*
Task. Implement the key randomisation countermeasure as discussed during the lecture.

Hint: When implementing key randomization, you need to ensure that your randomised key is functionally equivalent to the original key. You can also obtain inspiration from Exercise 2.

Exercise 2 (DPA countermeasure 1: Key Randomisation)**.** *2 points, via MyCourses* Which algorithms out of `kgen`, `enc` and `dec` of the El Gamal public-key encryption scheme[1] need to implement key randomization in the key randomization countermeasure? Justify your answer. Explain how to modify them and why to modify them in this way. How would one benchmark *in a machine-independent-way* how an implementation of the key randomization countermeasure affects the algorithm's performance? Discuss what would happen if the parameter for implementing this countermeasure is too small or too large.

Exercise 3 (DPA countermeasure 2: point blinding)**.** *2 points*
Task. Implement the countermeasure for point blinding as discussed in the lecture. Explain your implementation choices. How does the efficiency of the point blinding countermeasure compare with the efficiency of key randomisation, i.e., which one implies a bigger performance penalty?

Optional task (advanced) Think about possible side-channel attacks that may be successful in the presence of one countermeasure, but not in the presence of the other. For this you can come up with your own ideas or do a bit of research on scientific works.

[1] You can either read the code on Aplus and/or the Wikipedia page on El Gamal`https://en.wikipedia.org/wiki/ElGamal_encryption`