Mastermath
Spring 2019
Exam Selected Areas in Cryptology
Tuesday, 10 June 2019

Name :

Student number and home university :

Exercise	1	2	3	4	5	6	total
points							

Notes: Please 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 can reach 100 points.
You have from 10:00 – 13:00 to solve them.
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 paper provided by the university; 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 allowed to use any books and notes, e.g. your homework.
You are not allowed to use the textbooks of your colleagues.
You are allowed to use a calculator without networking abilities.
Usage of laptops and cell phones is forbidden.

1. This exercise is about code-based cryptography.

   (a) Code C is given by its parity check matrix

   $$H = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

   Compute a generator matrix for this code. 3 points

   (b) For a Goppa code with $m = 10$, n of maximal size, and degree of the irreducible polynomial $t = 50$, compute or give bounds for the length, dimension, and minimum distance. 4 points

2. Let H be the parity-check matrix of a code of length n, dimension k, and minimum distance $d = 2t + 1$. The school-book version of the Niederreiter system encrypts a message $m \in \mathbb{F}_2^n$ of Hamming weight t by computing the syndrome $s = H \cdot m$.

   You are given access to a decryption oracle. In the following two situations, show how to recover m and compute how many calls to the oracle are required.

   (a) The oracle decrypts any ciphertext $s' \neq s$ provided that $s' = H \cdot m'$ with m' of Hamming weight less than or equal to t. 5 points

   (b) The oracle decrypts any ciphertext $s' \neq s$ provided that $s' = H \cdot m'$ with m' of Hamming weight exactly equal to t. 10 points

3. This exercise is about the NTRU encryption system. Remember that all computations take place in $R = \mathbb{Z}[x]/(x^n - 1)$ and are done modulo 3 or modulo q. The secret key consists of $f(x), g(x) \in R$, where f is invertible in $R_q = R/q$ and R_3, and f has exactly d_f coefficients equal to 1 and $d_f - 1$ coefficients equal to -1 for some integer d_f. Similarly, g has d_g coefficients equal to 1 and the same number equal to -1. All other coefficients of f and g are 0. The public key is $h = 3g/f$ in R_q.

   To encrypt $m \in R$ with coefficients in $\{-1, 0, 1\}$ pick random $r \in R$ with d_r coefficients equal to -1, the same number equal to 1, and all others equal to 0. Then compute the ciphertext $c \equiv r \cdot h + m \bmod q$; move all coefficients to $(-q/2, q/2]$ to get a unique representative of c.

   To decrypt $c \in R_q$ compute $a = f \cdot c \bmod q$, again moving all coefficients to $(-q/2, q/2]$ (hence we use $=$ instead of $\equiv$) and compute $m = a/f \bmod 3$ with coefficients in $\{-1, 0, 1\}$.

   (a) Explain why decryption recovers m for sufficiently large choices of q and show how to choose q relative to d_f, d_g, and d_r to avoid decryption failures. You can assume that $d_r \leq d_g$. 6 points

(b) One tweak of NTRU is to use public key $h = g/F$ with $F = 1 + 3f$, where f is chosen to have d_f coefficients equal to 1 and the same number equal to -1. Explain how this simplifies the decryption procedure and compute lower bounds on q in terms of d_f, d_g, and d_r to avoid decryption failures. 10 points

4. This exercise is about hash-based signatures.

(a) Explain in your own words how the the Winternitz one-time-signature scheme works. 6 points

(b) A user accidentally uses his Winternitz signature key twice. Explain how an attacker can use these signatures to create a new signature. 6 points

5. This exercise is about differential cryptanalysis of the same toy cipher from the lectures. Using key $(k_1, k_2, k_3, k_4, k_5) \in (\{0,1\}^{16})^5$ it encrypts a plaintext $P = P_1||\ldots||P_{16} \in \{0,1\}^{16}$ as follows. Let S be the current state, we start with $S = P$. Rounds $i = 1, 2, 3$ perform key mixing

$$S \leftarrow S \oplus k_i,$$

substitution using a Sbox (Table 2)

$$S \leftarrow Sbox(S_1 \ldots S_4)||\ldots||Sbox(S_{12} \ldots S_{16}),$$

and finally applies permutation π_P (Table 1) on the state bits:

$$S \leftarrow S_{\pi_P(1)}||\ldots||S_{\pi_P(16)} = S_1||S_5||S_9||\ldots||S_{12}||S_{16}.$$

Round 4 applies key mixing with round key k_4, substitution using the sbox and finally applies another key mixing with round key k_5. After round 4, the cipher outputs the current state S as the ciphertext C.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\pi_P(i)$	1	5	9	13	2	6	10	14	3	7	11	15	4	8	12	16

Table 1: State bit permutation

In contrast to the lecture notes, we use the following SBox:

in	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
out	4	2	14	8	10	12	7	1	15	5	0	11	9	3	6	13

Note most significant bit is left most bit, so 12 represents '1100' in binary.

