CS 798

Assignment 2

due 11:30 AM, October 22

1. Consider the following mutual identification protocol:

   Protocol: PUBLIC-KEY MUTUAL AUTHENTICATION

   1. Bob chooses a random challenge, r_1. He also computes

      $$y_1 = sig_{Bob}(r_1)$$

      and sends $\mathbf{Cert}(Bob)$, r_1 and y_1 to Alice.
   2. Alice verifies Bob's public key, ver_{Bob}, on the certificate $\mathbf{Cert}(Bob)$. Then she checks that

      $$ver_{Bob}(r_1, y_1) = true.$$

      If not, then Alice outputs "reject" and quits. Otherwise, Alice chooses a random challenge, r_2. She also computes

      $$y_2 = sig_{Alice}(r_1)$$

      and

      $$y_3 = sig_{Alice}(r_2)$$

      and sends $\mathbf{Cert}(Alice)$, r_2, y_2 and y_3 to Bob.
   3. Bob verifies Alice's public key, ver_{Alice}, on the certificate $\mathbf{Cert}(Alice)$. Then he checks that

      $$ver_{Alice}(r_1, y_2) = true$$

      and

      $$ver_{Alice}(r_2, y_3) = true.$$

      If so, then Bob outputs "accept"; otherwise, Bob outputs "reject". Bob also computes

      $$y_4 = sig_{Bob}(r_2)$$

      and sends y_4 to Alice.
   4. Then she checks that

      $$ver_{Bob}(r_2, y_4) = true.$$

      If so, then Alice outputs "accept"; otherwise, Alice outputs "reject".

   Prove that the above protocol is insecure. (In particular, show that Oscar can impersonate Bob by means of a certain type of parallel session attack, assuming that Oscar has observed a previous session of the protocol between Alice and Bob.)

2. Prove that the Guillou-Quisquater Identification Scheme is honest-verifier zero-knowledge.

   Note: If you wish, you can assume in your proof that $\gcd(u, n) = 1$. This will be true with high probability for any secure implementation of the scheme. However, the proof should work regardless of whether you make this assumption.

3. Give a complete analysis of the adversary's probability of carrying out a successful impersonation attack in Protocol 2.4 from the notes.

   Note: The security assumption for MACs specifies the number of previous MACs observed by the adversary, before he attempts to forge a new MAC, to be q. There are two MACs created in each session of the identification protocol. Therefore the analysis of the protocol requires that we assume that at most $q/2$ sessions of the protocol take place before the adversary carries out an active attack.

4. Prove that it is impossible to design a secure two-flow mutual identification protocol of the general type studied in class (i.e., based on random challenges).

   Note: A two-flow protocol consists of one flow from Bob to Alice (say) followed by a flow from Alice to Bob. In a mutual identification protocol, both parties are required to "accept".

5. Suppose that the Blom scheme is implemented with security parameter k. Suppose that a coalition of k bad users, say $W_1, \ldots, W_k$, pool their secret information. Additionally, assume that a key $K_{U,V}$ is exposed, where U and V are two other users.

   (a) Describe how the coalition can determine the polynomial $g_U(x)$ by polynomial interpolation, using known values of $g_U(x)$ at $k + 1$ points.

   (b) Having computed $g_U(x)$, describe how the coalition can compute the bivariate polynomial $f(x, y)$ by bivariate polynomial interpolation.

   (c) Illustrate the preceding two steps, by determining the polynomial f in the following sample implementation of the Blom Scheme. $k = 2$, $p = 101$, $r_i = i$ $(1 \leq i \leq 4)$, $g_1(x) = 79 + 85x + 71x^2$, $g_2(x) = 31 + 45x + 53x^2$, $K_{3,4} = 18$.

6. Suppose that Alice uses the **Okamoto Scheme** with $q = 1201$, $p = 122503$, $t = 10$, $\alpha_1 = 60497$ and $\alpha_2 = 17163$. Suppose also that $v = 119504$.

   (a) Verify that
   $$\alpha_1^{70}\alpha_2^{1033}v^{877} \equiv \alpha_1^{248}\alpha_2^{883}v^{992} \pmod{p}.$$

   (b) Show how Olga can use this information to compute b_1 and b_2 such that
   $$\alpha_1^{-b_1}\alpha_2^{-b_2} \equiv v \pmod{p}.$$

   (c) Now suppose that Alice reveals that $a_1 = 484$ and $a_2 = 935$. Show how Alice and Olga together will compute $\log_{\alpha_1} \alpha_2$.

Note that the computations in the last two questions can be done very easily using Maple