Mastermath
Spring 2017
Exam Cryptology Course
Tuesday, 04 July 2017

Name :

Student number :

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 have from 14:00 – 17:00 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 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) State the parameters (length, dimension, minimum distance) of a binary Goppa code with $m = 14$, i.e. length $n = 2^{14}$, using an irreducible polynomial of degree 52.
   Make sure to state inequalities where appropriate. $\boxed{\text{3 points}}$

2. This exercise is about hash-based signatures.

   (a) Explain in your own words how the Lamport one-time signature scheme works to sign one bit. State the public key, the private key and why the system is secure. $\boxed{\text{4 points}}$

   (b) Explain in your own words how to extend Lamport's one-time signature scheme to sign multiple messages of length m bits using Merkle trees. State the public key, the private key and why the system is secure. $\boxed{\text{6 points}}$

   (c) We use Winternitz' scheme with hash chains of length 2^8, i.e., we process 8 bits at once. Compute the size (in bits) of the public key, the private key, and the signature for this scheme, assuming that the iteration function F and the hash function H have output length of 256 bits.
   Hint: Remember that you also need to sign the second component.
   $\boxed{\text{8 points}}$

3. 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 $P = P_1||\dots||P_{16} \in \{0,1\}^{16}$ 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)

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

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

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

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

Note <u>most significant</u> bit is <u>left most</u> 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																
	1																
	2																
	3				0	0	4	0	4	0	0	0	0	0	0	4	4
	4				4	0	0	0	0	0	4	0	4	0	4	0	0
	5				0	4	0	6	2	2	2	0	0	0	0	0	0
	6				8	0	0	2	2	2	2	0	0	0	0	0	0
	7				4	0	0	0	0	0	0	0	0	8	0	4	0
Δin	8				0	0	0	0	2	4	0	2	0	2	0	0	0
	9				0	0	2	0	4	2	0	0	0	2	4	0	0
	10				0	0	0	2	0	2	2	2	0	0	0	0	2
	11				0	0	2	2	2	0	2	0	0	0	4	0	2
	12				0	0	2	0	0	0	2	2	2	2	0	0	0
	13				0	2	2	2	0	0	0	2	0	2	0	4	0
	14				0	2	0	0	0	0	2	0	4	0	0	0	2
	15				0	4	0	2	0	0	0	0	2	0	0	4	2

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 and iterate over out instead of in.) 6 points

(b) Construct a differential for this cipher over the first three rounds with only one active SBox in the 2nd round and compute its estimated probability. 8 points

(c) Consider the boomerang with input plaintext difference

$$\Delta P = (0000\ 0111\ 0000\ 0000)$$

and output ciphertext difference

$$\Delta C = (0000\ 0000\ 0011\ 0000),$$

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. (Hint: in round 2 each Sbox has either input difference 0 or 4 (0100), so every *active* round 2 Sbox contributes a term $4 \times (4/16)^2$. Likewise, in round 3 each active Sbox has output difference 2 (0010).) 8 points

(d) Determine an example impossible differential by limiting which round 1 Sboxes and which round 3 Sboxes may be active, show why this is the case. 8 points

4. 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 p or modulo q. The secret key is $f(x) \in R$, $f \cdot f_p = 1$ in R/p, $f \cdot f_q = 1$ in R/q, $h = pf_q \cdot g$ in R/q, and $c = rh + m$ in R/q for random r and message m. (We used $p = 3$ in the lecture and also below).

   (a) Let $N = 4, p = 3$, and $f(x) = x^2 - x - 1$. Compute the inverse f_p of f in R/p and compute $f \cdot f_p$ in R/p to verify that the result is indeed 1.
   Hint: this needs a XGCD computation. Make sure to document the steps or state how you did this computation. Do *not* simply state the result or just a verification of the result. 10 points

   (b) Explain how to attack NTRU using an algorithm to find short lattice vectors, i.e., explain how to translate the problem of finding the secret key into a problem of finding short lattice vectors. Make sure to state the lattice involved. 8 points

5. This exercise is about password recovery. Let $h : \{0,1\}^* \to \{0,1\}^{512}$ be a fixed 512-bit hash function. A website stores for each user a username string u and a 512-bit hash $a = h(p)$ of the user's password string p. Let $\mathcal{P}$ be the set of all numeric (i.e., '0...9') passwords of length 15. The size of the set $\mathcal{P}$ is $10^{15} \approx 2^{49.82}$.

   (a) Explain how one can construct an efficient map $f : \{0,1\}^{512} \to \mathcal{P}$, computing a password from each possible hash. It has to be approximately balanced, i.e., preimage sizes have to be approximately equal: $|f^{-1}(p_1)| \approx |f^{-1}(p_2)|$ for all $p_1, p_2 \in \mathcal{P}$. 4 points

   (b) Hellman's time-memory trade-off attack uses multiple tables that require distinct reduction functions. Explain how to create $N < |\mathcal{P}|$ distinct reduction functions from f at very low computational cost. 4 points

(c) Explain how to apply Hellman's time-memory trade-off attack to h to recover passwords from the given password space $\mathcal{P}$ with success probability about 0.8. (Specify the following quantities: number of tables, number of trails, the length of each trail, and the offline and online complexity.)

6 points

(d) Assume an attacker can use a single high-end GPU for this attack that can compute 2^{30} evaluations of $f \circ h$ per second. Estimate the offline and online runtime complexity in wall clock time (days, hours, seconds) for this attack using this single high-end GPU as well as the storage requirements. Disregard the effect from 'false alarms' and assume RAM and GPU memory size are not an issue.

6 points

6. This exercise is about attacks on code-based cryptography. Let G be the generator matrix of an $[n, k, d]$ code with $d = 2t+1$. In the basic schoolbook-version of McEliece encryption, a message $m \in \mathbb{F}_2^k$ is encrypted by computing $y = mG + e$, where $e \in \mathbb{F}_2^n$ is randomly chosen of weight t.

   Alice and Bob use this method to send m but Eve intercepts $y_1 = mG + e_1$ and stops the transmission. After a while, Alice resends an encryption of m, using a different error vector e_2, so $y_2 = mG + e_2$, where both e_i have weight t.

   (a) Compute the average weight of $e_1 + e_2$, where $+$ denotes addition in $\mathbb{F}_2^n$, and the average weight of $e_1 \cdot e_2$, where $\cdot$ denotes componentwise multiplication in $\mathbb{F}_2^n$.

   5 points

   (b) Show how Eve can recover the message m.
   Hint 1: Eve's task should be stated as a decoding problem of a code of length less than n.
   Hint 2: First solve the problem assuming that e_1 and e_2 have no overlap in their non-zero positions.

   6 points