NAME

CMAC_CTX, CMAC_CTX_new, CMAC_CTX_cleanup, CMAC_CTX_free, CMAC_CTX_get0_cipher_ctx, CMAC_CTX_copy, CMAC_Init, CMAC_Update, CMAC_Final, CMAC_resume − create cipher−based message authentication codes

SYNOPSIS

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

The following functions have been deprecated since OpenSSL 3.0, and can be disabled entirely by defining **OPENSSL_API_COMPAT** with a suitable version value, see **openssl_user_macros**(7).

```
typedef struct CMAC_CTX_st CMAC_CTX;

CMAC_CTX *CMAC_CTX_new(void);
void CMAC_CTX_cleanup(CMAC_CTX *ctx);
void CMAC_CTX_free(CMAC_CTX *ctx);
EVP_CIPHER_CTX *CMAC_CTX_get0_cipher_ctx(CMAC_CTX *ctx);
int CMAC_CTX_copy(CMAC_CTX *out, const CMAC_CTX *in);
int CMAC_Init(CMAC_CTX *ctx, const void *key, size_t keylen,
              const EVP_CIPHER *cipher, ENGINE *impl);
int CMAC_Update(CMAC_CTX *ctx, const void *data, size_t dlen);
int CMAC_Final(CMAC_CTX *ctx, unsigned char *out, size_t *poutlen);
int CMAC_resume(CMAC_CTX *ctx);
```

DESCRIPTION

The low−level MAC functions documented on this page are deprecated. Applications should use the new **EVP_MAC**(3) interface. Specifically, utilize the following functions for MAC operations:

EVP_MAC_CTX_new(3) to create a new MAC context.
EVP_MAC_CTX_free(3) to free the MAC context.
EVP_MAC_init(3) to initialize the MAC context.
EVP_MAC_update(3) to update the MAC with data.
EVP_MAC_final(3) to finalize the MAC and retrieve the output.

Alternatively, for a single−step MAC computation, use the **EVP_Q_mac**(3) function.

The **CMAC_CTX** type is a structure used for the provision of CMAC (Cipher−based Message Authentication Code) operations.

CMAC_CTX_new() creates a new **CMAC_CTX** structure and returns a pointer to it.

CMAC_CTX_cleanup() resets the **CMAC_CTX** structure, clearing any internal data but not freeing the structure itself.

CMAC_CTX_free() frees the **CMAC_CTX** structure and any associated resources. If the argument is NULL, no action is taken.

CMAC_CTX_get0_cipher_ctx() returns a pointer to the internal **EVP_CIPHER_CTX** structure within the **CMAC_CTX**.

CMAC_CTX_copy() copies the state from one **CMAC_CTX** structure to another.

CMAC_Init() initializes the **CMAC_CTX** structure for a new CMAC calculation with the specified key, key length, and cipher type. Optionally, an **ENGINE** can be provided.

CMAC_Update() processes data to be included in the CMAC calculation. This function can be called multiple times to update the context with additional data.

CMAC_Final() finalizes the CMAC calculation and retrieves the resulting MAC value. The output is stored in the provided buffer, and the length is stored in the variable pointed to by *poutlen*. To determine the required buffer size, call with *out* set to NULL, which stores only the length in *poutlen*. Allocate a buffer of this size and call **CMAC_Final()** again with the allocated buffer to retrieve the MAC.

CMAC_resume() resumes a previously finalized CMAC calculation, allowing additional data to be

processed and a new MAC to be generated.

RETURN VALUES

CMAC_CTX_new() returns a pointer to a new **CMAC_CTX** structure or NULL if an error occurs.

CMAC_CTX_get0_cipher_ctx() returns a pointer to the internal **EVP_CIPHER_CTX** structure, or NULL if an error occurs.

CMAC_CTX_copy(), **CMAC_Init()**, **CMAC_Update()**, **CMAC_Final()** and **CMAC_resume()** return 1 for success or 0 if an error occurs.

HISTORY

All functions described here were deprecated in OpenSSL 3.0. For replacements, see **EVP_MAC_CTX_new**(3), **EVP_MAC_CTX_free**(3), **EVP_MAC_init**(3), **EVP_MAC_update**(3), and **EVP_MAC_final**(3).

COPYRIGHT