

**NAME**

HMAC, HMAC\_CTX\_init, HMAC\_Init, HMAC\_Init\_ex, HMAC\_Update, HMAC\_Final, HMAC\_CTX\_cleanup, HMAC\_cleanup - HMAC message authentication code

**SYNOPSIS**

```
#include <openssl/hmac.h>

unsigned char *HMAC(const EVP_MD *evp_md, const void *key,
int key_len, const unsigned char *d, int n,
unsigned char *md, unsigned int *md_len);

void HMAC_CTX_init(HMAC_CTX *ctx);

int HMAC_Init(HMAC_CTX *ctx, const void *key, int key_len,
const EVP_MD *md);
int HMAC_Init_ex(HMAC_CTX *ctx, const void *key, int key_len,
const EVP_MD *md, ENGINE *impl);
int HMAC_Update(HMAC_CTX *ctx, const unsigned char *data, int len);
int HMAC_Final(HMAC_CTX *ctx, unsigned char *md, unsigned int *len);

void HMAC_CTX_cleanup(HMAC_CTX *ctx);
void HMAC_cleanup(HMAC_CTX *ctx);
```

**DESCRIPTION**

HMAC is a MAC (message authentication code), i.e. a keyed hash function used for message authentication, which is based on a hash function.

*HMAC()* computes the message authentication code of the **n** bytes at **d** using the hash function **evp\_md** and the key **key** which is **key\_len** bytes long.

It places the result in **md** (which must have space for the output of the hash function, which is no more than **EVP\_MAX\_MD\_SIZE** bytes). If **md** is NULL, the digest is placed in a static array. The size of the output is placed in **md\_len**, unless it is NULL.

**evp\_md** can be *EVP\_sha1()*, *EVP\_ripemd160()* etc.

*HMAC\_CTX\_init()* initialises a **HMAC\_CTX** before first use. It must be called.

*HMAC\_CTX\_cleanup()* erases the key and other data from the **HMAC\_CTX** and releases any associated resources. It must be called when an **HMAC\_CTX** is no longer required.

*HMAC\_cleanup()* is an alias for *HMAC\_CTX\_cleanup()* included for back compatibility with 0.9.6b, it is deprecated.

The following functions may be used if the message is not completely stored in memory:

*HMAC\_Init()* initializes a **HMAC\_CTX** structure to use the hash function **evp\_md** and the key **key** which is **key\_len** bytes long. It is deprecated and only included for backward compatibility with OpenSSL 0.9.6b.

*HMAC\_Init\_ex()* initializes or reuses a **HMAC\_CTX** structure to use the function **evp\_md** and key **key**. Either can be NULL, in which case the existing one will be reused. *HMAC\_CTX\_init()* must have been called before the first use of an **HMAC\_CTX** in this function. **N.B. *HMAC\_Init()* had this undocumented behaviour in previous versions of OpenSSL - failure to switch to *HMAC\_Init\_ex()* in programs that expect it will cause them to stop working.**

*HMAC\_Update()* can be called repeatedly with chunks of the message to be authenticated (**len** bytes at **data**).

*HMAC\_Final()* places the message authentication code in **md**, which must have space for the hash function output.

**RETURN VALUES**

*HMAC()* returns a pointer to the message authentication code or NULL if an error occurred.

*HMAC\_Init\_ex()*, *HMAC\_Update()* and *HMAC\_Final()* return 1 for success or 0 if an error occurred.

*HMAC\_CTX\_init()* and *HMAC\_CTX\_cleanup()* do not return values.

**CONFORMING TO**

RFC 2104

**SEE ALSO**

[sha\(3\)](#), [evp\(3\)](#)

**HISTORY**

*HMAC()*, *HMAC\_Init()*, *HMAC\_Update()*, *HMAC\_Final()* and *HMAC\_cleanup()* are available since SSLeay 0.9.0.

*HMAC\_CTX\_init()*, *HMAC\_Init\_ex()* and *HMAC\_CTX\_cleanup()* are available since OpenSSL 0.9.7.

*HMAC\_Init\_ex()*, *HMAC\_Update()* and *HMAC\_Final()* did not return values in versions of OpenSSL before 1.0.0.