Problems

8.1. Understanding the functionality of groups, cyclic groups and subgroups is important for the use of public-key cryptosystems based on the discrete logarithm problem. That's why we are going to practice some arithmetic in such structures in this set of problems.

Let's start with an easy one. Determine the order of all elements of the multiplicative groups of:

1. $\mathbb{Z}_5^*$
2. $\mathbb{Z}_7^*$
3. $\mathbb{Z}_{13}^*$

Create a list with two columns for every group, where each row contains an element a and the order $\text{ord}(a)$.

(Hint: In order to get familiar with cyclic groups and their properties, it is a good idea to compute all orders "by hand", i.e., use only a pocket calculator. If you want to refresh your mental arithmetic skills, try not to use a calculator whenever possible, in particular for the first two groups.)

8.2. We consider the group $\mathbb{Z}_{53}^*$. What are the possible element orders? How many elements exist for each order?

8.3. We now study the groups from Problem 8.2.

1. How many elements does each of the multiplicative groups have?
2. Do all orders from above divide the number of elements in the corresponding multiplicative group?
3. Which of the elements from Problem 8.1 are primitive elements?
4. Verify for the groups that the number of primitive elements is given by $\phi(|\mathbb{Z}_p^*|)$.

8.4. In this exercise we want to identify primitive elements (generators) of a multiplicative group since they play a big role in the DHKE and and many other public-key schemes based on the DL problem. You are given a prime $p = 4969$ and the corresponding multiplicative group $\mathbb{Z}_{4969}^*$.

1. Determine how many generators exist in $\mathbb{Z}_{4969}^*$.
2. What is the probability of a randomly chosen element $a \in \mathbb{Z}_{4969}^*$ being a generator?
3. Determine the smallest generator $a \in \mathbb{Z}_{4969}^*$ with $a > 1000$.
   Hint: The identification can be done naïvely through testing *all* possible factors of the group cardinality $p-1$, or more efficiently by checking the premise that $a^{(p-1)/q_i} \neq 1 \bmod p$ for all prime factors q_i with $p-1 = \prod q_i^{e_i}$. You can simply start with $a = 1001$ and repeat these steps until you find a respective generator of $\mathbb{Z}_{4969}^*$.
4. What measures can be taken in order to simplify the search for generators for arbitrary groups $\mathbb{Z}_p^*$?

8.5. Compute the two public keys and the common key for the DHKE scheme with the parameters $p = 467$, $\alpha = 2$, and

1. $a = 3$, $b = 5$
2. $a = 400$, $b = 134$
3. $a = 228$, $b = 57$

In all cases, perform the computation of the common key for Alice *and* Bob. This is also a perfect check of your results.

8.6. We now design another DHKE scheme with the same prime $p = 467$ as in Problem 8.5. This time, however, we use the element $\alpha = 4$. The element 4 has order 233 and generates thus a subgroup with 233 elements. Compute k_{AB} for

1. $a = 400$, $b = 134$
2. $a = 167$, $b = 134$

Why are the session keys identical?

8.7. In the DHKE protocol, the private keys are chosen from the set

$$\{2, \ldots, p-2\}.$$

Why are the values 1 and $p-1$ excluded? Describe the weakness of these two values.

8.8. Given is a DHKE algorithm. The modulus p has 1024 bit and α is a generator of a subgroup where $\text{ord}(\alpha) \approx 2^{160}$.

1. What is the maximum value that the private keys should have?
2. How long does the computation of the session key take on average if one modular multiplication takes 700 μs, and one modular squaring 400 μs? Assume that the public keys have already been computed.
3. One well-known acceleration technique for discrete logarithm systems uses short primitive elements. We assume now that α is such a short element (e.g., a 16-bit integer). Assume that modular multiplication with α takes now only 30 μs. How long does the computation of the public key take now? Why is the time for one modular squaring still the same as above if we apply the square-and-multiply algorithm?

8.9. We now want to consider the importance of the proper choice of generators in multiplicative groups.

1. Show that the order of an element $a \in \mathbb{Z}_p$ with $a = p-1$ is always 2.
2. What subgroup is generated by a?
3. Briefly describe a simple attack on the DHKE which exploits this property.

8.10. We consider a DHKE protocol over a Galois fields $GF(2^m)$. All arithmetic is done in $GF(2^5)$ with $P(x) = x^5 + x^2 + 1$ as an irreducible field polynomial. The primitive element for the Diffie–Hellman scheme is $\alpha = x^2$. The private keys are $a = 3$ and $b = 12$. What is the session key k_{AB}?

