MD5.

Introducción.

El algoritmo MD5 está diseñado para ser bastante rápido en máquinas de 32 bits. Además, el algoritmo MD5 no requiere una gran sustitución mesas; el algoritmo se puede codificar de forma bastante compacta.
El algoritmo MD5 es una extensión del algoritmo de resumen de mensajes MD4 1,2]. MD5 es un poco más lento que MD4, pero es más "conservador" en diseño. MD5 fue diseñado porque se consideró que MD4 era quizás siendo adoptados para su uso más rápidamente de lo justificado por el existente revisión crítica; porque MD4 fue diseñado para ser excepcionalmente rápido, está "al límite" en términos de riesgo criptoanalítico exitoso ataque. MD5 retrocede un poco, cediendo un poco de velocidad por mucho mayor probabilidad de máxima seguridad. Incorpora algunos sugerencias hechas por varios revisores, y contiene más optimizaciones. El algoritmo MD5 se coloca en el dominio público para revisión y posible adopción como estándar. Para aplicaciones basadas en OSI, el identificador de objeto de MD5 es IDENTIFICADOR DE OBJETO md5 :: = iso (1) miembro-cuerpo (2) EE. UU. (840) rsadsi (113549) digestAlgorithm (2) 5} En el tipo AlgorithmIdentifier [3] de X.509, los parámetros para MD5 debe tener el tipo NULL.

Terminología y notación

Una "palabra" es una cantidad de 32 bits y un "byte" es un
cantidad de ocho bits. Una secuencia de bits se puede interpretar en un
de manera natural como una secuencia de bytes, donde cada grupo consecutivo
de ocho bits se interpreta como un byte con el orden superior (la mayoría
significativo) bit de cada byte enumerado primero. Del mismo modo, una secuencia de
bytes se pueden interpretar como una secuencia de palabras de 32 bits, donde cada
grupo consecutivo de cuatro bytes se interpreta como una palabra con el
Byte de orden bajo (menos significativo) dado primero.

Deje x_i denotar "x sub i". Si el subíndice es una expresión,
rodearlo entre llaves, como en x_ {i + 1}. Del mismo modo, usamos ^ para
superíndices (exponenciación), de modo que x ^ i denota x elevado a i-ésimo
poder.

Deje que el símbolo "+" denote la adición de palabras (es decir, módulo-2 ^ 32
adición). Sea X <<< s el valor de 32 bits obtenido circularmente
desplazando (girando) X a la izquierda por s posiciones de bit. No denote (X) el
complemento bit a bit de X, y sea X v Y denotar el OR bit a bit de X

e Y. Sea X xor Y el XOR bit a bit de X e Y, y sea XY
denotar el AND bit a bit de X e Y.

Descripción del algoritmo MD5

Comenzamos suponiendo que tenemos un mensaje de b-bit como entrada, y que deseamos encontrar el resumen de su mensaje. Aquí b es un arbitrario entero no negativo; b puede ser cero, no necesita ser un múltiplo de ocho, y puede ser arbitrariamente grande. Imaginamos los pedazos del mensaje escrito de la siguiente manera:

m_0 m_1 ... m_ {b-1}

Se realizan los siguientes cinco pasos para calcular el resumen del mensaje del mensaje.

Paso 1. Agregar bits de relleno

El mensaje se "rellena" (extendido) para que su longitud (en bits) sea congruente con 448, módulo 512. Es decir, el mensaje se extiende para que es sólo 64 bits menos que un múltiplo de 512 bits de longitud. El relleno siempre se realiza, incluso si la longitud del mensaje es ya congruente con 448, módulo 512.

El relleno se realiza de la siguiente manera: se agrega un solo bit "1" al mensaje, y luego "0" bits se añaden para que la longitud en bits de el mensaje relleno se vuelve congruente con 448, módulo 512. En total, en se añaden al menos un bit y como máximo 512 bits.

Paso 2. Agregar longitud

Una representación de 64 bits de b (la longitud del mensaje antes del se agregaron bits de relleno) se agrega al resultado de la anterior paso. En el improbable caso de que b sea mayor que 2 ^ 64, entonces solo se utilizan los 64 bits de orden inferior de b. (Estos bits se agregan como dos Palabras de 32 bits y la palabra de orden inferior agregada primero de acuerdo con la convenciones anteriores.)

En este punto, el mensaje resultante (después de rellenar con bits y tiene una longitud que es un múltiplo exacto de 512 bits. Equivalentemente, este mensaje tiene una longitud que es un múltiplo exacto de 16 (32 bits) palabras. Sea M [0 ... N-1] las palabras del mensaje resultante, donde N es un múltiplo de 16.

Paso 3. Inicializar MD Buffer

Se utiliza un búfer de cuatro palabras (A, B, C, D) para calcular el resumen del mensaje. Aquí cada uno de A, B, C, D es un registro de 32 bits. Estos registros son inicializado a los siguientes valores en bytes hexadecimales de orden inferior primero):

palabra A: 01 23 45 67
palabra B: 89 ab cd ef
palabra C: fe dc ba 98
palabra D: 76 54 32 10

Paso 4. Procesar mensaje en bloques de 16 palabras

Primero definimos cuatro funciones auxiliares que cada una toma como entrada tres palabras de 32 bits y producir como salida una palabra de 32 bits.

