Hash Functions — MD5 and SHA1

Orr Dunkelman

Computer Science Department

14 March, 2012

University of Haifa

Outline

The MD5 Hash Function

- A successor to MD4, designed by Rivest in 1992 (RFC 1321).
- Takes messages of size up to 2^{64} bits, and generates a digest of size 128 bits.
- Uses the Merkle-Damgård mode of iteration and a compression function (512-bit message block, 128-bit chaining value).
- The compression function is made in a Davies-Meyer mode (transformation of a block cipher into a compression function).

The MD5 Hash Function (cont.)

- To hash a message M the following steps are performed:
  1. M is padded with '1' as many 0's as needed (up to 512) and the original length of M encoded in 64 bits, such that the length of the padded message $pad(M)$ is divisible by 512.
  2. $pad(M)$ is divided into ℓ blocks of 512 bits, i.e., $pad(M) = m_1, m_2, \ldots, m_\ell$.
  3. The 128-bit chaining value h_0 is initialized.
  4. For $i = 1, 2, \ldots, \ell$, $h_i = H(h_{i-1}, m_i)$ (the compression function is applied).
  5. The output is h_ℓ

The MD5 IV

- The internal state (chaining value) of MD5, is treated as four words of 32-bit each: A, B, C, D.
- The initial value h_0 is:

$$\begin{aligned} A &= \texttt{67452301}_x \\ B &= \texttt{EFCDAB89}_x \\ C &= \texttt{98BADCFE}_x \\ D &= \texttt{10325476}_x \end{aligned}$$

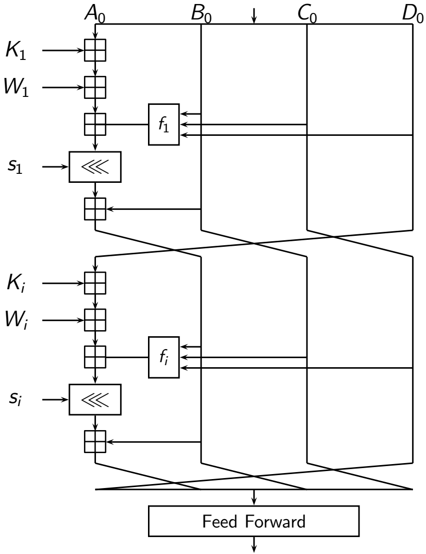
(this initial value is given in a little-endian manner)

The MD5 Compression Function

- Let $h_{i-1} = (A_0, B_0, C_0, D_0)$.
- Let the message block be $M_i = (W_0, W_1, \ldots, W_{15})$
- For $i = 0, 1, \ldots, 63$:
  1. $D_{i+1} \leftarrow C_i$
  2. $C_{i+1} \leftarrow B_i$
  3. $B_{i+1} \leftarrow B_i + (A_i + F_i(B_i, C_i, D_i) + K_i + W_{g(i)}) \lll s_i$
  4. $A_{i+1} \leftarrow D_i$
- $h_i \leftarrow (A_0 + A_{64}, B_0 + B_{64}, C_0 + C_{64}, D_0 + D_{64})$.

All additions are modulo 2^{32}, and $\lll$ stands for rotation to the left.

The MD5 Compression Function



The MD5 Compression Function (cont.)

- Each round, a different message word is used, a different round constant is used, and a different function and rotations:

$$
\begin{array}{rll}
0 \leq t \leq 15: & f_t(X,Y,Z) = XY \vee (\neg X)Z & g(t) = t \\
16 \leq t \leq 31: & f_t(X,Y,Z) = XY \vee (\neg Z)X & g(t) = (5 \cdot t + 1) \bmod 16 \\
32 \leq t \leq 47: & f_t(X,Y,Z) = X \oplus Y \oplus Z & g(t) = (3 \cdot t) \bmod 16 \\
48 \leq t \leq 63: & f_t(X,Y,Z) = Y \oplus (X \vee \neg Z) & g(t) = (7 \cdot t) \bmod 16
\end{array}
$$

The set of constants K_i is based on sin:

$$K_i = \lfloor |\sin(i+1)| \cdot 2^{32} \rfloor$$

The MD5 Compression Function (cont.)

