NAME

gnutls_certificate_set_rawpk_key_file - API function

SYNOPSIS

#include <gnutls/gnutls.h>

int gnutls_certificate_set_rawpk_key_file(gnutls_certificate_credentials_t *cred***, const char*** *rawpkfile***, const char*** *privkeyfile***, gnutls_x509_crt_fmt_t** *format***, const char *** *pass***, unsigned int** *key_usage***, const char **** *names***, unsigned int** *names_length***, unsigned int** *privkey_flags***, unsigned int** *pkcs11_flags***);**

ARGUMENTS

gnutls_certificate_credentials_t cred
: is a **gnutls_certificate_credentials_t** type.

const char* rawpkfile
: contains a raw public key in PKIX.SubjectPublicKeyInfo format.

const char* privkeyfile
: contains a file path to a private key.

gnutls_x509_crt_fmt_t format
: encoding of the keys. DER or PEM.

const char * pass
: an optional password to unlock the private key privkeyfile.

unsigned int key_usage
: an ORed sequence of **GNUTLS_KEY_*** flags.

const char ** names
: is an array of DNS names belonging to the public-key (NULL if none).

unsigned int names_length
: holds the length of the names list.

unsigned int privkey_flags
: an ORed sequence of **gnutls_pkcs_encrypt_flags_t**. These apply to the private key pkey.

unsigned int pkcs11_flags
one of gnutls_pkcs11_obj_flags. These apply to URLs.

DESCRIPTION

This function sets a public/private keypair read from file in the **gnutls_certificate_credentials_t** type to be used for authentication and/or encryption. *spki* and *privkey* should match otherwise set signatures cannot be validated. In case of no match this function returns **GNUTLS_E_CERTIFICATE_KEY_MISMATCH**. This function should be called once for the client because there is currently no mechanism to determine which raw public-key to select for the peer when there are multiple present. Multiple raw public keys for the server can be distinghuished by setting the *names* .

Note here that *spki* is a raw public-key as defined in RFC7250. It means that there is no surrounding certificate that holds the public key and that there is therefore no direct mechanism to prove the authenticity of this key. The keypair can be used during a TLS handshake but its authenticity should be established via a different mechanism (e.g. TOFU or known fingerprint).

The supported formats are basic unencrypted key, PKCS8, PKCS12, and the openssl format and will be autodetected.

If the raw public-key and the private key are given in PEM encoding then the strings that hold their values must be null terminated.

Key usage (as defined by X.509 extension (2.5.29.15)) can be explicitly set because there is no certificate structure around the key to define this value. See for more info **gnutls_x509_crt_get_key_usage()**.

Note that, this function by default returns zero on success and a negative value on error. Since 3.5.6, when the flag **GNUTLS_CERTIFICATE_API_V2** is set using **gnutls_certificate_set_flags()** it returns an index (greater or equal to zero). That index can be used in other functions to refer to the added key-pair.

RETURNS

On success, **GNUTLS_E_SUCCESS** (0) is returned, in case the key pair does not match **GNUTLS_E_CERTIFICATE_KEY_MISMATCH** is returned, in other erroneous cases a different negative error code is returned.

SINCE

3.6.6

REPORTING BUGS

Report bugs to <bugs@gnutls.org>.
Home page: https://www.gnutls.org

COPYRIGHT

SEE ALSO

The full documentation for **gnutls** is maintained as a Texinfo manual. If the /usr/local/share/doc/gnutls/ directory does not contain the HTML form visit

https://www.gnutls.org/manual/