CHALMERS — GÖTEBORGS UNIVERSITET

Cryptography

TDA352 (Chalmers) - DIT250 (GU)

12 Jan. 2017, 14:00 - 18:00

No extra material is allowed during the exam except for pens and a simple calculator (not smartphones). *No other electronic devices allowed.* Your answers in the exam must be written in *English.* Your language skills will not be graded (but of course we cannot grade your answer if we do not understand it), so try to give *clear answers.* Your thoughts and ways of reasoning must be clearly understood!

Teacher: Elena Pagnin
Examiner: Aikaterini Mitrokotsa
Questions during the exam: Elena Pagnin (phone 072 9681552)
Inspection of exam: See web page for announcement.

The exam has 4 *topics* and *some bonus questions* to gain extra points.
The total number of points is 100 points *(+ 8 bonus points).*
Grades are :

CTH Grades:	50-64 → 3,	65-89 → 4,	90 or above → 5
GU Grades:	50-89 → G,	90 or above → VG	

Good luck!

Symmetric Ciphers (20p)

1. Consider the message m = `HKPUFCMHY BHDDXZH`, and let $(\mathbf{E}, \mathbf{D})$ be a substitution cipher.

   (a) Decrypt m using the following (secret) substitution key: **(2p)**

plain	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
cipher	X	G	P	Y	H	Q	Z	I	R	A	J	S	B	K	T	C	L	U	D	M	V	E	N	W	F	O

   Using the given substitution key we see that the decryption of m is: `encrypted message`. (2p)

   (b) Can this cipher be broken by someone who has access to m but not to the secret key? Why? **(3p)**

   No (1p), the ciphertext is too short. (1p) Any message matching the following pattern is a possible plaintext: (1p)

   $$\boxed{?_1}?_2?_3?_4?_5?_6?_7\boxed{?_1}?_8 \quad ?_9\boxed{?_1}\boxed{?_{10}}\boxed{?_{10}}?_{11}?_{12}\boxed{?_1}$$

   Other possible decryptions are:
   `rightward broomer`, `embroaden lettuce`, `equilobed terrace`.

2. Let (**E**, **D**) be a (one-time) semantically secure cipher, where the messages, ciphertexts and keys are binary strings, e.g., you can think $\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0,1\}^n$, with $n \geq 2$. Are the following encryption schemes, derived from (**E**, **D**), semantically secure or not? Explain why (no need for formal proofs, but your motivations should be well-justified).
   Why do we require $n \geq 2$? Would $n = 1$ provide different answers? (1 bonus point)

   (a) $\mathbf{E}'(k, m) = \mathbf{E}(k, m) \oplus \mathbf{1}$, where $\mathbf{1}$ denotes the string with all ones. **(2p)**

   (b) $\mathbf{E}'(k, m) = \mathbf{E}(k, m)||RB(m)$, where $RB(m)$ gives back a random bit of the input m. **(2p)**

   (c) $\mathbf{E}'(k, m) = \mathbf{E}(k, m)||RB(k)$, where $RB(k)$ gives back a random bit of the input k. **(2p)**

   (a) The encryption scheme is semantically secure. The ciphertexts of $\mathbf{E}'$ are simply the ciphertexts of $\mathbf{E}$ flipped. Thus, it is easy to see that any attack against $\mathbf{E}'$ can be turn into an attack against $\mathbf{E}$ (which is semantically secure by assumption). (2p)

   (b) The encryption scheme is not semantically secure. The adversary could use the folowing strategy to win the semantic security game. Choose $m_0 = \mathbf{0}$ (the all-0 message) and $m_1 = \mathbf{1}$ (the all-1 message). Return $RB(m)$ as b', the guess for the b chosen by the adversary to produce the challenge ciphertext. (2p)

   (c) The encryption scheme is semantically secure. Although $\mathbf{E}'$ is leaking one bit of the key, it does not harm (given that the key is longer than 1 bit, $n \geq 2$ (1p)), since a *randomly* chosen keys, will have both 0 and 1 bits. Thus, the adversary cannot retrieve any useful information from $RB(k)$. (2p)

3. Does the OTP (One Time Pad) cipher achieve perfect secrecy? Prove it. **(9p)**
   (Hint: you can start by quickly describing how the OTP cipher works and how perfect secrecy is defined).

