CHALMERS | GÖTEBORGS UNIVERSITET

Cryptography

TDA352 (Chalmers) - DIT250 (GU)

24 August 2017, 8:30 - 12:30

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 plaintext *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.* You can draw diagrams to explain concepts like security games, cryptographic protocols or simple ciphers modes of operation. In any case, 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. Let $(\mathbf{E}, \mathbf{D})$ be a secure block cipher. Describe the CBC mode of operation (encryption and decryption). **(8p)**

   In CBC (Cipher Block Chaining) mode, the message M is divided in blocks $\{m_i\}$ of the same length. (1p) A random initialization vector IV is chosen to encrypt M, and each block is encrypted sequentially with $\mathbf{E}$ and the same key $\mathbf{sk}$. (2p) The IV is attached at the beginning of the ciphertext. (1p) If we denote with c_i the i-th encrypted block, we have that

   $$c_0 = \mathbf{E}(\mathbf{sk}, IV \oplus m_0), \quad c_i = \mathbf{E}(\mathbf{sk}, c_{i-1} \oplus m_i) \quad (2p)$$

   and

   $$m_0 = \mathbf{D}(\mathbf{sk}, m_0) \oplus IV, \quad m_i = \mathbf{D}(\mathbf{sk}, c_i) \oplus c_{i-1} \quad (2p)$$

2. Show that the One Time Pad (OTP) cipher is malleable, i.e., that and adversary can change the ciphertexts so that it decrypts to a different message. **(7p)**

   The OTP cipher does not achieve message integrity, since the ciphertexts are *intentionally* malleable. For instance, given $c = \mathbf{E}(k, m) = k \oplus m$ (2p), an adversary could substitute c with $c' = c \oplus a$, where $a \in \{0, 1\}^n$. (3p) Now the adversary can predict what entries of c have been affected by the change, and thus what entries of the original message m have been tampered, since $\mathbf{D}(k, c') = m \oplus a$. (2p) This is attack is particularly harmful if the adversary has some knowledge on the plaintext m (e.g., it contains the amount of money to be transferred).

3. Let $(\mathbf{E}$,$\mathbf{D})$ be a correct cipher. Write the formula for correct decryption in the encryption schemes below.
   (Hint: your solution should look like $\mathbf{D}'((k_1, k_2), c) = \mathbf{D}(k_1, c \oplus k_1) \oplus k_2$).

   (a) $\mathbf{E}'((k_1, k_2), m) = \mathbf{E}(k_1, m) \oplus k_2 \oplus k_1$. **(2p)**

   $\mathbf{D}'((k_1, k_2), c) = \mathbf{D}(k_1, c \oplus k_2 \oplus k_1)$ (2p)

   (b) $\mathbf{E}'((k_1, k_2, k_3), m) = \mathbf{E}(k_3, \mathbf{E}(k_2, \mathbf{E}(k_1, m)))$. **(3p)**

   $\mathbf{D}'((k_1, k_2, k_3), c) = \mathbf{D}(k_1, \mathbf{D}(k_2, \mathbf{D}(k_3, c)))$ (3p)

Public Key Encryption (30p)

4. This exercise is about RSA encryption. You are given two prime numbers $p = 11$ and $q = 17$, an RSA modulus $N = p \cdot q = 187$, and an encryption exponent $e = 3$.

   (a) Compute the RSA decryption exponent d. **(7p)**

   For how RSA is defined, the decryption exponent d is the inverse of e modulus $\varphi(N)$ (1p), where $\varphi(\cdot)$ is Euler's totient function. (1p) In this case we have $\varphi(N) = \varphi(p) \cdot \varphi(q) = (p-1)(q-1) = (11-1)(17-1) = 160$ (2p), since p and q are co-prime. To compute d we use the Extended Euclidean Algorithm between $\varphi(N) = 160$ and the encryption exponent $e = 3$: (1p)

   $$\begin{array}{rcll} 160 & = & 3(53) & +1 \\ 3 & = & 1(3) & +0 \end{array}$$

From the second-last row we see that $1 = 160 + 3(-53)$. If we read the equation mod 160 we directly get that the inverse of $e = 3 \mod 160$ is $-53 = 107 \mod 160 = d$.(2p).

(b) Encrypt the message $m = 100$ using RSA with the values given above. **(7p)**

In order to encrypt using RSA we need to compute $c = m^e \mod N$. (3p) That is 100^3 in $\mathbb{Z}_{187}$. The first two modular multiplications give $100 \cdot 100 = 89 \mod 187$, and $89 \cdot 100 = 111 \mod 187$. (3p) So the encrypted message is $c = 111$. (1p)

5. Give a pseudocode for Fermát primality test. **(8p)**
(Hint: Fermát's primality test is based on Fermát little theorem that states: $a^p = a \mod p$ if p is a prime number (and $a \in \mathbb{Z}_p \setminus \{0\}$))
We can write the Fermat primality test in algorithm form as:

