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. The latter includes communicating with chatbots such as ChatGPT and is thus 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/ukjhST9QJWZPVnR
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/.

RSA-CRT

This exercise is about RSA key generation and usage of the private key with the CRT method. The first exercise parts are numerical exercises, i.e., your answer consists of a single number. The last part requires answering with text and your calculation.
You will be gererating a key using primes $p = 1499$ and $q = 1579$ and public exponent $e = 2^{16} + 1 = e$.

0.5p **1a** See the general description for the parameters.

Answer this questions with n.

Answer

0.5p **1b**

Aswer this questionn with $\varphi(n)$.

Answer

2.0p **1c** Aswer this question with d.

Answer

1.0p **1d** Answer this question with d_p.

Answer

1.0p **1e** Answer this question with d_q.

Answer

5.0p **1f** You receive ciphertext $c = 1718641$ encrypted to your public key. Use the CRT method to decrypt it. The last two numerical exercises belong to the steps in the CRT decryption. The rest of the answer goes to the last question.

Use the CRT method to decrypt the ciphertext. In the previous two exercise parts you have already computed d_p and d_q, now compute c_p, c_q, m_p, m_q, u, and m. Make sure to state what you are computing.

Once you have computed m verify your solution by reencrypting it.State what computation you do or copy the calculation from your computer algebra system.

Edwards and Montgomery

This exercise is about arithmetic on elliptic curves in Edwards and Montogomery form.
The point $P = (x, y)$ is on the twisted Edwards curve $E : ax^2 + y^2 = 1 + dx^2y^2$ modulo $p = 2^{31} - 1$ for $a = p - 1$ and
x =2091124997

y =52661690
d =2045781447
Compute $2P$ on the Edwards curve and compute the birationally equivalent Montgomery curve M : $Bv^2 = u^3 + Au^2 + u$. Compute the images of P and $2P$ on M and double the image of P on M to see that it matches the image of $2P$.
Make sure to use positive integers in $[0, p-1]$ to represent the results.
The first exercise parts are numerical exercises and ask for intermediate results from this computation; answer each with a number (you cannot enter text into the field). The last part is an open question where you should enter text and computation details

2.0p **2a** Answer this exercise with the x-coordinate of $2P$.

Answer

2.0p **2b** Answer this exercise with the y-coordinate of $2P$.

Answer

1.0p **2c** Answer this exercise with A.

Answer

1.0p **2d** Answer this exercise with B.

Answer

1.0p **2e** Answer this exercise with the u-coordinate of the image of P.

Answer

1.0p **2f** Answer this exercise with the v-coordinate of the image of P.

Answer

1.0p **2g** Answer this exercise with the u-coordinate of the image of $2P$.

Answer

1.0p **2h** Answer this exercise with the v-coordinate of the image of $2P$.

Answer

1.0p **2i** Answer this exercise with the slope λ computed in doubling the image of P.

Answer

3.0p 2j Let $Q = (u, v)$ be the image of P on M.

Verify that Q is on M.

Show the computation of $2Q$ on the Montgomery curve. Document the formulas you use and the results you get.
You should check that the result matches what you computed above

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 $A = 749587589429431132509710 9$ over the finite field $\mathbb{F}_p$ for $p = 38685626227668216277324061$.
There are $n = 2^5 \cdot \ell$ points on the curve over $\mathbb{F}_p$ and the group is cyclic. Here ℓ is a large, 80-bitt prime $\ell = 1208925819614337375438193$

A generator for the group is
$P = [317361871595703261753400 31, 348575579505426639061446 16]$.
You are given
$Q = [156961769470302912811351 39, 360257872497908017153300 21]$,
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 3a [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 $a \equiv a_{2,0} + a_{2,1}2 + a_{2,2}2^2 + a_{2,3}2^3 + a_{2,4}2^4 \bmod 2^5$ 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 $a_{2,2}, a_{2,3}$ and $a_{2,4}$.

Explain your steps and verify your answer.

5.5p **3b** Explain how you would solve the ECDLP in the subgroup of order $\ell = 1208925819614337375438193$.

This explanation needs to include how you map the base point and target to this subgroup and what algorithm you would choose. Note that ℓ is large, so the question is not to actually perform this attack, but you should describe the steps and explain how expensive the attack is.

2.0p **3c** The DLP modulo ℓ gives $a \equiv 2323232342424242$. Combine this with your answer from part a) to compute a modulo n.

Verify your result.

Factorization

This exercise is about factoring integers. The integer n is a product of two primes.

1.0p **4a** Use the $p-1$ method to factor $n = 1247321099$ with base $a = 66533$ and exponent s the lcm of $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16\}$.
Make sure to compute the value for s and to compute the result b of the exponentiation modulo n.

For this part filll in the value for s.

This is a copy of the description above as the Ans Deft review after the exam does not seem to show the problem descriptions:
This exercise is about factoring integers. The integer n is a product of two primes.

Answer

1.0p **4b** This is a continuation of the previous exercise. For this answer fill in the value of b (the result after exponentiation, but before subtracting 1).

Answer

1.0p **4c** This is a continuation of the previous exercise. Fill in the factor you obtained from the gcd.

Answer

9.5p **4d** This is a continuation of the previous exercise.

Explain why the $p-1$ method was successful in factoring this n.

Consider whether the exponent s would have worked for any base a for these factors and if not, give conditions for which a it does work and how restrictve these are.

Hint: To give a proper argument you will need to compute the factorizations of $p-1$ and $q-1$.