Yes, the OTP cipher is perfect secret.

The OTP Cipher (**E**,**D**) over the message-space $\mathcal{M} = \{0,1\}^n$, the keys-pace $\mathcal{K} = \{0,1\}^n$ and the ciphertext-space $\mathcal{C} = \{0,1\}^n$ (where n is a positive integer), is defined as follows. For every key $k \xleftarrow{R} \mathcal{K}$: $\mathbf{E}(k,\cdot) : \mathcal{M} \longrightarrow \mathcal{C}$, $\mathbf{E}(k,m) = m \oplus k$, for any $m \in \mathcal{M}$. $\mathbf{D}(k,\cdot) : \mathcal{C} \longrightarrow \mathcal{M}$, $\mathbf{D}(k,c) = c \oplus k$, for any $m \in \mathcal{M}$. (2p)

The notion of perfect secrecy states that, for all $m_0, m_1 \in \mathcal{M}$ such that $len(m_0) = len(m_1)$, and for all $c \in \mathcal{C}$ (1p) it holds that:

$$\mathsf{Prob}[\mathbf{E}(k,m_0) = c] = \mathsf{Prob}[\mathbf{E}(k,m_1) = c]$$

where k is chosen uniformly at random from $\mathcal{K}$ ($k \xleftarrow{R} \mathcal{K}$). (2p)

In order to prove that the OTP cipher has perfect secrecy we need to show that $\mathsf{Prob}[\mathbf{E}(k,m) = c]$ is a constant value (the same for all possible messages $m \in \mathcal{M}$). (1p) Now, $\mathsf{Prob}[\mathbf{E}(k,m) = c]$ is the probability that there exists a key $k \in \mathcal{K}$ that encrypts m in the ciphertext c. For how the OTP encryption is defined, there is only one key that encrypts m in c, namely $k = m \oplus c$. Therefore,

$$\begin{aligned} \mathsf{Prob}[\mathbf{E}(k,m) = c] &= \frac{\text{number of keys that encrypt } m \text{ into } c}{\text{number of total keys}} \quad (1p) \\ &= \frac{|\, k \in \mathcal{K} : \mathbf{E}(k,m) = c \,|}{|\, \mathcal{K} \,|} \\ &= \frac{1}{|\, \mathcal{K} \,|} = \frac{1}{2^n} \quad (1p) \end{aligned}$$

By replacing m with m_0 and m_1 we get that:

$$\mathsf{Prob}[\mathbf{E}(k,m_0) = c] = \mathsf{Prob}[\exists\, k \in \mathcal{K} : k \oplus m_0 = c] = \frac{1}{|\, \mathcal{K} \,|} = \frac{1}{2^n} = \mathsf{Prob}[\mathbf{E}(k,m_1) = c].$$

Therefore OTP has perfect secrecy.(2p)

Public Key Encryption (30p)

4. Describe the ElGamal encryption scheme. **(6p)**
(Hint: write down input, output and behaviour of the algorithms).

The ElGamal encryption scheme is made by the three algorithm (**KeyGen**, **Enc**, **Dec**), defined as follows.
KeyGen$(\lambda) \rightarrow (\mathbf{pk}, \mathbf{sk})$, is the key generation algorithm. It takes as input the security parameter λ and generates the description of a cyclic group $G =< g >$ of order q (where q is a λ-bit long integer). Then it chooses a random value $x \xleftarrow{R} \{1, 2, \ldots q-1\}$ and computes $h = g^x \in G$. The output is the secret key $\mathbf{sk} = x$ and the public key $\mathbf{pk} = (G, g, q, h)$. (2p)
Enc$(\mathbf{pk}, m) \rightarrow c = (c_1, c_2)$, is the encryption algorithm. It takes as input the public key and a message $m \in G$ and outputs the ciphertext $c \in G \times G$. As a first step, the algorithm generates a random value $r \xleftarrow{R} \{1, 2, \ldots, q-1\}$ and computes $c_1 = g^r \in G$. Then it computes $c_2 = mh^r \in G$. (2p)
Dec$(\mathbf{sk}, c) \rightarrow m$ is the encryption algorithm. It takes as input the secret key and a ciphertext $c \in G \times G$. The decryption works as follows. First it computes $k = c_1^x \in G$. Then it computes $m = c_2 k^{-1} \in G$. (2p)

