CS 798

Solutions to Assignment 2

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 **Cert**(Bob), r_1 and y_1 to Alice.

   2. Alice verifies Bob's public key, ver_{Bob}, on the certificate **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 **Cert**$(Alice)$, r_2, y_2 and y_3 to Bob.

   3. Bob verifies Alice's public key, ver_{Alice}, on the certificate **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.)

Answer:

First, Oscar observes a session $\mathcal{S}_1$ between Bob (the initiator) and Alice (the responder). In the first flow of $\mathcal{S}_1$, Bob chooses a random number r_1, computes a signature $y_1 = sig_{Bob}(r_1)$, and sends r_1 and y_1 to Alice. Oscar records the values of r_1, y_1, and $\mathbf{Cert}(Bob)$ from $\mathcal{S}_1$.

Later Oscar initiates a session $\mathcal{S}_2$ with Alice, in which he pretends to be Bob. He sends r_1, y_1, and $\mathbf{Cert}(Bob)$ to Alice in the first flow of $\mathcal{S}_2$. Alice responds with some information which includes a random challenge r_2' and a signature $y_3' = sig_{Alice}(r_2')$.

Now Oscar has to obtain Bob's signature on r_2'. Oscar does this by initiating another session, $\mathcal{S}_3$, with Bob, in which he (Oscar) pretends to be Alice. Oscar sends r_2', y_3', and $\mathbf{Cert}(Alice)$ to Bob in the first flow of $\mathcal{S}_3$. Bob's response includes $sig_{Bob}(r_2')$. Oscar can forward this response to Alice as the final flow of $\mathcal{S}_2$.

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

Answer:

Let $\mathcal{T}$ consist of all possible triples $T = (\gamma, r, y)$, where $\gamma, y \in \mathbb{Z}_n$, $r \in \{0, \ldots, b-1\}$ and

$$\gamma \equiv v^r y^b \pmod{n}. \tag{1}$$

The values v and b are fixed in the scheme.

Clearly r and y determine γ uniquely in the congruence (1). There are b choices for r and n choices for y; therefore $|\mathcal{T}| = nb$.

A real session of Guillou-Quisquater yields a transcript $T \in \mathcal{T}$. However, the adversary can forge transcripts in $\mathcal{T}$ as follows:

(1) choose $y \in \mathbb{Z}_n$ randomly

(2) choose $r \in \{0, \ldots, b-1\}$ randomly

(3) compute $\gamma = v^r y^b \bmod n$.

Let $\mathsf{Pr}_{real}[T]$ denote the probability that the triple T is the transcript of a real session of Guillou-Quisquater; and let $\mathsf{Pr}_{forge}[T]$ denote the probability that the triple T is a forged transcript produced by the above-described forging algorithm.

We claim that $\mathsf{Pr}_{real}[T] = \mathsf{Pr}_{forge}[T] = 1/(nb)$ for all $T \in \mathcal{T}$. As a consequence, forged transcripts are indistinguishable from real transcripts, and hence the protocol is honest-verifier zero-knowledge.

It remains to prove the claim. It is clear that $\mathsf{Pr}_{forge}[T] = 1/(nb)$ for all $T \in \mathcal{T}$ by the way in which forged transcripts are constructed. So we just have to show that $\mathsf{Pr}_{real}[T] = 1/(nb)$ for all $T \in \mathcal{T}$.

This can be done in one of two ways: algebraically or by counting (i.e., using the pigeonhole principle).

Algebraic approach: Using the fact that b is a valid RSA exponent, it must be the case that $a = b^{-1} \bmod \phi(n)$ exists. Then it is not hard to verify that

$$y \equiv \gamma^a u^r \pmod{n}.$$

This proves that γ and r determine y uniquely. Moreover, for fixed r, there is a one-to-one correspondence (i.e., a bijection) between the n possible values of γ and the n possible values of y in a triple $T \in \mathcal{T}$.

Now we can compute $\mathsf{Pr}_{real}[T]$. k is chosen randomly, and hence $\gamma = k^b \bmod n$ is a random element of $\mathbb{Z}_n$. r is chosen randomly, and γ and r determine a unique value of y such that $(\gamma, r, y) \in \mathcal{T}$, using the result proved above. This proves that every real transcript occurs with the same probability, $1/(nb)$, and the proof is complete.

Counting approach: k is chosen randomly, and hence $\gamma = k^b \bmod n$ is a random element of $\mathbb{Z}_n$. r is also chosen randomly, and then y is computed to be $y = ku^r \bmod n$. The resulting triple $(\gamma, r, y) \in \mathcal{T}$. Since there are n choices for k and b choices for r, there are nb possible transcripts produced in this way, all having the same probability $1/(nb)$. However, $|\mathcal{T}| = nb$, so there are no other transcripts, and the proof is complete.

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.

   Answer:

   We assume the MAC is (q, ϵ)-*secure*: the adversary cannot construct a valid MAC for any new message with probability greater than ϵ, given that the adversary has previously seen valid MACs for at most q messages.

