(Applied) Cryptography

Tutorial #7

Manuel Barbosa (mbb@fc.up.pt) Rogério Reis (rvr@dcc.fc.up.pt)

MSI/MCC/MERSI – 2021/2022

1 - Use openSSL to generate Diffie-Hellman parameters at 128-bit security (4096-bit modulus) using option `dhparam`. Do not activate option `-dsaparam`.

2 - Repeat the exercise activating option `-dsaparam`.

3 - Why does the first approach take so much longer? Use Sage to check that the produced primes have the structure you describe in your answer.

4 - Use Sage to check that DH works for both parameter sets:

- generate exponents x, y in the range $[0 \dots q[$ where q is the order of the group generator
- compute $X = g^x \pmod p$ and $Y = g^y \pmod p$
- check that $X^y \pmod p = Y^x \pmod p$

5 - Alice and Bob agree to use the prime $p = 1373$ and the base $g = 2$ for a Diffie–Hellman key exchange. Alice sends Bob the value $X = 974$. Bob asks your assistance, so you tell him to use the secret exponent $y = 871$. What value Y should Bob send to Alice, and what is their secret shared value? Can you figure out Alice's secret exponent?

6 - Prove that an algorithm that solves the Computational Diffie–Hellman problem can be used to solve the Decisional Diffie–Hellman problem.