

## Assignment #5

Due: Tuesday, April 27, 2004.

**Problem 1 (25 pts)** Parties  $A_1, \dots, A_n$  and  $B$  wish to generate a secret conference key. All parties should know the conference key, but an eavesdropper should not be able to obtain any information about the key. They decide to use the following variant of Diffie-Hellman: there is a public prime  $p$  and a public element  $g \in \mathbb{Z}_p^*$  of order  $q$  for some large prime  $q$  dividing  $p - 1$ . User  $B$  picks a secret random  $y \in [1, q - 1]$  and computes  $\gamma = (g^y \bmod p)$ . Each party  $A_i$  picks a secret random  $x_i \in [1, q - 1]$  and computes  $\alpha_i = (g^{x_i} \bmod p)$ . User  $A_i$  sends  $\alpha_i$  to  $B$ . User  $B$  responds to party  $i$  by sending  $\beta_i = (\alpha_i^y \bmod p)$ .

- (5 pts) Show that party  $i$  given  $\beta_i$  (and  $\alpha_i$ ) can determine  $\gamma$ .
- (5 pts) Explain why (a hash of)  $\gamma$  can be securely used as the conference key. Namely, give a brief informal explanation why an eavesdropper cannot determine  $\gamma$ .
- (15 pts) Formally prove part b. Namely, show that if there exists an efficient algorithm  $\mathcal{A}$  that given the public values in the above protocol, outputs  $\gamma$ , then there also exists an efficient algorithm  $\mathcal{B}$  to break the Diffie Hellman protocol (using  $p$  and  $g$  as the public values). Note that  $\mathcal{B}$  takes  $(g^a \bmod p)$  and  $(g^b \bmod p)$  as input and should output  $(g^{ab} \bmod p)$ .

**Problem 2 (30 pts)** Let  $N = pq$  be an RSA composite. Let  $g \in [0, N^2]$  be an integer satisfying  $g = (aN + 1 \bmod N)$  for some  $a \in \mathbb{Z}_N^*$ . Consider the following encryption scheme. The public key is  $\langle N, g \rangle$ . The private key is  $\langle p, q, a \rangle$ . To encrypt a message  $m \in \mathbb{Z}_N$  do: (1) pick a random  $h \in \mathbb{Z}_{N^2}^*$ , and (2) compute  $C = g^m \cdot h^N \bmod N^2$ . Our goal is to develop a decryption algorithm.

- (10 pts) Show that the discrete log problem  $\bmod N^2$  base  $g$  is easy when knowing the private key. That is, show that given  $g$  and  $B = g^x \bmod N^2$  there is an efficient algorithm to recover  $x \bmod N$ . Use the fact that  $g = aN + 1$  for some integer  $a \in \mathbb{Z}_N^*$ .
- (10 pts) Show that given the public key and the private key, decrypting  $C = g^m \cdot h^N \bmod N^2$  can be done efficiently.  
Hint: consider  $C^{\phi(N)} \bmod N^2$ . Use the fact that by Euler's theorem  $x^{\phi(N^2)} = 1 \bmod N^2$  for any  $x \in \mathbb{Z}_{N^2}^*$ .
- (10 pts) Show that this encryption scheme enables limited computation on ciphertexts. Let  $a, b, c$  be integers in  $[1, N]$ . Show that given  $N$  and  $c$ , and the encryption of  $a$  and  $b$  it is possible to construct the encryption of  $a + b$  and the encryption of  $c \cdot a$ . More precisely, show that given  $N$  and an integer  $c$ , and ciphertexts  $C_1 = E[a]$ ,  $C_2 = E[b]$ , it is possible to construct the ciphertexts  $C_3 = E[a + b]$  and  $C_4 = E[c \cdot a]$ .

**Problem 3 (25 pts)** An earlier version of the ISO Public Key Three-Pass Mutual Authentication Protocol is as follows:

1. Alice  $\leftarrow$  Bob:  $N_B$
2. Alice  $\rightarrow$  Bob:  $\text{Cert}_A, N_A || N_B || B || \text{sig}_A(N_A || N_B || B)$ ;
3. Alice  $\leftarrow$  Bob:  $\text{Cert}_B, N'_B || N_A || A || \text{sig}_A(N'_B || N_A || A)$ ;

In the protocol,  $\text{Cert}_A$  is Alice's certificate,  $\text{Cert}_B$  is Bob's certificate,  $N_A$  and  $N_B$  are nonces generated by Alice and Bob respectively,  $\text{sig}_A(M)$  denotes Alice's digital signature on  $M$ , and  $\text{sig}_B(M)$  denotes Bob's signature on  $M$ .

- a. (20 pts) Describe an attack on this protocol that enables a Malicious party to initiate a communication with Alice and convince that it is Bob who initiated the communication.  
Hint: in the attack, the Malicious party also needs to communicate with  $B$ .
- b. (5 pts) Describe a fix of the problem.

**Problem 4 (20 pts)** The Woo-Lam Protocol is an authentication protocol using symmetric encryption and trusted third party Trent.

Alice and Trent share a symmetric key  $K_{AT}$ ;

Bob and Trent share a symmetric key  $K_{BT}$ .

The protocol is as follows:

1. Alice  $\rightarrow$  Bob:  $Alice$ ;
2. Alice  $\leftarrow$  Bob:  $N_B$ ;
3. Alice  $\rightarrow$  Bob:  $E_{K_{AT}}[N_B]$ ;
4. Trent  $\leftarrow$  Bob:  $Bob, E_{K_{BT}}[Alice, E_{K_{AT}}[N_B]]$ ;
5. Trent  $\rightarrow$  Bob:  $E_{K_{BT}}[N_B]$ ;
6. Bob decrypts what he receives in step 5 using  $K_{BT}$ , and accepts if the encryption returns his nonce sent in step 2 correctly; he rejects otherwise.

Assume that Carl and Trent also share a symmetric key  $K_{CT}$ .

Describe a parallel-session attack in which Carl starts two sessions with Bob (one as Carl and one faking as Alice) and can eventually make the faking session with Bob succeed, i.e., Bob believes that he is talking with Alice in that session. Describe the message sequences in the attack.

Hint: Assume that the communication between Trent and Bob is connection-less (e.g., through UDP); in other words, when Bob sends two messages in two sessions to Trent and receives two replies, Bob cannot link a reply with a particular session; he can only try to decrypt and see whether the reply is meaningful for that session. In this case, Bob will accept in a session when one of the replies is correct for that session.