Exercise 1

See Fig. 1.

Exercise 2

See Figures 2 and 3.

Exercise 3

Without the `dsaparam` option, the generation of DH parameters takes much longer because it generates a *strong prime*, *i.e.* a prime number p such that $(p-1)/2$ is also prime. This ensures that the multiplicative group $\mathbb{Z}_p^*$ does not contain small subgroups. The existence of small subgroups within a larger group in a cryptographic protocol is undesirable in the sense that it confines shared secrets to a smaller set of possible values than if it were to use the whole group $\mathbb{Z}_p^*$.

On the other hand, using the `dsaparam` DSA rather than DH parameters are read or generate, which are then they are converted to DH format, making the key exchange process more efficient.

Note: See Figs. 4 and 5.

Exercise 4

Running the following function with the previously generated DH parameters, we can see that $X^y \pmod p = Y^x \pmod p$ holds.

```
from sage.all import *

def ex4(p, q):
    x = randrange(q)
    y = randrange(q)

    X = Mod(q ** x, p)
    Y = Mod(q ** y, p)

    return Mod(X ** y, p) == Mod(Y ** x, p)
```

Note: See Figs. 6 and 7.

Exercise 5

We know that $p = 1373, g = 2, X = 974$ and $y = 871$. We can determine Y by computing the following:

$$Y = g^y \bmod p = 2^{871} \bmod 1373 = 805$$

Knowing the value of X and y, we can then determine the shared secret g^{xy} by computing the following:

$$g^{xy} = (g^x)^y = X^y = 974^{871} \bmod 1373 = 397$$

Finally, we can find Alice's secret exponent by solving:

$$X = g^x \bmod p \Leftrightarrow 974 = 2^x \bmod 1373 \Leftrightarrow x = 587$$

Exercise 6

The Computational Diffie-Hellman Problem (CHD) consists in computing the shared secret g^{ab} given only the public values g^a and g^b, and not any of the secret values a or b. The motivation is to ensure that even if an eavesdropper captures g^a and g^b, they will not be able to determine the shared secret g^{ab}.

The Decisional Diffie-Hellman Problem (DDH) is stronger than the CDH. To ensure that an attacker can't learn anything about the shared secret g^{ab}, this value needs only to be indistinguishable from a random group element. This being said, given g^a, g^b and a value that is either g^{ab} or g^c for a random value c, each with probability $1/2$, the DDH problem consists of determining whether g^{ab} was chosen.

An algorithm that solves the Computational Diffie–Hellman problem can be used to solve the Decisional Diffie–Hellman problem. Given g^a, g^b and g^c, and assuming we can solve CDH, we can derive g^{ab} from g^a and g^b. Then, to solve DDH, we only have to check if the result equals g^c.