For the factorizations and other computations in this exercise you can use a computer algebra system (Sage, Pari-GP, ...). You do not need to run Pollard's rho method or such for obtaining factorizations. Make sure to state what computations you made, what the answers were, and how they help in solving this question.

Word-size primes

Werner knows that integer multiplications are fastest if the integers fit in the word size of his computer. He is on a 64-bit computer, so computations modulo primes of up to 64 bits are very efficient. He also knows that public-key cryptography needs to use much larger primes to be achieve security but he comes up with an interesting system that uses many small fields.

In particular he uses the 64-bit primes $p_1 = 2^{64} - 59, p_2 = 2^{64} - 83, p_3 = 2^{64} - 95, p_4 = 2^{64} - 179, p_5 = 2^{64} - 189$, and $p_6 = 2^{64} - 257$. The finite fields $\mathbb{F}_{p_i}$ are generated by $g_1 = 2, g_2 = 2, g_3 = 3, g_4 = 2, g_5 = 2$, and $g_6 = 7$.
You do not need to check that the g_i are generators or that the p_i are prime.

Werner's scheme is based on ElGamal encryption and messages must be less than p_6. The keys are as large as for a prime of size $2^{6 \cdot 64}$ but he has found a trick to make ciphertexts shorter. So it is both faster than for a large prime and requires less bandwidth.
Here are the parts of the scheme. The primes p_i and generators g_i are fixed and known to everybody.

KeyGen: To generate a key, A picks $a_i \in [1, p_i - 1]$ for $i = 1, 2, \ldots, 5, 6$ and computes $h_i \equiv g_i^{a_i} \bmod p_i$. The private key is $[a_1, a_2, a_3, a_4, a_5, a_6]$ and the public key is $[h_1, h_2, h_3, h_4, h_5, h_6]$.

Encrypt: To encrypt message $m \in [1, p_6 - 1]$ to a user with public key $[h_1, h_2, h_3, h_4, h_5, h_6]$ perform the following steps:

- let $c = m$.
- for i = 6 down to 1:
  - pick a random k_i and compute $r_i \equiv g_i^{k_i}$
  - update $c \equiv h_i^{k_i} \cdot c \bmod p_i$, consider c as an integer in $[0, p_i - 1]$
- output $[r_1, r_2, r_3, r_4, r_5, r_6, c]$.

Decrypt: To decrypt $[r_1, r_2, r_3, r_4, r_5, r_6, c]$ using private key $[a_1, a_2, a_3, a_4, a_5, a_6]$ perform the following steps:

- let $m = c$
- for i = 1 to 6:

- update $m \equiv m/(r_i)^{a_i} \bmod p_i$, consider m as an integer in $[0, p_i - 1]$
- output m

4.0p **5a** To give you a numerical example we will do a abbreviated version of the scheme using just p_1 and p_2 so that encryption runs i = 2 down to 1 and decryption runs i = 1 to 2.

The private key of the user in the previous step is $[8234521262418220989, 14606849324963514516]$. Decrypt the ciphertext [393409703196277116, 10226847233493016439, 4355484846617436761]

Show all steps of the computation and the resulting plaintext.

4.0p **5b** Show that the scheme is correct, i.e., that the decryption of the encryption of m returns m. This answer should use the full scheme with all six primes.

Note: it is important here that $p_i > p_{i+1}$ and your arguments need to use this. Explain why decryption would not work correctly for all ciphertexts if $p_1 < p_2$.

12.0p **5c** Explain how to break the scheme and what the complexity of the attack is.

As a concrete target, here are the public key of a user and a ciphertext encrypted to that public key.
$[16952123817536195058, 582037740744937677 5, 12749820517516361691, 1273534869769649539, 819364422$
and ciphertext
$[12355222323261073666, 5475009980681203783, 10413851761635543882, 13700861180531269169, 7652336$
.
Demonstrate how to break the system by breaking the first two layers as if only the first two primes were used; you can ignore p_3, p_4, p_5, and p_6.

Note: You may use the full power of Sage, but make sure to document the all steps including what you typed and got as output.

Coppersmith's method

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

```
********** is your lucky number for today 240123 This is personally and randomly generated just for you
```

where ********** are 14 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('123456789abcde is your lucky number for today 240123 This is personally and randomly generated just for you',36)
sage: m
12350768511761770277717152514674145032178117965852450466198393649384453084717492451911369
sage: m.str(36)
'123456789abcdeisyourluckynumberfortoday240123thisispersonallyandrandomlygeneratedjustforyou'
```

You observe some ciphertext c from the system to Alice and you know that Alice's public key is (n, e) with
$n =$
37429655931265906452868596940328051929109473594248140330403673587272041011821621004978872
and $e = 3$.

15.0p **6a** 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 14 alphanumeric characters as well as n and e. Scroll up to see the parameters.
This is not identical to the stereotypical-message attack covered in the lecture because here we know the bottom characters rather than the top characters, so you need to adjust the attack to take this into account.
Hint: You do not only need to modify the polynomial to get to a monic $x + a$ but then also update the value for the ciphertext to match what happened on m.
Factoring n to obtain $\varphi(n)$ and thus the decryption key does not count as a solution.

9.0p **6b** Execute the attack you described in part a) for ciphertext
$c =$
18354297982249902818956298029417079749093126019392692672090815782800943391455711888935

and
$n =$
3742965593126590645286859694032805192910947359424814033040367358727204101182162100497

Make sure to document all computation and results.