CS 758

Assignment 2

due by Noon on Friday, October 18, 2013

General Instructions

- Assignments must be handed in by the designated time. I prefer an electronic copy (.pdf file) submitted by email, but I will also accept a hard copy that is dropped off in class or at my office. Due to the large number of students in the class, no extensions will be given.
- You are permitted to discuss assignment questions *informally* with a partner. However, when you hand in the assignments you must indicate who you had discussions with, and all solutions must be written up separately. No "copied" solutions or computer code are allowed; copied assignments are considered to be plagiarism and are subject to severe penalties.
- It is recommended that you attempt to solve every question on your own. If you do use an external information source (a book or a research paper, for example), give an appropriate citation.
- Computational questions can be done using any desired programming tools, languages or calculators. Maple is **strongly recommended** as the easiest way to do the questions. Please include all source code, and sufficient output so I can verify the main steps in the computations.
- I am available to provide extra help or hints if you get stuck on a question.

Questions

In the questions below, numbered theorems, etc., refer to the mathematics notes on the course web page.

1. Consider the following possible identification scheme. Alice possesses a secret key $n = pq$, where p and q are prime and $p \equiv q \equiv 3 \pmod 4$. The value of n will be stored on Alice's certificate. When Alice wants to identify herself to Bob, say, Bob will present Alice with a random quadratic residue modulo n, say x. Then Alice will compute a square root y of x and give it to Bob. Bob then verifies that $y^2 \equiv x \pmod n$.

   (a) Explain why this scheme is insecure against Bob, i.e., show how Bob can impersonate Alice with some non-negligible probability after Alice identifies herself to him.

   Answer: This is basically the same as the chosen-ciphertext attack against the Rabin Cryptosystem. Bob chooses a random $r \in \mathbb{Z}_n$. If $\gcd(r, n) > 1$ then Bob immediately obtains the factorization (this happens with negligible probability, however). Otherwise, Bob computes $x = r^2 \bmod n$, which is a random quadratic residue in $\mathbb{Z}_n$. Alice then gives Bob a value y which is a square root of x modulo n. Since Alice does not know the value of r that Bob initially chose, the probability that $y \not\equiv \pm r \bmod n$ is $1/2$. In this case, the computation of $\gcd(y + r, n)$ yields the factorization $n = pq$. Once Bob has this factorization, he can impersonate Alice.

(b) Is the scheme zero-knowledge from the point of view of an observer, say Oscar, who simply watches the protocol being executed by a verifier who is choosing challenges randomly? Explain.

Answer: Note that proving a scheme is zero-knowledge requires showing transcripts of sessions can be simulated.

Suppose we assume that Alice chooses the square root y of x randomly. Then it is easy for Oscar to simulate transcripts (x, y): he chooses a random $x \in \mathbb{Z}_n$ and checks if $\gcd(x, n) = 1$. If so, then Oscar computes $y = x^2 \bmod n$ and creates the transcript. Note that the value y is a random quadratic residue. The generated transcripts have the same probability distribution as the real transcripts created by Bob. (Note that I gave full marks for this answer, even if the above assumption regarding Alice was not stated.)

What happens if Alice does not choose the square root y randomly? (For example, Alice may compute the four square roots of x and then always return the smallest one.) There does not seem to be a way for Oscar to simulate transcripts if Alice behaves in some non-random way.

2. Consider the mutual authentication scheme defined in pages 21–25 of the FIPS 196 standard. (This document is not currently available from the NIST website due to the U.S. government shutdown, but a copy is posted on the course webpage.) Show that this scheme is insecure if the optional field $[A]$ is omitted from sS_B and the optional field $[B]$ is omitted from sS_A.

Answer: The basic structure of the protocol (with optional fields omitted) is as follows:

step 1 Bob sends a random challenge r_B to Alice.

step 2 Alice sends a random challenge r_A to Bob along with $sig_{Alice}(r_A, r_B)$.

step 3 Bob sends $sig_{Bob}(r_B, r_A)$ to Alice.