5. Define the IND-CCA security game (indistinguishability chosen ciphertext attack) and show that the ElGamal encryption scheme is not secure under IND-CCA. **(11p)**

In the IND-CCA security game, the adversary is given the public key of the scheme,

and he can ask for (at most $q \sim poly(\lambda)$) decryptions of messages of his choice to the challenger. After that, the adversary chooses two messages m_0, m_1 (of the same length) and sends them to the challenger. The challenger selects a random bit $b \xleftarrow{R} \{0,1\}$ and returns to the attacker the ciphertext $c \leftarrow \mathbf{Enc}(\mathbf{pk}, m_b)$. The adversary then can go through another query pahse, where anyhow he is not allowed to submit the challenge ciphertext c. The adversary wins the game if he can determine with non-negligible probability if c is an encryption of m_0 or of m_1 (i.e., guess the bit b chosen by the Challenger). (4p)
In what follows, we show a possible strategy for the attacker to win the game (there are several variants of this attack that are valid). The attacker can skip the first query phase, and choose two random messages $m_0, m_1 \in G$ such that $m_0 \neq m_1$, and send them to the Challenger. Let c denote the received challenge ciphertext. During the second query phase the Adversary can submit a modification of the challenge ciphertext, e.g., $c' = (c_1, d \cdot c_2)$, where $d \neq 1$ is an (of course invertible) element in G. The plaintext m' returned by the Challenger will then be $m' = d \cdot (c_2 \cdot c_1^{-x}) = d \cdot m_b$. Therefore the attacker can output the guess $b' = 0$ for b in case $d^{-1} \cdot m' = m_0$, and $b' = 1$ otherwise. (5p)
Following the above strategy, we can show that the adversary has a non-negligible advantage in the IND-CCA game. Let W_i denote the probability of the event ẗhe Challenger selects $b = i$ (i.e., c is an encryption of m_i, $i \in \{0,1\}$), and the adversary outputs as a guess $b' = 0$ ". Then, $Adv_{IND-CCA}[\mathcal{A}, ElGamal] = |W_0 - W_1| = |1 - 0| = 1$, which is indeed non-negligibly larger than $\frac{1}{2}$. (2p)

6. Consider the cyclic group $\mathbb{Z}_{37}^*$. *If you explain in details the functions / theorems / theory involved in this exercise you can gain a maximum of (3 bonus points)*

   (a) How many elements are in $\mathbb{Z}_{37}^*$, i.e., what is the order of the group? **(2p)**

   The order of $\mathbb{Z}_{37}^*$ is $\varphi(37)$, where φ denotes Euler's Phi (totient) function. (1p)
   Since 37 is a prime number, we have $\varphi(37) = 37 - 1 = 36$. (1p)

   (b) Is $\mathbb{Z}_{37}^*$ a cyclic group? How many generators does it have? **(4p)**

   Yes, $\mathbb{Z}_{36}^*$ is a cyclic group because 37 is a prime number. (1p) The number of generators corresponds to the number of coprime numbers between 1 and 36, that is $\mathbb{Z}_{36}^*$ has $\varphi(\varphi(37))$ generators (1p), more explicitly $\varphi(36) = \varphi(2^2 \cdot 3^2) = 2^{2-1}(2-1) \cdot 3^{2-1}(3-1) = 12$ generators. (2p)

