Mastermath
Spring 2015
Exam Cryptology Course
Tuesday, 30 June 2015

Name :

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 sheets with the exercises.
This exam consists of 7 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 $m = 15$ of length $n = 2^{15}$ using an irreducible polynomial of degree $s = 40$. 3 points

2. This exercise is about attacks on code-based cryptography.

   Leon's algorithm finds low-weight codewords. Assume for concreteness that the code contains a word of weight t and assume for simplicity that there is only one word c of weight t.

   The outer loop randomly permutes the columns of the parity-check matrix H and turns the rightmost $n-k$ columns into an $(n-k) \times (n-k)$ identity matrix (if these columns are not linearly independent another permutation is tried). The resulting matrix looks like $(H^*|I_{n-k})$. The algorithm also fixes a set $Z \subset \{1, 2, \ldots, n-k\}$ of size z. For concreteness assume that $Z = \{1, 2, \ldots, z\}$.

   The inner loop picks p of the remaining k columns and computes the sum of these p columns on the z positions indexed by Z. If this vector of length z is the all-zero vector then compute the sum of these p columns on all $n-k$ positions. The algorithm succeeds if the resulting vector has weight $t - p$.

   (a) Explain how to obtain the word c of weight t from the steps described above, i.e., assume that you have found p columns so that their sum has the top z positions equal to zero and so that is has weight $t - p$.
   Explain why it makes sense to first check the sum on just z positions and to compute the full length-$n-k$ column only if those positions are all zero. 7 points

   (b) Compute the probability that one choice of column permutation distributes the positions of c in such a way that the inner loop of Leon's algorithm can be successful. Remember that Leon checks the sum at full length only once the top z positions are all zero. Make sure to justify this answer, a product of binomial coefficients is not sufficient. 8 points

3. This exercise is about hash-based signatures. Let $H : \{0,1\}^* \to \{0,1\}^{256}$ be a cryptographic hash function. We use H as the hash function inside the Lamport and the Winternitz scheme and also as the hash function to compress messages before signing. For the random elements in the secret key you should assume that they each need 256 bits.

   (a) We use Lamport's one-time signature together with Merkle's tree construction. To sign $H(m)$ of length 256 we need to have a tree with 256 leaves. Compute the size (in bits) of the public key, the private key, and the the signature for this scheme. How many hash function computations are needed in signing and how many in verifying? 5 points

(b) We use Winternitz' scheme with parameter $k = 5$, i.e., we process 5 bits at once to sign $H(m)$ of length 256. Compute the size (in bits) of the public key, the private key, and the signature for this scheme.
Hint: Remember that you also need to sign the σ component.
How many hash function computations are needed in signing and how many in verifying? 6 points

(c) Compare the two answers above to using the Winternitz scheme with parameter $k = 8$ (also for $H(m)$ of 256 bits) in terms of the size of keys and signature and also in terms of how many times you need to evaluate H. 7 points

4. 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.

(a) Let $\mathcal{P}$ be the set of all alphanumeric (i.e., 'a...z,A...Z,0...9') passwords of length 9. Compute the size of the set $\mathcal{P}$. 2 points

(b) Explain how one can construct an efficient map $f : \{0,1\}^{512} \to \mathcal{P}$ from the hash space to the password space. It has to be balanced, i.e., preimage sizes have to be approximately equal: $|f^{-1}(p_1)| \approx |f^{-1}(p_2)|$ for all $p_1, p_2 \in \mathcal{P}$. 6 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. 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

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	A	B	C	D	E	F
out	0	3	5	8	6	C	B	7	A	4	9	E	F	1	2	D

Note most significant bit is left most bit and using hexadecimal notation.
So 'C' represents number 12 or '1100' in binary.

Table 2: Sbox

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

		out														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0																
	1																
	2																
	3		2	2	2	2	0	2	2	2	0	0	0	0	2	0	0
	4		0	0	2	0	4	2	0	0	0	0	2	0	0	2	4
	5		0	2	2	2	2	0	0	0	0	0	4	4	0	0	0
	6		0	0	2	4	0	2	0	2	2	0	2	0	0	0	2
	7		2	0	0	0	0	2	4	4	2	0	0	0	0	2	0
in	8		0	0	0	0	0	2	2	0	4	4	0	2	2	0	0
	9		2	0	2	2	2	2	2	0	2	0	2	0	0	0	0
	A		2	0	0	4	0	2	0	0	2	0	0	2	2	0	2
	B		2	2	0	0	0	0	0	2	0	4	2	0	0	4	0
	C		0	4	0	0	2	0	2	2	2	0	0	2	0	0	2
	D		2	2	0	0	2	2	0	2	0	2	0	2	0	2	0
	E		2	4	2	0	0	0	0	0	2	2	0	0	0	2	2
	F		2	0	0	2	2	0	2	2	0	0	0	0	4	0	2

Table 3: Sbox difference distribution table

(a) Complete the DDT. You only have to write down the missing numbers in a table. [6 points]

(b) Construct a differential for this cipher over the first three rounds with only one active SBox in the third round and compute its estimated probability. [6 points]

(c) Consider all 3-round differentials that have only 1 active Sbox in round 1 and only one active Sbox in round 3. Prove that all such 3-round differentials are impossible differentials. (Hint: the input difference of active round 2 sboxes is uniquely determined by which round 1 Sbox is active.) [8 points]

6. 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, $h = f_q \cdot g$ in R/q, and $c = p\phi h + m$ in R/q for random ϕ and message m.

(a) Let $N = 5, p = 2$, and $f(x) = x^3 - x + 1$. 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. [6 points]

(b) Let $p = 3, N = 503$, and $q = 256$. The encryption equation $c = p\phi h + m$ in R/q is the schoolbook version of NTRU and is not CCA-II secure. Show how you can use an oracle that decrypts any ciphertext but c to find m. Note that this has two parts: stating candidate ciphertexts c' to feed to the oracle and a verification whether the obtained message M matches the actual message m. [8 points]

7. This exercise is about differential cryptanalysis of the same toy cipher of question 5.

(a) Argue that extending the toy cipher from 4 rounds (using 3 normal rounds and 1 final round and a total of 5 round keys) to 7 rounds (using 6 normal rounds and 1 final round and a total of 8 round keys) is sufficient to ensure that the differential key recovery attack from the lecture notes does not work. That is, argue that there exist no 6-round differential with success probability strictly higher than 2^{-16}. (Hint: use the fact from 5(c) to argue a minimum number of active Sboxes over 6 rounds in combination with the DDT.) [10 points]