NAME

rijndael_makeKey, **rijndael_cipherInit**, **rijndael_blockEncrypt**, **rijndael_padEncrypt**, **rijndael_blockDecrypt**, **rijndael_padDecrypt** - AES encryption

SYNOPSIS

#include <sys/types.h>
#include <crypto/rijndael.h>

int
rijndael_makeKey(*keyInstance *key, uint8_t direction, int keyLen, char *keyMaterial*);

int
rijndael_cipherInit(*cipherInstance *cipher, uint8_t mode, char *IV*);

int
rijndael_blockEncrypt(*cipherInstance *cipher, keyInstance *key, uint8_t *input, int inputLen, uint8_t *outBuffer*);

int
rijndael_padEncrypt(*cipherInstance *cipher, keyInstance *key, uint8_t *input, int inputOctets, uint8_t *outBuffer*);

int
rijndael_blockDecrypt(*cipherInstance *cipher, keyInstance *key, uint8_t *input, int inputLen, uint8_t *outBuffer*);

int
rijndael_padDecrypt(*cipherInstance *cipher, keyInstance *key, uint8_t *input, int inputOctets, uint8_t *outBuffer*);

DESCRIPTION

The **rijndael_makeKey**() function is used to set up the key schedule in *key*. The *direction* (which may be DIR_ENCRYPT or DIR_DECRYPT) specifies the intended use of the key. The length of the key (in bits) is given in *keyLen*, and must be 128, 192 or 256. The actual key is supplied in the buffer pointed to by *keyMaterial*. This material may be raw binary data, or an ASCII string containing a hexadecimal rendition of the raw binary data, dependent on a compile-time option in the **rijndael_makeKey** sources, BINARY_KEY_MATERIAL.

RETURN VALUES

The **rijndael_makeKey**() function will return BAD_KEY_INSTANCE if a NULL *key* is passed,

BAD_KEY_DIR if *direction* is not DIR_ENCRYPT or DIR_DECRYPT, BAD_KEY_MAT if the key materials are not a hexadecimal string (and binary keys are not set), and TRUE otherwise.

AUTHORS

Mark R V Murray