6.857 Computer and Network Security **Fall Term, 1997**

Lecture 9 : October 2, 1997

Lecturer: Ron Rivest *Scribe: Jesse Kornblum*

Topics

- Hashing in Digital Signatures
- Hashing Security
- MD5
- Birthday Attacks
- SHA
- CvHP
- MACs and Hashing

1 Digital Signatures

When messages get very long, encrypting them to use as a digital signature with algorithms such as RSA can be very expensive. We'd like to sign our messages with something much shorter, a "fingerprint" of the message, also called a "message digest."

To create the fingerprint, we use a function (fingerprint function, MD-function, cryptographic hash function, etc.) that compresses messages of arbitrary length into a preset set. This function, $h(x)$, normally produces a fingerprint either 128 or 160 bits long. This shorter value is much easier to sign than the full message. Our goal is that:

- The function h is efficient
- The length of $h(x)$ is short
- The function $h(x)$ is secure (i.e. it has strong collision-resistance, defined below)

2 Hashing Security

The security of hash functions is measured by the existence of *collisions*. A collision is when two different messages produce the same hash. (i.e. $h(m_1) = h(m_2)$) Here are some terms used in describing the security hash functions:

Strong Collision-Resistance: It is infeasible for an attacker to find x and x' forming a collision. (i.e. $h(x) = h(x')$)

Weak Collision-Resistance: Given an x and $h(x)$, it is infeasible to find x' such that $h(x) = h(x')$. This implies that given $h(x)$, it is infeasible to find any x' such that $h(x) = h(x')$. (i.e. That h is a one-way function.)

3 Message Digest 5 (MD5)

MD5 is a hashing function developed by Ron Rivest. A much more detailed description can be found in handout 8 [1] or RFC 1321.

MD5 works by first padding the message until it is a multiple of 512 bits long. Padding is done as follows:

1. Append a '1' bit to the message.
2. Append '0' bits until the message is 64 bits shorter than a multiple of 512 bits
3. Append a 64-bit representation of the message's original length

The state of MD5 is kept in four 32-bit words, A, B, C, and D, all of which are initialized to magic constant values. MD5 processes the message in 512-bit blocks. As we process the ith block of message, we update A_{i-1}, B_{i-1}, C_{i-1}, and D_{i-1} to A_i, B_i, C_i, and D_i. The output of MD5, a 128 bit value, is the final state of A, B, C, and D concatenated.

For each block of message, we have four rounds of updates. Each round updates one of the four 32-bit words A, B, C, or D four times. (For a total of sixteen updates per block of message.) Initially on each round, $A_i \leftarrow A_{i-1}$, $B_i \leftarrow B_{i-1}$, etc. Each of the updates is something similar to $A_i \leftarrow B_i + ((A_i + F(B_i, C_i, D_i) + M_i + T_i <<< s)$, where

[1] Which actually comes from *Applied Cryptography* by B. Schneier

F is a function, [2] M_i is the ith block of the message, and T_i and s are magic constants. (The symbol <<< means "rotate left".) At the end of each round, we finish by updating all of the values one last time, namely: $A_i \leftarrow A_i + A_{i-1}$, $B_i \leftarrow B_i + B_{i-1}$, etc.

4 Secure Hash Algoritm (SHA)

SHA is a hash function developed by the US government. It's similar to MD5, but uses five 32-bit words and five rounds per message block to produce a 160-bit hash, using linear transformations of the message in each block.

5 Birthday Attacks

A birthday attack on a hash function attempts to use the birthday paradox to find collisions in a hash function. (i.e. creating random messages, taking their hash value, and checking if that hash value has been encountered before.) For MD5, as an example, an attacker could expect to find collisions after trying 2^{64} messages. Given today's computing power, this is a difficult, but not impossible problem. Hence the move to stronger hashing algorithms, like SHA.

6 CvHP

Yet another hash function is CvHP. It's full name is the Chaum-vanHeijst-Pfitzmann hash function. First, we need to choose prime numbers p and q such that $p = 2q + 1$. Next, we need an α and a β that are both an element of order q (i.e. $\alpha^q \equiv 1(\text{mod} p)$) and that $\alpha \neq \beta$. For security, it must be difficult to compute $\log_\alpha \beta$.

The hash function splits the message into two parts m_1 and m_2. $h(m_1, m_2) = \alpha^{m_1} * \beta^{m_2}(\text{mod} p)$. It is necessary that $1 \leq |m_1|, |m_2| \leq q$.

The heart of CvHP is that it's infeasible to find α and β. As an example, let's say that we have a collision:

$$\alpha^{m_1} * \beta^{m_2} = \alpha^{m_3} * \beta^{m_4}(\text{mod} p))$$

[2] As an example, $F(a, b, c) = (a \wedge b) \vee (\bar{a} \wedge c)$

We can transform this to be

$$\alpha^{m_1 - m_3} = \beta^{m_4 - m_2} (\mathrm{mod} p)$$

If we set $r = m_1 - m_3$ and $s = m_4 - m_2$, we get

$$\alpha^r = \beta^s (\mathrm{mod} p)$$

If then we choose t such that $t = s^{-1} (\mathrm{mod} q)$, we can use

$$\alpha^{rt} = \beta^{st} (\mathrm{mod} p)$$

to get $\alpha^{rt} = \beta (\mathrm{mod} p)$. Since we said it's difficult to compute $log_\alpha \beta$, an attacker cannot get past this stage.

7 MACs and Hash Functions

A hash function can be used to produce a Message Authentication Code. For example, a proposal by IBM for Internet messages is that a MAC for each packet should be formed as follows:

$HMAC = MD5(k_1, MD5(k_2, M))$, where k_1 and k_2 are two parts of a shared secret key.