TECHNISCHE UNIVERSITEIT EINDHOVEN
Faculty of Mathematics and Computer Science
Introduction to Cryptology, Monday 23 January 2017

Name :

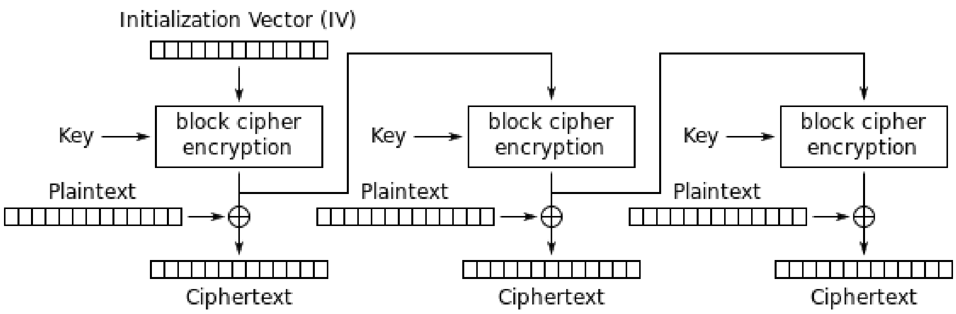
TU/e student number :

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 requires 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
$$s_{i+4} = s_{i+2} + s_{i+1} + s_i$$
   (a) Draw the LFSR corresponding this sequence. 2 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. 10 points
   (c) For each of the factors of f compute the order. 10 points
   (d) What is the longest period generated by this LFSR? Make sure to justify your answer. 4 points
   (e) State the lengths of all subsequences so that each state of n bits appears exactly once. Make sure to justify your answer. 8 points

2. This exercise is about modes. Here is a schematic description of the OFB (Output Feedback) mode.



Output Feedback (OFB) mode encryption

[Picture by White Timberwolf, public domain]

This encryption uses a block cipher of block size b. Let $\text{Enc}_k(m)$ denote encryption of a single block m using this block cipher with key k and let $\text{Dec}_k(c)$ denote decryption of a single block c using the block cipher with key k. Let IV be the initialization vector of length b, let m_i be the b-bit strings holding the message and c_i be the b-bit strings holding the ciphertexts.

(a) Describe how encryption and decryption of long messages work, i.e., write c_0, c_1, and a general c_i in terms of IV, m_0, m_1, m_i, and (if necessary) other m_j and c_j; and write m_0, m_1, and a general m_i in terms of IV, c_0, c_1, c_i, and (if necessary) other m_j and c_j. 6 points

(b) Ciphertexts are received with explicit sequence numbers (i, c_i). Assume that ciphertext c_j gets modified in transit. Show which messages get decrypted incorrectly. 4 points

3. This problem is about RSA encryption.

   (a) Alice's public key is $(n, e) = (14351, 5)$. Encrypt the message $m = 234$ to Alice using schoolbook RSA (no padding). 4 points

   (b) Let $p = 449$ and $q = 569$. Compute the public key using $e = 3$ 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 that the group is $(\mathbb{F}_{983}^*, \cdot)$ and that it is generated by $g = 5$.

   (a) Compute the public key belonging to the secret key $b = 20$. 4 points

   (b) Alice's public key is $h_a = 473$. Compute the shared DH key with Alice using b from the previous part. 6 points

5. The integer $p = 19$ is prime. You are the eavesdropper and know that Alice and Bob use the Diffie-Hellman key-exchange in $\mathbb{F}_{19}^*$ with generator $g = 3$. Alice's public key is $h_a = g^a = 10$. Use the Baby-Step Giant-Step method to compute Alice's private key a. Verify your result, i.e. compute g^a. 10 points

6. The poly-RSA cryptosystem is the polynomial analogue of the regular RSA cryptosystem. To generate a key, pick two irreducible polynomials $p(x), q(x) \in \mathbb{F}_2[x]$ and compute $N(x) = p(x) \cdot q(x)$.

   Let $\deg(p) = m, \deg(q) = n$. Select e with $\gcd(e, (2^m - 1) \cdot (2^n - 1)) = 1$ and compute $d \equiv e^{-1} \bmod (2^m - 1) \cdot (2^n - 1)$. The public key is $(e, N(x))$, the secret key is $(d, N(x))$.

   Messages are elements of $\mathbb{F}_2[x]$ of degree less than $m + n$. To encrypt $M(x)$ compute $C(x) \equiv M(x)^e \bmod N(x)$. To decrypt $C(x)$ compute $\bar{M}(x) \equiv C(x)^d \bmod N(x)$.

   (a) Explain why this scheme works, i.e., explain why $M(x) = \bar{M}(x)$. **Hint:** $\mathbb{F}_2[x]/p(x) \cong \mathbb{F}_{2^m}$ and $\mathbb{F}_2[x]/q(x) \cong \mathbb{F}_{2^n}$. 7 points

   (b) Explain why poly-RSA is not secure and show how to break it. 5 points

7. The ElGamal signature scheme works as follows. Let $G = \langle g \rangle$ be a group of order ℓ. User A picks a private key a and computes the matching public key $h_A = g^a$. To sign message m, A picks a random nonce k and computes $r = g^k$ and $s \equiv k^{-1}(r + \text{hash}(m)a) \bmod \ell$. The signature is (r, s).

   (a) Show how to compute a given m, r, s, and k. 6 points

   (b) Bob uses ElGamal signatures to authenticate his messages. He didn't pass the introduction to cryptology course and doesn't know how to generate random numbers, so he uses the same k for all messages. Show how to compute a given signatures (r, s_1) on m_1 and (r, s_2) on $m_2 \neq m_1$.
   Note: k and thus r are the same in both signatures. 6 points