F (X, Y, Z) = XY v no (X) Z
G (X, Y, Z) = XZ v Y no (Z)
H (X, Y, Z) = X x o Y xor Z
I (X, Y, Z) = Y xor (X v no (Z))

En cada posición de bit, F actúa como un condicional: si X entonces Y si no Z. La función F podría haberse definido usando + en lugar de v desde XY y no (X) Z nunca tendrá unos en la misma posición de bit).
Es interesante notar que si los bits de X, Y y Z son independientes e insesgado, cada bit de F (X, Y, Z) será independiente y imparcial.

Las funciones G, H e I son similares a la función F, en que actúan en "bit a bit paralelo" para producir su salida a partir de los bits de X, Y, y Z, de tal manera que si los bits correspondientes de X, Y, y Z son independientes e insesgados, entonces cada bit de G (X, Y, Z), H (X, Y, Z) e I (X, Y, Z) serán independientes e insesgados. Tenga en cuenta que la función H es la función "xor" o "paridad" bit a bit de su entradas.

Este paso utiliza una tabla T [1 ... 64] de 64 elementos construida a partir de función seno. Sea T [i] el elemento i-ésimo de la tabla, que es igual a la parte entera de 4294967296 multiplicado por abs (sin (i)), donde i está en radianes. Los elementos de la tabla se dan en el apéndice.

Ejemplo

```
/ * Procesar cada bloque de 16 palabras. * /
Para i = 0 a N / 16-1 hacer

  / * Copia el bloque i en X. * /
  Para j = 0 a 15 hacer
    Establezca X [j] en M [i * 16 + j].
  fin / * del bucle en j * /

  / * Guarde A como AA, B como BB, C como CC y D como DD. * /
  AA = A
  BB = B

  CC = C
  DD = D

  /* La ronda 1. */
  / * Deje que [abcd ksi] denote la operación
       a = segundo + ((a + F (segundo, c, re) + X [k] + T [i]) <<< s). * /
  / * Realice las siguientes 16 operaciones. * /
  [ABCD 0 7 1] [DABC 1 12 2] [CDAB 2 17 3] [BCDA 3 22 4]
  [ABCD 4 7 5] [DABC 5 12 6] [CDAB 6 17 7] [BCDA 7 22 8]
  [ABCD 8 7 9] [DABC 9 12 10] [CDAB 10 17 11] [BCDA 11 22 12]
  [ABCD 12 7 13] [DABC 13 12 14] [CDAB 14 17 15] [BCDA 15 22 16]

  /* La ronda 2. */
  / * Deje que [abcd ksi] denote la operación
       a = segundo + ((a + G (segundo, c, re) + X [k] + T [i]) <<< s). * /
  / * Realice las siguientes 16 operaciones. * /
  [ABCD 1 5 17] [DABC 6 9 18] [CDAB 11 14 19] [BCDA 0 20 20]
  [ABCD 5 5 21] [DABC 10 9 22] [CDAB 15 14 23] [BCDA 4 20 24]
  [ABCD 9 5 25] [DABC 14 9 26] [CDAB 3 14 27] [BCDA 8 20 28]
  [ABCD 13 5 29] [DABC 2 9 30] [CDAB 7 14 31] [BCDA 12 20 32]

  / * Ronda 3. * /
  / * Sea [abcd kst] la operación
       a = segundo + ((a + H (segundo, c, re) + X [k] + T [i]) <<< s). * /
  / * Realice las siguientes 16 operaciones. * /
  [ABCD 5 4 33] [DABC 8 11 34] [CDAB 11 16 35] [BCDA 14 23 36]
  [ABCD 1 4 37] [DABC 4 11 38] [CDAB 7 16 39] [BCDA 10 23 40]
  [ABCD 13 4 41] [DABC 0 11 42] [CDAB 3 16 43] [BCDA 6 23 44]
  [ABCD 9 4 45] [DABC 12 11 46] [CDAB 15 16 47] [BCDA 2 23 48]
```

```
/ * Ronda 4. * /
  / * Sea [abcd kst] la operación
       a = segundo + ((a + I (segundo, c, d) + X [k] + T [i]) <<< s). * /
  / * Realice las siguientes 16 operaciones. * /
  [ABCD 0 6 49] [DABC 7 10 50] [CDAB 14 15 51] [BCDA 5 21 52]
  [ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
  [ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
  [ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

  / * Luego realice las siguientes adiciones. (Eso es incremento cada
     de los cuatro registros por el valor que tenía antes de este bloque
     empezó.) */
  A = A + AA
  B = B + BB
  C = C + CC
  D = D + DD

fin / * del bucle en i * /
```

Paso 5. Salida

El resumen del mensaje producido como salida es A, B, C, D. Es decir, comenzar con el byte de orden inferior de A y terminar con el byte de orden superior de D.

Esto completa la descripción de MD5. Una implementación de referencia en C se da en el apéndice.

Resumen.

El algoritmo de resumen de mensajes MD5 es sencillo de implementar y proporciona una "huella digital" o resumen de un mensaje de longitud arbitraria.
Se conjetura que la dificultad de proponer dos mensajes tener el mismo resumen de mensajes es del orden de 2 ^ 64 operaciones, y que la dificultad de encontrar un mensaje que tenga un determinado El resumen del mensaje es del orden de 2 ^ 128 operaciones. El algoritmo MD5 ha sido cuidadosamente examinado en busca de debilidades. Sin embargo, es un algoritmo relativamente nuevo y más análisis de seguridad es, por supuesto justificado, como es el caso de cualquier nueva propuesta de este tipo.