qgx2009

qgx2009.blog.chinaunix.net

qgx2009

博客访问: 1131773
博文数量: 389
博客积分: 2874
博客等级: 少校
技术积分: 3577
用 户 组: 普通用户
注册时间: 2009-10-24 10:34

gmssl C语言制作证书

原创 分类: C/C++ 2018-07-04 15:40:32

重点是mkcert函数

点击(此处)折叠或打开

```
#include <stdio.h>
#include <openssl/ssl.h>
#include <openssl/err.h>
#include <openssl/pem.h>
#include <openssl/conf.h>
#include <openssl/x509v3.h>

#include <openssl/bn.h>
#include <openssl/evp.h>
#include <openssl/asn1.h>
#include <openssl/x509.h>
#include <openssl/objects.h>
#include <openssl/buffer.h>

#ifndef OPENSSL_NO_ENGINE
#include <openssl/engine.h>
#endif

void die_most_horribly_from_openssl_error (const char *func)
{
    /* This is the OpenSSL function that prints the contents of the
     * error stack to the specified file handle. */
    BIO *mem = BIO_new(BIO_s_mem());
    ERR_print_errors(mem);

    BUF_MEM *pp;
    BIO_get_mem_ptr(mem, &pp);
    printf("%s ERROR %s", func, pp->data);

    BIO_free(mem);
    exit (EXIT_FAILURE);
}

static void callback(int p, int n, void *arg)
{
    char c = 'B';

    if (p == 0) c = '.';
    if (p == 1) c = '+';
    if (p == 2) c = '*';
    if (p == 3) c = '\n';
    fputc(c, stderr);
}

/*
 * pkeyp: cert's public key
 * sign_privkeyp: sign private key
 * issuer_name: issuer name,use for cert chain,if self sign,this is null,else ca's name
 * cn : cert's subject cn entry
 * days:
 */
int mkcert(X509 **x509p, EVP_PKEY **pkeyp, EVP_PKEY **sign_privkeyp, X509_NAME *issuer_nam
e, const char* cn,int bits, int serial, int days)
{
    X509 *x;
    EVP_PKEY *pk;
    X509_NAME *name = NULL;

    if ((pkeyp == NULL) || (*pkeyp == NULL))
    {
        if ((pk = EVP_PKEY_new()) == NULL)
        {
            abort();
            return(0);
        }
    }
    else
        pk = *pkeyp;

    if ((x509p == NULL) || (*x509p == NULL))
    {
        if ((x = X509_new()) == NULL)
            goto err;
    }
    else
        x = *x509p;

    X509_set_version(x, 2);
    ASN1_INTEGER_set(X509_get_serialNumber(x), serial);
    X509_gmtime_adj(X509_get_notBefore(x), 0);
    X509_gmtime_adj(X509_get_notAfter(x), (long)60 * 60 * 24 * days);
    X509_set_pubkey(x, pk);

    name = X509_get_subject_name(x);


    /* This function creates and adds the entry, working out the
     * correct string type and performing checks on its length.
     * Normally we'd check the return value for errors...
     */
    X509_NAME_add_entry_by_txt(name, "C",
                               MBSTRING_ASC, "UK", -1, -1, 0);
    X509_NAME_add_entry_by_txt(name, "CN",
                               MBSTRING_ASC, cn, -1, -1, 0);

    /* Its self signed so set the issuer name to be the same as the
     * subject.
     */
    if (!issuer_name)
        X509_set_issuer_name(x, name);
    else
        X509_set_issuer_name(x, issuer_name);

    /* Add various extensions: standard extensions */
    add_ext(x, NID_basic_constraints, "critical,CA:TRUE");
    add_ext(x, NID_key_usage, "critical,keyCertSign,cRLSign");

    add_ext(x, NID_subject_key_identifier, "hash");

    /* Some Netscape specific extensions */
    add_ext(x, NID_netscape_cert_type, "sslCA");

    add_ext(x, NID_netscape_comment, "example comment extension");


#ifdef CUSTOM_EXT
    /* Maybe even add our own extension based on existing */
    {
        int nid;
        nid = OBJ_create("1.2.3.4", "MyAlias", "My Test Alias Extension");
        X509V3_EXT_add_alias(nid, NID_netscape_comment);
        add_ext(x, nid, "example comment alias");
    }
#endif

