TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculty of Mathematics and Computer Science
Introduction to Cryptology, Monday 03 February 2025

Name :

TU/e student number :

Exercise	1	2	3	4	5	6	total
points							

Notes: Please fill out and hand in this sheet at the end of the exam. You may keep the sheets with the exercises.
This exam consists of 6 exercises. You have from 09:00 – 12:00 to solve them. You can reach 100 points.
Make sure to justify your answers in detail and to give clear arguments. Document all steps and intermediate results, in particular of algorithms on the exam paper, do not use the scrap paper. It is not sufficient to state the correct result without the explanation and the steps that lead to it.
If the problem statement asks for usage of a particular algorithm other solutions will not be accepted even if they give the correct result.
All answers must be submitted on TU/e letterhead; should you require more sheets ask the proctor. State your name on every sheet.
Do not write in red or with a pencil.
You are not allowed to use any books, notes, or other material.
You are allowed to use a simple, non-programmable calculator without networking abilities. Usage of laptops and cell phones is forbidden.

1. This exercise is about LFSRs. Do the following subexercises for the sequence

$$s_{i+6} = s_{i+5} + s_{i+4} + s_{i+3} + s_i.$$

(a) Draw the LFSR corresponding this sequence. 3 points

(b) State the characteristic polynomial f and compute its factorization. You do not need to do a Rabin irreducibility test but you do need to argue for each factor why it is irreducible.
Reminder: Factors may appear with multiplicity larger than one. Looking for roots is insufficient for larger degrees. Reference to the list of irreducible polynomials given in the lecture is insufficient, you need to give an argument for each why it is irreducible. 13 points

(c) Write the factorization of f from (b) in the form $f = \prod f_i^{e_i}$ with integers $e_i > 0$ and f_i different irreducible polynomials, i.e., group equal factors.
For each of the $f_i^{e_i}$ compute the order. 6 points

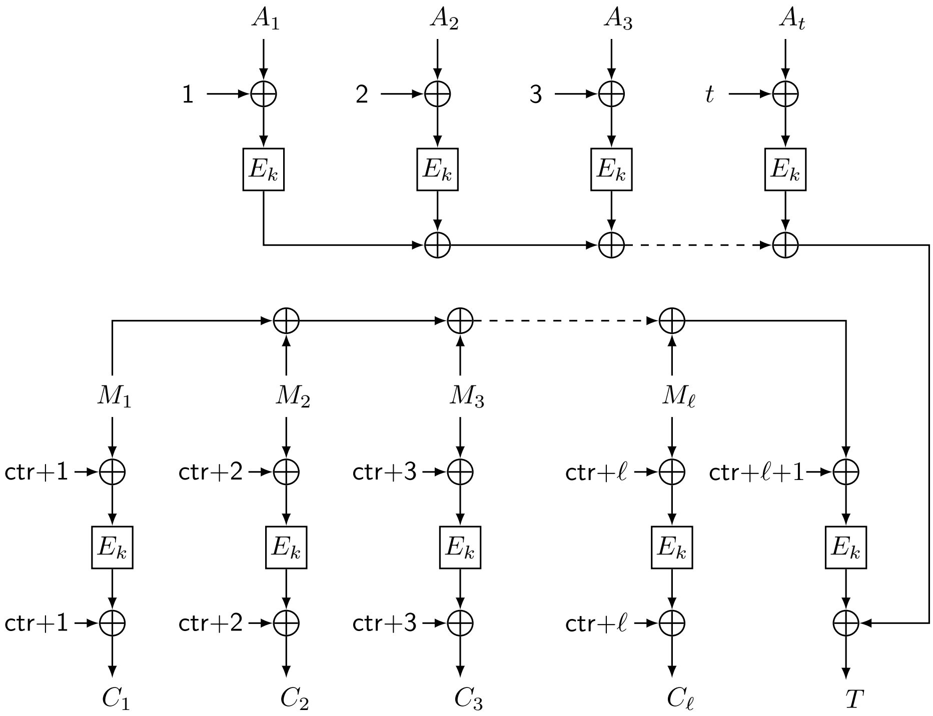
(d) What is the longest period generated by this LFSR?
Make sure to justify your answer. 3 points

