CS255: Cryptography and Computer Security **Winter 2004**

Assignment #3

Due: Monday, March 1st, 2004.

Problem 1. Repeated squaring Algorithm.

a. The number $p = 521$ is prime. Use Euler's theorem to decide whether 2 is a quadratic residue modulo this p. Show all intermediate steps in your calculation.

b. The number $N = 1517$ factors as $N = 37 * 41$. Is 2 a quadratic residue modulo this N? Show all intermediate steps in your calculation.

Problem 2 Parties $A_1, \ldots, A_n$ 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 $b \in [1, q-1]$ and computes $y = g^b \bmod p$. Each party A_i picks a secret random $a_i \in [1, q-1]$ and computes $x_i = g^{a_i} \bmod p$. User A_i sends x_i to B. User B responds to party i by sending $z_i = x_i^b \bmod p$.

a. Show that party i given z_i (and a_i) can determine y.

b. Explain why (a hash of) y can be securely used as the conference key. Namely, explain why at the end of the protocol all parties $A_1, \ldots, A_n$ and B know y and give a brief informal explanation why an eavesdropper cannot determine y.

c. 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 y, then there also exists an efficient algorithm $\mathcal{B}$ that breaks the Computational Diffie-Hellman assumption (using p and g as the public values). Use algorithm $\mathcal{A}$ as a subroutine in your algorithm $\mathcal{B}$. Note that algorithm $\mathcal{B}$ takes $g^a \bmod p$ and $g^b \bmod p$ as input and should output $g^{ab} \bmod p$.

Problem 3 Commitment schemes. A commitment scheme enables Alice to commit a value x to Bob. The scheme is *secure* if the commitment does not reveal to Bob any information about the committed value x. At a later time Alice may *open* the commitment and convince Bob that the committed value is x. The commitment is *binding* if Alice cannot convince Bob that the committed value is some $x' \neq x$. Here is an example commitment scheme:

Public values: (1) a 1024 bit prime p, and (2) two elements g and h of $\mathbb{Z}_p^*$ of prime order q.

Commitment: To commit to an integer $x \in [0, q-1]$ Alice does the following: (1) she picks a random $r \in [0, q-1]$, (2) she computes $b = g^x \cdot h^r \bmod p$, and (3) she sends b to Bob as her commitment to x.

Open: To open the commitment Alice sends (x, r) to Bob. Bob verifies that $b = g^x \cdot h^r \bmod p$.

Show that this scheme is secure and binding.

a. To prove security show that b does not reveal any information to Bob about x. In other words, show that given b, the committed value can be any integer x' in $[0, q-1]$.
Hint: show that for any x' there exists a unique $r' \in [0, q-1]$ so that $b = g^{x'} h^{r'}$.

b. To prove the binding property show that if Alice can open the commitment as (x', r') where $x \neq x'$ then Alice can compute the discrete log of h base g. In other words, show that if Alice can find an (x', r') such that $b = g^{x'} h^{r'} \bmod p$ then she can find the discrete log of h base g. Recall that Alice also knows the (x, r) used to create b.

Problem 4 Let's explore why in the RSA public key system each person has to be assigned a different modulus $N = pq$. Suppose we try to use the same modulus $N = pq$ for everyone. Each person is assigned a public exponent e_i and a private exponent d_i such that $e_i \cdot d_i = 1 \bmod \varphi(N)$. At first this appears to work fine: to encrypt a message to Bob, Alice computes $C = M^{e_{bob}}$ and sends C to Bob. An eavesdropper Eve, not knowing d_{bob} appears to be unable to decrypt C. Let's show that using e_{eve} and d_{eve} Eve can very easily decrypt C.

a. Show that given e_{eve} and d_{eve} Eve can obtain a multiple of $\varphi(N)$.

b. Show that given an integer K which is a multiple of $\varphi(N)$ Eve can factor the modulus N.
Hint: Consider the sequence $g^K, g^{K/2}, g^{K/4}, \ldots g^{K/\tau(N)} \bmod N$ where g is random in $\mathbb{Z}_N$ and $\tau(N)$ is the largest power of 2 dividing K. Use the the left most element in this sequence which is not equal to $1 \bmod N$.

c. Deduce that Eve can decrypt any RSA ciphertext encrypted using the modulus N intended for Alice or Bob (at this point this should be obvious).

