Ασφάλεια

ΛΙΑΓΚΟΥ ΒΑΣΙΛΙΚΗ

Εργαστήριο

A. Γράψτε πρόγραμμα με την χρήση openssl crypto εντολές

1. Γράψτε ένα απλό πρόγραμμα που θα εκτυπώσει "HELLO world!" και μεταγλωττίστε το με την ακόλουθη γραμμή εντολών

- Write a simple program that will print “HELLO world!” and compile it with the following command line
  - ./gcc –Wall first.c –lcrypto –o first
  - In your program add the following libraries
    - #include <stdio.h>
    - #include <openssl/evp.h>
    - #include <openssl/aes.h>
    - #include <openssl/err.h>
    - #include <string.h>

2. int main(int arc, char *argv[])

- {printf (“Hello WORLD\n”); return 1;
- }

Αρχικοποίηση

- int main(int arc, char *argv[])
- {
  - OpenSSL_add_all_algorithms(); //load all cipher algorithms
    ERR_load_crypto_strings(); //load human readable errors

  - //the above two lines are needed for initialization and to be able to use the libcrypto library

Κλήση Συνάρτηση κρυπτογράφησης

- Τι χρειαζόμαστε για τη συνάρτηση κρυπτογράφησης;
  - Αρχείο εισόδου για την κρυπτογραφηση-plaintext
  - Μήκος του απλού κειμένου
  - Κλειδί

- Η κρυπτογράφηση αποτελείται από τα ακόλουθα στάδια:
- Ρύθμιση περιβάλλοντος
- Αρχικοποίηση της λειτουργίας κρυπτογράφησης
- Παροχή απλού κειμένου byte για κρυπτογράφηση
- Ολοκλήρωση της λειτουργίας κρυπτογράφησης

Κλήση Encrypt() function με AES

- int main(int arc, char *argv[])
- {
    - OpenSSL_add_all_algorithms();
    - ERR_load_crypto_strings();
    - /* A 256 bit key */
    - static const unsigned char key[] = "01234567890123456789012345678901";
- unsigned char plaintext[] = "CS475 is an awesome course about computer security in the University of Cyprus.";
    - /* Buffer to store the ciphertext. The size may be different due to padding. */ unsigned char ciphertext[128];
    - /* Buffer for the decrypted text for verifying decryption. */ unsigned char decryptedtext[128];
- int decryptedtext_len = 0, ciphertext_len = 0;

Κλήση Encrypt() function με AES

- /* Encryption. */
  - ciphertext_len = encrypt(plaintext, strlen((char *)plaintext), key, ciphertext); printf("Ciphertext is:\n");
- BIO_dump_fp(stdout, (const char *)ciphertext, ciphertext_len);

- • }

Συνάρτηση encrypt εσωτερικά

- int encrypt(unsigned char *plaintext, int plaintext_len, const unsigned char*key, unsigned char *ciphertext)
- {EVP_CIPHER_CTX *ctx = NULL;
- / * Δημιουργήστε και αρχικοποιήστε το περιβάλλον και επιστρέφει ένα δείκτη για επιτυχία και μηδενικό αλλιώς */
- int len = 0, ciphertext_len = 0;
- if(!(ctx = EVP_CIPHER_CTX_new())) handleErrors();
- /* Initialize the encryption operation. */
- if(1 != EVP_EncryptInit_ex(ctx, EVP_aes_128_ecb(), NULL, key, NULL)) handleErrors();

Function: encrypt()

```
/* Initialize the encryption operation. */
if(1 != EVP_EncryptInit_ex(ctx, EVP_aes_128_ecb(), NULL, key, NULL))
        handleErrors();
```

Function: EVP_EncryptUpdate

/* Provide the message to be encrypted, and obtain the encrypted output.
* EVP_EncryptUpdate can be called multiple times if necessary*/

```
int EVP_EncryptUpdate(EVP_CIPHER_CTX *ctx, unsigned char *out,
        int *outl, const unsigned char *in, int inl);
```

