Permitted items:

- The following items are permitted
  - Books (physical 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 computer 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 exception 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. Use your own words, do not copy text. 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 recording (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/ZdpBGS6lzYwEaZP
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 hindered 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.

1.0p a Carry out the RSA key generation for primes p =1229 and q =1777 and exponent $e = 2^{16} + 1$. The results will be used in this and the following 2 exercise parts.

Answer this question with n.

Answer

1.0p b In the setting of part a),
answer this question with $\varphi(n)$.

Answer

2.0p c In the setting of part a),
answer this question with d.

Answer

2.0p d Bob has public key $(n, e) = (2852321, 65537)$ and private key $(n, d) = (2852321, 222353)$. He receives ciphertext $c = 2735721$ which was encrypted using schoolbook RSA to his public key. Decrypt c to compute the corresponding message.

Answer

2 LFSR

This exercise is about LFSRs.

20.0p a You are given an LFSR of state length 14
via its characteristic polynomial
$P(x) = f_1 \cdot f_2$ with
f1=x^6 + x^5 + x^4 + x + 1
f2=x^8 + x^6 + x^5 + x^2 + 1
in fully factored form, i.e., both of these factors are irreducible.

Determine the order of each factor with as little computation as possible. State which powers of x you needed to test. Solutions by brute force, e.g., trying all powers of x, will not be accepted.

You should provide full justifications for the correctness of the obtained orders using the results proved in the course, i.e., use the results proved in the course to substantiate why the computations you performed indeed result in the correct orders.

Note: the polynomials are provided in raw form so that you can easily copy and paste them. Please do not make typos by manually copying them.

4.0p b What is the longest period generated by the LFSR from exercise part a)?

Make sure to justify your answer.

12.0p c State the lengths of all subsequences of the LFSR from part a) so that each state of 14 bits appears exactly once.

For this exercise you need to justify the number of different sequences and their periods using the orders obtained in part a) and the theoretical results obtained in the course. Solutions by brute force/scripts running through all possible states will not be accepted.

3 Block cipher

This exercise is about blockciphers.

8.0p a The picture below shows the DES block cipher, a typical example of a Feistel cipher. At the beginning of DES the 64 input bits are permuted using the public permutation IP and at the end its inverse $FP = IP^{-1}$. Then the block of 64 bits is split into a left half of 32 bits and a right half of 32 bits. The functions f_i use parts of the key and are not invertible.

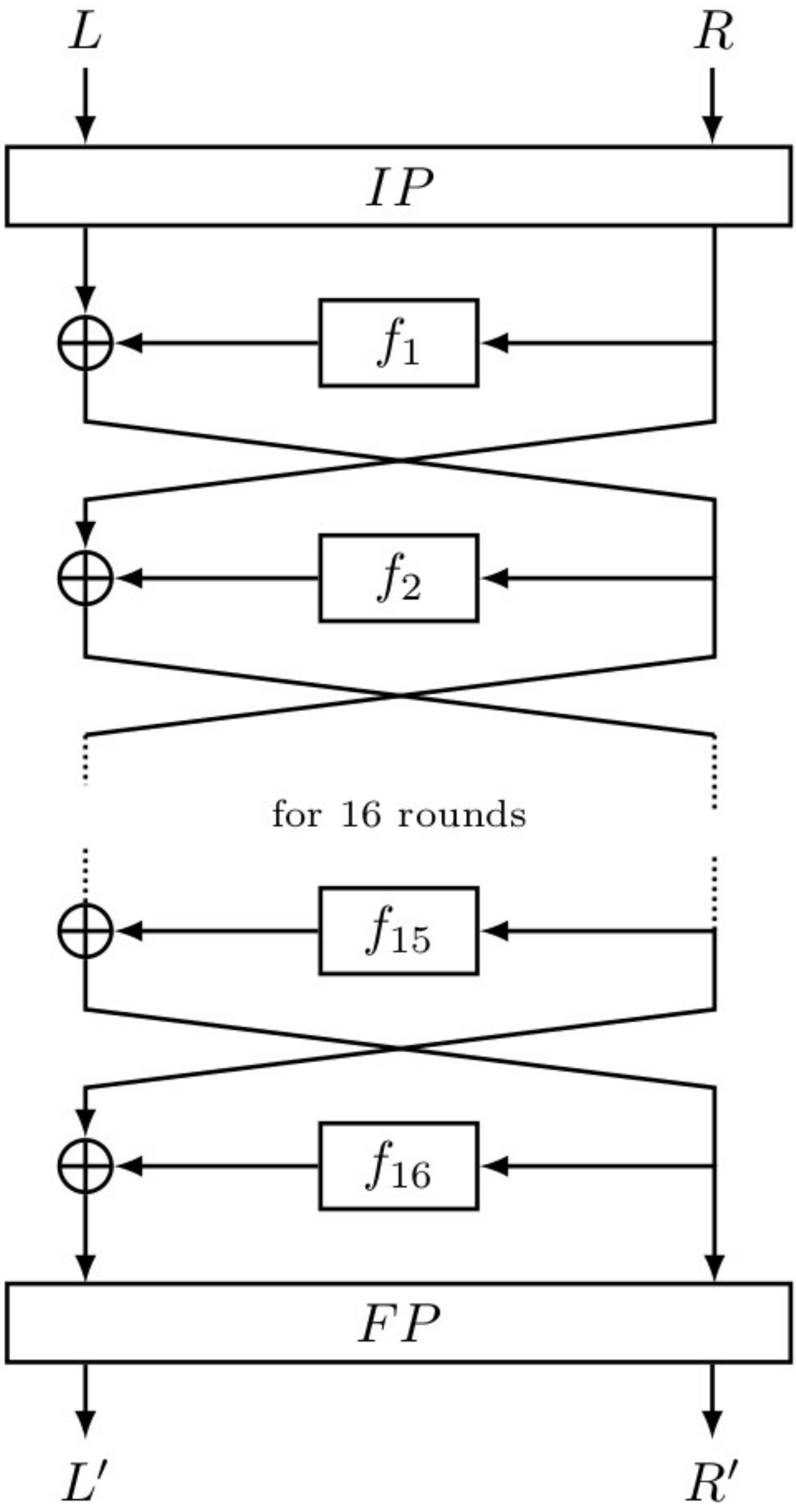

Image: Jérémy Jean at https://www.iacr.org/authors/tikz/

Explain in your own words how a user knowing all f_i can decrypt the ciphertext produced by DES. Note that the f_i are not invertible.