Appendices

Exercise 1

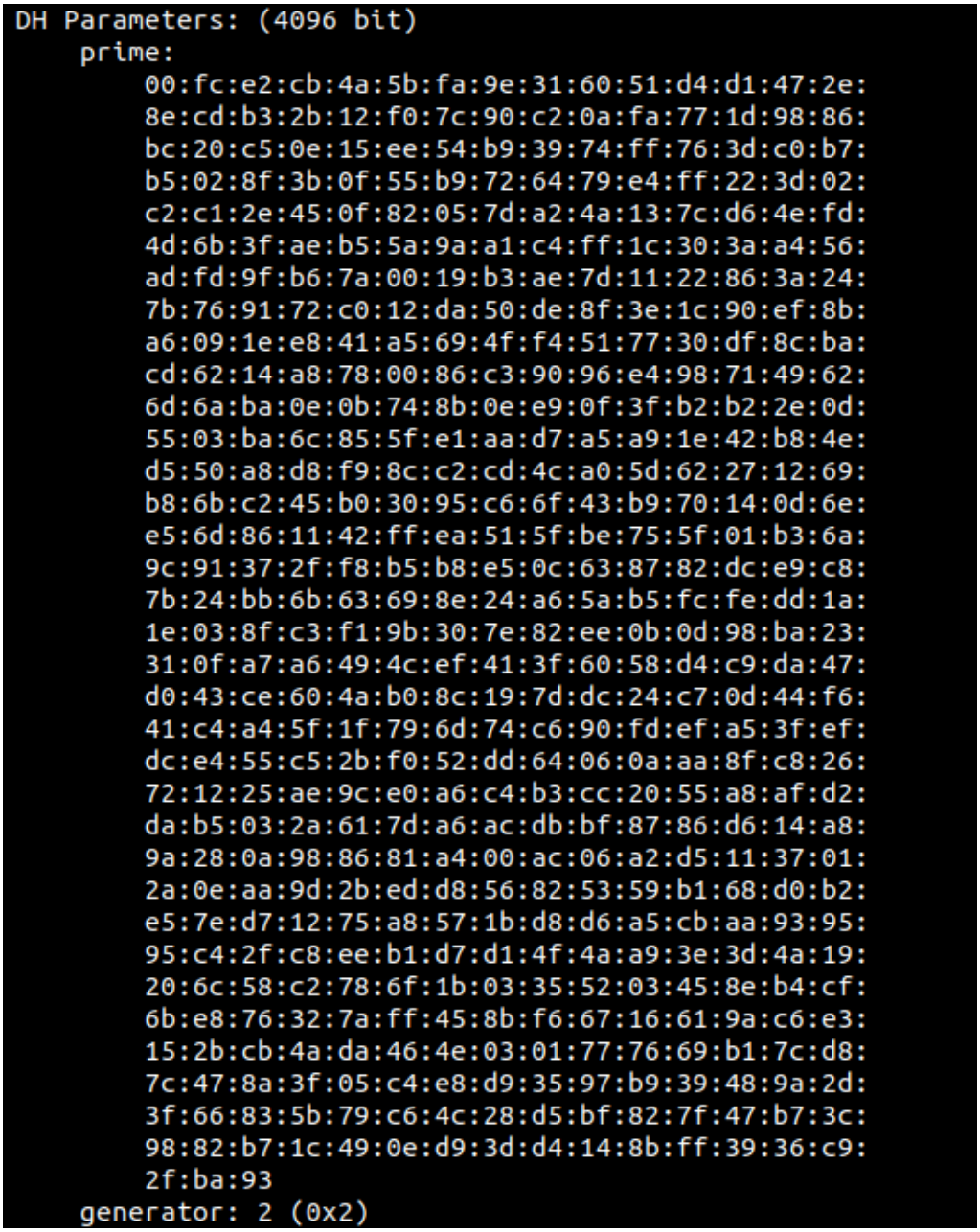

Figure 1: Generation of DH parameters without the `dsaparam` option

