Exercise Sheet Week 9

—Side Channel Analysis on Elliptic Curve Cryptography—

Submission deadline: 23:59, via eMail

Each exercise can give up to two participation points, 2 for a mostly correct solution and 1 point for a good attempt. Overall, the exercise sheet gives at most 4 participation points. We encourage to **choose** exercises which seem **interesting** and/or adequately challenging to you.

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*
Task. Implement the key randomisation countermeasure as discussed during the lecture. Which algorithms out of `kgen`, `enc` and `dec` of the El Gamal public-key encryption scheme need to implement this countermeasure? Explain why you modified the algorithms the way you modified them. How can you benchmark *in a machine-independent-way* how your implementation of the countermeasure affects the algorithm's performance? Discuss what would happen if the parameter for implementing this countermeasure is too small or too large.

Hint: When implementing key randomization, you need to ensure that your randomised key is functionally equivalent to the original key.

Chris: I think that the exercise can also be done on pseudo-code for those who don't feel comfy with Python. Maybe, we should add a pseudo-code variant of the exercise here.

Exercise 2 (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.