CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Katerina Mitrokotsa

DIT250/TDA352/TDA351

Exam in Cryptography

Wednesday April 6, 2016, 8:30 – 12.30.
Teacher: Katerina Mitrokotsa, phone 076 200 11 68.
Tillåtna hjälpmedel: Typgodkänd räknare. Annan minnestömd räknare får användas efter godkännande av kursansvarig vid dennes besök i skrivsalen.
Allowed aids: Approved calculator. Other calculators with cleared memory may be used after approval by the responsible teacher.

The exam has 5 problems with a total of 50 points. 22/31/40 points are needed for grade 3/4/5.

Answers must be given in English and should be clearly justified.

1. Alice uses a block cipher E where both the block size and the key size are 64 bits, but is worried that the key size is too small and thus brute force attacks are feasible. She therefore decides to use two keys (k_0, k_1) and encrypt a message m as follows:

$$c = E_{k_0}(m) \oplus k_1$$

Suppose that an adversary gets access to two plaintext/ciphertext pairs (i.e., (m,c) and (m',c')) and is able to perform a brute-force attack on the original block cipher E and recover the key in a known plaintext attack.

(a) Show that the adversary can also break Alice's "improved" cipher and recover her extended key. (2 p)

(b) Does the attack against the "improved" cipher require much more effort than an attack against the block cipher E? Explain why. (2 p)

Solution: Suppose Alice has the plaintext/ciphertext pairs (m,c) and (m',c') from the extended cipher, i.e. $c = E_{k_0} \oplus k_1$ and $c' = E_{k_0}(m') \oplus k_1$. By xoring the two equations we get:

$$c \oplus c' = E_{k_0}(m) \oplus k_1 \oplus E_{k_0}(m') \oplus k_1 = E_{k_0}(m) \oplus E_{k_0}(m')$$

This way k_1 has been eliminated and the adversary can all possible k_0 and check for this equation to hold. This is only twice as much computation as an attack against E alone, so is certainly feasible. Thus, k_0 is determined and it is easy to use $c = E_{k_0}(m) \oplus k_1$ to determine k_1.

2. Alice and Bob use a block cipher for encryption and need to choose between two modes of operation either CBC mode or counter mode.

- CBC mode: Here an n block plaintext $M_1M_2M_3 \ldots M_n$ is encrypted to an $n+1$ block ciphertext $C_0C_1C_2 \ldots C_n$ where C_0 is an initialisation vector and $C_i = E_K(M_i \oplus C_{i-1})$ for $i > 0$.

- Counter mode: Here an n block plaintext $M_1M_2M_3\ldots M_n$ is encrypted to an n block ciphertext $C_1C_2\ldots C_n$ where:

$$K_i = E_K(IV||i)$$

$$C_i = M_i \oplus K_i$$

An adversary is able to intercept and change messages sent between Alice and Bob. Now consider the following scenarios.

(a) In some messages sent by Bob, it is the case that the last block is a randomly generated secret key. Decide for the two modes whether the adversary can corrupt messages sent, so that Alice receives a message that looks good after decryption, but contains the wrong key. (2 p)

Solution: For both modes it is the case that the adversary can replace the last ciphertext block with any other block. When Alice decrypts the message all previous blocks will be unchanged and the message looks good; the last block will be corrupt, but since it is random, there is no way for Alice to discover this.

(b) In some messages sent by Bob, the adversary may know the first block M_1 and want to replace it by another block A_1 of his choice, leaving the rest of the message unchanged. Show that the adversary can achieve this if Counter mode is used. Do you think he can do it with CBC mode? (4 p)

Solution: The adversary can achieve this if the encryption is in Counter mode. The encryption of the first block is $C_1 = M_1 \oplus E_K(IV||1)$, from which he can compute $E_K(IV||1) = M_1 \oplus C_1$. He wants to replace C_1 by $C_1' = A_1 \oplus E_K(IV||1)$ and can easily compute $C_1' = A_1 \oplus M_1 \oplus C_1$. The other blocks are not affected by this.

For CBC mode, we have $M_1 = D_K(C_1) \oplus C_0$. The adversary cannot change C_1, since that would affect Alice's decryption of C_2. Instead he must try to find C_0' such that $A_1 = D_K(C_1) \oplus C_0'$. Solving for C_0', we get:
$C_0' = A_1 \oplus D_K(C_1) = A_1 \oplus M_1 \oplus C_0$

3. Consider that we are in $\mathbb{Z}_p$ where p is a large prime and $p-1$ has a prime divisor q. Further, g is a generator for a subgroup of order q of $\mathbb{Z}_p^*$. A community of users share parameters p, q and g. Typically, p is a 1024 bit number, while q has only 160 bits. Each user has a private key $x < q$.