Exercise 2

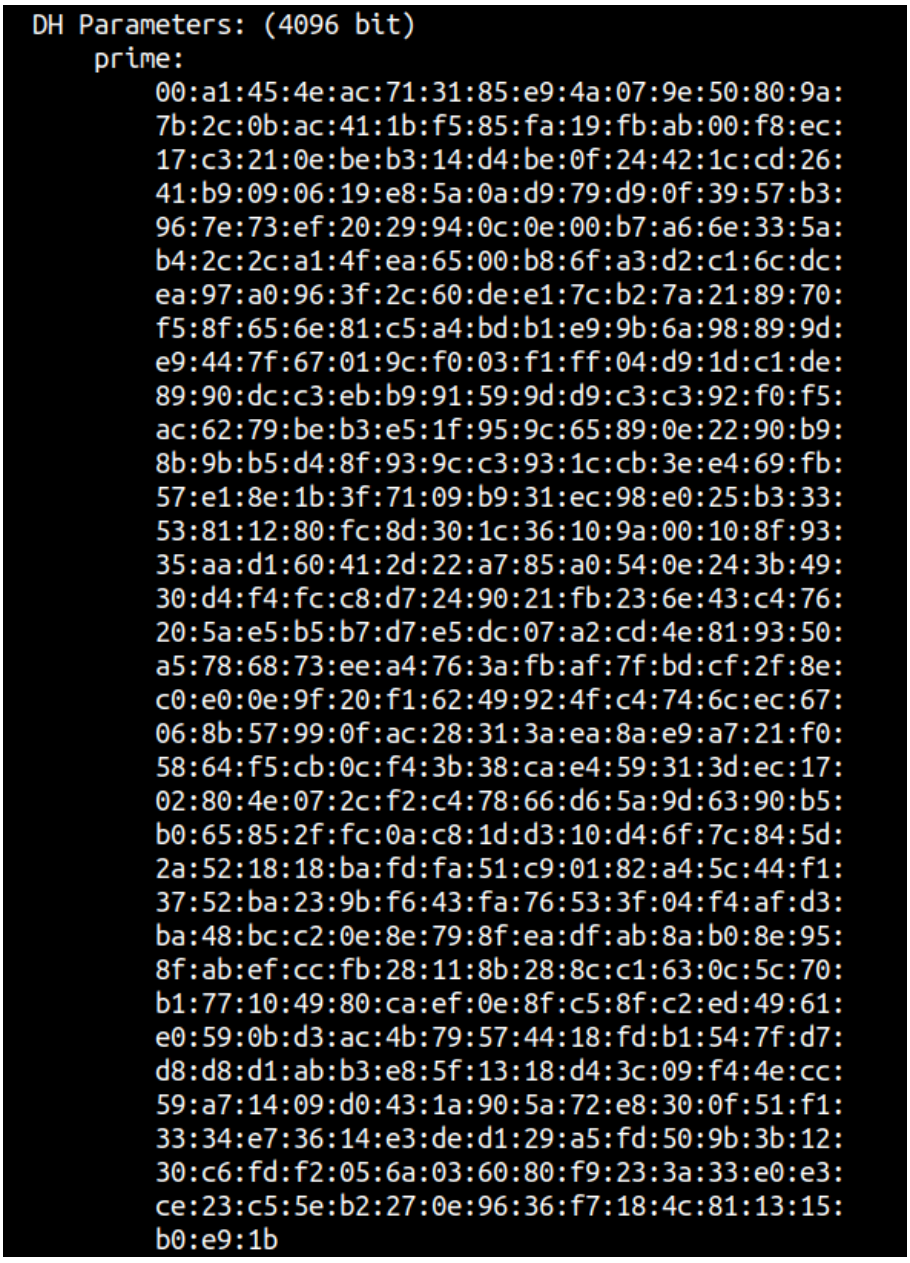

Figure 2: Prime number generated using DH parameter generation with the `dsaparam` option

