CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Björn von Sydow INN150/TDA351

Solutions to exam in Cryptography 091217

1. (a) The secret key is installed as the initial state in the LFSR. Then the LFSR is run to produce a bitstream $s = s_0 s_1 s_2 \ldots$. Finally, message m is encrypted by xoring with the keystream, i.e. $c = m \oplus s$. Thus we need only run the LFSR to produce as many bits as the message contains.

   (b) From $c = m \oplus s$ we get $s = m \oplus c$. The adversary can thus compute the initial part of the keystream. This can be used to set up a system of linear equations for the unknown taps $c_1, c_2, \ldots c_L$. With knowledge of $2L$ bits of s we can form a system of L equations for the L unknowns.

2. A MAC is an efficiently computable function that takes a secret key and an arbitrarily long message as arguments and computes a fixed size value (typically 160 or 256 bits). The required security property is that, even with access to a large number of (message,MAC) pairs, it should be infeasible to compute a pair consisting of a new message and its MAC without knowledge of the secret key.

   A MAC is used for verification of the data integrity of a message, i.e. that it has not been changed, and its authenticity, i.e. that it originates from the claimed sender.

3. (a) The protocol is as follows
   - Alice chooses a random integer $x < p-1$, computes $X = g^x$ and sends X to Bob.
   - Bob chooses a random integer $y < p-1$, computes $Y = g^y$ and sends Y to Alice.
   - Both Alice and Bob can now compute $K = g^{xy} = X^y = Y^x$ and use K as session key.

   (b) The adversary intercepts the communication between Alice and Bob, and communicates with each of them, claiming to be the other. The adversary can then do one Diffie-Hellman key negotiation with Alice and one with Bob; thereafter he can forward messages between the parties, decrypting and reencrypting the messages. Of course, he can also change messages at will.

   (c) A certificate contains at least the name and public key of a user, typically also e.g. expiration date. It is usually issued by a trusted certification authority.

   (d) g is a generator for $\mathbb{Z}_p^*$ if g^n, $n = 0, 1, 2 \ldots p-2$ gives all elements of $\mathbb{Z}_p^*$ (or, equivalently, if g has order $p-1$).

   (e) The discrete log problem is the problem to find x given y, where $y = g^x \in \mathbb{Z}_p^*$ (p and g are known).

4. (a) 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) 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

$$C_0' = A_1 \oplus D_K(C_1) = A_1 \oplus M_1 \oplus C_0,$$

   so also in CBC mode it is possible.

5. Alice proceeds as follows.

   - Decrypt c using textbook RSA decryption and parse the result as $t||u$, where t has $m+k$ bits:
$$t||u = c^d \bmod N.$$
   - Compute $r = u \oplus H(t)$. This will give the value r used by the sender.
   - Compute $M' = t \oplus G(r)$. This will give $M||0^k$.
   - Check that M' ends with k zeros. If not, decryption fails. If it does, strip these zeros to get the plaintext M.

6. (a) First we note that, since $x = X^{-d}$, we have $x^{-e} = X^{ed} = X$.
   Victor will get
$$y^e \cdot X^c = (r \cdot x^c)^e \cdot (x^{-e})^c = r^e = R,$$
   so he will accept Peggy.

   (b) This is to avoid a false Peggy who chooses $R = 0$ and $y = 0$.

   (c) If the false Peggy can guess c, she may pick y arbitrarily and then choose $R = y^e \cdot X^c$.

7. The adversary eavesdrops and hears the two first messages. He then notices he can construct a valid message 4, as follows: He uses the encrypted part of message 2, i.e. $\{A, N_A, T_B\}_{K_{BT}}$, as the first part of his message. It has the correct structure, with A first and B:s timestamp T_B last, and in the middle the nonce N_A playing the role of K_{AB}. To complete the message, he just needs to encrypt N_B using N_A as key (both these were sent unencrypted in messages 1 and 2). The complete protocol run is

$$\begin{array}{lll}
1. & A \to B & : A, N_A \\
2. & B \to T & : B, \{A, N_A, T_B\}_{K_{BT}}, N_B \\
3. & T \to A & : \ldots \\
4. & C(A) \to B & : \{A, N_A, T_B\}_{K_{BT}}, \{N_B\}_{N_A}
\end{array}$$

   The third message, from T to A, plays no role in the attack.