

**NAME**

**es384\_pk\_new**, **es384\_pk\_free**, **es384\_pk\_from\_EC\_KEY**, **es384\_pk\_from\_EVP\_PKEY**,  
**es384\_pk\_from\_ptr**, **es384\_pk\_to\_EVP\_PKEY** - FIDO2 COSE ES384 API

**SYNOPSIS**

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

```
#include <fido/es384.h>
```

```
es384_pk_t *
```

```
es384_pk_new(void);
```

```
void
```

```
es384_pk_free(es384_pk_t **pkp);
```

```
int
```

```
es384_pk_from_EC_KEY(es384_pk_t *pk, const EC_KEY *ec);
```

```
int
```

```
es384_pk_from_EVP_PKEY(es384_pk_t *pk, const EVP_PKEY *pkey);
```

```
int
```

```
es384_pk_from_ptr(es384_pk_t *pk, const void *ptr, size_t len);
```

```
EVP_PKEY *
```

```
es384_pk_to_EVP_PKEY(const es384_pk_t *pk);
```

**DESCRIPTION**

ES384 is the name given in the CBOR Object Signing and Encryption (COSE) RFC to ECDSA over P-384 with SHA-384. The COSE ES384 API of *libfido2* is an auxiliary API with routines to convert between the different ECDSA public key types used in *libfido2* and *OpenSSL*.

In *libfido2*, ES384 public keys are abstracted by the *es384\_pk\_t* type.

The **es384\_pk\_new**() function returns a pointer to a newly allocated, empty *es384\_pk\_t* type. If memory cannot be allocated, NULL is returned.

The **es384\_pk\_free**() function releases the memory backing \**pkp*, where \**pkp* must have been previously allocated by **es384\_pk\_new**(). On return, \**pkp* is set to NULL. Either *pkp* or \**pkp* may be NULL, in which case **es384\_pk\_free**() is a NOP.

The **es384\_pk\_from\_EC\_KEY()** function fills *pk* with the contents of *ec*. No references to *ec* are kept.

The **es384\_pk\_from\_EVP\_PKEY()** function fills *pk* with the contents of *pkey*. No references to *pkey* are kept.

The **es384\_pk\_from\_ptr()** function fills *pk* with the contents of *ptr*, where *ptr* points to *len* bytes. The *ptr* pointer may point to an uncompressed point, or to the concatenation of the x and y coordinates. No references to *ptr* are kept.

The **es384\_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, **es384\_pk\_to\_EVP\_PKEY()** returns NULL.

## RETURN VALUES

The **es384\_pk\_from\_EC\_KEY()**, **es384\_pk\_from\_EVP\_PKEY()**, and **es384\_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)**, **fido\_assert\_verify(3)**, **fido\_cred\_pubkey\_ptr(3)**, **rs256\_pk\_new(3)**