CS255: Cryptography and Computer Security **Winter 2009**

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. Does counter mode encryption require a PRP, or is a PRF sufficient? Justify your answer.

b. Does CBC mode encryption require a PRP, or is a PRF sufficient? Justify your answer.

c. Why is it that symmetric key ciphers use relatively short keys (e.g. 128 bits), while public key algorithms such as RSA use much larger keys (e.g. 1024 bits)?

d. Let (E, D) be a block cipher defined over $(\mathcal{K}, \mathcal{K}, \mathcal{C})$. Recall that the first argument to E is the key and the second argument is the message. Define the following functions:

$$h_1(x) := E(x, 0) \quad h_2(x) := E(0, x) \quad h_3(x, y) := E(x, 0) \oplus y$$

Is h_1 a one-way function? If so, explain why. If not explain why not.
Is h_2 a one-way function? If so, explain why. If not explain why not.
Is h_3 a one-way function? If so, explain why. If not explain why not.

e. Let F be a secure PRF and H a hash function. Show that if H is not collision resistant then $F'(k, x) := F(k, H(x))$ is not a secure PRF.

f. Consider the encrypted CBC MAC built from AES. Suppose we compute the tag for a long message m comprising of n AES blocks. Let m' be the n-block message obtained from m by flipping the last bit of m (i.e. if the last bit of m is b then the last bit of m' is $b \oplus 1$). Show how to compute the tag for m' from the tag for m (and the MAC key), using at most 4 applications of AES.

Problem 2. Disk encryption. In a disk encryption system, each plaintext sector is encrypted independently of other sectors. More precisely, there is a disk-wide key used to encrypt all sectors. When sector number i is encrypted, the resulting ciphertext replaces the plaintext data in sector i. Since the sector size is fixed (often 512 bytes), the encrypted sector must be exactly the same size as the plaintext sector. Here we will consider attackers who steal the disk and are then able to read the physical bits on disk.

a. As a first (incorrect) approach, the designers decided to use counter mode encryption to encrypt each sector. Moreover, they decided to use a fixed disk-wide IV. In more detail, all disk sectors are encrypted with the same key and IV, but the IV is chosen at random the very first time the system is initialized. Explain why this is a bad idea.

b. Their next idea is to use CBC encryption where the IV is the sector number. That is, sector number i is CBC encrypted with a disk-wide key k and an IV set to i. They (incorrectly) used the sector number as direct input to CBC, i.e. the first 128 bits of the sector are xored with the IV and then encrypted with AES.

Suppose the attacker knows that file F will be sequentially written to sectors 15, 16, 17, and so on (i.e. the first 512 bytes of F will be written to sector 15, the next 512 bytes of F will be written to sector 16, and so on). Explain how the attacker can create a file F so that when encrypted, the resulting ciphertext data in sectors 15, 16, and 17 is identical (i.e. all three sectors contain exactly the same bits).

This is often called watermarking: an attacker who steals the disk can tell that encryption was applied, since sectors 15, 16, and 17 are all equal and contain random looking data.

c. Explain how to correctly use CBC so as to prevent the attack from part (b).

d. None of these proposals so far provide data integrity. Assuming no prior information is known about the data on disk, is it possible to ensure integrity, if the ciphertext size must be equal to the plaintext size? Justify your answer.

Problem 3. Trapdoor functions.

a. What is a trapdoor function? Give a precise definition.

b. Explain how to use trapdoor functions for (unauthenticated) key exchange secure against eavesdropping. Explain why the scheme is secure; that is, show how to use an attacker A who can recover the exchanged secret to break the trapdoor function.

c. Explain how to construct semantically secure public key encryption from a trapdoor function (assuming random oracles). No need to prove security, just state the construction.

d. Explain how to construct a secure digital signature scheme from a trapdoor function (assuming random oracles). No need to prove security, just state the construction.

Problem 4. A classic identification protocol works as follows. First, we fix a cyclic group G of prime order q. Let g be a generator of G. Next we choose a random $x \in \mathbb{Z}_q$ and set the prover's secret key to $\text{sk} := x$ and the verifier's key to $\text{vk} := g^x$. Now, the protocol works as follows:

1. the prover chooses a random $r \in \mathbb{Z}_q$ and sends $R := g^r$ to the verifier;
2. the verifier chooses a random challenge $c \in \{1, \ldots, B\}$ (for some fixed integer B, say $B = 2^{80}$) and sends c to the prover;
3. the prover computes $z := x \cdot c + r \in \mathbb{Z}_q$ and sends z to the verifier;
4. the verifier outputs "yes" if $g^z = (\text{vk})^c \cdot R$.

a. Show that an honest prover who follows the protocol will always be accepted by the verifier (i.e. the verifier will output "yes").

b. Suppose steps (1) and (2) of the protocol are swapped (i.e. step (2) takes place before step (1)). Show that the resulting protocol is insecure. That is, an attacker who knows vk, but does now know the secret x, can cause the verifier to accept.

c. Suppose the prover accidentally responds to two challenges c and c' without changing R. That is, the prover outputs z and z' such that $g^z = (\text{vk})^c \cdot R$ and $g^{z'} = (\text{vk})^{c'} \cdot R$. Show that the verifier can use $(\text{vk}, R, c, c', z, z')$ to recover x.