The rotation constants (s_i) are

Rotation Constants

7	12	17	22	7	12	17	22	7	12	17	22	7	12	17	22
5	9	14	20	5	9	14	20	5	9	14	20	5	9	14	20
4	11	16	23	4	11	16	23	4	11	16	23	4	11	16	23
6	10	15	21	6	10	15	21	6	10	15	21	6	10	15	21

The SHA-1 Hash Function

- Designed by the NSA, following the structure of MD4 and MD5.
- The first standard was SHA (now called SHA-0), first published in 1993.
- Shortly after, it was later changed slightly to SHA-1, due to some unknown weakness found by the NSA.
- Today, the SHA family contains four more hash functions (the SHA-2 family), and in 2012, NIST is expected to select SHA-3.

The SHA-1 Hash Function (cont.)

- SHA-1 is a Merkle-Damgård hash function:
  1. **Padding**: Given an m-bit message, a single bit "1" is appended as the $m+1$th bit and then $(448-(m+1)) \bmod 512$ (between 0 and 511) zero bits are appended. As a result, the message becomes 64-bit short of being a multiple of 512 bits long.
  2. **Merkle-Damgård Strengthening** Append the length: A 64-bit representation of the original length of m is appended, making the result a multiple of 512 bits long.
  3. **Division into Blocks** The result is divided into 512-bit blocks, denoted by $M_1, M_2, \ldots, M_\ell$.

The SHA-1 Hash Function (cont.)

The internal state of SHA-1 is composed of five 32-bit words A, B, C, D and E, used to keep the 160-bit chaining value h_i.

- **Initialization:** The initial value (h_0) is (in hexadecimal)

$$\begin{aligned} A &= \texttt{67452301}_x \\ B &= \texttt{EFCDAB89}_x \\ C &= \texttt{98BADCFE}_x \\ D &= \texttt{10325476}_x \\ E &= \texttt{C3D2E1F0}_x. \end{aligned}$$

- **Compression:** For each block, the compression function $h_i = H(h_{i-1}, M_i)$ is applied on the previous value of $h_{i-1} = (A, B, C, D, E)$ and the message block.
- **Output:** The hash value is the 160-bit value $h_\ell = (A, B, C, D, E)$.

The Compression Function H of SHA-1

1. Divide M_i into 16 32-bit words: $W_0, W_1, W_2, \ldots, W_{15}$.
2. for $t = 16$ to 79 compute
   $W_t = (W_{t-3} \oplus W_{t-8} \oplus W_{t-14} \oplus W_{t-16}) \lll 1$.
   Remark The one-bit rotate in computing W_t was not included in SHA, and is the only difference between SHA and SHA-1.

The Compression Function H of SHA-1 (cont.)

3. Set $(A_0, B_0, C_0, D_0, E_0) \leftarrow h_{i-1}$.
4. For $t = 0$ to 79 do
    1. $T = A_t \lll 5 + f_t(B_t, C_t, D_t) + E_t + W_t + K_t$.
    2. $E_{t+1} = D_t,\ D_{t+1} = C_t,\ C_{t+1} = B_t \lll 30,\ B_{t+1} = A_t,\ A_{t+1} = T$.
5. Output $A = A_0 + A_{80},\ B = B_0 + B_{80},\ C = C_0 + C_{80},\ D = D_0 + D_{80}$, and $E = E_0 + E_{80}$ (modulo 2^{32}).
6. The function f_t and the values K_t used above are:

$0 \le t \le 19$:	$f_t(X,Y,Z) = XY \vee (\neg X)Z$	$K_t = \texttt{5A827999}$
$20 \le t \le 39$:	$f_t(X,Y,Z) = X \oplus Y \oplus Z$	$K_t = \texttt{6ED9EBA1}$
$40 \le t \le 59$:	$f_t(X,Y,Z) = XY \vee XZ \vee YZ$	$K_t = \texttt{8F1BBCDC}$
$60 \le t \le 79$:	$f_t(X,Y,Z) = X \oplus Y \oplus Z$	$K_t = \texttt{CA62C1D6}$

The Compression Function H of SHA-1 (cont.)