Inputs: n, k; where n is the number we want to test for primality, (1p) $n > 3$; and k is a parameter that determines the number of times to test for primality (accuracy) (1p)
Output: non-prime if n is composite, probably prime otherwise (2p)

```
for $i = 1, 2, \ldots, k$ do
   pick $a$ randomly in the range $\{2, 3, \ldots, n-1\}$
   compute $GCD(a, n) = d$,
      if $d \neq 1$ return non-prime
      else compute $c = a^{n-1} \mod n$
         if $c \neq 1$ return non-prime
         end if
      end if
end for (4p)
```

6. Use the Chinese Remainder Theorem (CRT) to solve the following system of linear congruences. **(8p)**

$$\begin{cases} x = 1 \mod 7 \\ x = 3 \mod 11 \end{cases}$$

Since 7 and 11 are relative primes, the CRT guarantees that the system admits unique solution modulus $77 = 11 \cdot 7$. (2p) To find it, we compute the coefficients of Bézout identity (using the Extended Euclidean Algorithm): $11(2) + 7(-3) = 22 - 21 = 1$. (3p) We have $x = 1(22) + 3(-21) = -41 = 36 \mod 77$. (3p).

7. *Bonus question: describe the IND-CCA security game (indistinguishability chosen ciphertext attack)* *(4 bonus points)*

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 $b \in \{0, 1\}$ 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)

Data Integrity (20p)

8. Consider the following signature scheme. The setting is a cyclic group $\mathbb{Z}_q^*$, for a large prime q. Let g be a generator for $\mathbb{Z}_q^*$.
The **KeyGen** algorithm picks a random value $\mathbf{sk} = x \in \mathbb{Z}_q^*$, computes the corresponding public key $\mathbf{pk} = X = g^x \in \mathbb{Z}_q^*$, and outputs $(\mathbf{pk}, \mathbf{sk})$.
The **Sign** algorithm takes as input $\mathbf{sk}$ and a message $m \in \{0,1\}^n$, and proceed as follows. First it computes $h = H(m) \in \mathbb{Z}_q^*$ for some hash function $H : \{0,1\}^n \longrightarrow \mathbb{Z}_q^*$. Secondly, it computes $z = xh^{-1}$ in $\mathbb{Z}_q^*$. Finally, it outputs the signature $\sigma = g^z$.

   (a) Define the correctness property of a signature scheme. **(2p)**

   A signature scheme is said to be correct if $\mathbf{Verify}(\mathbf{pk}, m, \mathbf{Sign}(\mathbf{sk}, m)) = \mathsf{true}$ for all possible messages (2p).

   (b) What computations should the **Verify** algorithm perform for the above signature scheme to be correct? **(3p)**

   In order for the given signature to be correct, the verification algorithm should perform the following steps. First, compute $h = H(m) \in \mathbb{Z}_q^*$. (1p) Then return true if $\sigma^h = X = \mathbf{pk}$, (1p) otherwise return false. (1p)

   (c) Is it computationally infeasible for an attacker to produce a valid signature for an arbitrary message m^*, without knowing the secret key x? **(5p)**

   No. (1p) Chosen a message m^*, the attacker can compute $H(m^*) = h^*$ and make the signature $\sigma^* = X^{(h^*)^{-1}}$. (2p) This is computationally feasible since the evaluation of the hash function H is efficient and computing $(h^*)^{-1}$ from h^* can be done using the Extended Euclidean Algorithm (which is known to be efficient). (2p).

9. Let $2 < N < 100$ be a positive integer such that $GCD(N, 3) = 1$. Consider the function $h : \mathbb{Z} \longrightarrow \mathbb{Z}_N$, defined as $h(m) = 3m + 1 \mod N$.

   (a) Is h such that, given a message digest y, it is computationally infeasible to find an m with $h(m) = y$? Why? (i.e., is h one-way / pre-image resistant?). **(5p)**

   No, h is not pre-image resistant. (2p) For example, given a digest $y \in \mathbb{Z}_N$ we can compute a possible pre-imagine as $m = (y-1)3^{-1} \mod N$, (2p) where 3^{-1} denotes the inverse of 3 modulus N (1p) (which exists because, by assumption, N and 3 are coprime).

   (b) Is h such that it is computationally infeasible to find two distinct messages $m_1, m_2 \in \mathbb{Z}$ such that $h(m_1) = h(m_2)$? Why? (i.e., is h collision-free?). **(5p)**

   No, h is not collision-resistant. (1p) Indeed, for any message $m_1 \in \mathbb{Z}$ we have that $m_2 = m_1 + N \in \mathbb{Z}$ (1p) has the same digest: $h(m_1) = 3m_1 + 1 \mod N = 3m_1 + 3N + 1 \mod N = 3(m_1 + N) + 1 \mod N = h(m_2)$ (2p) since $N = 0 \mod N$. Actually, all the messages of the form $m_1 + kN$ for $k \in \mathbb{Z}$ have the same digest. (1p)

Advanced Topics in Cryptography (30p)

10. Consider the following identification protocol based on the discrete logarithm problem. $G =< g >$ is a cyclic group of prime order q, the prover (called Peggy) has a private key $x \in \{1, 2, \dots, q-1\}$ and publishes the corresponding public key $h = g^x \in G$. The purpose of the protocol is to convince the verifier (called Victor) that Peggy knows the secret value x:

    -1- Peggy chooses a random $r \in \{1, 2, \dots, q-1\}$, computes $R = g^r$ and $S = g^{x-r}$. She sends R and S to Victor.