    if (!X509_sign(x, *sign_privkeyp, EVP_sha1()))
        goto err;

    *x509p = x;
    *pkeyp = pk;
    return(1);
err:
    return(0);
}

/* Add extension using V3 code: we can set the config file as NULL
 * because we wont reference any other sections.
 */
int add_ext(X509 *cert, int nid, char *value)
{
    X509_EXTENSION *ex;
    X509V3_CTX ctx;
    /* This sets the 'context' of the extensions. */
    /* No configuration database */
    X509V3_set_ctx_nodb(&ctx);
    /* Issuer and subject certs: both the target since it is self signed,
     * no request and no CRL
     */
    X509V3_set_ctx(&ctx, cert, cert, NULL, NULL, 0);
    ex = X509V3_EXT_conf_nid(NULL, &ctx, nid, value);
    if (!ex)
        return 0;

    X509_add_ext(cert, ex, -1);
    X509_EXTENSION_free(ex);
    return 1;
}

int main()
{
    OpenSSL_add_all_ciphers();
    OpenSSL_add_all_digests();
    ERR_load_crypto_strings();

    SSL_CTX *ctx = SSL_CTX_new (SSLv23_server_method ());
    SSL_CTX_set_options (ctx,
                         SSL_OP_SINGLE_DH_USE |
                         SSL_OP_SINGLE_ECDH_USE |
                         SSL_OP_NO_SSLv2);

    /* Cheesily pick an elliptic curve to use with elliptic curve ciphersuites.
     * We just hardcode a single curve which is reasonably decent.
     * See http://www.mail-archive.com/openssl-dev@openssl.org/msg30957.html */
    EC_KEY *ecdh = EC_KEY_new_by_curve_name (NID_sm2p256v1);
    if (! ecdh)
        die_most_horribly_from_openssl_error ("EC_KEY_new_by_curve_name");

    //PEM_write_bio_ECPrivateKey(BIO *bp, EC_KEY *x, const EVP_CIPHER *enc,
    // unsigned char *kstr, int klen, pem_password_cb *cb, void *u);
    //ec_key_simple_generate_key(ecdh);
    EC_KEY_generate_key(ecdh);

    BIO *bio_out = BIO_new_file("SM2_private.pem", "w");

    PEM_write_bio_ECPrivateKey(bio_out, ecdh, NULL, NULL, 0, NULL, NULL);

    BIO_free(bio_out);

    //----------------make self sign cert--------------
    EVP_PKEY *evk = EVP_PKEY_new();
    EVP_PKEY_set1_EC_KEY(evk, ecdh);
    X509 *x509 = NULL;

    mkcert(&x509, &evk, &evk,0,"Openssl Group", 256, 2345, 3650);
    BIO *bio_x = BIO_new_file("cert.pem", "w");
    PEM_write_bio_X509(bio_x, x509);

    BIO_free(bio_x);

    EVP_PKEY_free(evk);
    //----------------make server cert ---------------------
    evk = EVP_PKEY_new(); //ca key
    EVP_PKEY_set1_EC_KEY(evk, ecdh);

    EC_KEY *ec_server = EC_KEY_new_by_curve_name (NID_sm2p256v1);
    if (! ec_server)
        die_most_horribly_from_openssl_error ("EC_KEY_new_by_curve_name");
```

```
208.
209.      EC_KEY_generate_key(ec_server);
210.      bio_out = BIO_new_file("server.pem", "w");
211.
212.      PEM_write_bio_ECPrivateKey(bio_out, ec_server, NULL, NULL, 0, NULL, NULL);
213.      BIO_free(bio_out);
214.
215.      //
216.      EVP_PKEY *evserv = EVP_PKEY_new();
217.      EVP_PKEY_set1_EC_KEY(evserv, ec_server);
218.
219.      X509 *serv_x509 = NULL;
220.      X509_NAME *is_name = X509_get_subject_name(x509);
221.
222.      mkcert(&serv_x509,&evserv , &evk,is_name,"gd.com", 256, 2345, 3650);
223.      BIO *bio_s = BIO_new_file("server-cert.pem", "w");
224.      PEM_write_bio_X509(bio_s, serv_x509);
225.
226.      BIO_free(bio_s);
227.      EVP_PKEY_free(evserv);
228.      EC_KEY_free(ec_server);
229.      EVP_PKEY_free(evk);
230.      //----------------
231.
232.      EC_KEY_free(ecdh);
233.      SSL_CTX_free(ctx);
234.      return 0;
235.  }
```

参考github上的代码。

上一篇:C语言的位域
下一篇:C语言跳表(skiplist)实现

给主人留下些什么吧！~~