(a) Describe what is each user's El Gamal public key, how El Gamal encryption of a message m and decryption of the corresponding ciphertext works and in which hard problem the security relies on. (3 p)

Solution: Each user has public key $X = g^x \mod p$. To encrypt a message m for this user, the sender chooses a random number $y < q$ and encrypts the message as $(c_1, c_2) = (g^y, m \cdot X^y)$. To decrypt it computes $c_1^x = g^{xy}$ and then computes $\frac{c_2}{c_1^x} = m \cdot g^{xy} g^{xy} = m$. The security is based on the discrete log problem i.e., it is hard to compute the secret key x if you are given the public key $X = g^x$.

(b) Show how to compute m^q given the encryption of m. (3 p)

Solution: We have that $c_2^q = (m \cdot (g^x)^y)^q = m^q \cdot (g^q)^{xy} = m^q$ since $g^q = 1$ (g is a generator of a group of order q).

(c) Show that El Gamal is not secure against a chosen ciphertext attack. (3 p)

Solution: The attacker could have asked for the decryption of a modified ciphertext. To see one possibility, we just multiply the equation above by 2 to get $2m = 2c_2 \cdot c_1^{-x}$ and we see that a suitable choose is $c' = (c_1, 2c_2)$. If we get the plaintext m' back, we know that $m' = 2m$, i.e. $m = m'/2$

4. (a) Describe what does it mean that a public key encryption scheme is semantically secure and provide the definition of the advantage of the adversary. (4 p)

Hint: Use the standard game between a challenger and an adversary.

Solution: In the semantic security game, the adversary A submits two messages (plaintexts) m_0 and m_1 of the same length to the challenger. The challenger, who possess a secret key k, picks a random $b \in \{0,1\}$ and returns to A one ciphertext, $c = Enc(k, m_b)$. The game ends with A outputting a guess b' for b. A is said to break the semantic security if he can guess the correct b with non-negligible probability. More formally, let W_b denote the event that the challenger returns $c = Enc(k, m_b)$ for $b \in \{0,1\}$, the advantage of A in the semantic security game is defined as $Adv(A) = |Prob[A(W_0) = 1] - Prob[A(W_1) = 1]|$.

(b) We consider RSA encryption. It is often recommended to choose a small public key exponent to increase efficiency. A common choice is $e = 3$. Explain why $e = 2$ is not suitable. (3 p)

Solution: It is required that $gcd(e, \Phi(N)) = 1$ for the RSA system to work properly. But $\Phi(N) = (p-1)(q-1)$ is an even number so $gcd(2, \Phi(N)) = 2$. In particular, it would not be possible to choose d with $ed = 1 \pmod{\Phi(N)}$.

(c) We consider double RSA encryption using a common modulus N an two public keys e_1 and e_2 with corresponding private keys. Thus, a message m is encrypted first using RSA encryption with the key e_1; the result is encrypted again using key e_2. Explain why this approach does not increase security. (3 p)

Solution: The general argument against double encryption is that it is subject to the meet-in-the-middle attack, which has time complexity similar to that of a single brute force attack. In the particular case of RSA encryption, double encryption is also meaningless, since the double encryption is equivalent to the single RSA encryption with public key $e_1 e_2$ and private key $d_1 d_2$. It is easy to verify this since it holds $(m^{e_1} \pmod N)^{e_2} \pmod N = m^{e_1 e_2} \pmod N$

5. Assume that we have three parties P_1, P_2 and P_3 and that we tolerate $t = 1$ corrupted party. Assume that we work in $\mathbb{Z}_{11}$ and each of the parties have a secret value $a = 2$, $b = 4$ and $c = 1$ correspondingly. The three parties want to compute the sum $\sigma = a + b + c$ while keeping their corresponding value secret. Using Shamir's secret sharing show how to calculate the sharing of a, b, c and of their sum σ.

More precisely if we denote by a_1, a_2, a_3 the shares of the secret value a and we denote similarly the shares of b, c and σ. Then:

(a) Fill in the following table: (5 p)

	P_1	P_2	P_3
$a=2$	a_1	a_2	a_3
$b=4$	b_1	b_2	b_3
$c=1$	c_1	c_2	c_3
$\sigma=a+b+c$	σ_1	σ_2	σ_3

Solution: P_1 chooses polynomial $f(x)$ of degree at most 1 to share a, and P_2 and P_3 choose correspondingly polynomials $g(x)$ and $h(x)$ to share b and c correspondingly. The chosen polynomials are kept secret by each party. The only restriction in the choice of the polynomials is that $f(0)=a=2$, $g(0)=b=4$ and $h(0)=c=1$. Let the chosen polynomials be:

$f(x)=2+2x$, $g(x)=4+x$, and $h(x)=1+3x$

Then P_1 computes $a_1=f(1)=4$, $a_2=f(2)=6$ and $a_3=f(3)=8$ and sends a_2 to P_2 and a_3 to P_3.

Similarly P_2 computes $b_1=g(1)=5$, $b_2=g(2)=6$ and $b_3=g(3)=7$ and sends b_1 to P_1 and b_3 to P_3.

Finally, P_3 computers $c_1=h(1)=4$, $c_2=h(2)=7$ and $c_3=h(3)=10$ and sends c_1 to P_1 and c_2 to P_2.

