CS255: Cryptography and Computer Security **Winter 2000**

Final Exam

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

Problem 1 Questions from all over.

a. In his book "The road ahead" Bill Gates writes that the security of RSA is based on the "difficulty of factoring large primes". Is the problem of factoring large primes difficult?

b. Both a CA and a Key Distribution Center (KDC) are trusted entities that are needed for secure key exchange. Briefly explain the differences between the two in terms of scalability and trust.

c. One of SSL's key exchange modes supports "forward secrecy". Briefly explain the term forward secrecy.

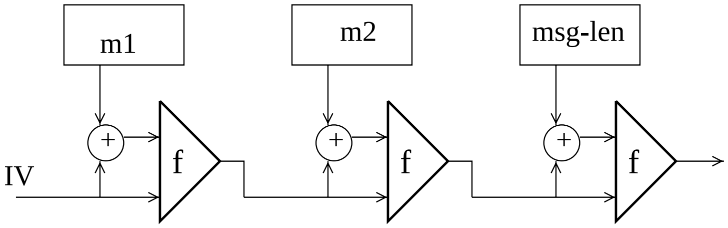
d. Consider the following combination of encryption and MAC on a plaintext M

$$C = E_{k_1}(M) \parallel MAC_{k_2}(M)$$

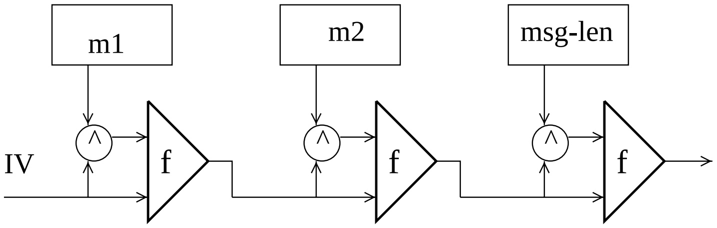
Is this secure? Justify your answer.

Problem 2 A certain organization tried to modify the Merkle-Damgard construction in two ways.

Method 1:



Method 2:



In the first construction $+$ is a bit-by-bit Xor. In the second construction $\wedge$ is a bit-by-bit and. The IV is fixed and public in both constructions. Suppose f is a collision resistant compression function that takes a 512 bit message block, and a 512 bit chaining value. The function outputs a 512 bit result. Show that one of the constructions above is a collision resistant hash function while the other isn't.

a. Show how to construct collisions for the one that is not collision resistant.

b. Prove that the other construction is collision resistant.

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 generators g and h of $\mathbb{Z}_p^*$.

Commitment: To commit to an integer $x \in [1, p-1]$ Alice does the following: (1) she picks a random $r \in [1, p-1]$, (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 value x' in $[1, p-1]$.
Hint: show that for any x' there exists a unique $r' \in [1, p-1]$ 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 In class we mentioned various security notions for MACs. Here we consider two notions: (1) MACs that are secure against existential forgery under a *chosen* message attack (CMA), and (2) MACs that are secure against existential forgery under a *random* message attack (RMA). Clearly MACs that are secure under CMA are also secure under RMA. What about the converse? Show that the converse is false.
Hint: Suppose $F_k(M)$ is a MAC secure under RMA. Construct a new MAC $G_k(M)$ (using $F_k(M)$) that is still secure under RMA but is obviously insecure against CMA.

Problem 5 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 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.

Problem 6 Let $E(M, k)$ be a block cipher using 56-bit keys. Suppose Alice sends the ciphertext $C_1 = E(M_1, k_1)$ to Bob and sends the ciphertext $C_2 = E(M_2, k_2)$ to Charlie. An eavesdropper, Eve, intercepts the two ciphertexts C_1 and C_2. Suppose Eve knows $\Delta = M_1 \oplus M_2$.

Eve's goal is to find k_1 and k_2. Assume the messages M_1 and M_2 are sufficiently long that given C_1, C_2 and Δ the pair of keys (k_1, k_2) is uniquely determined. By trying all possible pairs of keys Eve can find (k_1, k_2) using 2^{112} applications of the decryption function (simply try all k_1', k_2' until a pair satisfying $D(C_1, k_1') \oplus D(C_2, k_2') = \Delta$ is found).

Show that given C_1, C_2 and Δ Eve can find (k_1, k_2) using only 2^{57} application of the decryption function. Your algorithm may use as much memory space as you wish.