   (c) Is 4 a generator of $\mathbb{Z}_{37}^*$? Prove it. **(7p)**
   An element $g \in \mathbb{Z}_{37}^*$ is a generator of the group if $g^{36} = 1$ in $\mathbb{Z}_{37}^*$ and $g^i \neq 1$ for all $i \in \{1, 2, \ldots, 35\}$. Therefore, in order to check if 4 is a generator we need to see if $4^i \neq 1$ in $\mathbb{Z}_{37}^*$ for all $i \in \{1, 2, \ldots, 35\}$. By Lagrange theorem we know that the order of an element always divides the order of the group. Since $\mathbb{Z}_{37}^*$ has order $36 = 2^2 \cdot 3^2$ we only need to check the powers $1, 2, 3, 4, 6, 9, 12, 18$ (proper divisors of 36). (2p)
   $4^1 = 4 \neq 1 \mod 37$, $4^2 = 16 \neq \pm 1 \mod 37$ (thus 4 does not have order 2 or 4), $4^3 = -10 \neq \pm 1 \mod 37$ (thus 4 does not have order 3 or 6).
   $4^9 = (4^3)^3 = -1000 + 37 * 27 = -1 \mod 37$ from which we can derive that 4 has order $2 * 9 = 18$ (4p) and thus 4 is not a generator of $\mathbb{Z}_{37}^*$. (1p)

Data Integrity (20p)

7. Describe the RSA digital signature scheme. **(10p)**
(Hint: write down input, output and behaviour of the algorithms).

The RSA digital signature scheme is made by the three algorithm (**KeyGen**, **Sign**, **Verify**), defined as follows.

KeyGen$(\lambda) \rightarrow (\mathbf{pk}, \mathbf{sk})$, is the key generation algorithm. Given λ, the security parameter of the scheme, the algorithm does:
1. Generate two, distinct λ-bit primes p, q; (1p) and compute $N = pq$ and $\varphi(N)$, where $\varphi(\cdot)$ denotes Euler Totient function. (1p)
2. Choose a random integer $e \xleftarrow{R} \mathbb{Z}_{\varphi(N)}$ satisfying GCD$(e, \varphi(N))$=1.(1p) Compute $d = e^{-1} \mod \varphi(N)$. (1p)
3. Set $\mathbf{pk} = (N, e)$ and $\mathbf{sk} = d$. (1p)*(note: the role of e and d is interchangeable, as long as* **Sign** *takes as input the secret value, and* **Verify** *the public one).*

Sign$(\mathbf{sk}, m) \rightarrow \sigma$, is the signing algorithm. It takes as input the secret key d and a message $m \in \mathbb{Z}_{\varphi(N)}$ and outputs the signature $\sigma = m^d \mod N$. (2p)

Verify$(\mathbf{pk}, m, \sigma) \rightarrow \{0, 1\}$ is the verification algorithm. It takes as input the public key e, a message m and a signature σ; it outputs 1 if the signature is correct (verifies) and 0 otherwise. (1p) More formally, the verification algorithm checks whether $\sigma^e = m \mod N$. (2p) If the equality holds **Verify** returns 1, otherwise it returns 0.

8. Let $N > 2$ be a positive integer. Consider the function $h : \mathbb{Z} \longrightarrow \mathbb{Z}_N$, defined as $h(m) = m \mod N$. To check if h is a cryptographic hash function we need to assure that h satisfies (at least) the following three properties:

   (2a) Given a message m, the message digest $y = h(m)$ can be computed in an efficient way.

   (2b) Given a message digest y, it is computationally infeasible to find an m with $h(m) = y$ (in other words, h is a one-way, or pre-image resistant function).

   (2c) It is computationally infeasible to find two dinstint messages $m_1, m_2 \in \mathbb{Z}$ such that $h(m_1) = h(m_2)$ (in this case, the function h is said to be collision-free).

