NAME

fido_cred_new, **fido_cred_free**, **fido_cred_pin_minlen**, **fido_cred_prot**, **fido_cred_fmt**, **fido_cred_rp_id**, **fido_cred_rp_name**, **fido_cred_user_name**, **fido_cred_display_name**, **fido_cred_authdata_ptr**, **fido_cred_authdata_raw_ptr**, **fido_cred_clientdata_hash_ptr**, **fido_cred_id_ptr**, **fido_cred_aaguid_ptr**, **fido_cred_largeblob_key_ptr**, **fido_cred_pubkey_ptr**, **fido_cred_sig_ptr**, **fido_cred_user_id_ptr**, **fido_cred_x5c_ptr**, **fido_cred_attstmt_ptr**, **fido_cred_authdata_len**, **fido_cred_authdata_raw_len**, **fido_cred_clientdata_hash_len**, **fido_cred_id_len**, **fido_cred_aaguid_len**, **fido_cred_largeblob_key_len**, **fido_cred_pubkey_len**, **fido_cred_sig_len**, **fido_cred_user_id_len**, **fido_cred_x5c_len**, **fido_cred_attstmt_len**, **fido_cred_type**, **fido_cred_flags**, **fido_cred_sigcount** **-** FIDO2 credential API

SYNOPSIS

#include <fido.h>

*fido_cred_t **
fido_cred_new(*void*);

void
fido_cred_free(*fido_cred_t **cred_p*);

size_t
fido_cred_pin_minlen(*const fido_cred_t *cred*);

int
fido_cred_prot(*const fido_cred_t *cred*);

*const char **
fido_cred_fmt(*const fido_cred_t *cred*);

*const char **
fido_cred_rp_id(*const fido_cred_t *cred*);

*const char **
fido_cred_rp_name(*const fido_cred_t *cred*);

*const char **
fido_cred_user_name(*const fido_cred_t *cred*);

*const char **
fido_cred_display_name(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_authdata_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_authdata_raw_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_clientdata_hash_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_id_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_aaguid_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_largeblob_key_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_pubkey_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_sig_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_user_id_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_x5c_ptr(*const fido_cred_t *cred*);

*const unsigned char **
fido_cred_attstmt_ptr(*const fido_cred_t *cred*);

size_t
fido_cred_authdata_len(*const fido_cred_t *cred*);

size_t
fido_cred_authdata_raw_len(*const fido_cred_t *cred*);

size_t

fido_cred_clientdata_hash_len(*const fido_cred_t *cred*);

size_t
fido_cred_id_len(*const fido_cred_t *cred*);

size_t
fido_cred_aaguid_len(*const fido_cred_t *cred*);

size_t
fido_cred_largeblob_key_len(*const fido_cred_t *cred*);

size_t
fido_cred_pubkey_len(*const fido_cred_t *cred*);

size_t
fido_cred_sig_len(*const fido_cred_t *cred*);

size_t
fido_cred_user_id_len(*const fido_cred_t *cred*);

size_t
fido_cred_x5c_len(*const fido_cred_t *cred*);

size_t
fido_cred_attstmt_len(*const fido_cred_t *cred*);

int
fido_cred_type(*const fido_cred_t *cred*);

uint8_t
fido_cred_flags(*const fido_cred_t *cred*);

uint32_t
fido_cred_sigcount(*const fido_cred_t *cred*);

DESCRIPTION

FIDO2 credentials are abstracted in *libfido2* by the *fido_cred_t* type. The functions described in this page allow a *fido_cred_t* type to be allocated, deallocated, and inspected. For other operations on *fido_cred_t*, please refer to fido_cred_set_authdata(3), fido_cred_exclude(3), fido_cred_verify(3), and fido_dev_make_cred(3).

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

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

If the CTAP 2.1 FIDO_EXT_MINPINLEN extension is enabled on *cred*, then the **fido_cred_pin_minlen**() function returns the minimum PIN length of *cred*. Otherwise, **fido_cred_pin_minlen**() returns zero. See fido_cred_set_pin_minlen(3) on how to enable this extension.

If the CTAP 2.1 FIDO_EXT_CRED_PROTECT extension is enabled on *cred*, then the **fido_cred_prot**() function returns the protection of *cred*. Otherwise, **fido_cred_prot**() returns zero. See fido_cred_set_prot(3) for the protection policies understood by *libfido2*.

The **fido_cred_fmt**() function returns a pointer to a NUL-terminated string containing the attestation statement format identifier of *cred*, or NULL if *cred* does not have a format set.

The **fido_cred_rp_id**(), **fido_cred_rp_name**(), **fido_cred_user_name**(), and **fido_cred_display_name**() functions return pointers to NUL-terminated strings holding the relying party ID, relying party name, user name, and user display name attributes of *cred*, or NULL if the respective entry is not set.

The **fido_cred_authdata_ptr**(), **fido_cred_authdata_raw_ptr**(), **fido_cred_clientdata_hash_ptr**(), **fido_cred_id_ptr**(), **fido_cred_aaguid_ptr**(), **fido_cred_largeblob_key_ptr**(), **fido_cred_pubkey_ptr**(), **fido_cred_sig_ptr**(), **fido_cred_user_id_ptr**(), **fido_cred_x5c_ptr**(), and **fido_cred_attstmt_ptr**() functions return pointers to the CBOR-encoded and raw authenticator data, client data hash, ID, authenticator attestation GUID, "largeBlobKey", public key, signature, user ID, x509 certificate, and attestation statement parts of *cred*, or NULL if the respective entry is not set.

The corresponding length can be obtained by **fido_cred_authdata_len**(), **fido_cred_authdata_raw_len**(), **fido_cred_clientdata_hash_len**(), **fido_cred_id_len**(), **fido_cred_aaguid_len**(), **fido_cred_largeblob_key_len**(), **fido_cred_pubkey_len**(), **fido_cred_sig_len**(), **fido_cred_user_id_len**(), **fido_cred_x5c_len**(), and **fido_cred_attstmt_len**().

The authenticator data, x509 certificate, and signature parts of a credential are typically passed to a FIDO2 server for verification.

The **fido_cred_type**() function returns the COSE algorithm of *cred*.

The **fido_cred_flags**() function returns the authenticator data flags of *cred*.

The **fido_cred_sigcount**() function returns the authenticator data signature counter of *cred*.

RETURN VALUES

The authenticator data returned by **fido_cred_authdata_ptr**() is a CBOR-encoded byte string, as obtained from the authenticator. To obtain the decoded byte string, use **fido_cred_authdata_raw_ptr**().

If not NULL, pointers returned by **fido_cred_fmt**(), **fido_cred_authdata_ptr**(), **fido_cred_clientdata_hash_ptr**(), **fido_cred_id_ptr**(), **fido_cred_aaguid_ptr**(), **fido_cred_largeblob_key_ptr**(), **fido_cred_pubkey_ptr**(), **fido_cred_sig_ptr**(), and **fido_cred_x5c_ptr**() are guaranteed to exist until any API function that takes *cred* without the *const* qualifier is invoked.

SEE ALSO

fido_cred_exclude(3), fido_cred_set_authdata(3), fido_cred_set_pin_minlen(3), fido_cred_set_prot(3), fido_cred_verify(3), fido_credman_metadata_new(3), fido_dev_largeblob_get(3), fido_dev_make_cred(3)