Permitted items:

- The following items are permitted
    - Books (phyisical or pdf), printouts, digital documents on the computer or online, handwritten notes
    - Your homeworks and the corrections you received
    - Blank paper for taking notes (no upload of pictures)
    - Pens, pencils, etc
    - Calculators
    - You may run comuter algebra systems as well as your own code on the computer and in online calculators
    - You may use spell-checking tools and pepare text in other editors.
- You may **not** communicate with any other person regarding the exercises by any means during the exam. As an exeption you may contact Tanja Lange if you encounter any problems.
- Looking up existing webpages is permitted; posting the questions or answers counts as communication and is not permitted.
- You may visit the bathroom during the exam time and you may have food and drink on your desk.

Instructions for answering questions:
All answers should be entered into the answer fields in Ans; do not write on paper and upload photos of your answers.

The exam has numerical questions, i.e. questions you answer with a single number, and open questions, i.e. questions where you get a text field and can type arbitrary text. For the latter type of questions, make sure to justify your answers in detail and to give clear arguments. Document all steps, in particular of algorithms. It is not sufficient to state the correct result without explanation.

You may copy instructions and outputs from your computer algebra system into the answers but need to explain what they do and why you invoke them.If an exercise requires usage of a particular algorithm, other approaches will not be accepted even if they give the correct result.

Video upload:
After this first part finishes you should record a video of you explaning your solution. Choose 3 exercise parts which are not numerical questions and aim for 5 min of recordinng (no longer than 10 min). Show your student ID and state your name at the beginning of the video.
Please use https://surfdrive.surf.nl/files/index.php/s/nn9BxdmBv58YNdF
for uploading your video. Name the file as
ID_[student ID]_[Last name].[file format]
filling in your TU/e student ID, your last name, and the file format (mp4, webm) instead of the brackets.
If your connection is too weak, store the video on your computer and compute the SHA-256 checksum of it and mail that to Tanja Lange at t.lange@tue.nl.

Support:
If you want to indicate that any unwanted disturbances occurred that might be registered as an irregularity, or if your exam does not go as expected due to technical problems that indered your exam (for example power or Internet failure in the region), you can report this within 24 hours to the Examination Committee via the Webform Online Exam at https://educationguide.tue.nl/studying/corona/webform-online-exams/.

1 RSA

This exercise is about the RSA cryptosystem.

0.5p a Carry out the RSA key generation for primes p =1823 and q =1549 and exponent $e = 2^{16} + 1$. The results will be used in this and the following 2 exercise parts.

Answer this questions with n.

Answer

0.5p b Aswer this questionn with $\varphi(n)$ in the setting of part a).

Answer

2.0p c Aswer this questionn with d in the setting of part a).

Answer

2.0p d Bob has public key $(n, e) = (2760427, 65537)$ and private key $(n, d) = (2760427, 1459889)$. He receives ciphertext $c = 2154271$ which was encrypted using schoolbook RSA to his public key. Decrypt c to compute the corresponding message.

Answer

2 Elliptic-curve discrete logarithm

This exercise is about the elliptic-curve discrete-logarithm problem (ECDLP).

For this exercise we will be considering an elliptic curve given in Weierstrass form $M : y^2 = x^3 + Ax^2 + x$ with $A = 2097$ over the finite field $\mathbb{F}_p$ for $p = 3373$.
There are $n = 3280 = 2^4 \cdot 5 \cdot 41$ points on the curve over $\mathbb{F}_p$ and the group is cyclic.

A generator for the group is $P = [1873, 421]$. You are given $Q = [2574, 1329]$, another point on this curve, and the task through this exercise is to compute the discrete logarithm of Q with base P, i.e., compute a with $Q = aP$.

12.0p a [Scroll up to see the definitions of M, P, Q etc. if you navigated here without seeing them.]

The following is - up to notation - a more detailed instruction of the Pohlig-Hellman computation for prime 2.

Compute $a \equiv a_{2,0} + a_{2,1}2 + a_{2,2}2^2 + a_{2,3}2^3 \bmod 2^4$ by first determining images of the base P and target Q in the subgroup of order 2 that allow to compute $a_{2,0}$, and then updating the target to another element of in the subgroup of order 2 to compute $a_{2,1}$ using the same table of multiples of P as in the first step. Continue the same for $a_{2,2}$ and $a_{2,2}$.

Explain your steps and verify your answer.

5.0p b [Scroll up to see the definitions of M, P, Q etc. if you navigated here without seeing them.]

Compute $a \bmod 5$.

12.0p c [Scroll up to see the definitions of M, P, Q etc. if you navigated here without seeing them.]

Let $\ell = 41$. Use the baby-step giant-step algorithm to compute $a \bmod \ell$.

For this, start by first determining images of the base P and target Q in the subgroup of order ℓ. Then compute and state the table of baby steps. Finally compute giant steps till you can compute $a \bmod \ell$.

Verify your answer.

4.0p d [Scroll up to see the definitions of M, P, Q etc. if you navigated here without seeing them.]

Combine the results from the previous three exercise parts to compute a with $Q = aP$. Verify your answer.

If you do not have all results, combine those that you do have and perform the verification on that part, i.e. in the matching subgroup.

5.0p e [Scroll up to see the definitions of M, P, Q etc. if you navigated here without seeing them.]

The elliptic curve M is a Montgomery curve. Compute the twisted Edwards curve E which is birationally equivalent to it and compute the images of P and Q on it, i.e., compute the coefficients a and d and images P_E and Q_E so that P_E and Q_E are on $E: ax^2 + y^2 = 1 + dx^2y^2$.

