

## Assignment #6

Due: Tuesday, December 9, 2004.

**Problem 1: (15 pts)** Exercise 7.6 on Page 312 of the Textbook; 5 pts for each of a, b, and c. For part c, we don't require formal reduction-type proof; instead, consider what happens if one tries to forge a signature.

**Problem 2: (15 pts)**

Consider the following signature scheme: A public key is  $(p, g, y)$ , and the corresponding private key is  $x$ , which is chosen randomly from  $[1, p - 2]$ , where  $p$  is a large prime,  $g$  is a generator of  $\mathbb{Z}_p^*$ , and  $y = g^x \pmod{p}$ .

To sign a message  $M$ , compute  $h = H(M)$ , the hash code of the message. We require that  $\gcd(h, p - 1) = 1$ . If not, append the hash to the message and calculate a new hash. Continue this process until a hash code is produced that is relatively prime to  $(p - 1)$ . Then calculate  $z$  to satisfy  $z \times h \equiv x \pmod{(p - 1)}$ . The signature of the message is  $\sigma = g^z$ . To verify the signature  $\sigma$ , check whether  $y \equiv \sigma^h \pmod{p}$ .

- **a.** (4 pts) Show that if the signature is generated as stated, the verification succeeds.
- **b.** (4 pts) Why we require that  $\gcd(h, p - 1) = 1$ ?
- **c.** (7 pts) Show that the scheme is unacceptable by describing a simple technique for forging a user's signature on an arbitrary message.

**Problem 3: (10 pts)**

Describe the signature scheme derived from the Fiat-Shamir protocol, by converting it to a non-interactive proof. Define the space of the public/private key pairs, how to compute the signature of a message  $M$ , and how to verify the signature.

**Problem 4 (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:  $\text{Cert}_A, N_A \parallel N_B \parallel B \parallel \text{sig}_A(N_A \parallel N_B \parallel B)$ ;
3. Alice  $\leftarrow$  Bob:  $\text{Cert}_B, N'_B \parallel N_A \parallel A \parallel \text{sig}_B(N'_B \parallel N_A \parallel A)$ ;

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 5 (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.

**Problem 6 (15 pts)** In the Bellare-Micali 1-out-of-2 Oblivious Transfer protocol,  $A$  has two inputs  $x_1, x_2 \in G$ , where  $G$  is a group of prime order  $p$ , and  $B$  has  $b \in \{0, 1\}$ . The protocol is as follows:  $A$  publishes a random  $c \in G$ ,  $B$  chooses random  $k \in \mathbb{Z}_p$  and sends  $PK_b = g^k$  and  $PK_{1-b} = c/g^k$  to  $A$ ,  $A$  checks that  $PK_0 \cdot PK_1 = c$ , and sends  $C_0 = \langle g^{r_0}, H(PK_0^{r_0}) \oplus x_0 \rangle$ ,  $C_1 = \langle g^{r_1}, H(PK_0^{r_1}) \oplus x_1 \rangle$  to  $B$ .

- a. Describe how  $B$  can get  $x_b$ ?
- b. Prove that this protocol is oblivious.
- c. Why  $B$  cannot recover both  $x_0$  and  $x_1$ ?
- d. Assume that in the protocol,  $A$  does not check  $PK_0 \cdot PK_1 = c$ ; describe how  $B$  can get both  $x_0$  and  $x_1$ .