CHALMERS UNIVERSITY OF TECHNOLOGY
Dept of Computing Sciences

Home assignment 2, Cryptography course

0. Introduction

In this assignment you will work with the ElGamal encryption system. To gain an understanding of this system and the algorithms for modular exponentiation and extended gcd, you will do various computations by hand. Of course, this implies that we will need to work with very small numbers. In fact, the setting of the assignment, common for all of you, is $\mathbb{Z}_{23}^*$. The assignment has three parts.

1. In the first part, you will investigate your given generator and show that it is in fact not a generator of the whole group $\mathbb{Z}_{23}^*$, but rather a certain subgroup. This contradicts our lecture description of ElGamal encryption; however, it turns out that in practice one often uses such subgroups. The (minor) differences with working in a subgroup will be explained below; the advantages will be discussed in a lecture week 4.

2. In the second part, you will act as a sender and encrypt a message for a receiver whose public key is known.

3. In the third part, you will act as receiver and your task is to decrypt a given message.

The generator g, the keys and the messages will again be computed from your personnummer, so the assignments differ. Since we work in such a small set, we cannot have different generators for all of you; that would require more than 100 generators. Similarly, keys and messages cannot be all different, but we expect that there will not be two assignments that are identical in all respects. Before you start your work on the assignment, you need to generate your assignment data. This is done by giving your personnummer as command line argument to a program on our Linux machines. You can

- visit this webpage (http://www.cse.chalmers.se/edu/year/2016/course/TDA352/cryptoAss2.php). Here, you insert your civic registration number (personnummer). If you are a foreign student and do not yet have a personnummer, you may construct one on the form YYMMDDXXXX, where YYMMDD is your birth date and XXXX are any four digits. Of course, you must include your personnummer in the solution, so that we can check your solution.

- remotely log in to one of these, open a terminal window and do as follows (where $ is the shell prompt):

```
$ ~brunetta/cryptoAss2 1234567890
Home assignment 2, cryptography.
Data for 1234567890.
Your generator is g=12.
```

```
For task 2, the receiver's public key is X=16 and the message is m=12.
For task 3, your private key is x=10 and the ciphertext is (2,17)(18,22)(16,21).
```

Here, you replace 1234567890 with your civic registration number (personnummer). If you are a foreign student and do not yet have a personnummer, you may construct one on the form YYMMDDXXXX, where YYMMDD is your birth date and XXXX are any four digits. Of course, you must include your personnummer in the solution, so that we can check your solution.

1. Analyzing the generator.

Your assignment data will contain an element $g \in \mathbb{Z}_{23}^*$. Your first step is to check whether it really generates the whole group. In this small setting, we can do this by simply calculating, in $\mathbb{Z}_{23}^*$, the powers g^i for $0 \leq i \leq 21$. (Careful: do not start from scratch for each i, but make use of the fact that $g^{i+1} = g^i \cdot g$).

Task 1A: Do this and include a table of the results in your report.

For g to be a generator, the computed powers should be a permutation of all the 22 elements of $\mathbb{Z}_{23}^*$. However, we have given you numbers g that fail to be generators of the whole group. In fact, only 11 different numbers appear and then the sequence repeats. Let S be the set consisting of these 11 numbers. It turns out that the set S has the property that for any two elements $a, b \in S$ also their product $a \cdot b \in S$. We say that S is *closed* under multiplication. Thus, we can work with only this subset and ignore the other 11 numbers in $\mathbb{Z}_{23}^*$.

Task 1B: Choose at least one of the following two tasks (you need not do both):

1. Just check at least five pairs of numbers from S for this property, i.e. that their product is also in S. Include your tests in the report.
2. (Much more useful!) Prove that if $q > 0$ and $g \in \mathbb{Z}_p^*$ has the property that $g^q = 1$, then the set $\{g^i | 0 \leq i < q\}$ is closed under multiplication.

There is a lot of further structure here for those who want to explore $\mathbb{Z}_{23}^*$ in more detail. Examples:

1. From Fermat's little theorem we know that $a^{22} = 1$ for all $a \in \mathbb{Z}_{23}^*$. We have now found a subset of 11 elements for which $a^{11} = 1$. For all elements in S except one (which one?), 11 is the least power that gives result 1.
2. There is a subgroup of $\mathbb{Z}_{23}^*$ with two elements, namely $\{1, 22\}$ (check it!).
3. It is not a coincidence that $\mathbb{Z}_{23}^*$ has 22 elements, that $22 = 2 \cdot 11$ and that there is one subgroup with 11 elements and one with two elements.
4. All elements of S are squares, i.e. for every $a \in S$ there is an $x \in \mathbb{Z}_{23}^*$ (in fact, two) such that $a = x^2$.

2. Encrypting a message

Your assignment data will contain a public key X for receiver Alice and a message m to encrypt. Since the key is actually g^x for some secret x, we know that $X \in S$ and that actually Alice chose an $0 < x \leq 10$, since it is pointless to have $11 \leq x \leq 21$ (since anyhow $g^x = g^{x-11}$).

Task 2A: Since you have computed all powers of g you actually can find out x. Which number is it? (In a realistic system where p is very large, you of course cannot find x from X; this is the discrete logarithm problem.)
The message m can be any number in $\mathbb{Z}_{23}^*$, also outside S. A sample message m is given in your assignment data.
Task 2B: Choose an encryption key k and encrypt this m for Alice. Show not only the result, but also what computations you performed.
Task 2C: To check that your computation in the previous task is correct, you should take on the role of Alice and decrypt the message and check that you recover m. For realistic key sizes, you cannot do this, but from Task 2A you know x. Perform this decryption and include it in your report, showing your computations.

3. Decrypting a message.

Your assignment data will contain also a choice of private key for you (sorry that we did not allow you to choose it yourself!).

Task 3A: Which is your public key?

Finally, your assignment data contains a ciphertext for you, consisting of three ElGamal pairs encrypted using your public key. If you decrypt them, you get a plaintext consisting of three numbers in $\mathbb{Z}_{23}^*$. Interpreting these numbers as letters using 1=A, 2=B, etc. you will get a plaintext word, which in all assignments is an English name (or perhaps a short form of a name). Since we only have 22 elements, we cannot encrypt words containing the last four letters of the alphabet, but let us not worry about that.

Task 3B: Decrypt the message, showing the computations you did.