

# Exam Security

18 January 2010, 8:30 – 10:30

You can score a maximum of 100 points. Each question indicates how many points it is worth. You may answer both in Dutch and in English. Please write clearly, and don't forget to put your name on each page.

1. **(10 points, 2 per item)** During an exam a student sitting next to you copies part of your answers. Say yes/no, with one line of explanation, for each of the following security goals, depending on whether they are relevant in this situation.

- (a) authenticity
- (b) integrity
- (c) confidentiality
- (d) availability
- (e) non-repudiation.

2. Consider the following block encryption mechanism (due to Karn), based on only a secure hash function  $h$  and bitwise XOR  $\oplus$ . Assume  $h$  produces hash values of length 256 bits. Assume a symmetric/shared key  $k$  of even length.

Let  $m$  be a plaintext message, of length 512 bits. Split  $m$  halfway in two parts  $m_1, m_2$ , both of length 256, so that  $m = m_1 | m_2$ , where  $|$  is concatenation. Split also the key  $k$  halfway in two parts  $k = k_1 | k_2$ .

Now define  $\{m\}_k = c_1 | c_2$  where

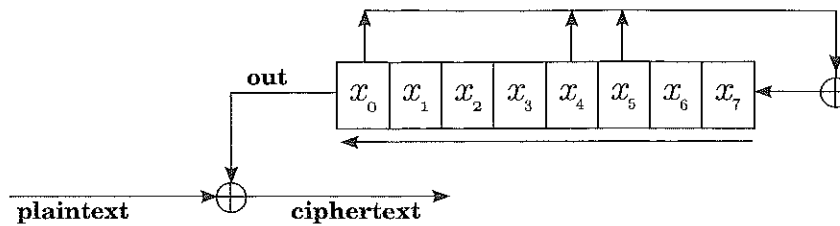
$$c_1 = m_1 \oplus h(m_2 | k_1) \quad \text{and} \quad c_2 = m_2 \oplus h(c_1 | k_2).$$

- (a) **(8 points)** Show how to decrypt: how to obtain  $m$  from  $\{m\}_k$ , assuming that you know  $k$ .
  - (b) **(8 points)** Suppose now you do *not* know the key  $k$  but you have intercepted cipher text  $\{m\}_k$ . Describe how one would proceed to break the encryption via brute force. You may assume that one can (automatically) recognise when a plaintext makes sense (for instance, when it is proper english).
3. Alice and Bob live together but work at different places. On the way home from work they try to decide who has to cook dinner, and use the following protocol.

$$\begin{aligned} A &\rightarrow B: h(n_A), \\ B &\rightarrow A: n_B, \\ A &\rightarrow B: n_A. \end{aligned}$$

where  $h$  is a hash function, and  $n_A, n_B$  are numbers chosen by Alice and Bob. If  $n_A + n_B$  is even, Alice has to cook, and Bob otherwise.

- (a) **(8 points)** Does the size of the chosen numbers affect the security of this protocol. Give short answers, both for Alice and for Bob.
  - (b) **(8 points)** Which security properties of the hash function  $h$  are relevant for this protocol? Give brief answers.
4. Consider the following simple Linear Feedback Shift Register (LFSR). The plaintext is bitwise XOR-ed with the output bits of the LFSR which first **computes**  $x_0 \oplus x_4 \oplus x_5$  and **then shifts** such that  $x_0$  falls out.



**Example:**

The state 

0	1	1	0	1	1	0	1
---	---	---	---	---	---	---	---

  
 is followed by 

1	1	0	1	1	0	1	0
---	---	---	---	---	---	---	---

  
 and outputs 

0
---

- (a) (8 points) Describe the next five states of the state in the above example box. The first successor state is already given as illustration, so you have to give the four subsequent ones.
  - (b) (8 points) Also do “rollback” and compute the *previous* 4 states, starting from the above example state.
  - (c) (8 points) Assume you know the cipher is in this example state, and you intercept 1111 as resulting ciphertext. Reconstruct the 4 bits of plaintext that produces this ciphertext 1111.
5. (ElGamal) Consider  $\mathbb{Z}_{19}^*$ , the multiplicative group of integers modulo 19 and a generator  $g = 2$  in this group.
- (a) (5 points) How many elements does the group  $\mathbb{Z}_{19}^*$  have? List them.
  - (b) (5 points) For which number  $n$  do we have  $2^n = 1 \pmod{19}$ ?
  - (c) (12 points) Compute ElGamal encryption of a message  $m = 3$  using the public key  $g^x = 5$  and randomness  $r = 3$ . Thus: compute the ciphertext  $(c_1, c_2) = \{3\}_5$ .  
 (The private key  $x$  is not given, since it is not necessary for encryption.)
  - (d) (12 points) It is well-known that the security of ElGamal is lost when the same randomness  $r$  is used twice. Imagine that Alice is silly enough to ignore this and sends the following ciphertexts:  $(c_1, c_2) = (10, 8)$  and  $(c'_1, c'_2) = (10, 13)$ . Exploit this mistake to recover the second plaintext message  $m'$ , knowing that the plaintext of the first message is  $m = 2$ .  
 (Hint: in order to do so you might need to use division modulo 19).