

Assignment #3

Due: Tuesday, March 2, 2004.

Problem 1 (15 pts) 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 x one uses the following tree construction. The message is first divided into N blocks, then starting from the beginning, apply f to every pair of adjacent blocks, resulting in $\lceil N/2 \rceil$ blocks. Repeat until one gets one block a , then apply f to $a||\text{msg-len}$ and get the hash value.

For example, suppose the message has 3100 bits; it thus has 7 blocks, with the last block padded with 484 0's. Let the 7 blocks be x_0, x_1, \dots, x_6 . One first compute $c_0 = f(x_0, x_1)$, $c_1 = f(x_2, x_3)$, $c_2 = f(x_4, x_5)$, $c_3 = x_6$. One then compute $b_0 = f(c_0, c_1)$, $b_1 = f(c_2, c_3)$. One then compute $a_0 = f(b_0, b_1)$. The hash value of the message x is $f(a_0, \text{msg-len})$, where msg-len is the binary representation of 3100, padded with 0's.

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

Hint: The proof is similar to that of the Merkle-Damgard construction.

Problem 2 (40 pts) Constructing MAC's from hash functions.

In this problem we explore the different ways of constructing a MAC out of a non-keyed hash function. Let $h : \{0, 1\}^* \rightarrow \{0, 1\}^m$ be a hash function constructed by iterating a collision resistant compression function using the Merkle-Damgard construction.

- a. (10 pts) Show that defining $MAC_k(M) = h(k||M)$ results in an insecure MAC. That is, show that given a valid text/MAC pair (M, H) one can efficiently construct another valid text/MAC pair (M', H') without knowing the key k .
- b. (10 pts) Recall that in the Merkle-Damgard iterated construction one uses a fixed Initial Vector (0^{m+1}) as the initial chaining variable. Consider the following MAC construction from a collision-free compression function based on the Merkle-Damgard construction: the Initial Vector is set to be the secret key k . Show that this construction is an insecure MAC.
- c. (20 pts) Consider the MAC defined by $MAC_k(M) = h(M||k)$. Show that in expected time $O(2^{m/2})$ it is possible to construct two messages M and M' such that given $MAC_k(M)$ it is possible to construct $MAC_k(M')$ without knowing the key k .

Problem 3 (15 pts) Let \bar{A} denote the bitwise complement of A . (I.e., if $A = 1101$, $\bar{A} = 0010$.) Let \mathcal{E} be the DES encryption function. Prove that if $C = \mathcal{E}_k[M]$, then $\bar{C} = \mathcal{E}_k[\bar{M}]$.

Hints:

- Use the following facts: $\bar{\bar{A}} = A$ and $\overline{A \oplus B} = \bar{A} \oplus \bar{B}$.

- Let k_1, k_2, \dots, k_{16} be the round keys for DES encryption using key k , then $\overline{k_1}, \overline{k_2}, \dots, \overline{k_{16}}$ are the round keys for DES encryption using key \overline{k} .
- No assumption on the S-boxes and the permutation is needed for this problem.

Problem 4 (30 pts + extra 20 pts) Constructing hash functions from block ciphers.

Consider the Davies and Price construction of a hash function from a block cipher \mathcal{E} . (The construction is described in Stallings 11.4.) A message x is divided into fixed-size blocks x_1, x_2, \dots, x_k .

$$\begin{aligned} H_0 &= \text{Initial Vector} \\ H_i &= \mathcal{E}_{x_i}[H_{i-1}] \oplus H_{i-1} \text{ for } 1 \leq i \leq k \\ H_k &\text{ is the hash value.} \end{aligned}$$

We use $h(x, y)$ to denote the hash value of message x when using y as the initial vector. For example, given two message blocks x_1, x_2 , then

$$\begin{aligned} h(x_1, y) &= \mathcal{E}_{x_1}[y] \oplus y \\ h(x_1 || x_2, y) &= h(x_2, h(x_1, y)) = \mathcal{E}_{x_2}[h(x_1, y)] \oplus h(x_1, y). \end{aligned}$$

In this problem, we assume that DES is used as \mathcal{E} . Therefore, each message block has 56 bits and the initial vector and the hash value have 64 bits.

- a. (10pts)** An attacker is given a message x consisting of blocks $x_{1,2}, \dots, x_k$, an initial vector y and the hash code $h(x, y)$. Show that the attacker can easily find another initial vector y' and a message x' such that $h(x', y') = h(x, y)$.

Hint: Use the property about DES proved in Problem 2.

- b. (20 pts)** Describe an algorithm that can generate an initial vector y and an infinite sequence of messages x^1, x^2, x^3, \dots such that $h(x^1, y) = h(x^2, y) = h(x^3, y) = \dots$.

Hint: find a message block x_1 and a 64 bit block y such that $h(x_1, y) = y$.

- c. (extra 20 pts)** Describe a variation of the above attack with expected running time $O(2^{28})$ to attack the hash function when the initial vector value is fixed to a value y_0 . The attack algorithm, when given y_0 , finds an infinite sequence of messages x^1, x^2, x^3, \dots such that $h(x^1, y_0) = h(x^2, y_0) = h(x^3, y_0) = \dots$.

Hint: find two message blocks x_1 and x_2 and a block y such that $h(x_1, y_0) = y = h(x_2, y_0)$.