CCA – COMPETENCE CENTRE

HTL Anichstraße

FSST-OpenSSL

Fabio Plunser

17. Februar 2021

Inhaltsverzeichnis

Abbildungsverzeichnis

Code

1 Aufgabenstellung

```
1 # deciphers to "Schoene Crypto Welt" with IV=BBBBBBBBBBBBBBBB and key=
    BBBBBBBBBBBBBBBB aes128-cbc
2 cyphertext = "
    AAE365272C81078AB6116B361831D0F6A5D3C8587E946B530B7957543107F15E"
3 bc = binascii.unhexlify(cyphertext)
4 data = b'D' + bytes([len(bc)]) + binascii.unhexlify(cyphertext) + b'X'
```

Listing 1: Angabge

Der cyphertext soll entschlüsselt „Schöne Crypto Welt“ bedeuten. Um dies zu überprüfen kann https://www.openssl.org/ verwendet werden.

Schreiben Sie ein Programm das unter Verwendnung von openssl obige Aussage überprüft, verbessen Sie ihr Program in dem Sinne dass sie key/iv/plaintexte/ciphertexte als Argumente/Dateien/Usereingaben verarbeiten.

Hinweise

- Sie benötigen die openssl Biblotheksheader, unter Ubuntu 20.04 können Sie diese installieren via:

```
1 $ sudo apt install libssl-dev
```

Listing 2: Angabge

- em Linker muss mitgeteilt werden dass sie in Ihrem Programm Funktionen verwenden die in einer externen Bibliothek bereit liegen, verwenden sie dazu das flag -l (klein-L) und den Namen der Bibliothek OHNE das führende lib. openssl besteht aus mehreren Bibliotheken, die für AES notwendingen Funktionen befinden sich in libcrypto.

```
1 $ gcc my_code.c -lbibliothek -o my_executable
```

Listing 3: Angabge

Sie können sich die gelinkten Bibliotheken dann via ldd Kommando ansehen

```
1 $ ldd my_executable
```

Listing 4: Angabge

2 Theorie

2.1 OpenSSL

2.2 AES

3 Programm

```
1  // Author: FabioPlunser //
2  // Date: 17.2.2020 //
3  // GIT-Repo: https://github.com/FabioPlunser/FSST_Lezuo
4  // Specific Git-location: https://github.com/FabioPlunser/FSST_Lezuo/tree/
       main/Programme/openssl/openssl-Programm //
5  // Compiled with make, in WSL using Ubuntu 20.0.4, as you can see in my Repo
       //
6
7  // openssl //
8
9  //$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
10
11 //Basierend auf http://www.firmcodes.com/how-do-aes-128-bit-cbc-mode-
       encryption-c-programming-code-openssl/
12 //und
13 #include <stdio.h>
14 #include <string.h>
15 #include <stdlib.h>
16 #include <unistd.h>
17 #include <sys/types.h>
18
19 #include <openssl/aes.h>
20 #include <openssl/evp.h>
21 #include <openssl/err.h>
22
23
24 int do_decrypt(char *ciphertext, int ciphertext_len, char *key, char *iv,
       char* plaintext);
25
26 int main()
27 {
28
29     unsigned char key[16];
30     unsigned char iv[16];
31
32     memset(key, 'B', 16);
33     memset(iv, 'B', 16);
34
35     unsigned char ciphertext[128] = {
36     0xAA, 0xE3, 0x65, 0x27, 0x2C, 0x81, 0x07, 0x8A, 0xB6, 0x11, 0x6B, 0x36,
           0x18, 0x31, 0xD0, 0xF6,
37     0xA5, 0xD3, 0xC8, 0x58, 0x7E, 0x94, 0x6B, 0x53, 0x0B, 0x79, 0x57, 0x54,
           0x31, 0x07,0xF1, 0x5E
```

```
38      };
39
40      unsigned char* plaintext="Schoene Crypto Welt";
41      unsigned char decryptedtext[128];
42      int decryptedtext_len, ciphertext_len;
43
44      decryptedtext_len = do_decrypt(ciphertext, sizeof(ciphertext)/4, key, iv
            , decryptedtext);
45      decryptedtext[decryptedtext_len] = '\0';
46      printf("EVP:\nDecrypted test is: %s\n", decryptedtext);
47
48      AES_KEY dec_key;
49      AES_set_decrypt_key(key, sizeof(key)*8, &dec_key);
50          AES_cbc_encrypt(ciphertext, decryptedtext, sizeof(ciphertext)/4, &
               dec_key, iv, AES_DECRYPT);
51      printf("\nAES_KEY:\nDecrypted test is: %s\n", decryptedtext);
52
53  }
```

Listing 5: main.c

```
1   #include <stdio.h>
2   #include <string.h>
3   #include <stdlib.h>
4   #include <unistd.h>
5   #include <sys/types.h>
6
7   #include <openssl/aes.h>
8   #include <openssl/evp.h>
9   #include <openssl/err.h>
10
11
12  void Error_handling(void)
13  {
14      ERR_print_errors_fp(stderr);
15      abort();
16  }
17
18  int do_decrypt(char *ciphertext, int ciphertext_len, char *key, char *iv,
        char* plaintext)
19  {
20      EVP_CIPHER_CTX *ctx;
21      int len;
22      int plaintext_len;
23
24      if(!(ctx = EVP_CIPHER_CTX_new())) Error_handling();
25
26      EVP_CIPHER_CTX_set_padding(ctx, 0);
27
28      if(1 != EVP_DecryptInit_ex(ctx, EVP_aes_128_cbc(), NULL, key, iv))
            Error_handling();
29
```

```
30      if(1 != EVP_DecryptUpdate(ctx, plaintext, &len, ciphertext,
            ciphertext_len)) Error_handling();
31      plaintext_len = len;
32
33      if(1 != EVP_DecryptFinal_ex(ctx, plaintext+len, &len)) Error_handling();
34      plaintext_len += len;
35
36      ERR_print_errors_fp(stderr);
37      EVP_CIPHER_CTX_cleanup(ctx);
38      return plaintext_len;
39  }
```

Listing 6: EVP.c

3.0.1 Programm Output:

```
peppi@Peppi:/mnt/c/Users/fplun/GoogleDrive/Schule/2020_21/FSST/FSST_Lezuo/Programme/openssl/openssl-Programm$ ./main
EVP:
Decrypted test is: Schoene Crypto Welt

AES_KEY:
Decrypted test is: Schoene Crypto Welt
```

Abbildung 1: Programm-Output