   Suppose the adversary has (passively) observed $q/2$ sessions between Bob and Alice, in which a total of q valid MACs have been transmitted between them. Then the adversary attempts to have Alice or Bob "accept" in some new session, say $\mathcal{S}_{new}$, in which the adversary is active.

   Suppose that the random challenges are (positive) k-bit integers. Under these conditions, we can easily give an upper bound on the adversary's probability of deceiving at least one of Alice or Bob. The adversary can reuse a MAC from an old session if a challenge is repeated by coincidence, or he can try to guess (i.e., forge) a MAC for a new challenge.

   (a) The probability that Bob repeats one of his challenges, say r_1, from a previous session, say $\mathcal{S}$, is at most $(q/2) \times 2^{-k}$. In the session $\mathcal{S}$, Alice responded with r_2 and $y_1 = MAC_K(Alice \parallel r_1 \parallel r_2)$, for some r_2. Oscar can use the same r_2 and y_1 in the session $\mathcal{S}_{new}$ to deceive Bob.

   (b) The probability that Alice repeats one of her challenges, say r_2, from a previous session, say $\mathcal{S}$, is at most $(q/2) \times 2^{-k}$. In the session $\mathcal{S}$, Bob responded with $y_2 = MAC_K(Bob \parallel r_2)$. Oscar can use the same y_2 in the session $\mathcal{S}_{new}$ to deceive Alice.

   (c) Oscar can guess a new MAC, $MAC_K(Alice \parallel r_1 \parallel r_2)$, where r_1 and r_2 are the (new) challenges chosen by Alice and Bob in the session $\mathcal{S}_{new}$, with probability at most ϵ. A correct guess will allow Oscar to deceive Bob.

   (d) Oscar can guess a new MAC, $MAC_K(Bob \parallel r_2)$, where r_2 is the (new) challenge chosen by Alice in the session $\mathcal{S}_{new}$, with probability at most ϵ. A correct guess will allow Oscar to deceive Alice.

Summing up, Oscar's probability of deceiving Alice or Bob is at most $2 \times ((q/2)/2^k + \epsilon) = q/2^k + 2\epsilon$.

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".

   Answer:

   Suppose that Oscar observes a session $\mathcal{S}_1$ between Alice and Bob, in which they both accept. Let y_1 denote the information that Alice receives from Bob in the first flow. Alice must "accept" based only on the information y_1, because that is all the information she receives from Bob in the session $\mathcal{S}_1$. Oscar can record the value y_1 and send it to Alice in a new session $\mathcal{S}_2$ in which he is pretending to be Bob. Alice will "accept" y_1 in $\mathcal{S}_2$, and therefore Oscar is successfully able to impersonate Bob.

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.

   Answer: The coalition knows the k values

$$\begin{aligned} g_U(r_{W_1}) &= g_{W_1}(r_U) \\ g_U(r_{W_2}) &= g_{W_2}(r_U) \\ &\vdots \\ g_U(r_{W_k}) &= g_{W_k}(r_U). \end{aligned}$$

   Additionally, the coalition knows

$$g_U(r_V) = K_{U,V}.$$

   Therefore they have $k+1$ points on the polynomial $g_U(x)$, which has degree at most k, and the coalition can compute $g_U(x)$ by the interpolation formula.

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

   Answer: The coalition knows the $k+1$ polynomials

$$\begin{aligned} g_{W_1}(x) &= f(x, r_{W_1}) \\ g_{W_2}(x) &= f(x, r_{W_2}) \\ &\vdots \\ g_{W_k}(x) &= f(x, r_{W_k}), \quad \text{and} \\ g_U(x) &= f(x, r_U). \end{aligned}$$

   Therefore the coalition can compute $f(x,y)$ by the bivariate interpolation formula.

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

Answer: First, $g_4(1) = g_1(4) = 40$, $g_4(2) = g_2(4) = 49$ and $g_4(3) = K_{3,4} = 18$. Interpolating, $g_4(x) = 81x^2 + 69x + 92$. Next, interpolating $g_1(x), g_2(x)$ and $g_4(x)$, we obtain

$$f(x, y) = 11 + 25x + 25y + 43x^2 + 43y^2 + 9xy + 51x^2y + 51xy^2 + 78x^2y^2.$$

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

$$\alpha_1{}^{70}\alpha_2{}^{1033}v^{877} \equiv \alpha_1{}^{248}\alpha_2{}^{883}v^{992} \pmod{p}.$$

Answer: Both sides of the congruence equal 31939.

(b) Show how Olga can use this information to compute b_1 and b_2 such that

$$\alpha_1{}^{-b_1}\alpha_2{}^{-b_2} \equiv v \pmod{p}.$$

Answer:

$$\begin{aligned} b_1 &= (70 - 248)(877 - 992)^{-1} \bmod 1201 = 1119, \quad \text{and} \\ b_2 &= (1033 - 883)(877 - 992)^{-1} \bmod 1201 = 312. \end{aligned}$$

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

Answer:

$$\log_{\alpha_1} \alpha_2 = (484 - 1119)(312 - 935)^{-1} \bmod 1201 = 562.$$