TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculty of Mathematics and Computer Science
Introduction to Cryptology, Monday 23 January 2023

Name :

TU/e student number :

Exercise	1	2	3	4	5	6	7	total
points								

Notes: Please hand in this sheet at the end of the exam. You may keep the sheet with the exercises.
This exam consists of 7 exercises. You have from 13:30 – 16:30 to solve them. You can reach 100 points.
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 the explanation. 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

$$s_{i+6} = s_{i+4} + s_{i+3} + s_{i+2} + s_i.$$

   (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 why a factor is irreducible. **12 points**

   (c) For each of the factors of f compute the order. **9 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 check that all 2^6 states are covered. **10 points**

2. This exercise is about modes.

   Here is a schematic description of the Cipher Feedback (CFB) mode.

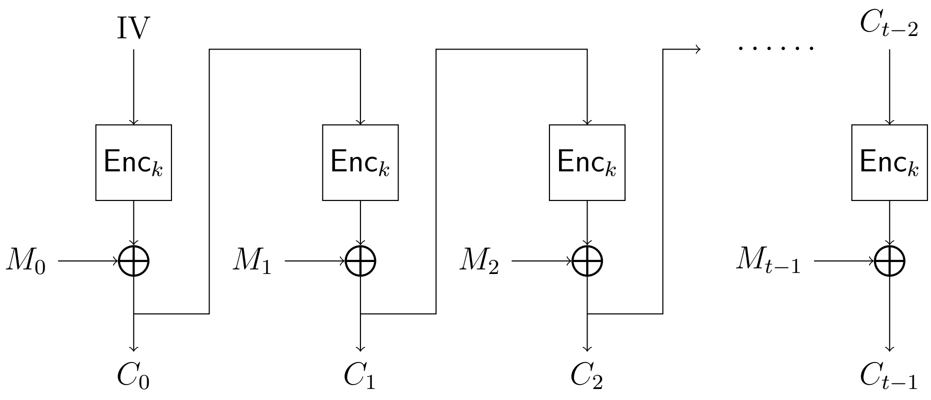


   Image credit: adapted from Jérémy Jean.

   Enc is an n-bit block cipher. Alice and Bob share a key k for Enc. Let $\mathsf{Enc}_k(m)$ denote encryption of a single block m using this block cipher with key k and let $\mathsf{Dec}_k(c)$ denote decryption of a single block c using the block cipher with key k. Let IV be the initialization vector of length n, let $M_i, i = 0, 1, 2, \ldots$ be the n-bit blocks holding the message and $C_i, i = 0, 1, 2, \ldots$ be the n-bit blocks holding the ciphertexts.

(a) Describe how encryption of long messages works by writing C_0 and a general C_i in terms of IV, M_0, M_i, and (if necessary) other M_j and C_j. Describe how decryption of long messages works by writing M_0 and a general M_i in terms of IV, C_0, C_i, and (if necessary) other M_j and C_j. 5 points

(b) Assume that ciphertext C_j gets modified in transit. Show which message blocks get decrypted incorrectly and explain why others get decrypted correctly. 5 points

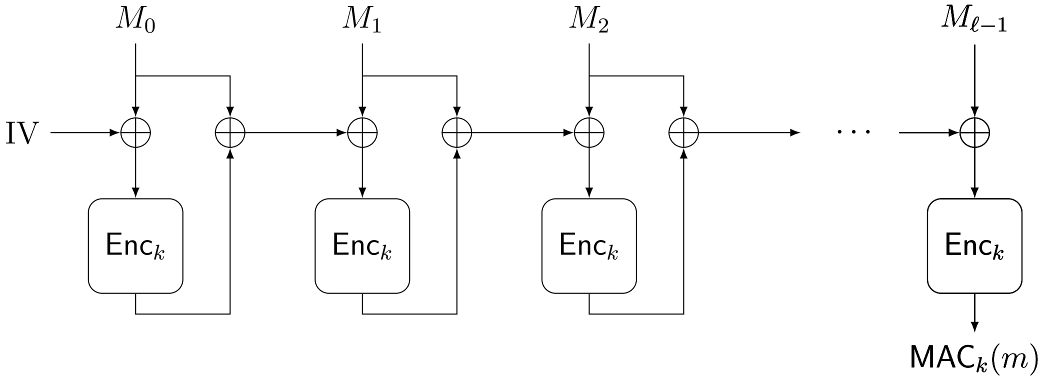
3. This problem is about RSA encryption. Let $p = 347$ and $q = 419$. Compute the public key using $e = 3$ and the corresponding private key. **Reminder:** The private exponent d is a positive number. 6 points

4. This problem is about the DH key exchange. The public parameters are the group G and generator g, where $G = (\mathbb{F}^*_{1009}, \cdot)$ and $g = 11$. Alice's public key is $h_A = 510$. Bob's private key is $b = 32$, Compute the DH key that Bob shares with Alice. 8 points