8.11. In this chapter, we saw that the Diffie–Hellman protocol is as secure as the Diffie–Hellman problem which is probably as hard as the DL problem in the group $\mathbb{Z}_p^*$. However, this only holds for passive attacks, i.e., if Oscar is only capable of eavesdropping. If Oscar can manipulate messages between Alice and Bob, the key agreement protocol can easily be broken! Develop an active attack against the Diffie–Hellman key agreement protocol with Oscar being the man in the middle.

8.12. Write a program which computes the discrete logarithm in $\mathbb{Z}_p^*$ by exhaustive search. The input parameters for your program are p, α, β. The program computes x where $\beta = \alpha^x \bmod p$.

Compute the solution to $\log_{106} 12375$ in $\mathbb{Z}_{24691}$.

8.13. Encrypt the following messages with the Elgamal scheme ($p = 467$ and $\alpha = 2$):

1. $k_{pr} = d = 105$, $i = 213$, $x = 33$
2. $k_{pr} = d = 105$, $i = 123$, $x = 33$
3. $k_{pr} = d = 300$, $i = 45$, $x = 248$
4. $k_{pr} = d = 300$, $i = 47$, $x = 248$

Now decrypt every ciphertext and show all steps.

8.14. Assume Bob sends an Elgamal encrypted message to Alice. Wrongly, Bob uses the same parameter i for all messages. Moreover, we know that each of Bob's cleartexts start with the number $x_1 = 21$ (Bob's ID). We now obtain the following ciphertexts

$$(k_{E,1} = 6, y_1 = 17),$$
$$(k_{E,2} = 6, y_2 = 25).$$

The Elgamal parameters are $p = 31, \alpha = 3, \beta = 18$. Determine the second plaintext x_2.

8.15. Given is an Elgamal crypto system. Bob tries to be especially smart and chooses the following pseudorandom generator to compute new i values:

$$i_j = i_{j-1} + f(j) \ , \ 1 \leq j \tag{8.5}$$

where $f(j)$ is a "complicated" but known pseudorandom function (for instance, $f(j)$ could be a cryptographic hash function such as SHA or RIPE-MD160). i_0 is a true random number that is not known to Oscar.

Bob encrypts n messages x_j as follows:

$$k_{E_j} = \alpha^{i_j} \bmod p,$$
$$y_j = x_j \cdot \beta^{i_j} \bmod p,$$

where $1 \leq j \leq n$. Assume that the last cleartext x_n is known to Oscar and all ciphertext.

Provide a formula with which Oscar can compute any of the messages x_j, $1 \leq j \leq n-1$. Of course, following Kerckhoffs' principle, Oscar knows the construction method shown above, including the function $f()$.

8.16. Given an Elgamal encryption scheme with public parameters $k_{pub} = (p, \alpha, \beta)$ and an unknown private key $k_{pr} = d$. Due to an erroneous implementation of the random number generator of the encrypting party, the following relation holds for two temporary keys:

$$k_{M,j+1} = k_{M,j}^2 \mod p.$$

Given n consecutive ciphertexts

$$(k_{E_1}, y_1), (k_{E_2}, y_2), ..., (k_{E_n}, y_n)$$

to the plaintexts

$$x_1, x_2, ..., x_n.$$

Furthermore, the first plaintext x_1 is known (e.g., header information).

1. Describe how an attacker can compute the plaintexts $x_1, x_2, ..., x_n$ from the given quantities.
2. Can an attacker compute the private key d from the given information? Give reasons for your answer.

8.17. Considering the four examples from Problem 8.13, we see that the Elgamal scheme is nondeterministic: A given plaintext x has many valid ciphertexts, e.g., both $x = 33$ and $x = 248$ have the same ciphertext in the problem above.

1. Why is the Elgamal signature scheme nondeterministic?
2. How many valid ciphertexts exist for each message x (general expression)? How many are there for the system in Problem 8.13 (numerical answer)?
3. Is the RSA crypto system nondeterministic once the public key has been chosen?

8.18. We investigate the weaknesses that arise in Elgamal encryption if a public key of small order is used. We look at the following example. Assume Bob uses the group $\mathbb{Z}_{29}^*$ with the primitive element $\alpha = 2$. His public key is $\beta = 28$.

1. What is the order of the public key?
2. Which masking keys k_M are possible?
3. Alice encrypts a text message. Every character is encoded according to the simple rule a $\rightarrow$ 0,..., z $\rightarrow$ 25. There are three additional ciphertext symbols: ä $\rightarrow$ 26, ö $\rightarrow$ 27, ü $\rightarrow$ 28. She transmits the following 11 ciphertexts (k_E, y):

$$(3,15),(19,14),(6,15),(1,24),(22,13),(4,7),$$
$$(13,24),(3,21),(18,12),(26,5),(7,12)$$

Decrypt the message without computing Bob's private key. Just look at the ciphertext and use the fact that there are only very few masking keys and a bit of guesswork.