NAME

rs256_pk_new, **rs256_pk_free**, **rs256_pk_from_RSA**, **rs256_pk_from_EVP_PKEY**, **rs256_pk_from_ptr**, **rs256_pk_to_EVP_PKEY** - FIDO2 COSE RS256 API

SYNOPSIS

#include <openssl/rsa.h>
#include <fido/rs256.h>

*rs256_pk_t **
rs256_pk_new(*void*);

void
rs256_pk_free(*rs256_pk_t **pkp*);

int
rs256_pk_from_EVP_PKEY(*rs256_pk_t *pk, const EVP_PKEY *pkey*);

int
rs256_pk_from_RSA(*rs256_pk_t *pk, const RSA *rsa*);

int
rs256_pk_from_ptr(*rs256_pk_t *pk, const void *ptr, size_t len*);

*EVP_PKEY **
rs256_pk_to_EVP_PKEY(*const rs256_pk_t *pk*);

DESCRIPTION

RS256 is the name given in the CBOR Object Signing and Encryption (COSE) RFC to PKCS#1.5 2048-bit RSA with SHA-256. The COSE RS256 API of *libfido2* is an auxiliary API with routines to convert between the different RSA public key types used in *libfido2* and *OpenSSL*.

In *libfido2*, RS256 public keys are abstracted by the *rs256_pk_t* type.

The **rs256_pk_new**() function returns a pointer to a newly allocated, empty *rs256_pk_t* type. If memory cannot be allocated, NULL is returned.

The **rs256_pk_free**() function releases the memory backing **pkp*, where **pkp* must have been previously allocated by **rs256_pk_new**(). On return, **pkp* is set to NULL. Either *pkp* or **pkp* may be NULL, in which case **rs256_pk_free**() is a NOP.

The **rs256_pk_from_EVP_PKEY**() function fills *pk* with the contents of *pkey*. No references to *pkey* are kept.

The **rs256_pk_from_RSA**() function fills *pk* with the contents of *rsa*. No references to *rsa* are kept.

The **rs256_pk_from_ptr**() function fills *pk* with the contents of *ptr*, where *ptr* points to *len* bytes. No references to *ptr* are kept.

The **rs256_pk_to_EVP_PKEY**() function converts *pk* to a newly allocated *EVP_PKEY* type with a reference count of 1. No internal references to the returned pointer are kept. If an error occurs, **rs256_pk_to_EVP_PKEY**() returns NULL.

RETURN VALUES

The **rs256_pk_from_EVP_PKEY**(), **rs256_pk_from_RSA**(), and **rs256_pk_from_ptr**() functions return FIDO_OK on success. On error, a different error code defined in <*fido/err.h*> is returned.

SEE ALSO

eddsa_pk_new(3), es256_pk_new(3), es384_pk_new(3), fido_assert_verify(3), fido_cred_pubkey_ptr(3)