NAME

MD4Init, **MD4Update**, **MD4Pad**, **MD4Final**, **MD4End**, **MD4File**, **MD4FileChunk**, **MD4Data** - calculate the RSA Data Security, Inc., ‘‘MD4’’ message digest

LIBRARY

Message Digest (MD4, MD5, etc.) Support Library (libmd, -lmd)

SYNOPSIS

#include <sys/types.h>
#include <md4.h>

void
MD4Init(*MD4_CTX *context*);

void
MD4Update(*MD4_CTX *context*, *const void *data*, *unsigned int len*);

void
MD4Pad(*MD4_CTX *context*);

void
MD4Final(*unsigned char digest[16]*, *MD4_CTX *context*);

*char **
MD4End(*MD4_CTX *context*, *char *buf*);

*char **
MD4File(*const char *filename*, *char *buf*);

*char **
MD4FileChunk(*const char *filename*, *char *buf*, *off_t offset*, *off_t length*);

*char **
MD4Data(*const void *data*, *unsigned int len*, *char *buf*);

DESCRIPTION

The MD4 functions calculate a 128-bit cryptographic checksum (digest) for any number of input bytes. A cryptographic checksum is a one-way hash-function, that is, you cannot find (except by exhaustive search) the input corresponding to a particular output. This net result is a "fingerprint" of the input-data, which does not disclose the actual input.

MD4 is the fastest and MD5 is somewhat slower. MD4 has now been broken; it should only be used where necessary for backward compatibility. MD5 has not yet (1999-02-11) been broken, but sufficient attacks have been made that its security is in some doubt. The attacks on both MD4 and MD5 are both in the nature of finding "collisions" - that is, multiple inputs which hash to the same value; it is still unlikely for an attacker to be able to determine the exact original input given a hash value.

The **MD4Init**(), **MD4Update**(), and **MD4Final**() functions are the core functions. Allocate an *MD4_CTX*, initialize it with **MD4Init**(), run over the data with **MD4Update**(), and finally extract the result using **MD4Final**(), which will also erase the *MD4_CTX*.

The **MD4Pad**() function can be used to pad message data in same way as done by **MD4Final**() without terminating calculation.

The **MD4End**() function is a wrapper for **MD4Final**() which converts the return value to a 33-character (including the terminating '\0') ASCII string which represents the 128 bits in hexadecimal.

The **MD4File**() function calculates the digest of a file, and uses **MD4End**() to return the result. If the file cannot be opened, a null pointer is returned. The **MD4FileChunk**() function is similar to **MD4File**(), but it only calculates the digest over a byte-range of the file specified, starting at *offset* and spanning *length* bytes. If the *length* parameter is specified as 0, or more than the length of the remaining part of the file, **MD4FileChunk**() calculates the digest from *offset* to the end of file. The **MD4Data**() function calculates the digest of a chunk of data in memory, and uses **MD4End**() to return the result.

When using **MD4End**(), **MD4File**(), or **MD4Data**(), the *buf* argument can be a null pointer, in which case the returned string is allocated with malloc(3) and subsequently must be explicitly deallocated using free(3) after use. If the *buf* argument is non-null it must point to at least 33 characters of buffer space.

ERRORS

The **MD4End**() function called with a null buf argument may fail and return NULL if:

[ENOMEM] Insufficient storage space is available.

The **MD4File**() and **MD4FileChunk**() may return NULL when underlying open(2), fstat(2), lseek(2), or MD4End(3) fail.

SEE ALSO

md4(3), md5(3), ripemd(3), sha(3), sha256(3), sha512(3), skein(3)

R. Rivest, *The MD4 Message-Digest Algorithm*, RFC 1186.

R. Rivest, *The MD5 Message-Digest Algorithm*, RFC 1321.

H. Dobbertin, "Alf Swindles Ann", *CryptoBytes*, 1(3):5, 1995.

MJ. B. Robshaw, "On Recent Results for MD2, MD4 and MD5", *RSA Laboratories Bulletin*, 4, November 12, 1996.

HISTORY

These functions appeared in FreeBSD 2.0.

AUTHORS

The original MD4 routines were developed by RSA Data Security, Inc., and published in the above references. This code is derived directly from these implementations by Poul-Henning Kamp <*phk@FreeBSD.org*>.

Phk ristede runen.

BUGS

The MD5 algorithm has been proven to be vulnerable to practical collision attacks and should not be relied upon to produce unique outputs, *nor should they be used as part of a cryptographic signature scheme.*