CS255: Cryptography and Computer Security **Winter 2001**

Final Exam

Instructions
− Answer **four** of the following six problems. Do not answer more than four.
− The exam is open book and open notes.
− You have two hours.

Problem 1 Questions from all over.

a. Data compression is often used in data storage or transmission. Suppose you want to use data compression in conjunction with encryption. Does it make more sense to first encrypt the data and then compress or visa versa?

b. Explain the purpose of a public salt in password storage.

c. Explain the purpose of a secret salt in password storage.

d. Explain why reusing a stream cipher key for multiple messages is a bad idea.

e. Explain the purpose of the IV in CBC encryption. Recall that a different IV must be used for every message being encrypted. Explain what goes wrong if the same IV is used for encrypting all messages.

Problem 2 Recall that the standard Diffie-Hellman protocol is defined modulo a prime p. In this problem we will show that breaking the Diffie-Hellman protocol when working modulo a composite N is as hard as factoring N.

a. Let $N = pq$ be a large composite. Show that if there is an efficient algorithm for computing square roots modulo N then there is also an efficient randomized algorithm for factoring N. That is, suppose there is an efficient algorithm $\mathcal{A}$ that given $x \in \mathbb{Z}_N$ outputs some $y \in \mathbb{Z}_N$ such that $y^2 = x \bmod N$. Show that there is an efficient algorithm $\mathcal{B}$ (that uses $\mathcal{A}$) such that given N, algorithm $\mathcal{B}$ outputs p and q.

b. Let $N = pq$ and let $g \in \mathbb{Z}_N$ be an element of order $\varphi(N)/4$. Further assume that $\varphi(N)/4$ is odd. Let $h = g^2$. Suppose there is an efficient algorithm $\mathcal{A}$ to break the Diffie-Hellman protocol base h. That is, given h^x and h^y algorithm $\mathcal{A}$ will output $h^{xy} \bmod N$. Show that there is an algorithm $\mathcal{B}$ for computing square roots in $\mathbb{Z}_N$ for all elements in the group generated by h.

c. Briefly explain how parts (a) and (b) can be used to reduce the problem of factoring N to the problem of computing the Diffie-Hellman function base $h \in \mathbb{Z}_N$.

Problem 3 MAC Construction. Let p be a 128-bit prime. Let E_k be a symmetric encryption scheme. Define the following MAC for messages $(m_1, m_2) \in \mathbb{Z}_p^2$: $\text{MAC}(m_1, m_2) = E_k[am_1 + bm_2 \bmod p]$ where k and $a, b \in \mathbb{Z}_p$ are shared random secrets between Alice and Bob. We will show that this is a secure MAC.

a. Let $f_{a,b}(x_1, x_2) = ax_1 + bx_2 \bmod p$. Show that for any two distinct messages (m_1, m_2) and (m_1', m_2') we have that

$$\Pr_{a,b}[f_{a,b}(m_1, m_2) = f_{a,b}(m_1', m_2')] = 1/p$$

Here the probability is over the random choice of a, b in $\mathbb{Z}_p$.

b. Show that this MAC is existentially secure under a chosen message attack assuming E_k is an ideal cipher. More precisely, use part (a) to argue that any MAC forging algorithm $\mathcal{A}$ will succeed in creating an existential forgery with probability at most q^2/p where q is the number of chosen message queries issued by the attacker.

Problem 4 Explain what Certificate Revocation Trees (CRTs) are used for and describe how they work. Describe their advantages and disadvantages when compared to OCSP.

Problem 5 Let E_k be a symmetric encryption scheme encrypting messages in $\{0,1\}^n$. We wish to construct a symmetric encryption scheme $\hat{E}_k$ (based on E_k) for encrypting messages in $\{0,1\}^{n-1}$. To encrypt $M \in \{0,1\}^{n-1}$ we do the following: (1) compute $C_1 = E_k(0\|M)$, (2) if the MSB of C_1 is zero set the ciphertext $\hat{E}_k(M)$ to be the remaining $n-1$ bits of C_1 and stop, (3) otherwise, compute $C_2 = E_k(C_1)$, (4) if the MSB of C_2 is zero set the ciphertext to be the remaining $n-1$ bits of C_2 and stop, (5) repeat this process until the algorithm stops.

a. Show that $\hat{E}_k$ is well defined: show that for any k we have that $\hat{E}_k$ is a permutation on $\{0,1\}^{n-1}$. That is, given a ciphertext $C \in \{0,1\}^{n-1}$ show how to decrypt C. You may assume the algorithm for computing $\hat{E}_k$ always terminates.

b. Show that this construction is secure. To do so, show that if E_k is a random permutation π on $\{0,1\}^n$ (chosen uniformly from the set of $2^n!$ permutations) then $\hat{E}_k$ is a random permutation $\hat{\pi}$ on $\{0,1\}^{n-1}$ (chosen uniformly from the set of $2^{n-1}!$ permutations).
Hint: show that shrinking the domain size from 2^n to $2^n - 1$ using this method is secure. Then use induction to show that shrinking the domain all the way to 2^{n-1} using this method is also secure.

Problem 6 Let $N = pq$ be an RSA composite. Let $g \in [0, N^2]$ be an integer satisfying $g = 1 \bmod N$. Consider the following encryption scheme. The public key is $\langle N, g\rangle$. To encrypt a message $m \in \mathbb{Z}_N$ do: (1) pick a random $h \in \mathbb{Z}_{N^2}$, and (2) compute $C = g^m \cdot h^N \bmod N^2$. Our goal is to develop a decryption algorithm.

a. Show that the discrete log problem base g is easy. That is, show that given g and $B = g^x \bmod N^2$ there is an efficient algorithm to recover $x \bmod N$. Recall that $g = aN + 1$ for some integer a.

b. Show that given g and the factorization of N, decrypting $C = g^m \cdot h^N \bmod N^2$ can be done efficiently.
Hint: consider $C^{\varphi(N)} \bmod N^2$. Use the fact that by Euler's theorem $x^{\varphi(N^2)} = 1 \bmod N^2$ for any $x \in \mathbb{Z}^*_{N^2}$.