Verify that both image points satisfy this curve equation.

3 Pollard rho for factorization

This exercise is about factoring integers.

8.0p a Let n =1149653

Use Pollard's rho method for factorization and Floyd's cycle-finding method with starting point ρ_0 =863593 and step update constant c =63 to factor n,i.e., use iteration function $\rho_{k+1} = \rho_k^2 + c$ for the above constants ρ_0 and c.

Document all steps you have computed and your code.

Compute the other factor of n as well.

8.0p b Describe in your own words how and why Pollard's rho method for factorization works and what the expected runtime is depending on the factors of n.

Use this to explain why you managed to factor n in part a) and explain why the p-1 method would likely not have succeded for factoring n when used with $B_1 = 10$, i.e., $s =$lcm$(1, 2, 3, \ldots, 10)$.

This latter part should refer to the concrete numbers you handled in part a).

4 Coppersmith's method

This exercise is about Coppersmith's method for recovering parts of stereotyped messages if a large-enough part of the message is known.

A company handling event tickets is using schoolbook RSA without padding and you
learn that each message has the form

```
The ticket code for entry to the event is
```

which is followed by 6 alphanumeric characters, i.e., numbers in [0,9] and letters in [a,z]. The message is then encoded in base 36. Note that this ignores the
spaces between words. Here is an example

```
sage: m = Integer('The ticket code for entry to the event is
0123456789abcdef',36)
sage: m
148628258710492576550287732171601957821090429961446452406550345151634854179 43
sage: m.str(36)
'theticketcodeforentrytotheeventis0123456789abcdef'
```

You observe some ciphertext c from the system to Alice and you know that Alice's public key is (n, e) with

$n =$
22025482441888726596151828037580210653496214995318500244177792057815419084860452102794997828952857497081479129671501030354465069182775254612751

and $e = 5$.

9.0p a Explain how and why you can use Coppersmith' method to compute the missing part of m from c given the above information, i.e., the known part of m and that the event codes have 6 alphanumeric characters as well as n and e. Scroll up to see the parameters.

Note that factoring n to obtain $\varphi(n)$ and thus the decryption key does not count as a solution.

8.0p **b** Execute the attack you described in part a) for ciphertext
$c =$
18026784157801727131455776136196465618870787523053645052562578424425084803098429518969764363704209533679484842464484114144371404287037590112297

and
$n =$
22025482441888726596151828037580210653496214995318500244177792057815419084860452102794997828952857497081479129671501030354465069182775254612751

Make sure to document all computation and results.

5 PIN system for authentication

Tom wants to build a chat system. He has learned that normal Diffie-Hellman key exchange does not offer any authentication and he is concerned about Eve mounting a man-in-the-middle attack. He understands the 3DH system and that he could use signatures to sign the ephemeral DH keys, however, he wants to make it possible for his users to use public computers to communicate and this means that they cannot store long-term keys for identification.

One thing to make his situation a bit easier is that the users know each other and so each pair of users can agree on a 6-digit pin to be used to secure their connection. For technical reasons the pin 000000 is not alowed.

The system that Tom builds works as follows:
Alice starts the program on her computer using her username and password. If she wants to chat with Bob she enters his username. If Bob happens to be online then Bob and Alice are prompted to enter their 6-digit pin code. The system uses the pin code and some randomness to create a symmetric key which is then used with a stream cipher and MAC to secure their chat.

Internally, the system works as follows:
The system uses a big prime $p =$
$596456925263842318066947352208796711390717753666295391038185689795719525460739622964368026001$
and all computations take place in the multiplicative group $G = \mathbb{F}_p^*$.
The pin code is taken as an integer in $[1, 10^6 - 1]$ and then used as an element $c \in G$, e.g., pin code 123456 is taken as $c = 123456$ and all computations involving c are done modulo p.
After Alice enters the pin code c shared with Bob, the system picks a random $a \in [0, p-2]$, computes c^a, and sends it to Bob. Likewise, after Bob enters c the system picks a random $b \in [0, p-2]$, computes c^b, and sends it to Alice. Each side can now compute c^{ab} and $k = H(c^{ab})$ is used as the symmetric key, where H is a cryptographic hash function.
The first message from Alice is $H(alice, k)$ and from Bob is $H(bob, k)$, where $alice$ and bob are their respective usernames. Both locally compute the respective other message and if these messages do not match the incoming one, the connection closes. If their values match, k is used to encrypt and authenticate their chat.

8.0p **a** Explain why the system works if Alice and Bob enter the same pin code c.
Explain what goes wrong if Charlie tries to impersonate Bob and enters $c' \neq c$.

4.0p **b** Tom seems to have missed several lessons on secure DL systems as the system does not actually prove that the sender knows the pin code. Observe the range of exponents that a can be taken from and find a value for c^a that Eve can send to Bob to impersonate Alice, i.e., a value so that she can compute the matching $k = H(c^{ab})$ without knowing c and independent of Bob's value for c^b.

12.0p **c** Even if Tom manages to avoid the issue you found in b) there are still problems with the system.

Note that $p - 1$ factors as

```
2^4 * 3^3 * 5^3 * 7 * 11 * 13 * 19 * 89 * 139 * 197 * 1297 * 2357 * 3677
* 6037 * 8747 * 10957 * 13901 * 14411 * 17291 * 23593 * 62473 * 172603 *
224501 * 405001 * 592621 * 825551 * 934721
```

Show how you can determine c from observing just one exchange c^a, c^b, and $H(alice, k)$ and doing some feasible local computation.

Note that feasible means under 10 minutes of computation (after Eve spent some time writing suitable code).