NAME

EVP_MAC−CMAC − The CMAC EVP_MAC implementation

DESCRIPTION

Support for computing CMAC MACs through the **EVP_MAC** API.

This implementation uses EVP_CIPHER functions to get access to the underlying cipher.

Identity

This implementation is identified with this name and properties, to be used with **EVP_MAC_fetch()**:

‘‘CMAC’’, ‘‘provider=default’’ or ‘‘provider=fips’’

Supported parameters

The general description of these parameters can be found in ‘‘PARAMETERS’’ in **EVP_MAC** (3).

The following parameter can be set with **EVP_MAC_CTX_set_params()**:

‘‘key’’ (**OSSL_MAC_PARAM_KEY**) <octet string>
: Sets the MAC key. Setting this parameter is identical to passing a*k ey* to **EVP_MAC_init** (3).

‘‘cipher’’ (**OSSL_MAC_PARAM_CIPHER**) <UTF8 string>
: Sets the name of the underlying cipher to be used.

‘‘properties’’ (**OSSL_MAC_PARAM_PROPERTIES**) <UTF8 string>
: Sets the properties to be queried when trying to fetch the underlying cipher. This must be given together with the cipher naming parameter to be considered valid.

The following parameters can be retrieved with **EVP_MAC_CTX_get_params()**:

‘‘size’’ (**OSSL_MAC_PARAM_SIZE**) <unsigned integer>
: The ‘‘size’’ parameter can also be retrieved with with **EVP_MAC_CTX_get_mac_size()**. The length of the ‘‘size’’ parameter is equal to that of an **unsigned int**.

‘‘block-size’’ (**OSSL_MAC_PARAM_SIZE**) <unsigned integer>
: Gets the MAC block size. The ‘‘block-size’’ parameter can also be retrieved with **EVP_MAC_CTX_get_block_size()**.

SEE ALSO

EVP_MAC_CTX_get_params (3), **EVP_MAC_CTX_set_params** (3), ‘‘PARAMETERS’’ in **EVP_MAC** (3), **OSSL_PARAM** (3)

COPYRIGHT