

**NAME**

libssh2\_sign\_sk - Create a signature from a FIDO2 authenticator.

**SYNOPSIS**

```
#include <libssh2.h>
```

```
int
```

```
libssh2_sign_sk(LIBSSH2_SESSION *session,  
               unsigned char **sig,  
               size_t *sig_len,  
               const unsigned char *data,  
               size_t data_len,  
               void **abstract);
```

```
typedef struct _LIBSSH2_PRIVKEY_SK {  
    int algorithm;  
    uint8_t flags;  
    const char *application;  
    const unsigned char *key_handle;  
    size_t handle_len;  
    LIBSSH2_USERAUTH_SK_SIGN_FUNC((*sign_callback));  
    void **orig_abstract;  
} LIBSSH2_PRIVKEY_SK;
```

**DESCRIPTION**

*session* - Session instance as returned by **libssh2\_session\_init\_ex(3)**

*sig* - A pointer to a buffer in which to place the signature. The caller is responsible for freeing the signature with LIBSSH2\_FREE.

*sig\_len* - A pointer to the length of the sig parameter.

*data* - The data to sign.

*data\_len* - The length of the data parameter.

*abstract* - A pointer to a pointer to a LIBSSH2\_PRIVKEY\_SK. See description below.

Create a signature from a FIDO2 authenticator, using either the sk-ssh-ed25519@openssh.com or sk-ecdsa-sha2-nistp256@openssh.com key exchange algorithms.

The abstract parameter is a pointer to a pointer due to the internal workings of libssh2. The LIBSSH2\_PRIVKEY\_SK must be completely filled out, and the caller is responsible for all memory management of its fields.

*algorithm* - The signing algorithm to use. Possible values are LIBSSH2\_HOSTKEY\_TYPE\_ED25519 and LIBSSH2\_HOSTKEY\_TYPE\_ECDSA\_256.

*flags* - A bitmask specifying options for the authenticator. When LIBSSH2\_SK\_PRESENCE\_REQUIRED is set, the authenticator requires a touch. When LIBSSH2\_SK\_VERIFICATION\_REQUIRED is set, the authenticator requires a PIN. Many servers and authenticators do not work properly when LIBSSH2\_SK\_PRESENCE\_REQUIRED is not set.

*application* - A user-defined string to use as the RP name for the authenticator. Usually "ssh:".

*key\_handle* - The key handle to use for the authenticator's allow list.

*handle\_len* - The length of the key\_handle parameter.

*abstract* - User-defined data. When a PIN is required, use this to pass in the PIN, or a function pointer to retrieve the PIN.

*key\_handle* The decoded key handle from the private key file.

*handle\_len* The length of the key\_handle parameter.

*sign\_callback* - Responsible for communicating with the hardware authenticator to generate a signature. On success, the signature information must be placed in the 'sig\_info' sig\_info parameter and the callback must return 0. On failure, it should return a negative number. See

**libssh2\_userauth\_publickey\_sk(3)**

for more information.

*orig\_abstract* - User-defined data. When a PIN is required, use this to pass in the PIN, or a function pointer to retrieve the PIN.

## RETURN VALUE

Return 0 on success or negative on failure.

## SEE ALSO

**libssh2\_userauth\_publickey\_sk(3)**