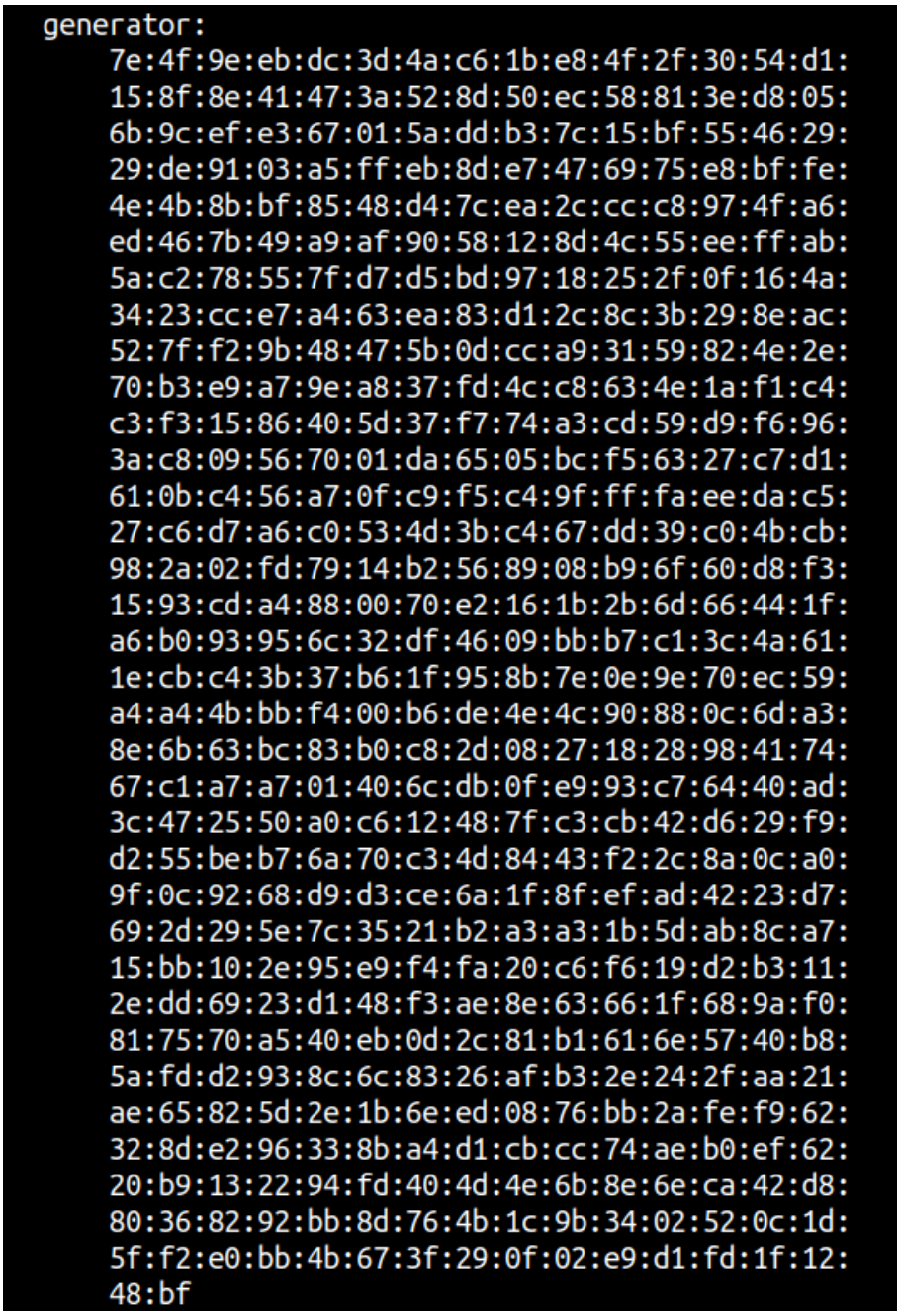

Figure 3: Group generator generated using DH parameter generation with the `dsaparam` option

Exercise 3

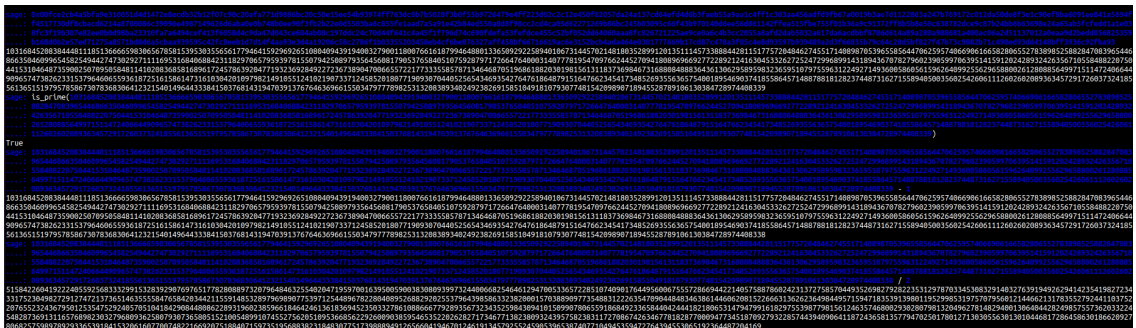

Figure 4: Prime number generated using DH parameter generation without the `dsaparam` option

