Institute of Computer Science, University of Bern
Cryptographic Protocols, FS 2021

Prof. Christian Cachin
Jovana Mićić

Exercise 4

4.1 Textbook ElGamal encryption in Python (4pt)

Consider the sample Python code that implements the Diffie-Hellman key exchange, which is available in the file `diffie_hellman.py`.

Use this code as the basis to implement the textbook ElGamal encryption scheme for messages in $\mathcal{Z}_p^*$. You may use Python or any other language of your choice.

4.2 Additively homomorphic ElGamal encryption (6pt)

In class we have shown how to extend the ElGamal cryptosystem so that it becomes additively homomorphic for ciphertexts that are small numbers.

a) Using the code of the first problem, implement additively homomorphic ElGamal public-key encryption. Test it with $|q| = 160$ and $|p| = 1024$ for numbers up to 10^6 (or 2^{20}).

b) Measure and plot the decryption time for encrypting numbers in $[1, max]$, where *max* ranges from ca. 256 to 10^6 (or 2^{20}). Use $|q| = 160$ and $|p| = 1024$.

Try also with $|q| = 256$ and $|p| = 2048$, but stop with a smaller value of *max* if it would take too long on your platform. (On a Linux laptop that reports 4200 bogomips, decryption takes up to about 100 s for a domain of size 10^7.)

c) *(Bonus question: +2pt)* Describe different approaches to reduce the time taken for decryption.