CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Björn von Sydow DIT250/TDA351

Exam in Cryptography

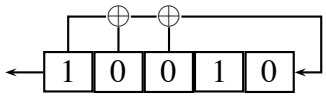
Saturday December 10, 2011, 8.30 – 12.30.
Teacher: Björn von Sydow, phone 0722 39 14 01.

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. Find the shortest LFSR that generates the same output sequence as the following:



   (6 p)

2. We consider cryptographic signatures.

   (a) Why does one usually sign the hash of a message, rather than the message itself? Give two reasons. (2 p)

   (b) What is meant by non-repudiation in the context of signatures? (1 p)

   (c) What is a certificate? What data does it contain, who issues it and how does one normally verify it? (4 p)

3. In RSA, each user chooses a modulus $N = p \cdot q$, where p and q are (large) primes, and an encryption exponent e.

   (a) Describe how the decryption exponent d is computed and the condition on e that guarantees that d can be computed successfully. (3 p)

   (b) Would we still get a good public key cipher if we let N be a prime number, rather than a product of two? (2 p)

   (c) Same question if we let N be a product of *three* prime numbers? (2 p)

4. We consider iterative cryptographic hash functions, which compute n bit hash values for arbitrarily long messages. As we have seen, hash functions are often defined using a *compression* function

$$h : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n.$$

The definition of the hash function also uses an n bit initialization vector IV and splits the message to be hashed into k bit blocks. We first ignore the need for padding, i.e. we assume that the message length is a multiple of k bits. To hash $M_1 M_2 \ldots M_p$, of length $p \cdot k$ bits, one defines the recurrence

$$\begin{aligned} H_0 &= IV \\ H_i &= h(M_i, H_{i-1}), \quad i = 1, 2, \ldots p. \end{aligned}$$

The hash value is H_p.

(a) One way to try to find collisions for this hash function is to search for a block B, such that $h(B, IV) = IV$. Show that if one finds such a block, then one can construct (many) collisions for the hash function. (3 p)

(b) Now we take padding into consideration and assume that messages are padded, to make the total message length a multiple of k. We don't specify padding exactly, except that the last 64 bits of padding will always contain the length in bits of the unpadded message. Thus we can only hash messages shorter than 2^{64} bits, a restriction without practical importance. We also note that padding must always be at least 64 bits; if fewer bits remain in the last block, we add an extra pad block.

Explain why we now cannot anymore construct collisions easily, given a block B as in (a), i.e. with $h(B, IV) = IV$. (3 p)

(c) (a) and (b) discuss attempts to find collisions by expoiting the actual definition of the hash function. A completely different method to find collisions, which applies to all hash functions, is the birthday attack. Describe briefly this attack. How many messages would one expect to have to generate?
(2 p)

5. We consider the block cipher mode CBC, where an n block message $M_1 M_2 \ldots M_n$ is encrypted to an $n+1$ block message $C_0 C_1 C_2 \ldots C_n$, where

$$\begin{aligned} C_0 &= IV \\ C_k &= E_K(M_k \oplus C_{k-1}), \quad k = 1, 2, \ldots n. \end{aligned}$$

(a) How is decryption affected if the first ciphertext block C_0 is removed from the ciphertext? (1 p)

(b) How is decryption affected if the last ciphertext block C_n is removed from the ciphertext? (1 p)

(c) We already have the encryption of a message as above. We want to add one specified plaintext block M_0 in the beginning of the message, so that the extended message is $M_0 M_1 M_2 \ldots M_n$. Show that a legitimate user (i.e,

someone who knows the key K) can achieve this by just adding one ciphertext block C_{-1} in the beginning of the ciphertext, to get encryption $C_{-1}C_0C_1\ldots C_n$. (3 p)

(d) Recall the setting of home assignment 1, i.e. a server that provides the adversary with the following service: For any proposed ciphertext encrypted with CBC, the server will check whether the decrypted message is correctly padded or not and inform the adversary about the outcome. Describe how, in this scenario, also the adversary can achieve the goal in (c). (2 p)

6. We consider the following protocol, intended to provide mutual authentication between Alice and Bob, and establishment of a session key K_{AB}, with the assistance of a trusted third party Trent:

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

Some comments to the protocol:

- Alice initiates the protocol by encrypting a nonce together with her and Bob's names with the key she shares with Trent, and sends this to Bob.
- Bob forwards Alice's encrypted part together with his own encrypted nonce to Trent.
- Trent decrypts the messages, chooses a session key K_{AB}, encrypts this key for Alice and Bob together with their respective nonces and sends all this to Bob.
- Bob forwards the part intended for Alice to her.

Now consider the following scenario. The adversary Charlie takes control over Bob's network connection and starts a protocol run, pretending to be Alice:

$$\begin{array}{lrcl} 1. & C(A) \to B & : & A, B, \{N_C, C, B\}_{K_{CT}} \\ 2. & B \to C(T) & : & A, B, \{N_C, C, B\}_{K_{CT}}, \{N_B\}_{K_{BT}} \\ 2'. & C(B) \to T & : & C, B, \{N_C, C, B\}_{K_{CT}}, \{N_B\}_{K_{BT}} \end{array}$$

(a) Explain how this will continue and lead to the result that Charlie and Bob share a fresh session key, but Bob believes that he shares it with Alice. Why will Bob not suspect that anything is wrong? (5 p)

(b) Suggest some simple change to the protocol that prevents the attack. (2 p)

7. A simple idea for hybrid encryption of a long message m is to choose a key K for a block cipher at random, encrypt this key using Elgamal encryption and then encrypt m using the block cipher with key K. In this problem we consider an

improvement of this scheme, proposed by Fujisaki and Okamoto. The improvement allowed them to prove CCA2 security of their scheme in the random oracle model.

The cipher is based on several primitives:

- A subgroup G of Z_p^* of order q with generator g, where q is a k_1 bit prime.
- The Elgamal encryption function

$$E_X(r,y) = (g^y, r \cdot X^y).$$

  Recall that Elgamal encryption of message $r < p$ for a user with public key X consists in choosing a random positive integer $y < q$ and then computing the encryption $E_X(r,y)$.
- A secret key cipher B_K for arbitrarily long messages, where key K has size k_k bits. Think of a block cipher together with a suitable mode of operation.
- Two hash functions H_1 and H_2, producing hash values of size k_1 and k_2 bits, respectively.

Each user of this system chooses a private key $x < q$ and computes the public key $X = g^x$.

To encrypt a (possibly long) message m for this user one proceeds as follows:

- Choose a random $r < p$.
- Compute $C_1 = E_X(r, H_1(r||m))$.
- Compute $C_2 = B_{H_2(r)}(m)$.
- The ciphertext is $C_1||C_2$.

(a) Describe in detail how decryption is done. (5 p)

(b) What does it mean that an encryption scheme is CCA2 secure? (3 p)