5. The integer $p = 23$ is prime. You are the eavesdropper and know that Alice and Bob use the Diffie-Hellman key-exchange in $\mathbb{F}^*_{23}$ with generator $g = 5$. Alice's public key is $h_A = g^a = 17$. Use the Baby-Step Giant-Step method to compute Alice's private key a.
Verify your result, i.e. compute g^a. 12 points

6. This exercise is about a message authentication code built on top of the PCBC mode.

(a) Enc is an n-bit block cipher. Alice and Bob share a key k for Enc. IV is an n-bit initialization vector chosen freely by the sender when generating $\mathsf{MAC}_k(m)$. Assume for simplicity that all messages have length a multiple of n. Let m split into t blocks of n bits as $m = (M_0, M_1, M_2, \ldots, M_{\ell-1})$.



This picture shows how $\mathsf{MAC}_k(m)$ is computed. The input are the key k and message $m = (M_0, M_1, M_2, \ldots, M_{\ell-1})$. The output are the chosen IV and $\mathsf{MAC}_k(m)$.

Describe in words and formulas how $\mathsf{MAC}_k(m)$ is computed for PCBC-MAC (shown on the picture above). 3 points

(b) PCBC-MAC as defined here is not a secure MAC.

Show how Eve can use a valid message-MAC pair

$$(m, t) = ((M_0, M_1, M_2, \ldots, M_{\ell-1}), (\mathrm{IV}, \mathsf{MAC}_k(m)))$$

that she obtains by recording a session between Alice and Bob. to compute a different valid message-MAC pair

$$(m', t') = ((M'_0, M'_1, M'_2, \ldots, M'_{\ell-1}), (\mathrm{IV}', \mathsf{MAC}_k(m')))$$

for M'_0 of her choice, without knowing k. I.e. describe how to choose $M'_1, M'_2, \ldots, M'_{\ell-1}, \mathrm{IV}', \mathsf{MAC}_k(m')$ and why this will pass verification by Alice and Bob. 8 points

7. Patrick observes that most people tend to send short messages and that there are problems with RSA if the message is too short. He thus designs a new padding scheme as follows:

   As always, the message m is assumed to be an integer in $[1, n-1]$. Let n have ℓ bits and m have $k \leq \ell$ bits. Assume that $\ell - k \geq 5$. To obtain the encoding M of m, m is written in binary, the next larger 4 bits are set to 0, the top bit is set to 0, and the remaining $\ell - k - 5$ bits are set to Hash(m), where Hash : $\{0,1\}^* \to \{0,1\}^{\ell-k-5}$.

   Hence

$M =$	0	$H_{\ell-k-6}$	$\cdots$	H_1	H_0	0	0	0	0	m_{k-1}	$\cdots$	m_1	m_0

   where Hash$(m) = (H_{\ell-k-6} \cdots H_1 H_0)$.

   Let M be the encoded message and consider it as an integer. Because the top bit is set to 0, M is in $[1, n-1]$. The integer M is then encrypted with RSA as usual as $c \equiv M^e \bmod n$ using the public key (n, e) of the receiver.

   The receiver decrypts c as usual, obtaining $M \equiv c^d \bmod n$ and then needs to figure out which part is m and which part is the padding. To do so, they first check that the top bit is 0, if not the message is invalid. Then they scan M starting from the bottom bit till they find 4 consecutive 0s. Then they check whether the top bits (apart from the top most) match the hash of those potential message bits. If so, they output m, if not they continue scanning further to the left for the next block of four consecutive 0s and repeat the procedure there. If they reach the left end of M without having output a valid message they declare that M is invalid.

   Note that seeing 6 consecutive 0s means possibly having to try each of the 3 possible starting positions for the run of four 0s until one works.

   (a) As a warm-up example, take $m = 23$ which has 5 bits and assume that n has 10 bits. Write the encoding M of m. **2 points**

   (b) Explain why this padding method works, i.e., why the encoding of a message m can be decoded correctly if m has no more than $\ell - 5$ bits. **6 points**

   (c) The decoding procedure with all the trial decodings is cumbersome and takes more time than users Sally and Rick like. They thus agree that Sally will send messages of exactly 8 bits so that Rick will only need to do a single trial for the decoding by taking as

m the bottom eight bits of M and checking that the remaining $\ell - 8 - 5$ bits match the hash of m.

This sure saves time and makes it much easier to implement the scheme and most of the time Sally's answer fits into a single character anyways.

Knowing Rick, Eve suspects such an arrangement. Find an attack on this restricted scheme and describe how Eve can determine m given c and how much effort this takes. You can assume that (n, e) are chosen to have cryptographic sizes, so n has at least 2048 bits. Remember that c is the encryption of M, so the message has full length. 8 points