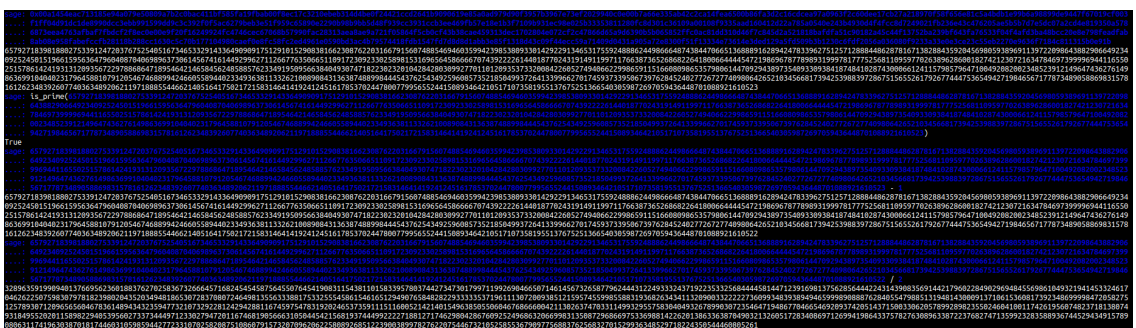

Figure 5: Prime number generated using DH parameter generation with the `dsaparam` option

Exercise 4

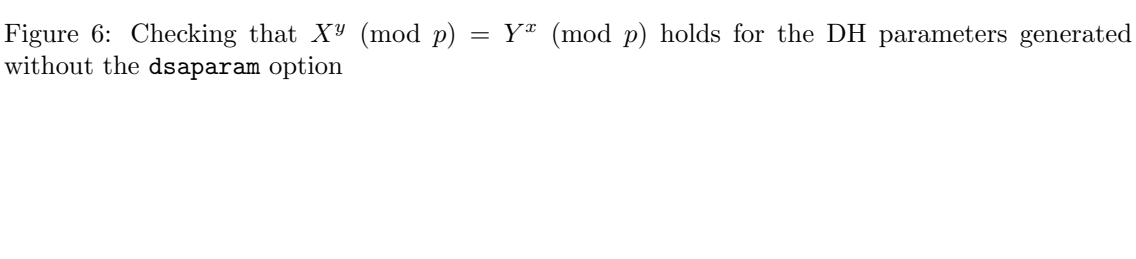

Figure 6: Checking that $X^y \pmod p = Y^x \pmod p$ holds for the DH parameters generated without the `dsaparam` option

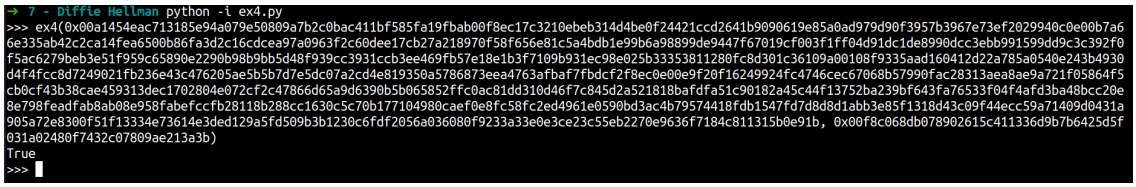

Figure 7: Checking that $X^y \pmod{p} = Y^x \pmod{p}$ holds for the DH parameters generated with the `dsaparam` option