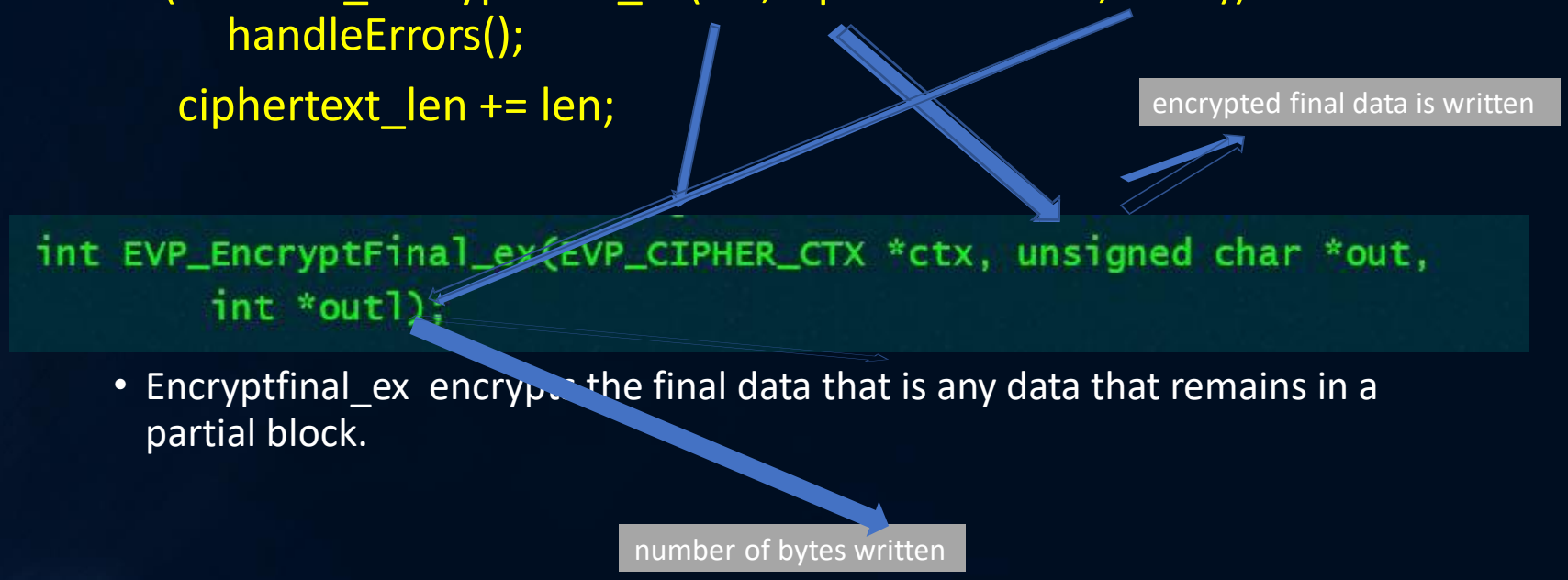
```
if(plaintext)
 {
if(1 != EVP_EncryptUpdate(ctx, ciphertext, &len, plaintext, plaintext_len))
 handleErrors();
ciphertext_len = len;
 }
```

encrypted version

actual number of bytes written

Function: EVP_EncryptFinal_ex

```
if(1 != EVP_EncryptFinal_ex(ctx, ciphertext + len, &len))
        handleErrors();
    ciphertext_len += len;
```



- Encryptfinal_ex encrypts the final data that is any data that remains in a partial block.

Function: EVP_CIPHER_CTX_free

```
/* Clean up */
EVP_CIPHER_CTX_free(ctx);
return ciphertext_len;
}
```

```
void EVP_CIPHER_CTX_free(EVP_CIPHER_CTX *ctx);
```

EVP_CIPHER_CTX_FREE clears all information from a cipher context and free up any allocated memory associate with it, including **ctx** itself.