Problem 5 Let N be a 1024 bit RSA modulus, and d a secret signing exponent. To protect the private key d one may wish to split it into three pieces and store each piece on a different server. An attacker who breaks into one of the servers should learn no information about d. Consider the following scheme: pick three random numbers d_1, d_2, d_3 in $[0, \varphi(N)]$ so that $d_1 + d_2 + d_3 = d \bmod \varphi(N)$. Store d_i on server i.

a. Suppose Alice wants to sign a message M. Show that Alice can do the following: (1) she sends M to the three servers, (2) each server i performs a local computation (using d_i) and responds with S_i to Alice, and (3) given S_1, S_2, S_3 Alice can easily construct the signature S. Explain how server i computes S_i and how Alice combines S_1, S_2, S_3 to obtain S. You have just shown that a digital signing key can be broken into three pieces and used without ever reconstructing the key.

b. To provide fault tolerance, show how the key d can be shared among the three servers so that any two of the three can be used to sign a message M as in part (a). You may store multiple d_i's on each server. An attacker who breaks into one of the servers should learn no information about d. As in part (a), your solution should not reconstruct the key d and there should be no interaction between the servers.

c. Briefly explain how your solution for part (b) can be generalized to provide a t-out-of-k solution in the same settings. Namely, explain how the key can be shared among k servers so that any t of them can be used to sign M while an attack on $t-1$ servers reveals no information about d.

Problem 6 Recall that a simple RSA signature $S = H(M)^d \bmod N$ is computed by first computing $S_1 = H(M)^d \bmod p$ and $S_2 = H(M)^d \bmod q$. The signature S is then found by combining S_1 and S_2 using the Chinese Remainder Theorem (CRT). Now, suppose a Certificate Authority (CA) is about to sign a certain certificate C. While the CA is computing $S_1 = H(C)^d \bmod p$, a glitch on the CA's machine causes it to produce the wrong value $\tilde{S}_1$ which is not equal to S_1. The CA computes $S_2 = H(C)^d \bmod q$ correctly. Clearly the resulting signature $\tilde{S}$ is invalid. The CA then proceeds to publish the newly generated certificate with the invalid signature $\tilde{S}$.

a. Show that any person who obtains the certificate C along with the invalid signature $\tilde{S}$ is able to factor the CA's modulus.
Hint: Use the fact that $\tilde{S}^e = H(C) \bmod q$. Here e is the public verification exponent.

b. Suggest some method by which the CA can defend itself against this danger.

Extra credit: In the lecture we defined the Decision Diffie-Hellman problem (DDH) as follows: let G be a group of prime order q. An algorithm $\mathcal{A}$ ϵ-solves the DDH problem in G if:

$$\left|\Pr[\mathcal{A}(g, g^a, g^b, g^{ab}) = \text{“yes''}] - \Pr[\mathcal{A}(g, g^a, g^b, g^c) = \text{“yes''}]\right| \geq \epsilon$$

where $g \neq 1$ is uniform in G and a, b, c are uniform in $\mathbb{Z}_q^*$. In other words, $\mathcal{A}$ is able to distinguish between a distribution of Diffie-Hellman tuples and a distribution of random tuples in G.

We also said that an algorithm $\mathcal{B}$ ϵ-breaks the semantic security of a public-key encryption scheme $\mathcal{E}$ if $\mathcal{B}$ wins the following game with probability at least $\frac{1}{2} + \epsilon$:
(1) $\mathcal{B}$ is given a public-key generated by the key generation algorithm of $\mathcal{E}$,
(2) $\mathcal{B}$ outputs two messages M_0, M_1,
(3) $\mathcal{B}$ is given the public key encryption of M_b under the public key from step (1) where b is random in $\{0, 1\}$,
(4) $\mathcal{B}$ returns a $b' \in \{0, 1\}$ and wins the game if $b = b'$.

Consider the original ElGamal encryption scheme where the encryption of a message $M \in G$ is $C = [g^r,\ M \cdot y^r]$ where $\langle g, y \rangle \in G$ is the public key and r is random in $\mathbb{Z}_q$. Show that this ElGamal encryption scheme is semantically secure assuming DDH in G is hard. In other words, show that if an algorithm $\mathcal{B}$ ϵ-breaks semantic security of

ElGamal in G then there is an algorithm $\mathcal{A}$ with approximately the same running time as $\mathcal{B}$ that ϵ-breaks DDH in G. (your goal is to design algorithm $\mathcal{A}$ for DDH in G that uses $\mathcal{B}$ as a subroutine).