NAME

SKEIN256_Init, **SKEIN256_Update**, **SKEIN256_Final**, **SKEIN256_End**, **SKEIN256_File**, **SKEIN256_FileChunk**, **SKEIN256_Data**, **SKEIN512_Init**, **SKEIN512_Update**, **SKEIN512_Final**, **SKEIN512_End**, **SKEIN512_File**, **SKEIN512_FileChunk**, **SKEIN512_Data**, **SKEIN1024_Init**, **SKEIN1024_Update**, **SKEIN1024_Final**, **SKEIN1024_End**, **SKEIN1024_File**, **SKEIN1024_FileChunk**, **SKEIN1024_Data** **-** calculate the ‘‘SKEIN’’ family of message digests

LIBRARY

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

SYNOPSIS

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

void
SKEIN256_Init(*SKEIN256_CTX *context*);

void
SKEIN256_Update(*SKEIN256_CTX *context*, *const unsigned char *data*, *size_t len*);

void
SKEIN256_Final(*unsigned char digest[32]*, *SKEIN256_CTX *context*);

*char **
SKEIN256_End(*SKEIN256_CTX *context*, *char *buf*);

*char **
SKEIN256_File(*const char *filename*, *char *buf*);

*char **
SKEIN256_FileChunk(*const char *filename*, *char *buf*, *off_t offset*, *off_t length*);

*char **
SKEIN256_Data(*const unsigned char *data*, *unsigned int len*, *char *buf*);

void
SKEIN512_Init(*SKEIN512_CTX *context*);

void
SKEIN512_Update(*SKEIN512_CTX *context*, *const unsigned char *data*, *size_t len*);

void
SKEIN512_Final(*unsigned char digest[64]*, *SKEIN512_CTX *context*);

*char **
SKEIN512_End(*SKEIN512_CTX *context*, *char *buf*);

*char **
SKEIN512_File(*const char *filename*, *char *buf*);

*char **
SKEIN512_FileChunk(*const char *filename*, *char *buf*, *off_t offset*, *off_t length*);

*char **
SKEIN512_Data(*const unsigned char *data*, *unsigned int len*, *char *buf*);

void
SKEIN1024_Init(*SKEIN1024_CTX *context*);

void
SKEIN1024_Update(*SKEIN1024_CTX *context*, *const unsigned char *data*, *size_t len*);

void
SKEIN1024_Final(*unsigned char digest[128]*, *SKEIN1024_CTX *context*);

*char **
SKEIN1024_End(*SKEIN1024_CTX *context*, *char *buf*);

*char **
SKEIN1024_File(*const char *filename*, *char *buf*);

*char **
SKEIN1024_FileChunk(*const char *filename*, *char *buf*, *off_t offset*, *off_t length*);

*char **
SKEIN1024_Data(*const unsigned char *data*, *unsigned int len*, *char *buf*);

DESCRIPTION

Skein is a new family of cryptographic hash functions based on the Threefish large-block cipher. Its design combines speed, security, simplicity, and a great deal of flexibility in a modular package that is easy to analyze. Skein is defined for three different internal state sizes--256 bits, 512 bits, and 1024 bits--and any output size. This allows Skein to be a drop-in replacement for the entire SHA family of hash functions.

The **SKEIN256_Init**(), **SKEIN256_Update**(), and **SKEIN256_Final**() functions are the core functions. Allocate an *SKEIN256_CTX*, initialize it with **SKEIN256_Init**(), run over the data with **SKEIN256_Update**(), and finally extract the result using **SKEIN256_Final**(), which will also erase the *SKEIN256_CTX*.

SKEIN256_End() is a wrapper for **SKEIN256_Final**() which converts the return value to a 33-character (including the terminating '\0') ASCII string which represents the 256 bits in hexadecimal.

SKEIN256_File() calculates the digest of a file, and uses **SKEIN256_End**() to return the result. If the file cannot be opened, a null pointer is returned. **SKEIN256_FileChunk**() is similar to **SKEIN256_File**(), 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, **SKEIN256_FileChunk**() calculates the digest from *offset* to the end of file. **SKEIN256_Data**() calculates the digest of a chunk of data in memory, and uses **SKEIN256_End**() to return the result.

When using **SKEIN256_End**(), **SKEIN256_File**(), or **SKEIN256_Data**(), 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.

The SKEIN512_ and SKEIN1024_ functions are similar to the SKEIN256_ functions except they produce a 512-bit, 65 character, or 1024-bit, 129 character, output.

ERRORS

The **SKEIN256_End**() function called with a null buf argument may fail and return NULL if:

[ENOMEM] Insufficient storage space is available.

The **SKEIN256_File**() and **SKEIN256_FileChunk**() may return NULL when underlying open(2), fstat(2), lseek(2), or SKEIN256_End(3) fail.

SEE ALSO

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

HISTORY

These functions appeared in FreeBSD 11.0.

AUTHORS

The core hash routines were imported from version 1.3 of the optimized Skein reference implementation written by Doug Whiting as submitted to the NSA SHA-3 contest. The algorithms were developed by Niels Ferguson, Stefan Lucks, Bruce Schneier, Doug Whiting, Mihir Bellare, Tadayoshi Kohno, Jon Callas, and Jesse Walker.