(e) State the lengths of all subsequences so that each state of 6 bits appears exactly once.
Make sure to justify your answer and to show a check that all 2^6 states are covered. 7 points

2. This exercise is about modes.

   OCB3 is a mode for authenticated encryption which permits to authenticate additional data, which is not encrypted but only authenticated. OCB3 is specified for a block cipher E_k with block length $n = 128$. Let k denote the key shared by Alice and Bob. The scheme uses a 96-bit nonce per message. Remember, a nonce is a string that must never repeat. This nonce takes the role of the initialization vector.

   Here is a schematic description of a simplified version of OCB3. We assume that the message length is a multiple of 128, here 128ℓ, and that the length of the associated data is a multiple of 128, here $128t$.



   Let $E_k(M)$ denote encryption of a single block M using the block cipher with key k and let $D_k(C)$ denote decryption of a single block C using the block cipher with key k.

   Let $A_1, A_2, \ldots, A_t$ be some associated data to be authenticated, $M_i, i = 1, 2, \ldots, \ell$ be the ℓ blocks of 128 bits each holding the message, $C_i, i = 1, 2, \ldots \ell$ be the 128-bit blocks holding the ciphertexts, and T hold the authentication tag.

   The ciphertext sent is $N, A_1, A_2, \ldots, A_t, C_1, C_2, \ldots, C_\ell, T$.

   The schematic description uses $\oplus$ to denote bitwise addition of bit strings of length 128, which is the same as addition in $\mathbb{F}_2^{128}$. When an input to $\oplus$ is an integer the integer is first written in binary and then added.

The value of ctr depends on the shared key k and the nonce as follows

$$\mathsf{ctr} = E_k(N||0^{32}).$$

(a) Describe how authenticated encryption of long messages works by writing C_1 and a general C_i in terms of ctr, M_1, M_i, and (if necessary) other M_j and C_j. 4 points

(b) Write T in terms of $\mathsf{ctr}, A_1, \dots A_t, M_1, \dots, M_\ell, C_1, \dots, C_\ell$. 2 points

(c) Describe how decryption of long messages and verification of the authentication tag works by writing M_1 and a general M_i in terms of ctr, C_1, C_i, and (if necessary) other M_j and C_j and describe how the authentication tag T confirms the authenticity of the message and the additional data A. 4 points

(d) Assume that ciphertext C_j gets modified in transit. Show which message blocks get decrypted incorrectly and explain why others get decrypted correctly. Show how the authentication tag T catches this error. 5 points

(e) Assume that some block of the associated data $A_1, A_2, \dots, A_t$ gets modified in transit. Show which message blocks get decrypted incorrectly and explain why others get decrypted correctly. Show how the authentication tag T catches this error. 3 points

3. This problem is about RSA encryption. Let $p = 337$ and $q = 419$. Compute the public key using $e = 65537$ and the corresponding private key.
Reminder: The private exponent d is a positive number. 8 points

4. This problem is about the DH key exchange. The public parameters are the group G and generator g, where $G = (\mathbb{F}_{1031}^*, \cdot)$ and $g = 37$. Alice's public key is $h_A = 123$. Bob's private key is $b = 23$, Compute the DH key that Bob shares with Alice. 8 points

5. The integer $p = 31$ is prime. You are the eavesdropper and know that Alice and Bob use the Diffie-Hellman key-exchange in $\mathbb{F}_{31}^*$ with generator $g = 3$. Alice's public key is $h_A = g^a = 11$. Use the Baby-Step Giant-Step method to compute Alice's private key a.
   Verify your result, i.e. compute g^a. State the intermediate results to prove that you actually did the computation. $\boxed{\text{12 points}}$

6. The exercise introduces (a simplified version of) the BlackBasta ransomware, which targets Windows computers and ESXi hosts running virtual machine workloads by adversarially encrypting their data. Files are renamed by appending a fixed ending, but the original file name and file type remain visible in the name.

   The simplification consists in substituting some cryptographic systems with equivalent ones and scaling down the example. The issues you will find exist in the original. The exercise focuses on the cryptography and skips some other details.

