CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Björn von Sydow INN150/TDA351

Exam in Cryptography

Thursday December 17 2009, 14.00 – 18.00.
Teacher: Björn von Sydow, phone 1040.

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 points are needed to pass.

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

1. A certain cryptographic device contains a linear feedback shift register (LFSR) of length *L* bits with secret tap sequence.

   (a) How does one make use of this LFSR to construct a stream cipher with secret keys of length *L*, i.e. how is a message *m* encrypted using this cipher? (2 p)

   (b) Assume that an adversary has access to a plaintext/ciphertext pair (m, c) from this cipher. How many bits long must *m* be in order for the adversary to be able to completely determine the tap sequence? Remember to give motivation for your answer! (3 p)

2. What is a message authentication code (MAC)? Describe the most important requirements on a MAC and also what it is used for. You do *not* have to describe how MAC's are implemented. (5 p)

3. We consider the Diffie-Hellman key agreement protocol in its original form, i.e. using a generator g in $\mathbb{Z}_p^*$ for some big prime p.

   (a) Describe how Alice and Bob can agree on a session key using this protocol. (3 p)

   (b) Describe the man-in-the-middle attack against the protocol. (3 p)

   (c) To protect against this attack, Diffie-Hellman must be improved in some way. Such improvements often involve certificates. What is a certificate, i.e. what information does it contain and who issues it? (2 p)

   (d) What does it mean that g is a generator for $\mathbb{Z}_p^*$? (2 p)

   (e) Formulate the discrete logarithm problem in this setting. (2 p)

4. 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 $M_1M_2M_3\dots M_n$ is encrypted to an $n+1$ block ciphertext $C_0C_1C_2\dots C_n$, 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 $M_1M_2M_3\dots M_n$ is encrypted to an n block ciphertext $C_1C_2\dots C_n$, 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. (4 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)

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. We have seen the Fiat-Shamir protocol, which is based on the infeasibility of computing square-roots of composite numbers, and the Schnorr protocol, based on the difficulty of the discrete log problem. Not surprisingly, one can also base such protocols on the difficulty of the RSA problem. We will now look at one such protocol, proposed by Guillou and Quisquater.

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 $y \neq 0$? (1 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. We consider yet another published, flawed protocol for authentication and session key agreement, the Neuman-Stubblebine protocol. It employs a trusted third party and runs as follows:

$$\begin{array}{lll}
1. & A \rightarrow B & : & A, N_A \\
2. & B \rightarrow T & : & B, \{A, N_A, T_B\}_{K_{BT}}, N_B \\
3. & T \rightarrow A & : & \{B, N_A, K_{AB}, T_B\}_{K_{AT}}, \{A, K_{AB}, T_B\}_{K_{BT}}, N_B \\
4. & A \rightarrow B & : & \{A, K_{AB}, T_B\}_{K_{BT}}, \{N_B\}_{K_{AB}}
\end{array}$$

The protocol employs both timestamps and nonces. Some remarks:

- Alice initiates the run in message 1, sending her name and a nonce N_A to Bob.
- Bob contacts the trusted third party Trent, forwarding Alice's information and adding a nonce N_B of his own and a timestamp T_B. Part of the message is encrypted with the key K_{BT} shared by Bob and Trent.
- Trent generates a session key K_{AB} to be used by Alice and Bob and sends to Alice a message with two encrypted parts, one for Alice and one for Bob, and Bob's nonce in the clear. The part encrypted for Bob, $\{A, K_{AB}, T_B\}_{K_{BT}}$, is called the *ticket*.

- Alice checks her nonce and forwards the ticket to Bob, together with Bob's nonce encrypted with the session key. This last piece convinces Bob both that the message is fresh and that the sender is Alice.

However, the system is flawed. Assume that keys and nonces have the same sizes in bits. Show how an adversary, eavesdropping on messages 1 and 2 of the initial protocol, may intercept and himself send a valid message 4 to Bob, claiming to be Alice, and thus complete the initial protocol and communicate with Bob using encryption with a session key that Bob believes he shares with Alice. (6 p)

Remark: The reason for using both nonces and timestamps was that, after running the above initial protocol, Alice should be able to open many new sessions with Bob using the same session key, *without* communicating with Trent, until the ticket expires. Such repeated authentication uses a separate three-message protocol:

$$\begin{array}{lll} 1. \; A \to B & : & \{A, K_{AB}, T_B\}_{K_{BT}}, N'_A \\ 2. \; B \to A & : & \{N'_A\}_{K_{AB}}, N'_B \\ 3. \; A \to B & : & \{N'_B\}_{K_{AB}} \end{array}$$