UHBC_TP_REASEAU

List of Students

- Kellouche Dhiya - IL G02

List of contents

AES

AES-128

AES is a symmetric block cipher algorithm and it is a successor of DES. AES is a symmetric key encryption cipher. It is available in different key lengths. AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages, while AES-192 uses a 192-bit key length and AES-256 a 256-bit key length to encrypt and decrypt messages.

Lib Crypto.Cipher

The Crypto.Cipher package provides algorithms and components for performing encryption and decryption on streams. It includes both high level packages that perform a specific function and lower level functions that may be used by advanced programmers to construct their own encryption and decryption schemes.

Example

```
# aes.py
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes

# get user input
message = input("Enter your message: ")

# generate a random key
key = get_random_bytes(16)

# create cipher
cipher = AES.new(key, AES.MODE_EAX)

# encrypt message
ciphertext, tag = cipher.encrypt_and_digest(message.encode('ascii'))
print("Encrypted message: ", ciphertext)
```

```
# output
python3 aes.py

Enter your message: Hello World
Encrypted message:
b'\x8c\x8a\x1c\x1c\x8f\x8f\x1f\x1f\x8f\x8f\x1f\x1f\x8f\x8f\x1f\x1f'
```

AES in C++

Example

```
// aes.cpp
#include <iostream>
#include <string>
#include <cryptopp/aes.h>
#include <cryptopp/modes.h>
#include <cryptopp/filters.h>
#include <cryptopp/hex.h>
#include <cryptopp/osrng.h>

using namespace std;
using namespace CryptoPP;

int main(int argc, char* argv[]) {
    // get user input
    string message;
    cout << "Enter your message: ";
    getline(cin, message);

    // generate a random key
    AutoSeededRandomPool rnd;
    byte key[AES::DEFAULT_KEYLENGTH];
    rnd.GenerateBlock(key, sizeof(key));

    // create cipher
    byte iv[AES::BLOCKSIZE];
    rnd.GenerateBlock(iv, sizeof(iv));
    CFB_Mode<AES>::Encryption cfbEncryption(key, sizeof(key), iv);
    StreamTransformationFilter stfEncryptor(cfbEncryption, new
StringSink(message));
    stfEncryptor.Put(reinterpret_cast<const unsigned char*>
(message.c_str()), message.length() + 1);
    stfEncryptor.MessageEnd();

    // encrypt message
    string cipher = "";
    StringSource(message, true, new HexEncoder(new StringSink(cipher)));
    cout << "Encrypted message: " << cipher << endl;

    return 0;
}
```

```
# output
g++ aes.cpp -o aes -lcryptopp
./aes

Enter your message: Hello World
Encrypted message:
'\x8c\x8a\x1c\x1c\x8f\x8f\x1f\x1f\x8f\x8f\x1f\x1f\x8f\x8f\x1f\x1f'
```

$X \RightArrow Y + Z$