In the attack, Oscar impersonates Bob while identifying himself to Alice (Oscar legitimately identifies himself to Bob).

step 1 Bob sends a random challenge r_B to Oscar, who forwards it to Alice.

step 2 Alice sends a random challenge r_A to Oscar (who is impersonating Bob) along with $sig_{Alice}(r_A, r_B)$. Oscar forwards r_A to Bob but he replaces $sig_{Alice}(r_A, r_B)$ with $sig_{Oscar}(r_A, r_B)$.

step 3 Bob sends $sig_{Bob}(r_B, r_A)$ to Oscar, who forwards it to Alice.

Note that Bob accepts after Oscar is active.

It is also possible for Oscar to impersonates Alice while identifying himself to Bob (Oscar legitimately identifies himself to Alice).

step 1 Bob sends a random challenge r_B to Oscar, who forwards it to Alice.

step 2 Alice sends a random challenge r_{Alice} to Oscar along with $sig_{Alice}(r_A, r_B)$. Oscar forwards r_A and $sig_{Alice}(r_A, r_B)$ to Bob.

step 3 Bob sends $sig_{Bob}(r_B, r_A)$ to Oscar (impersonating Alice) Oscar sends $sig_{Oscar}(r_B, r_A)$ to Alice.

Here, Alice accepts after Oscar is active.

Other attacks are also possible.

3. Give a complete analysis of the adversary's probability of successful impersonation in the *Public-Key Mutual Identification Protocol* from slides 134–135. You should explicitly state all your assumptions, and your answer should be given in terms of q, k and ϵ. You can assume that the adversary is passive during an information-gathering phase and the signature scheme is secure in a known-message attack. The security assumption for the signature scheme will specify the number of previous signatures observed by the adversary, before he attempts to forge a new signature, to be q. There are two signatures created in each session of the identification protocol, one by each participant. These signatures are

created using two different keys. Therefore, in the analysis of the protocol, it is reasonable to make the assumption that at most q sessions of the protocol take place before the adversary carries out an impersonation attack.

Answer:

We assume the signature scheme is (q, ϵ)-*secure*: the adversary cannot construct a valid signature for any new message with probability greater than ϵ, given that the adversary has previously seen valid signatures for at most q messages. This assumption holds for Alice's signature scheme, as well as for Bob's signature scheme.

Suppose the adversary has (passively) observed q sessions between Bob and Alice, in which there are a total of q messages signed by Bob and q messages signed by Alice. Then the adversary attempts to have Alice or Bob "accept" in some new session, say $\mathcal{S}_{\text{new}}$, 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 signature from an old session if a challenge is repeated by coincidence, or he can try to guess (i.e., forge) a signature for a new challenge.

First we consider a new session $\mathcal{S}_{\text{new}}$ from Bob's point of view. He chooses r_1 and receives r_2 and $sig_{\text{Alice}}(\text{ID(Bob)} \parallel r_1 \parallel r_2)$. He has to consider the probability that this signature is

(a) newly created by Oscar,

(b) copied from the second flow of some other session $\mathcal{S}_{\text{old}}$, or

(c) copied from the third flow of some other session $\mathcal{S}_{\text{old}}$.

In case (a), Oscar can guess the new signature with probability at most ϵ. A correct guess will allow Oscar to deceive Bob.

In case (b), Oscar can copy the signature if Bob repeated his challenge r_1 from the session $\mathcal{S}_{\text{old}}$. In the session $\mathcal{S}_{\text{old}}$, Alice responded with r_2 and $y_1 = sig_{\text{Alice}}(\text{Bob} \parallel r_1 \parallel r_2)$, for some r_2. Oscar can use the same r_2 and y_1 in the session $\mathcal{S}_{\text{new}}$ to deceive Bob. Here, Bob must repeat his challenge from a previous session. Therefore the probability of (b) holding is at most $q/2^k$.

