ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Family Name: .....

First Name: .....

Section: .....

Cryptography and Security Course

(Crypto Part)

Final Exam

March 2nd, 2006

This document consists of 8 pages.

Instructions

Electronic devices are *not* allowed.

Answers must be written on the exercises sheet.

This exam contains 2 *independent* exercises.

Answers can be either in French or English.

Questions of any kind will certainly *not* be answered. Potential errors in these sheets are part of the exam.

You have to put your full name on *each* page and you have to do it *now*.

Implementation of the Diffie-Hellman Protocol

We call a communication channel *secure* if it ensures the *authenticity* and the *integrity* of the messages it transmits. Note that a secure communication channel does *not* ensure confidentiality. We call a communication channel *private* if it is secure and protect confidentiality.

We would like to establish a private channel based on a secure one.

1. How to do this by using standard cryptographic primitives?

We consider the Diffie-Hellman protocol in the group $\mathbf{Z}_p^*$ generated by g, where p is a prime number. We assume that p and g are some standard numbers, agreed and safely stored by Alice and Bob.

2. Recall how the Diffie-Hellman protocol works between Alice and Bob over a secure channel.

3. What can happen if the channel is no longer secure, i.e., if an adversary can tamper with the communication between Alice and Bob? Describe an attack.

For a participant (either Alice or Bob), we call *viewed transcript* of the protocol the string obtained by concatenation of all incoming and outgoing messages of the protocol in chronological order.

Since secure communication is more expensive than insecure communication, we would like to minimize the amount of data sent over the secure channel. For this, Alice and Bob run the regular Diffie-Hellman protocol over an insecure channel and then authenticate the session by checking that their respective viewed transcripts are identical. For that, Alice and Bob hash the viewed transcript of the Diffie-Hellman protocol and exchange it in both directions through a secure channel.

4. Assuming that the hash function is collision resistant, prove that the previous attack no longer works.

5. What is now the amount of data (in terms of bits) that has to be transmitted on the secure channel (typically)?

To further reduce the quantity of information transmitted over the secure channel, we want to truncate the digest down to $k = 20$ bits.

6. What is the complexity of a second preimage attack against the truncated hash function?

7. Show that one can mount a man-in-the-middle attack.

8. Propose another (safer) value for k.

We now assume that the parameters p and g are no longer standard, but selected by Alice and sent to Bob together with the first message in the Diffie-Hellman protocol. We assume that the protocol developer forgot to include p and g in the transcript to authenticate, so that an adversary can make Bob accept fake parameters p' and g'.

9. Show that if the adversary carefully selects p' and fixes $g' = 1$, then Alice and Bob terminate the protocol on some consistent Diffie-Hellman key which can be computed by the adversary.

10. To thwart this attack, Bob should make sure that p' is prime and that g' generates a subgroup of large order in $\mathbf{Z}^*_{p'}$. Explain how Bob can achieve this.

A Provably Secure Hash Function

Let p and q be two distinct prime numbers of ℓ bits, $n = p \cdot q$, and g be an element of the group $\mathbf{Z}_n^*$ selected as in question 4. Let

$$\begin{array}{rccc} h: & \mathbf{Z} & \longrightarrow & \mathbf{Z}_n^* \\ & x & \longmapsto & h(x) = g^x \end{array}$$

be a hash function, hashing bit strings of arbitrary length (considered as integers). We assume that p and q are discarded after the computation of n so that no one knows p and q.

Implementation Parameters

1. Explain how one can generate p and q. Give the name of the main algorithm involved in this computation and the complexity of whole operation in terms of ℓ.

2. Given a message x, what is the algorithm that should typically be used during hashing? What is the complexity of hashing this message x?

3. What is the typical length of a hash value? What is the complexity of a collision search on h by brute force? What is the complexity of a first preimage attack by brute force?

Let g_1 and g_2 be generators of $\mathbf{Z}_p^*$ and $\mathbf{Z}_q^*$ respectively. From now on, we consider that g is such that $g \bmod p = g_1$ and $g \bmod q = g_2$.

Factoring Implies Collisions

4. Provide a formula that allows to compute g from g_1, g_2, p, and q.

5. Express the order λ of g in $\mathbf{Z}_n^*$ in terms of p and q.

Factoring Equivalent to Collisions

6. Given the knowledge of λ, show that we can find collisions on h.

7. Given a collision on h, show that we can deduce a multiple of λ.

9. Deduce that finding a collision on h is equivalent to factoring n.

10. Propose a minimal value of ℓ to obtain a secure construction.