CHALMERS TEKNISKA HÖGSKOLA
Datavetenskap
Björn von Sydow DIT250/TDA351

Exam in Cryptography

Thursday December 16, 2010, 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.

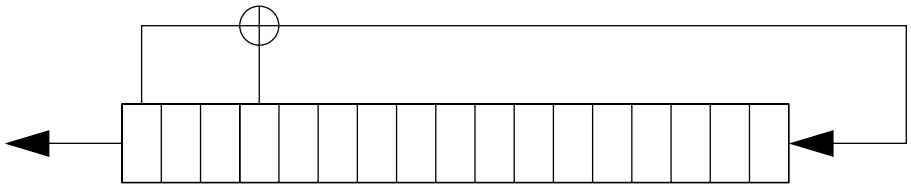
Answers must be given in English and should be clearly motivated.

1. We consider cryptographic hash functions.
   (a) What is meant by a collision attack against a hash function? (2 p)
   (b) Explain why any hash function which produces n bit hash values can only provide $n/2$ bits of security against a collision attack. (3 p)

2. This question concerns block ciphers.
   (a) Which block cipher was most commonly used during the last decades of the last century? (1 p)
   (b) Why is it now considered insecure? (1 p)
   (c) Describe how one can still use it with satisfactory security using multiple encryption. (3 p)
   (d) To encrypt large files with a block cipher one can split the file into a sequence of blocks and encrypt each block separately. This is the so called *electronic code book* mode and has several disadvantages. Describe some of these. (3 p)

3. In RSA, each user creates a key pair, consisting of a public encryption key and a private decryption key. To do this, the user first finds two large prime numbers p and q and computes $N = p \cdot q$. She then chooses an integer $e \in \mathbb{Z}_N^*$ and the public key is (e, N). Describe how the user computes the decryption key d. (3 p)

4. Alice uses a stream cipher with an LFSR of size 17 as key stream generator. The LFSR taps two bits of the state:



   (a) The adversary knows how Alice's cipher is constructed, but does not know the key. He intercepts a ciphertext $c_1c_2\ldots c_{2000}$ of size 2000 bits, for which he knows some of the corresponding plaintext bits, namely $m_{1001}m_{1002}\ldots m_{1100}$. Explain how he can decrypt the whole message. (5 p)

   (b) The choice of taps above actually makes the key stream period maximal for that size of LFSR. How long is this period? (1 p)

5. 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:

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

   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)

6. Consider the scenario where Bob receives a contract m from Alice, together with Alice's signature s on this document. Bob can then show to anyone that Alice has signed m. This property is unsuitable for some applications, where Alice's commitment to m may be sensitive. For such situations, David Chaum suggested *undeniable* signatures. An undeniable signature can only be verified with cooperation from the signer. To verify signature s, the verifier Victor engages in a protocol with Alice. If Victor is someone with whom Alice does not want (or need) to deal, she may refuse to participate in the protocol, but if she accepts to interact, there are three possible outcomes of a protocol run:

(1) The signature is verified;

(2) Victor is convinced that the signature is invalid, i.e. was not issued by Alice;

(3) Victor is convinced that the signature is valid, but Alice is trying to deny this fact.

The setting is a cyclic group G of prime order q with generator g and a hash function H with hash values in G. Each signer chooses a private key $x \in \mathbb{Z}_q^*$ and computes the public key $X = g^x$. The signature on m is $s = (H(m))^x$.

(a) The verification protocol is as follows:

Victor chooses random integers a and b, both smaller than q, and computes the challenge $c = s^a X^b$, which is sent to the signer Alice. Alice responds by sending back $r = c^{x'}$, where x' is the inverse of x in $\mathbb{Z}_q^*$, i.e. $x \cdot x' = 1 (\bmod q)$.

Which equality does Victor then check to verify the signature? Victor needs to compute some expressions based on what is known to him, i.e. q, g, H, m, s, X, a, b and r, and then check an equality that will hold if the signature is valid, but is very unlikely to hold if it is not. (5 p)

(b) If the verification fails, then either the signature is invalid or it is valid, but Alice is trying to deny that she did produce it. In this case, Victor runs the protocol once again, with new random values a_1 and b_1. If the response is r_1, then Victor checks whether

$$(r \cdot g^{-b})^{a_1} = (r_1 \cdot g^{-b_1})^a.$$

If so, Victor believes that the signature is invalid; if not, that Alice is trying to cheat.

Show that, if Alice is honest, the equality holds.

(You do not need to show that it is infeasible for a dishonest Alice to find an r_1 such that the equality holds, but this can be proved.) (5 p)

(c) How would you recommend to choose the group G? Motivate your choice. (2 p)

7. We consider again the Padding Oracle attack, which you studied in home assignment 2. Recall that the attack was as follows: Alice and Bob share a secret key K_{AB} for a block cipher E with block size 64 bits and exchange messages encrypted by that cipher using CBC mode. Before encryption, the message is padded. If the message consists of n bytes before padding, the padding consists of $8 - (n \bmod 8)$ bytes, each with the value $7 - (n \bmod 8)$ (as an eight-bit integer). So if n is a multiple of 8, padding consists of eight bytes, each with value 7 ($= 00000111_2$). (This is the same padding method as in home assignment 2.)

   Now, let M_p be block number p in a message sent from Alice to Bob and let C_{p-1} and C_p be blocks $p-1$ and p in the corresponding ciphertext. p is here arbitrary. By creating a sequence of carefully chosen messages to send to Bob, the adversary can decrypt and recover M_p, if, for each "ciphertext" message sent by the adversary, Bob informs the adversary whether that message becomes correctly padded after CBC decryption.

We now consider a variant of this attack: if Bob serves as a padding oracle as above, the adversary can also *encrypt* any message with K_{AB} – without knowing this key! So, we assume that the adversary has a message with k blocks, $A_1A_2\ldots A_k$, that he wants to encrypt. The corresponding ciphertext is $C_0C_1\ldots C_k$, where C_0 is a random initialization vector chosen by the adversary and

$$C_i = E_{K_{AB}}(A_i \oplus C_{i-1}), \quad \text{for } i = 1,2,\ldots k.$$

However, the adversary does not know K_{AB} and has to proceed differently.

(a) In order to construct $C_0C_1\ldots C_k$, the adversary starts by choosing the *last* block C_k arbitrarily. Show how he can then perform a Padding Oracle attack against Bob and determine $D_{K_{AB}}(C_k)$, where D is the block cipher decryption function. You must show which messages the adversary constructs and how he uses Bob's answers to determine $D_{K_{AB}}(C_k)$. (5 p)

(b) Show that he can now construct a suitable C_{k-1}, such that the last block of the decrypted message will be A_k. After this step, the adversary has constructed C_k and C_{k-1}. (3 p)

(c) Describe how the adversary continues to encrypt the whole message $A_1A_2\ldots A_k$. (3 p)