   Check if h is a cryptographic hash function, i.e., for each of the properties ((8a), (8b) and (8c)) show if h satisfies it or not. **(10p)**

   Computing the remainder modulus N of a number $m \in \mathbb{Z}$ can be done in an efficient way using, e.g., the Euclidean Algorithm. (2p) Therefore h satisfies property (8a). (1p)
   Property (8b) is obviously not satisfied, (1p) since $m = y$ itself is a possible pre-image of the digest y. (2p) More formally, for every $y \in \{0, 1, \ldots, N-1\} \subset \mathbb{Z}$ it holds that $h(y) = y \mod N = y$. (1p)
   From the observation above, it is immediate to see that h is not collision-resistant. (1p) Indeed, for any message $m_1 \in \mathbb{Z}$ we have that $m_2 = m_1 + N \in \mathbb{Z}$ has the same digest: $h(m_1) = m_1 \mod N = m_1 + N \mod N = h(m_2)$. (2p) Actually, all the messages of the form $m_1 + kN$ have the same digest (the remainder modulus N).

Advanced Topics in Cryptography (30p)

9. Describe in your own words (or give the definition of):

   (a) Unconditional and provable security. Also, give at least one example of a cryptosystem in each category. **(6p)**

Unconditional security: The cryptosystem cannot be broken even by an adversary with unlimited computational power. (2p) Examples of unconditionally secure crypto systems are: Shamir's and Mignotte's Secret Sharing Scheme., the OTP. (1p)
Provable security: It is possible to prove that an attack that breaks the cryptosystem, can be used to solve some well-known problem widely believed to be (computationally) hard. (2p) Examples of provably secure cryptosystems are: ElGamal (reduced to the Discrete-Log assumption), RSA (reduced to the Discrete-Log assumption and the factoring assumption). (1p)

(b) The three main properties of the Fiat-Shamir identification protocol (Completeness, Soundness and Zero-Knowledge). **(8p)**

Completeness: An (interactive) identification protocol is complete if an honest prover P succeeds in convincing a honest verifier V that a true statement is true. (2p)
Soundness: An (interactive) identification protocol is sound if no dishonest prover P succeeds in convincing an honest verifier V that a false statement is true. (2p)
Zero-Knowledge: An honest prover P can convince the verifier V of the validity of a statement without revealing any information beyond the truth of the statement. In other words, an honest-but-curious verifier is not able to extract any useful information (e.g. the secret key) from the prover while, at the same time, the verifier will be convinced that the prover knows the secret. (4p)

10. Consider the Secure Multiparty Computation (SMPC) protocol for addition, based on the Shamir Secret Sharing Scheme, seen in class. Assume that there are $n = 4$ parties (P_1, P_2, P_3, P_4), that the system tolerates $t = 3$ corrupted parties, and that all computations are done in $\mathbb{Z}_{13}$.

(a) Imagine you are P_1, and your secret input to the computation is $a = 5$. Explain how you would share your secret value a with the other parties and what you expect to receive from each other party (note that no explicit computation is required for this step, just a formal description of how the scheme works). **(4p)**

In order to share the secret $a \in \mathbb{Z}_{13}$ using the Shamir Secret Sharing Scheme, the party P_1 needs to first select a *random* polynomial $f(x) \in \mathbb{Z}_{13}[x]$ satisfying $f(0) = a$ and $3 = t = deg(f)$. (2p) In other words, P_1 generates $f(x)$ as $f(x) = a + r_1x + r_2x^2 + r + 3x^3$ with $r_1, r_2, r_3 \xleftarrow{R} \mathbb{Z}_{13}$. Then P_1 computes f on the points $i \in \{1, 2, 3, 4\}$ and gives to party P_i the share $a_i = f(i)$. (1p) Each party follows the same procedure, thus P_1 receives the shares of each party's secret input computed in $i = 1$, namely b_1, c_1 and d_1. (1p)

(b) Now, imagine you are P_1 and hold the table below (which corresponds to your view of the protocol). Compute the value $S = a+b+c+d$ using the information contained in the table. **(12p)**

	P_1	P_2	P_3	P_4
$a = 5$	$a_1 = 5$	$a_2 = 12$	$a_3 = 7$	$a_4 = 10$
$b =?$	$b_1 = 4$	?	?	?
$c =?$	$c_1 = 12$	?	?	?
$d =?$	$d_1 = 9$	?	?	?
S	$s_1 = 4$	$s_2 = 6$	$s_3 = 1$	$s_4 = 7$

