CS255: Cryptography and Computer Security **Winter 2010**

Final Exam

Instructions
– **Answer all four questions.**
– The exam is open book and open notes. Wireless devices are not allowed.
– You have two hours.

Problem 1. Questions from all over.

a. Suppose a symmetric cipher uses 128 bit keys to encrypt 1024 bit messages. Can the cipher be perfectly secure? Justify your answer.

b. Suppose a symmetric cipher uses 128 bit keys to encrypt 1024 bit messages. Can the cipher be semantically secure? Justify your answer.

c. Suppose a cipher encrypts 1024 bit messages and the resulting ciphertext is 1024 bits long. Can the cipher be semantically secure under a chosen plaintext attack? (i.e. can we use one key to encrypt more than one message?) Justify your answer.

d. What is the smallest possible positive value of e that can be used to define the RSA trapdoor function? Explain why a smaller value of e cannot be used.

e. Prove that the function $F(k, x) = k \oplus x$ is not a secure PRF.

f. Consider MAC-based challenge-response authentication where the shared secret between the prover and verifier is a human memorizable password. Can a passive eavesdropper who listens in on communications between the prover and verifier carry out a dictionary attack against the prover's password? If not, explain why not. If yes, explain how.

Problem 2. Hash functions.

a. Let's explore what goes wrong if the padding block in the Merkle-Damgard construction does not include the message length. Let $h : X \times M \to X$ be a compression function used in the Merkle-Damgard construction to build a hash function H taking inputs in X^*. Let IV be the initial value used by H. Suppose one finds a fixed point for h, namely a message block m such that $h(\text{IV}, m) = \text{IV}$. Explain how this fixed point can be used to construct collisions for H, assuming the padding block does not contain the length.

b. Recall that SHA-256 uses the Davies-Mayer compression function. Davies-Mayer builds a compression function h from a block cipher (E, D) by defining $h(x, m) := x \oplus E(m, x)$. Show that the designers of SHA-256 could easily choose an IV for which they have a fixed point for h.

c. Prove that if H_1 and H_2 are collision resistant then so is $H(x) := H_1(H_2(x))$.

d. Prove that if H_1 and H_2 are collision resistant, $H(x) := H_1(x) \oplus H_2(x)$ need not be.
Hint: given a collision resistant hash function T, construct collision resistant functions H_1, H_2 such that H is not collision resistant.

e. Recall that in CBC-MAC the IV is fixed. Suppose we chose a random IV for every message being signed and include the IV in the MAC, i.e. $S(k,m) := (r, \text{CBC}_r(k,m))$ where $\text{CBC}_r(k,m)$ refers to the raw CBC function using r as the IV. Describe an existential forgery on the resulting MAC.

Problem 3. Time lock. Our goal in this question is to build a mechanism by which Alice can encrypt a secret S that can be decrypted only after a certain amount of time has passed (e.g. a week, a year, a 100 years).

a. Alice's first solution is as follows. Let (E,D) be a symmetric cipher built from AES. Alice chooses a random AES key k and publishes (C,T) where $C \leftarrow E(k,S)$ and T contains all but t bits of k. Then by exhaustive search the attacker can decrypt C and recover S in time 2^t. By tuning t Alice can choose the time it will take for S to be revealed.

Unfortunately, this approach does not work. Briefly explain how an attacker can recover S in time $2^t/L$ for some L of the attacker's choosing.
Hint: think parallel processing.

b. Alice then remembers that she read somewhere that the best algorithm for computing g^x requires $O(\log x)$ sequential multiplications and that parallel processing cannot speed this up much. She decides to use the following approach. First, she generates two primes p and q and sets $n \leftarrow pq$. Next, she chooses a random g in $\mathbb{Z}_n^*$. Finally, she publishes (n, g, C) where

$$C \leftarrow S + g^{(2^{(2^t)})} \quad \in \mathbb{Z}_n$$

Describe an algorithm that enables anyone to recover S from (n,g,C) using 2^t multiplications in $\mathbb{Z}_n$. Hence, by tuning t Alice can make the puzzle take as long as she wants, even if the attacker mounts your attack from part (a).

c. Finally, show that Alice need not spend time 2^t herself to prepare the puzzle. Show that Alice can use her knowledge of $\varphi(n)$ to construct C using only $O(t)$ multiplications in $\mathbb{Z}_n$.

d. After setting this up Alice wondered if she could use a prime p in place of the RSA modulus n in the system above. Will the resulting time-lock system remain secure if n is replaced by p? If so, explain why. If not, describe an attack.

e. A different approach to time-lock: suppose a trusted party generates ℓ public/private encryption key pairs (e.g. $\ell = 10^5$). It publishes all its public keys at time 0 and keeps all secret keys to itself. Every day at midnight the trusted party publishes one of its secret keys: at day number i it publishes secret key number i on its list. Explain how Alice, and anyone else, can use this trusted party to implement time lock.

Problem 4 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 group G of prime order q and a generator g of G. User B picks a secret random b in $\{0, \ldots, q-1\}$ and computes $y := g^b$. Each party A_i picks a secret random a_i in $\{0, \ldots, q-1\}$ and computes $x_i := g^{a_i}$. For $i = 1, \ldots, n$ user A_i sends x_i to B and B responds to A_i by sending $z_i := x_i^b$.

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

b. Explain why this implies that y can be used as the conference key. Namely, (1) explain why at the end of the protocol all parties know y and (2) write down the complexity assumption you would need to argue that an eavesdropper cannot distinguish y from a random element in G.

c. show that the scheme is not secure against a man-in-the-middle attack. Explain how an active man-in-the-middle can eavesdrop on the conference call.

d. Can you suggest a way to make the scheme secure against a man-in-the-middle attack? You may assume the existence of a Certificate Authority.