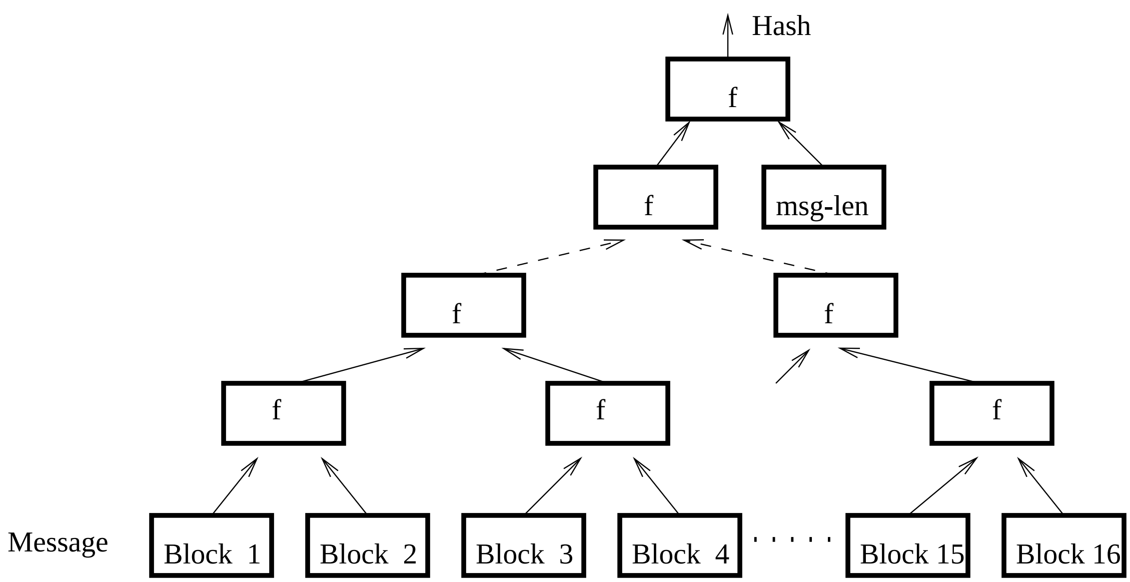
CS255: Cryptography and Computer Security **Winter 2006**

Assignment #2

Due: Wednesday, February 22nd, 2006.

Problem 1 Merkle hash trees.

Merkle suggested a parallelizable method for constructing hash functions out of compression functions. Let f be a compression function that takes two 512 bit blocks and outputs one 512 bit block. To hash a message M one uses the following tree construction:



Prove that if one can find a collision for the resulting hash function then one can find collisions for the compression function.

Problem 2 Let $h : \{0,1\}^* \to \{0,1\}^b$ be a hash function constructed by iterating a collision resistant compression function using the Merkle-Damgård construction. Show that defining $S(k,m) = h(k \parallel m)$ results in an insecure MAC. That is, show that given a valid msg/tag pair (m,t) one can efficiently construct another valid msg/tag pair (m',t') without knowing the key k.

Problem 3 Suppose Alice and Bob share a secret key k. A simple proposal for a MAC algorithm is as follows: given a message M do: (1) compute 128 different parity bits of M (i.e. compute the parity of 128 different subsets of the bits of M), and (2) AES encrypt the resulting 128-bit checksum using k. Naively, one could argue that this MAC is existentially unforgeable: without knowing k an attacker cannot create a valid

message-MAC pair. Show that this proposal is flawed. Note that the algorithm for computing the 128-bit checksums is public, i.e. the only secret unknown to the attacker is the key k.
Hint: show that an attacker can carry out an existential forgery given one valid message/MAC pair (where the message is a kilobyte long).

Problem 4 In the lecture we saw that Davies-Meyer is often used to convert an ideal block cipher into a collision resistant compression function. Let $E(k, m)$ be a block cipher. Show that the following method does not work:

$$f(x, y) = E(y, x) \oplus y$$

That is, show an efficient algorithm for constructing collisions for f. Recall that the block cipher E and the corresponding decryption algorithm D are both known to you.

Problem 5 Suppose user A is broadcasting packets to n recipients $B_1, \ldots, B_n$. Privacy is not important but integrity is. In other words, each of $B_1, \ldots, B_n$ should be assured that the packets he is receiving were sent by A. User A decides to use a MAC.

a. Suppose user A and $B_1, \ldots, B_n$ all share a secret key k. User A MACs every packet she sends using k. Each user B_i can then verify the MAC. Using at most two sentences explain why this scheme is insecure, namely, show that user B_1 is not assured that packets he is receiving are from A.

b. Suppose user A has a set $S = \{k_1, \ldots, k_m\}$ of m secret keys. Each user B_i has some subset $S_i \subseteq S$ of the keys. When A transmits a packet she appends m MACs to it by MACing the packet with each of her m keys. When user B_i receives a packet he accepts it as valid only if all MAC's corresponding to keys in S_i are valid. What property should the sets $S_1, \ldots, S_n$ satisfy so that the attack from part (a) does not apply? We are assuming all users $B_1, \ldots, B_n$ are sufficiently far apart so that they cannot collude.

c. Show that when $n = 6$ (i.e. six recipients) the broadcaster A need only append 4 MAC's to every packet to satisfy the condition of part (b). Describe the sets $S_1, \ldots, S_6 \subseteq \{k_1, \ldots, k_4\}$ you would use.

Problem 6 Strengthening hashes and MAC's.

a. Suppose we are given two hash functions $H_1, H_2 : \{0,1\}^* \rightarrow \{0,1\}^n$ (for example SHA1 and MD5) and are told that both hash functions are collision resistant. We, however, do not quite trust these claims. Our goal is to build a hash function $H_{12} : \{0,1\}^* \rightarrow \{0,1\}^m$ that is collision resistant assuming *at least* one of H_1, H_2 are collision resistant. Give the best construction you can for H_{12} and prove that a collision finder for your H_{12} can be used to find collisions for both H_1 and H_2 (this will prove collision resistance of H_{12} assuming one of H_1 or H_2 is collision resistant). Note that a straight forward construction for H_{12} is fine, as long as you prove security in the sense above.

b. Same questions as part (a) for Message Authentication Codes (MACs). Prove that an existential forger under a chosen message attack on your MAC_{12} gives an existential forger under a chosen message attack for both MAC_1 and MAC_2. Again, a straight forward construction is acceptable, as long as you prove security. The proof of security here is a bit more involved than in part (a).