This includes explaining how to recover (L_{i-1}, R_{i-1}) from (L_i, R_i) for all i and how to deal with the final and initial permutation FP and IP.

6.0p b Explain in your own words why ECB is not a good mode and what a good mode needs to ensure. Comment also on the function of the IV.

4 LFSR recovery

This exercise is about reconstructing the LFSR state from IV and output.

12.0p a You intercept a ciphertext encrypted with an LFSR of state-size 4. The first 4 bits are the IV $[s_0, s_1, s_2, s_3] = [0, 0, 0, 1]$. Knowing the plaintext you also recover the first 4 output bits of running the LFSR $[s_4, s_5, s_6, s_7] = [0, 1, 1, 0]$.

Recover the coefficients c_1, c_2, c_3 of the LFSR. Note that $c_0 = 1$. Confirm your result by re-computing s_7 using the coefficients you computed.

Make sure to describe your approach and document the steps in the computation.

5.0p b Explain in your own words why using an LFSR alone, without a non-linear component, does not give a secure stream cipher. Which of the properties are violated?

5 Schoolbook RSA

This is an exercise about schoolbook RSA.

15.0p Patty is not giving up on having private parties and using schoolbook RSA encryption, i.e., there is no padding in the message. Her friends continue to use RSA keys with the same public exponent, namely $e = 3$, and their individual moduli.

Patty wants to send each of them the same message m but has learned that this is a problem with attacks. She thus decides to randomize the messages by picking random numbers $r_1, r_2,$ and r_3 for her friends and encrypting the respective messages $m_i = r_i \cdot m$ to each of them. Of course the friends need to obtain the real message m and thus she sends each of them a second ciphertext with the encryption of their respective r_i.

The public keys of her friends are $(n_1, e) = (38203916642025540226434431416882251770903184383431323185235822185071924117399, 3)$, $(n_2, e) = (25591072467291370116440326033920039740438347162169210396333773404708625432897, 3)$, and $(n_3, e) = (30034401563249273728049078929565649828583769490700036579043480961820142421373, 3)$.

You observe ciphertexts $c_{11} = 37220484279799098082895400981688034902126827294069099176998627906892130560749,$ $c_{21} = 38061594956458811327147638644090527351348153010577152426026335442439115584343$ being sent to the user with key $(n_1, 3)$, where the c_{11} is the encryption of $r_i \cdot m$ and c_{21} is the encryption of r_1 .

With the same meaning and r_2 in place of r_1 you observe $c_{12} = 17480850855499496369391028607602839523603290475334762110806234759497068032805,$ $c_{22} = 12789626243104192775603068349192883002177556406218211131661641145430476920048$ being sent to the user with key $(n_2, 3)$ and finally $c_{13} = 6286177664163589080223788893811387567931149324788545998408290604424425008581,$ $c_{23} = 21410710525297314080553589583043319363613628312742695452500198179846398349317$ sent to the user with key $(n_3, 3)$ and using r_3 in place of r_1.

Compute the message m that Patty has encrypted to them.

Verify your answer by reencrypting m to at least one of the public keys.

You can use base36 encoding to see the message, but that is not required for the solution. To see m in base 36 type m.str(36) in Sage.

You *do not need to* document intermediate steps in XGCD or CRT, or exponentiation or such.

You *do need to* say what numbers you do what computation on and why and you need to document the results of divisions, exponentiations, and CRT. Also document the commands you use in the computations.

6 ElGamal encryption

Remember that the ElGamal encryption system is defined using a group $\mathbb{F}_p^*$ with generator g. Each user has a private and a public key, e.g. Alice has private key $a < p-1$ and public key $h_A = g^a$. To encrypt message m to Alice the sender picks a random nonce k and computes $r = g^k$ and $c = h_A^k \cdot m$. The ciphertext is (r, c). (Note that this means that messages are elements of $\mathbb{F}_p^*$.)

This exercise uses $p = 675708505243$ and $g = 2$.
Stijn is stingy and he has learned that randomness is expensive, so he tries to reduce his usage of randomness. He is sure that he will not send more than one message per minute, typically less, and thus devises the following scheme so that he doesn't need more than one random number per hour:

Every hour he picks a fresh random number k. The random number at minute i is then defined as $k_i = k + f_0 \cdot i + f_1$, for $f_0 = 44$ and $f_1 = 4$.

Apart from this change he follows the protocol for ElGamal encryption.

During the same hour you observe two ciphertexts from Stijn to Bob:

- $(r_1, c_1) = (186224024260, 612295167635)$ at minute 6
- $(r_2, c_2) = (32018022611, 560600849633)$ at minute 38

Bob has public key $h_B = 401856742657$.

6.0p a Verify that the ciphertexts are compatible with Stijn using his scheme.

6.0p b You learn that (r_1, c_1) was used to send the message $m_1 = 117137944963$.
Use the knowledge of m_1 and how the randomness is generated to obtain the message encrypted in (r_2, c_2). Approaches breaking the discrete logarithm problem will not be accepted.