    -2- Victor chooses a random challenge bit $c \in \{0, 1\}$, and sends c to Peggy.

    -3- Peggy replies to Victor with the value $z = cx - r$.

    (a) What computations should Victor do in order to check Peggy's values? **(4p)**

    In order to check Peggy's values Victor should check that:
    (1) $R \cdot S = h$, since $R \cdot S = g^{r+(x-r)} = g^x = h$, (2p)
    (2) either $R = g^z$ (if $c = 0$) (1p) or $S = g^z$ (if $c = 1$) (1p).
    This is equivalent to the correctness of the protocol.

    (b) Show that if Eve, who does not know Peggy's secret key x, can predict Victor's challenge, than she has probability 1 to pass the identification protocol (i.e., be accepted by Victor). **(6p)**
    (Hint: for the case $c = 1$, you can try to swap the role of R and S)

    Given that the Eve can predict Victor's challenge bit c, she can construct R in such a way that the final answer z will never involve computations with the secret x. (2p) For instance, if Eve predicts that $c = 0$, she can pick r at random, and send $R = g^r$, $S = R^{-1}h$ in message -1-. (1p) If Eve predicts $c = 1$, then she needs to exchange R and S, i.e., send $S = g^r$, $R = S^{-1}h$. (2p) Her values will always pass Victor's check: in both cases the reply is $z = r$. (1p)

    (c) Show that if an honest Peggy chooses the same randomness r twice, an honest-but-curious Victor can retrieve Peggy's secret x. **(6p)**

    Let $(R^{(1)}, S^{(1)}, c^{(1)}, z^{(1)})$ indicate the first transcript of the identification protocol. If Victor, in a subsequent run of the protocol, sees that message -1-. equals $R^{(1)}, S^{(1)}$ (2p), i.e., $(R^{(2)}, S^{(2)}) = (R^{(1)}, S^{(1)})$, then he can send as challenge $c^{(2)} = c^{(1)} \oplus \mathbf{1}$ (the flipped challenge) (2p), and retrieve $x = (c^{(1)} - c^{(2)})(z^{(1)} - z^{(2)})$. (2p)

11. Consider Shamir Secret Sharing Scheme. 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 the Dealer. Explain how you would share your secret value a among the four parties (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 Dealer needs to first select a *random* polynomial $f(x) \in \mathbb{Z}_{13}[x]$ satisfying $f(0) = a$ and $3 = t = deg(f)$. (3p) In other words, the Dealer generates $f(x)$ as

$f(x) = a + r_1 x + r_2 x^2 + r + 3x^3$, with $r_1, r_2, r_3 \in \mathbb{Z}_{13}$ chosen at random. Then the Dealer computes f on the points $i \in \{1, 2, 3, 4\}$ and gives to party P_i the share $a_i = f(i)$. (1p)

(b) Now, imagine you are P_1 and your share is $a_1 = 5$. Suppose you also learn the other parties' shares: $a_2 = 12, a_3 = 7, a_4 = 10$. Can you recover the secret value a shared among the four parties? Show how or explain why not. **(10p)**

Yes, (3p) party P_1 can compute the secret value a using Lagrange interpolation on the shares. Formally, $a = \sum_{i=1}^{4} a_i \delta_i^{\{1,2,3,4\}}(0)$, where $\delta_i^{\{1,2,3,4\}}(0)$ denote the Lagrange interpolation polynomials (evaluated at 0) (2p) and defined as: $\delta_i^{\{1,2,3,4\}}(0) = \prod_{j \in \{1,2,3,4\} \setminus \{i\}} j(j-i)^{-1} \mod 13$. (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: (0.5p) for correct computation. Total (2p).

Substituting the numbers in the Largange interpolation formula, we get:
$5 \cdot 4 + 12 \cdot 7 + 7 \cdot 4 + 10 \cdot 12 = 5 \mod 13$. Thus $a = 5$. (2p).

12. *Bonus question: describe textbook Diffie-Hellman key exchange.* *(4 bonus points)*

The global public parameter is the (description of a) cyclic group $G = \langle g \rangle$ of order q (note that G is known to both A and B). (1p)
The Diffie-Hellman key exchange protocol works as follows.

-1- A chooses a random $a \in \{1, \cdots, q-1\}$, computes $A = g^a$ in G and sends it to B.

-2- B chooses a random $b \in \{1, \cdots, q-1\}$, computes $B = g^b$ in G and sends it to A. (1p)

-3- A computes the shared secret key $B^a = \mathbf{sk} = g^{ab}$.

-4- B computes the shared secret key $A^b = \mathbf{sk} = g^{ba}$. (2p)