Case (c) cannot occur, because the information being signed in the two flows has a different structure.

Now we consider a session $\mathcal{S}_{\text{new}}$ from Alice's point of view. She receives r_1 and then she chooses r_2 and computes $sig_{\text{Alice}}(\text{ID(Bob)} \parallel r_1 \parallel r_2)$. Finally, she receives $sig_{\text{Bob}}(\text{ID(Alice)} \parallel r_2)$. She has to consider the probability that this signature is

(d) newly created by Oscar,

(e) copied from the second flow of some other session $\mathcal{S}_{\text{old}}$, or

(f) copied from the third flow of some other session $\mathcal{S}_{\text{old}}$.

In case (d), Oscar can guess a new signature, $sig_{\text{Bob}}(\text{Alice} \parallel r_2)$, with probability at most ϵ. A correct guess will allow Oscar to deceive Alice.

Case (e) cannot occur, because the information being signed in the two flows has a different structure.

In case (f), Alice repeats one of her challenges, say r_2, from the session $\mathcal{S}_{\text{old}}$, is at most $q \times 2^{-k}$. In the session $\mathcal{S}_{\text{old}}$, Bob responded with $y_2 = sig_{\text{Bob}}(\text{Alice} \parallel r_2)$. Oscar can use the same y_2 in the session $\mathcal{S}_{\text{new}}$ to deceive Alice. In case (f), Alice must repeat her challenge from a previous session. Therefore the probability of (f) holding is at most $q/2^k$.

Summing up, Oscar's probability of deceiving Alice or Bob is at most $2 \times (q/2^k + \epsilon) = q/2^{k-1} + 2\epsilon$.

4. Show that the *Schnorr Identification Scheme* is not secure against an active adversary who changes the messages that are sent *from Alice to Bob*.

Answer: There are many attacks; here is one. In step 1, Oscar intercepts γ and replaces it by $\gamma' = \alpha\gamma$. Then in step 3, Oscar intercepts y and replaces it by $y' = y + 1$. Since $\gamma = \alpha^y v^r$, it follows that

$$\gamma' = \alpha\gamma = \alpha\alpha^y v^r = \alpha^{y+1} v^r = \alpha^{y'} v^r.$$

5. Consider the following *Diffie-Hellman* type identification scheme. Alice has a public key $v = g^a$ and a private key a. Assume that g is a primitive element in a ${\mathbb{Z}_p}^*$ where p is prime, and assume that the Discrete Logarithm Problem in ${\mathbb{Z}_p}^*$ is infeasible. Bob chooses a random b, computes $w = g^b$, and sends w to Alice. Alice computes $K = w^a$ and sends it to Bob. Bob accepts if and only if $K = v^b$.

   (a) Is the above-described scheme *zero-knowledge* against an honest verifier (i.e., a verifier who chooses the challenges w as described above)?

   Answer: A transcript representing the view of an *honest* verifier can be written as a triple (b, w, k), where b is random, $w = g^b$ and $k = v^b$. Clearly the honest verifier can generate random triples of this form, so the scheme is zero-knowledge for an honest verifier.

   (b) Is the above-described scheme *zero-knowledge* against a dishonest verifier (i.e., a verifier who chooses the challenges w in some other way)?

   Answer: (Note: I treated this as a bonus question.) Suppose a dishonest verifier chooses challenges $w \in {\mathbb{Z}_p}^*$ (for which he does not know the corresponding b-values) according to some *non-uniform* probability distribution, without choosing b first. Such a verifier has no apparent way to compute the correct k values for a given v. Furthermore, he has no apparent way to generate this probability distribution on the v-values by first choosing b (according to some probability distribution) and then computing $w = g^b$. Therefore there does not seem to be any way for him to simulate valid transcripts of the form (v, k).