CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Katerina Mitrokotsa DIT250/TDA351

Exam in Cryptography

Tuesday April 15, 2015, 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 7 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 and Bob use a block cipher for encryption and need to choose a mode of operation. Recall the following two modes:

   - 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

   $$\begin{aligned} K_i &= E_K(IV||i) \\ C_i &= M_i \oplus K_i. \end{aligned}$$

   An adversary is able to intercept and changes 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)

   (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 use. Do you think he can do it with CBC mode? (5 p)

2. (a) Textbook RSA is multiplicatively homomorphic, i.e.,

$$RSA_{pk}(m_1) \cdot RSA_{pk}(m_2) = RSA_{pk}(m_1 \cdot m_2) \quad (1)$$

Define how RSA encryption is done, and show that RSA is multiplicatively homomorphic, i.e., that equation **??** holds. (3 p)

(b) Explain how textbook RSA can be used for digital signatures, i.e., explain how to sign and how to verify the signature. (3 p)

(c) Signatures based only on textbook RSA are not secure due to RSA being multiplicatively homomorphic. Show an existential forgery attack on textbook RSA signatures, i.e., given messages, m_i, and the corresponding signatures, σ_i, show how to construct a new message m and a corresponding valid signature σ without having access to the key. (2 p)

(d) How can we change the way we produce RSA signatures to protect against the above existential forgery attack? Give the construction and show that the attack no longer works. (3 p)

3. We consider ElGamal encryption using a generator g for $\mathbb{Z}_p^*$ for some large prime p. Every user chooses a random private key $x < p$ and computes the public key $X = g^x$. To encrypt message m for a user with public key X, the sender chooses a random $y < p$ and computes the encryption $(g^y, m \cdot X^y)$.

(a) Describe how decryption is done. (2 p)

(b) In an adaptive chosen ciphertext attack the Adversary wants to decrypt a message c and is allowed to ask for, and get, decryption of *any* message *except* c. Show in detail that both ElGamal and textbook RSA are not secure against such an attack. (6 p)

4. The Schnorr protocol works as follows. Setting: Group of prime order q with generator g. P has private key x and public key $X = g^x$.

- P generates random $r \in \mathbb{Z}_q$, computes $R = g^r$ and sends R to V.
- V generates random $c \in \mathbb{Z}_q$ and sends it to P.
- P computes $z = c \cdot x + r$ and sends it to V.

After this sequence of messages V checks if $g^z = R \cdot X^c$.

(a) Justify why this protocol works. (2 p)

(b) Show that if an eavesdropper is able to get two runs of the protocol that are using the same commitment r the eavesdropper can compute the secret x, i.e., show how to compute x from (R, c_1, z_1) and (R, c_2, z_2). (3 p)

5. Encryption in RSA-OAEP proceeds as follows. Let (e,N) be the public key and d the private key of Alice, where the modulus N is an n bit integer. We assume that message length is fixed to m bits and let $k = (n-m)/2$. G and H are two fixed hash functions, with hash values of size $m+k$ and k bits, respectively.

   Here is how to encrypt message M for Alice.

   - Pick a random string r of k bits.
   - Compute $t = (M||0^k) \oplus G(r)$, where 0^k denotes the bit string consisting of k zeros.
   - Compute $u = r \oplus H(t)$ and let $s = t||u$. This will be an n bit number; if $s \geq N$, start from the beginning and pick a new r.
   - The encrypted message is $c = s^e \bmod N$.

   Describe in detail how Alice decrypts the message. (6 p)

6. We consider protocols where Peggy proves her identity to Victor by giving evidence that she knows a secret x.

   The system involves a trusted third party T. Initially, T chooses primes p and q as in RSA and computes $N = p \cdot q$ and a RSA key pair (e,d). N and e are made public and can be used by a whole community of Peggies and Victors. T keeps the private key d for himself. All computations below are in $\mathbb{Z}_N^*$.

   Whenever (a new) Peggy wants to use the system, she chooses a public key $X \in \mathbb{Z}_N^*$ (which could be based on her name, email address etc, using some public way of transforming this to a number in $\mathbb{Z}_N^*$). She sends X to T, who computes Peggy's secret key $x = X^{-d}$ and sends it to her in some secure way. Peggy then announces her public key X.

   When Peggy wants to identify herself to Victor, the following protocol is used:

   1. Commitment: Peggy chooses a random $r \in \mathbb{Z}_N^*$, computes $R = r^e$ and sends R to Victor.
   2. Challenge: Victor chooses a random c with $1 \leq c \leq e$ and sends c to Peggy.
   3. Response: Peggy computes $y = r \cdot x^c$ and sends y to Victor.

   Victor now checks that $y \neq 0$ and $R = y^e \cdot X^c$; if this holds he believes that the other party is Peggy.

   (a) Show that a true Peggy, following the protocol, will be identified correctly by Victor. (4 p)

   (b) Why does Victor check that $y \neq 0$? (2 p)

   (c) Show that a false Peggy, who does not know x, but correctly guesses c before she makes her commitment, can arrange to be identified by Victor as Peggy. (2 p)

   Remark: Thus, the security level of the system can be decided by choosing e suitably. A false Peggy who guesses c has probability $1/e$ of success.

7. Alice and Bob both share a secret key with the trusted third party Trent; these keys are K_{AT} and K_{BT}, respectively. They have designed the following protocol, which allows Alice to send an encrypted message to Bob, where the message is encrypted using a session key K chosen by Alice:

$$\begin{array}{lll} 1. & A \to T & : \quad A, B, \{K\}_{K_{AT}} \\ 2. & T \to A & : \quad \{K\}_{K_{BT}} \\ 3. & A \to B & : \quad \{K\}_{K_{BT}}, \{M\}_K \end{array}$$

Alice uses Trent to encrypt the session key for Bob and includes this encrypted key in message 3, together with the encrypted message.

Unfortunately, the protocol is vulnerable to an attack by the adversary Cedric, who is an insider, i.e. he also shares a key K_{CT} with Trent and can participate in runs of the protocol. Explain how Cedric, after eavesdropping on the above run, can go on to decrypt the message. Then suggest some modification to the protocol that prevents the attack you found. (5 p)