- https://www.dynamsoft.com/codepool/how-to-use-openssl-generate-rsa-keys-cc.html

RSA κρυπτογράφηση και αποκρυπτογράφηση.

- Παρακάτω είναι το OpenSSL API για δημόσια κρυπτογράφηση και ιδιωτική αποκρυπτογράφηση.

- int RSA_public_encrypt (int flen, unsigned char * from,
- χωρίς υπογραφή char * to, RSA * rsa, int padding);

- int RSA_private_decrypt (int flen, unsigned char * from,
- χωρίς υπογραφή char * to, RSA * rsa, int padding);
-

RSA κρυπτογράφηση και αποκρυπτογράφηση.

```
RSA * createRSA(unsigned char * key,int public){
    RSA *rsa= NULL;
    BIO *keybio ;
    keybio = BIO_new_mem_buf(key, -1);
        if (keybio==NULL)   {
            printf( "Failed to create key BIO");
            return 0;   }
        if(public)   {
            rsa = PEM_read_bio_RSA_PUBKEY(keybio, &rsa,NULL, NULL);   }
        else   {
            rsa = PEM_read_bio_RSAPrivateKey(keybio, &rsa,NULL, NULL);   }
     return rsa;}
```

RSA κρυπτογράφηση και αποκρυπτογράφηση.

- Εάν θέλετε να δημιουργήσετε RSA με όνομα αρχείου κλειδιού, μπορείτε να χρησιμοποιήσετε αυτήν τη λειτουργία
- RSA * createRSAWithFilename(char * filename,int public){
- FILE * fp = fopen(filename,"rb");
- if(fp == NULL) {
- printf("Unable to open file %s \n",filename);
- return NULL; }
- RSA *rsa= RSA_new() ;
- if(public) {
- rsa = PEM_read_RSA_PUBKEY(fp, &rsa,NULL, NULL); }
- else {
- rsa = PEM_read_RSAPrivateKey(fp, &rsa,NULL, NULL); }
- return rsa;}

RSA κρυπτογράφηση και αποκρυπτογράφηση.

- Για κρυπτογράφηση μπορούμε να χρησιμοποιήσουμε padding, παρακάτω είναι η λίστα των υποστηριζόμενων paddings.

- **RSA_PKCS1_PADDING**
  PKCS # 1 v1.5 επένδυση. Αυτή τη στιγμή είναι η πιο ευρέως χρησιμοποιούμενη λειτουργία.
  RSA_PKCS1_OAEP_PADDING
  EME-OAEP όπως ορίζεται στο PKCS # 1 v2.0 με SHA-1, MGF1 και μια κενή παράμετρο κωδικοποίησης. Αυτή η λειτουργία συνιστάται για όλες τις νέες εφαρμογές.
  RSA_SSLV23_PADDING
  PKCS # 1 v1.5 padding με ειδική τροποποίηση SSL που υποδηλώνει ότι ο διακομιστής είναι ικανός για SSL3.
  RSA_NO_PADDING
  Raw RSA κρυπτογράφηση. Αυτή η λειτουργία θα πρέπει να χρησιμοποιείται μόνο για την εφαρμογή κρυπτογραφικά ήχων padding στον κώδικα εφαρμογής. Η κρυπτογράφηση δεδομένων χρήστη απευθείας με RSA δεν είναι ασφαλής.

RSA κρυπτογράφηση και αποκρυπτογράφηση.

- int RSA_public_encrypt(int flen, unsigned char *from,
- unsigned char *to, RSA *rsa, int padding);
-
- int RSA_private_decrypt(int flen, unsigned char *from,
- unsigned char *to, RSA *rsa, int padding);

Πηγες

- https://www.openssl.org/docs/man1.1.0/crypto/EVP_EncryptInit.html
- https://wiki.openssl.org/index.php/Libcrypto_API
- http://hayageek.com/rsa-encryption-decryption-openssl-c/#generate