Table 2: Sbox

This SBox has the following Difference Distribution Table (Table 3:

		Δout															
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
	0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	0	8	0	0	0	4	4	0	0	0	0
	2	0	0	0	0	0	0	0	0	0	0	4	0	0	4	4	4
	3	0	0	0	0	4	4	0	0	0	0	0	4	4	0	0	0
	4	0	0	0	0	0	0	8	0	0	4	0	0	0	0	4	0
	5	0	0	0	0	0	0	0	0	4	0	0	0	4	4	0	4
	6	0	0	0	4	4	0	0	0	4	4	0	0	0	0	0	0
	7	0	0	8	4	0	4	0	0	0	0	0	0	0	0	0	0
Δin	8	0	2	0	4	0	0	0	2	0	0	0	2	2	0	2	2
	9	0	2	0	0	0	4	0	2	2	2	2	0	0	2	0	0
	10	0	4	2	0	2	0	0	0	0	2	0	0	2	2	2	0
	11	0	0	2	0	2	0	0	4	2	0	2	2	0	0	0	2
	12	0	2	0	0	0	4	0	2	2	2	2	0	0	2	0	0
	13	0	2	0	4	0	0	0	2	0	0	0	2	2	0	2	2
	14	0	0	2	0	2	0	0	4	2	0	2	2	0	0	0	2
	15	0	4	2	0	2	0	0	0	0	2	0	0	2	2	2	0

Table 3: Sbox difference distribution table

(a) Complete the DDT. You only have to write down the missing numbers in a table. (Hint: to fill a column: fix Δout; iterate over *out* instead of *in*.) 8 points

(b) Construct a differential trail for this cipher over the first three rounds with only one active SBox in the second round and compute its estimated probability. 10 points

(c) Consider the boomerang with input plaintext difference

$$\Delta P = (0000\ 0100\ 0000\ 0000)$$

and output ciphertext difference

$$\Delta C = (0000\ 0000\ 0110\ 0000),$$

then a quartet $(P^{(1)}, P^{(2)}, P^{(3)}, P^{(4)})$ satisfies this boomerang if

$$P^{(1)} \oplus P^{(2)} = \Delta P, \quad P^{(3)} \oplus P^{(4)} = \Delta P, \quad \text{and}$$

$$C^{(1)} \oplus C^{(3)} = \Delta C, \quad C^{(2)} \oplus C^{(4)} = \Delta C.$$

Compute the total success probability of finding such quartets over all round 1 & 2 differentials with the given ΔP and all round 3 & 4 differentials with the given ΔC, i.e., compute

$$p_{success} = \left(\sum_{(\Delta P, \Delta O_1, \Delta O_2)} \Pr[(\Delta P, \Delta O_1, \Delta O_2)]^2 \right) \cdot \left(\sum_{(\Delta O_2, \Delta O_3, \Delta C)} \Pr[(\Delta O_2, \Delta O_3, \Delta C)]^2 \right)$$

(Hint: use the fact that in round 2 each Sbox has either input difference 0 or 4 (0100), so every *active* round 2 Sbox contributes a term

$$\sum_{(\Delta In = 4, \Delta Out \in \{0,\ldots,15\})} \Pr[(\Delta In, \Delta Out)]^2 = 2 \times (4/16)^2 + 1 \times (8/16)^2.$$

Likewise, in round 3 each active Sbox has output difference 2 (0010).
) $\boxed{\text{10 points}}$

6. This exercise is about applying generic cryptanalytic algorithms. Let $h : \{0,1\}^* \to \{0,1\}^{128}$ be a 128-bit hash function. A website stores for each user a username string u and a 128-bit hash $a = h(p)$ of the user's password string p, it only allows numeric passwords (i.e., '0...9') of length 13.

   (a) Explain how to apply Hellman's time-memory trade-off attack to h to recover passwords from the given password space $\mathcal{P}$. $\boxed{\text{10 points}}$

   (b) What is the memory complexity and online time complexity? (expressed in Bytes and in evaluations of h, respectively). $\boxed{\text{4 points}}$

   (c) Hellman's attack allows preimage attacks against a known small preimage set like passwords. However it turns out that a generic cryptanalytic attack can be used against h with complexity significantly lower than the desired 2^{128}. This is because h is a Merkle-Damgard construction with a secure blockcipher $E(K, P)$ used as a compression function *without using the Davies-Meyer feedforward*. That is, for a message M that is padded and split into blocks $M_1, \ldots, M_r$, its hash $h(M)$ is computed as:

   $$CV_0 = IV, \quad CV_i = E(M_i, CV_{i-1}), \quad h(M) = CV_r.$$

   Explain how to compute a preimage significantly faster than 2^{128} evaluations. (Hint: given a hash s, consider a message consisting of two blocks M_1, M_2 and use $CV_2 = s$.) $\boxed{\text{8 points}}$