Party P_1 can compute the sum S by using Lagrange interpolation on the partial sums s_1, s_2, s_3 and s_4 in the formula $S = \sum_{i=1}^{4} s_1 \delta_i^{\{1,2,3,4\}}(0)$, where $\delta_i^{\{1,2,3,4\}}(0)$ denote the Lagrange interpolation polynomials (evaluated at 0) (4p) and defined as: $\delta_i^{\{1,2,3,4\}}(0) = \prod_{j \in \{1,2,3,4\} \setminus \{i\}} j(j-i)^{-1}$ (here k^{-1} denotes the inverse of k modulus 13). (1p)

$\delta_1^{\{1,2,3,4\}}(0) = 2(2-1)^{-1} \cdot 3(3-1)^{-1} \cdot 4(4-1)^{-1} = 4 \mod 13$
$\delta_2^{\{1,2,3,4\}}(0) = 1(1-2)^{-1} \cdot 3(3-2)^{-1} \cdot 4(4-2)^{-1} = 7 \mod 13$
$\delta_3^{\{1,2,3,4\}}(0) = 1(1-3)^{-1} \cdot 2(2-3)^{-1} \cdot 4(4-3)^{-1} = 4 \mod 13$
$\delta_4^{\{1,2,3,4\}}(0) = 1(1-4)^{-1} \cdot 2(2-4)^{-1} \cdot 3(3-4)^{-1} = 12 \mod 13$
For each δ_i: (1p) for correct computation, (0.5p) for using EEA / Bézout identity to find the inverses in $\mathbb{Z}_{13}$. Total (6p).

Substituting the numbers in the Largange interpolation formula, one gets: $4 \cdot 4 + 6 \cdot 7 + 1 \cdot 4 + 7 \cdot 12 = 3 \mod 13$. Thus $S = 3$. (1p).

(c) *Bonus question: Looking at the table in point (10b), are you able to determine what was the polynomial f chosen by P_1 to share a? Why? Compute the polynomial f, if possible.* *(4 bonus points)*

Yes, it is possible to retrieve the polynomial f chosen by P_1, because table (10b) contains $n = 4$ points on which the degree $t = 3$ polynomial (chosen by P_1 to share the secret input a) is evaluated, and there exists only one degree t polynomial passing on $t + 1 = 4$ points (1p). It is possible to compute $f(x) \in \mathbb{Z}_{13}[x]$ using the Lagrange interpolation on the points $f(1) = a_1 = 5$, $f(2) = a_2 = 12 \equiv -1 \mod 13$, $f(3) = a_3 = 7$, $f(4) = a_4 = 10$, as follows:

$$f(x) = \sum_{i=1}^{4} f(i) \delta_i^{\{1,2,3,4\}}(x) \mod 13$$

where $\delta_i^{\{1,2,3,4\}}(x) = \prod_{j \in \{1,2,3,4\} \setminus \{i\}} (x-j)(i-j)^{-1}$ (1p). By substituting the numbers in the above expressions we get:

$\delta_1^{\{1,2,3,4\}}(x) = \frac{x-2}{1-2} \cdot \frac{x-3}{1-3} \cdot \frac{x-4}{1-4} = 2\big(x^3 + (-4-3-2)x^2 + (6+8+12)x + 2\big)$
$= 2x^3 + 8x^2 + 4 \mod 13$ (0.5p)
$\delta_2^{\{1,2,3,4\}}(x) = 7x^3 + 9x^2 + 3x + 7 \mod 13$ (0.5p)
$\delta_3^{\{1,2,3,4\}}(x) = 6x^3 - 3x^2 + 6x + 4 \mod 13$ (0.5p)
$\delta_4^{\{1,2,3,4\}}(x) = -2x^3 - x^2 + 4x - 1 \mod 13$ (0.5p)
Thus, $f(x) = -x^3 + x + 5$.