Thus, P_i has shares a_i, b_i and c_i and can compute $\sigma_i=a_i+b_i+c_i \pmod{11}$.

Thus we can easily fill the table as follows:

	P_1	P_2	P_3
$a=2$	4	6	8
$b=4$	5	6	7
$c=1$	4	7	10
$\sigma=a+b+c$	13	19	25

(b) We want that P_1 only learns $\sigma=7$ and nothing about b and c. Show that if P_1 makes the hypothesis that $b=3$ and $c=2$ then this **cannot be excluded** from the possible solutions. P_1 has the following view:

	P_1	P_2	P_3
$a=2$	a_1	a_2	a_3
assumes $b=3$	b_1	b_2	b_3
assumes $c=2$	c_1	c_2	c_3
$\sigma=a+b+c$	σ_1	σ_2	σ_3

where the blue values (i.e., $a, a_1, a_2, a_3, b_1, c_1, \sigma, \sigma_1, \sigma_2, \sigma_3$) denote the values that P_1 already knows (you have them if you solve question (a)) This means that P_1 does not know only the values: b, c, b_2, b_3, c_2, c_3. (5 p)

Solution: If we consider that P_1 is corrupted, then he knows $b_1=g(1)=5$ and $c_1=h(1)=6$.

By making the hypothesis that $b=g(0)=3$ and $c=h(0)=2$.

Then, this would mean that: P_1 assumes that $g'(x)=g(x)$ where $g'(x)=3+2x$ and similarly that $h'(x)=h(x)$ where $h'(x)=2+2x$.

Thus, P_1's view can be summarized in the following table:

	P_1	P_2	P_3
$a=2$	$a_1=4$	$a_2=6$	$a_3=8$
assumes $b=3$	$b_1=5$	$b_2'=7$	$b_3'=9$
assumes $c=2$	$c_1=4$	$c_2'=6$	$c_3'=8$
$\sigma=a+b+c$	$\sigma_1=13$	$\sigma_2'=19$	$\sigma_3'=25$

We should note here that b_2', b_3', c_2' and c_3' are incorrect values.

Thus, by computing $\sigma'_2 = a_2 + b'_2 + c'_2$ and $\sigma'_3 = a_3 + b'_3 + c'_3$, he gets $\sigma'_2 = \sigma_2$ and $\sigma'_3 = \sigma_3$. Thus, the hypothesis that $b = 3$ and $c = 2$ cannot be excluded from the possible solutions.

6. (a) Explain briefly the differences between a MAC and a signature. (2 p)

   Solution: A MAC is employed in secret key cryptography while a signature in public key cryptography. A MAC does not provide the non-repudiation property that the signature provides.

   (b) Explain how using the CBC mode of a block cipher E you can construct a CBC-MAC. (2 p)

   Solution: See slide 8 in

   `http://www.cse.chalmers.se/edu/year/2015/course/TDA352/lectures/lect06.pdf`

   We have a block cipher $E : K \times X \to X$ we get $F_{ECBC} : K^2 \times X \to K$ The figure that is given on the slide and describes how the CBC-MAC is calculated should be provided.

   (c) What is the main advantage of a CBC-MAC over a simple MAC? (1 p)

   Solution: It is much shorter.

7. We recall the Fiat-Shamir authentication protocol. Let $N = p \cdot q$, where p and q are primes. The prover P wants to convince the verifier V that he knows a square-root of $y \in \mathbb{Z}_N^*$, i.e., a number x such that $y = x^2 \in \mathbb{Z}_N^*$, without revealing x to V. They use the following protocol. All computations are in $\mathbb{Z}_N^*$.

   - P generates a random r, computes $R = r^2 \pmod N$ and sends R to V (the commitment).
   - V generates a uniformly random bit b (i.e., b is either 0 or 1 uniformly at random) and sends it to P (the challenge).
   - P responds with $z = r \cdot x^b \pmod N$.

   (a) What computation will V perform to check P's values? (2 p)

   Solution: V checks if it holds $z^2 = R \cdot y \pmod N$.

   (b) Discuss how a cheating P, who does not know x, can achieve a probability of 0.5 of passing the test. (1 p)

   Solution: If P does not know x, she can produce good values if she can predict b. P chooses z at random and

   - If $b = 0$, set $R = z^2$.
   - If $b = 1$, set $R = z^2 \cdot y^{-1}$.

   If P's prediction is right, V will verify her values. If not P has lost. Thus, someone pretending to be P has 0.5 probability of fooling V, if V chooses b with equal probabilities.

   (c) How can you decrease the success probability of a cheating P, who does not know x? (1 p)

   Solution: By iterating the protocol k times, the probability of a false P being accepted is reduced to 2^{-k}. Thus the soundness error is 2^{-k} for k iterations.

(d) P happens to use the same r in two executions of the protocol. Can V learn anything about x? If yes what and how? (2 p)

Solution: Yes. Since r is used twice it holds: $\frac{z'^2}{x^{b'^2}} = \frac{z^2}{x^{b^2}}$ thus it is possible to solve for x and disclose the key.