

**NAME**

**eddsa\_pk\_new**, **eddsa\_pk\_free**, **eddsa\_pk\_from\_EVP\_PKEY**, **eddsa\_pk\_from\_ptr**,  
**eddsa\_pk\_to\_EVP\_PKEY** - FIDO2 COSE EDDSA API

**SYNOPSIS**

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

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

```
eddsa_pk_t *
```

```
eddsa_pk_new(void);
```

```
void
```

```
eddsa_pk_free(eddsa_pk_t **pkp);
```

```
int
```

```
eddsa_pk_from_EVP_PKEY(eddsa_pk_t *pk, const EVP_PKEY *pkey);
```

```
int
```

```
eddsa_pk_from_ptr(eddsa_pk_t *pk, const void *ptr, size_t len);
```

```
EVP_PKEY *
```

```
eddsa_pk_to_EVP_PKEY(const eddsa_pk_t *pk);
```

**DESCRIPTION**

EDDSA is the name given in the CBOR Object Signing and Encryption (COSE) RFC to EDDSA over Curve25519 with SHA-512. The COSE EDDSA API of *libfido2* is an auxiliary API with routines to convert between the different EDDSA public key types used in *libfido2* and *OpenSSL*.

In *libfido2*, EDDSA public keys are abstracted by the *eddsa\_pk\_t* type.

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

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

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

The **eddsa\_pk\_from\_ptr()** function fills *pk* with the contents of *ptr*, where *ptr* points to *len* bytes. No references to *ptr* are kept.

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

## RETURN VALUES

The **eddsa\_pk\_from\_EVP\_PKEY()** and **eddsa\_pk\_from\_ptr()** functions return FIDO\_OK on success. On error, a different error code defined in *<fido/err.h>* is returned.

## SEE ALSO

es256\_pk\_new(3), es384\_pk\_new(3), fido\_assert\_verify(3), fido\_cred\_pubkey\_ptr(3), rs256\_pk\_new(3)