   The ransomware uses both public-key and symmetric key cryptography. The public-key part is using Diffie-Hellman key exchange in a subgroup of $\mathbb{F}_p^*$ generated by some fixed base g. The symmetric-key part uses a stream cipher

   For each victim, the ransomware picks a unique public key $h_V = g^v$ which is included in the ransomware program running on the victim's computer.

   For each file m_i on the victim's computer it does the following:

   i) Pick a random r_i.
   ii) Compute $R_i = g^{r_i}$ in $\mathbb{F}_p^*$.
   iii) Compute $k_i = \mathsf{Hash}(h_V^{r_i}, h_V, R_i)$.
   iv) Compute (c_i, t_i) as the ciphertext and MAC tag of m_i under key k_i using a stream cipher and MAC.
   v) Replace m_i with (c_i, t_i, R_i).
   vi) Erase k_i, r_i, and m_i.

   Once the ransomware has made all (or most) files unusable it displays a ransom note, demanding payment. They promise to send a decryption program upon payment.

(a) The ransom note includes the key h_V and instructs the victim to send that key, once the payment is done.
Explain how the decryption program will work and what information the victim misses to write it themselves. 6 points

(b) You are called in to help a victim and start inspecting the encrypted files. Luckily the victim has an uncompromised backup with most of their files.

Here is an example of a small image. The data is in the middle, shown in hexadecimal. The left block are just the line numbers, the right block is displaying ascii strings, so focus on the middle.

The original png file:

```
0000000: 8950 4e47 0d0a 1a0a 0000 000d 4948 4452  .PNG........IHDR
0000010: 0000 0001 0000 0001 0802 0000 0090 7753  ..............wS
0000020: de00 0000 0970 4859 7300 0003 b100 0003  .....pHYs.......
0000030: b101 f583 ed49 0000 0007 7449 4d45 07df  .....I....tIME..
0000040: 0401 0319 3a3d ca0b 0c00 0000 0c69 5458  ....:=.......iTX
0000050: 7443 6f6d 6d65 6e74 0000 0000 00bc aeb2  tComment........
0000060: 9900 0000 0f49 4441 5408 1d01 0400 fbff  .....IDAT.......
0000070: 00ff 0000 0301 0100 c706 926f 0000 0000  ...........o....
0000080: 4945 4e44 ae42 6082 4e49 4e4a 4144 4f47  IEND.B`.NINJADOG
0000090: 4532 340a                                E24.
```

The encrypted file, skipping the tag t_i and DH key R_i:

```
0000000: 8d54 4a43 090e 1e0e 0404 0409 4d4c 4056  .TJC........ML@V
0000010: 0404 0405 0404 0405 0c06 0404 0494 7357  ..............sW
0000020: de00 0000 0970 4859 7300 0003 b100 0003  .....pHYs.......
0000030: b101 f583 ed49 0000 0007 7449 4d45 07df  .....I....tIME..
0000040: 0401 0319 3a3d ca0b 0c00 0000 0c69 5458  ....:=.......iTX
0000050: 7443 6f6d 6d65 6e74 0000 0000 00bc aeb2  tComment........
0000060: 9d04 0404 0b4d 4045 500c 1905 0004 fffb  .....M@EP.......
0000070: 04fb 0404 0705 0504 c302 966b 0404 0404  ...........k....
0000080: 4945 4e44 ae42 6082 4e49 4e4a 4144 4f47  IEND.B`.NINJADOG
0000090: 4532 340a                                E24.
```

Describe what you observe about how the symmetric encryption works. Remember that this is using a stream cipher.

Hint: Compare matching lines. In particular, take a look at the second lines. 6 points

(c) For some database the backup is incomplete and new entries had been added since the last backup. The database works by adding entries at the end of the file.

Based on your observations from part 6.b), describe how you can help in decrypting, assuming that the file on backup has some minimal length. Include a statement of the minimal length of the backup 5 points

(d) Some files are entirely new since the last backup. Describe a strategy of how you can try to recover the data and what essential parts you need to find.

5 points