NANYANG TECHNOLOGICAL UNIVERSITY

SEMESTER I EXAMINATION 2017-2018

MH4311 – Cryptography

December 2017 TIME ALLOWED: 2 HOURS

INSTRUCTIONS TO CANDIDATES

1. This examination paper contains **FOUR (4)** questions and comprises **FOUR (4)** printed pages.
2. Answer all questions. The marks for each question are indicated at the beginning of each question.
3. Answer each question beginning on a **FRESH** page of the answer book.
4. This is a **RESTRICTED OPEN BOOK** exam. You are allowed to bring into the examination hall **ONE (1)** piece of A4-size paper written or printed on both sides.
5. Candidates may use calculators. However, they should write down systematically the steps in the workings.

Question 1. Hash function and MAC (20 marks)

(a) SHA-256 is applied to hash a message with length of 3000 bits. How many compression function operations are needed in the hashing?

(5 marks)

(b) HMAC-SHA-256 is applied to compute the authentication tag of a message with length of 3000 bits. How many compression function operations are needed?

(5 marks)

(c) At a website, each user's password P is hashed together with a salt S into a password image PI. The password images are stored at the website. Suppose that each salt is a 256-bit random number. The following algorithm is used to hash the password and the salt:

$$t1 = SHA\text{–}256(P) \oplus S;$$
$$t2 = SHA\text{–}256(t1) \oplus P;$$
$$t3 = SHA\text{–}256(t2) \oplus t1;$$
$$t4 = SHA\text{–}256(t3) \oplus t2;$$
$$PI = t3||t4;$$

Is this password hashing algorithm secure? Please justify your answer.

(10 marks)

Question 2. (15 marks)

(a) In AES, the irreducible polynomial with binary coefficients, $x^8 + x^4 + x^3 + x + 1$, is used to define $GF(2^8)$. Find the inverse of 5 in this field.

(10 marks)

(b) Suppose that you are required to implement AES to encrypt files on your computer. The encryption and decryption is provided by the user. When a user inputs the decryption key, your program should check whether the key is correct or not, and the decryption is performed only when the key is correct. Briefly explain how to implement it.

(5 marks)

Question 3. (20 marks)

(a) In a toy RSA encryption scheme, the public key (n, e), the private key is d. It is given that $n = 3149 = 47 \times 67$. You are required to generate a pair (e, d).

(10 marks)

Consider the following interpolation problem. Let

$$p(x) = a_{n-1}x^{n-1} + a_{n-2}x^{n-2} + \cdots + a_1 x + a_0$$

be a polynomial. The graph of the corresponding function $x \mapsto p(x)$ passes through the points $(x_0, y_0), (x_1, y_1), \ldots, (x_{n-1}, y_{n-1})$. Adam wrote the following Sage code to return the list $[a_{n-1}, \ldots, a_0]$ of coefficients of the polynomial $p(x)$. Here, `ylist` is the list $[y_0, y_1, \ldots, y_{n-1}]$ and `xlist` is the list $[x_0, x_1, \ldots, x_{n-1}]$.

```
def get_coeff(xlist, ylist):
    n = len(xlist)
    def f(i, j):
        return xlist[i]^j
    M = matrix(RDF, n, n, f)
    yvec = vector(RDF, ylist)
    return M.solve_right(yvec)
```

(i) Adam is getting an incorrect answer from `get_coeff` when he is trying to get the coefficients of a degree two polynomial which passes through the points $(1, 5), (2, 10), (3, 17)$. Find the error(s) in the function `get_coeff` due to which Adam is getting the incorrect answer. Give the corrections.

(ii) Is there any degree 2 polynomial $p(x)$ for which the *incorrect* function `get_coeff` would still give a correct solution? If so, then give an example of such a polynomial. If such a polynomial cannot be obtained, explain why this is the case.

Question 4. (20 marks)
Eve decided to compute a certain function using recursion. The function Eve wrote is the following:

```
def compute(a, b):
    if a < 0 or b < 0 or a < b:
        return 0
    if b == 0:
        return 1
    return a*compute(a-1, b-1)/b
```

(i) What mathematical function is Eve's function `compute()` evaluating?

(ii) Write a non-recursive version of Eve's function `compute()`.

Question 5. (20 marks)
Let π be a permutation of the set $I = \{0, \ldots, n-1\}$. The *orbits* of π on I are the equivalence classes of the binary relation $\equiv_\pi$ on I, so that $x \equiv_\pi y$ if and only if there exist $i \geq 0$ such that $\pi^i(x) = y$. Here π^i denotes the i-th iteration of π, i.e. $\pi^0(x) = x$, $\pi^1(x) = \pi(x)$, $\pi^2(x) = \pi(\pi(x))$, etc. Write a Sage function that takes π as a list of length n of numbers in I and returns the list of lengths of the orbits of π on I.

END OF PAPER