**컴퓨터 보안**

[AES,

MODE OF BLOCK CIPHER OPERATION]

학번 : 12141579

학과 : 컴퓨터정보공학과

이름 : 윤찬미

연락처 : 010-9287-1679

E-mail : [cmmcme0604@gmail.com](mailto:cmmcme0604@gmail.com)

1. **개요**
   1. **요구사항**

AES를 구현하여 5개의 MODE OF BLOCK CIPHER OPERATION 의 Encryption, Decryption을 구현한다.

* 1. **개발 환경**
     1. Operating System : Window 8.1
     2. IDE : Visual Studio 2015 community
     3. Processor : 2.5 GHz intel Core i7
     4. Memory : 16GB 1600 MHz DDR3
     5. 언어 : C++11

1. **프로그램 설명**
   1. **멤버 변수**

1) word w[44]

- 각 라운드에서 사용하는 키의 값이 들어가있다. KeyExpansion을 이용해 확장한 11번의 AddRoundKey에서 사용하는 4\*11 개의 Key가 있다

2) word Rcon[11]

- InverseS-Box를 만들어 놓은 배열이다

3) const byte MC\_MATRIX[4][4]

- MixColumn matrix

3) const byte IMC\_MATRIX[4][4]

- Inverse MixColumn matrix

3) int Sbox[256]

- S-Box를 만들어놓은 배열이다.

4) int ISbox[256]

- InverseS-Box를 만들어놓은 배열이다.

5) const byte COLUMN\_VECTOR

- S-BOX를 만들 때 XOR를 하는데 사용하는 값

6) const byte SUBTI\_MATRIX

- S-BOX를 만들 때 사용하는 행렬의 첫 번째 행의 값

* 1. **멤버 함수**

1. AES(vector<byte> &key)

생성자로 KEY의 값을 파라미터로 사용한다. S-Box와 Inverse S-Box 를 초기화 하고 그것 들을 만드는 함수(GetSBOX()) 와 키를 확장시키는 함수(KeyExpansion())을 호출한다

1. void SubstituteBytes(vector<vector<byte>>&)

Subtitute Byte로 각 Byte를 S-Box를 이용해 치환한다.

1. void ShiftRow(vector<vector<byte>>&)

각 행을 Shift 시키는 함수이다. 0번째 행은 그대로, 1번째 행은 왼쪽으로 1칸, 2 번째 행은 왼쪽으로 2칸, 3번째 행은 왼쪽으로 3칸 Shift 한다.

1. void MixColumn(vector<vector<byte>>&)

Multiple 함수를 이용하여 MC\_MATRIX와 행렬 곱을 구한다.

1. void AddRoundKey(vector<vector<byte>>&, byte)

각 Round에 해당하는 Key와 BLOCK를 XOR 한다.

1. void InverseSubstituteBytes(vector<vector<byte>>&)

InverseSubtituteByte로 각 Byte를 IS-Box를 이용해 치환한다.

1. void InverseShiftRow(vector<vector<byte>>&)

각 행을 InverseShiftRow 하는 함수이다. 0번째 행은 그대로, 1번째 행은 오른쪽으로 1칸, 2 번째 행은 오른쪽으로 2칸, 3번째 행은 오른쪽으로 3칸 Shift 한다.

1. void InverseMixColumn(vector<vector<byte>>&)

Multiple 함수를 이용하여 IMC\_MATRIX와 행렬 곱을 구한다.

1. word rotWord(word)

KeyExpansion에서 쓰이는 Rotation 함수이다.

1. word subWord(word)

KeyExpansion에서 쓰이는 S-Box를 이용해 치환하는 함수이다. 4바이트의 word를 4개의 byte로 나누어 치환한 후return 한다.

1. void KeyExpansion(const vector<byte>&)

Key를 확장시키는 함수이다. 책 177페이지의 의사코드를 함수로 나타내었다.

1. byte Multiple(byte, byte)

Galois Field 상에서의 곱셈을 하는 함수이다. Shift를 하며 연산을 진행하며 Shift중 bit OverFlow가 발생하면 normal과 XOR연산을 한다. 총 8번의 연산(8비트)을 한 값을 모두 XOR시킨 결과값을 return 한다.

1. int getMSB(int)

Galois Field 상에서의 나누기를 하기 위해서는 최고차항의 계수를 알아야한다. 방정식을 비트열로 나타내었는데 가장 큰 비트 MSB가 최고차항의 계수가 된다. 즉 이 함수는 최고차항의 계수를 구하는데 사용된다.

1. int Mod(int, int, int&)

S-BOX를만들기 위해서는 Galois Field 상에서의 역원을 구해야한다. 역원은 확장 유클리드로 구할 수 있고 확장 유클리드를 사용하기 위해서는 몫과 나머지를 알아야한다. Mod함수는 나머지를 리턴하며 함수의 세번째 인자가 몫의 결과를 갖게된다.

1. Solution ExtendedEuclid(int, int)

Galois Field상에서 ax + by = 1을 구하는 확장 유클리드 함수이다. Solution 구조체의 x와 y 에는 결과값이 저장된다.

1. byte getInverseElement(byte)

역원을 구하기 위하여 확장 유클리드 함수에 어떤 수와 를 넣으면 어떤 수에 대한 역원을 구할 수 있다. 이렇게 구한 역원을 return 한다.

1. void getSBOX()

확장유클리드를 통해 역원을 구한 후, 역원과 SUBTI\_MATRIX와 행렬 연산을 하고 COLUMN\_VECTOR와 XOR을 하여 SBox를 구한다. 이 결과를 반대로 저장하면 ISBox를 구성할 수 있다.

1. void Encrypt(vector<byte>&, int)

암호화를 시키는 함수로 Encrypt에 사용되는 함수를 순서대로 호출한다.

1. void Decrypt(vector<byte>&, int)

복호화를 시키는 함수로 Decrypt에 사용되는 함수를 순서대로 호출한다.

1. void inputFile(FILE \*)

파일 입출력을 도와주는 함수이다. 파일을 1바이트씩 읽어서 vector에 넣어줌

1. void increase(vector<byte>&)

CTR모드에서 Counter를 1씩 증가시켜주는 함수이다.

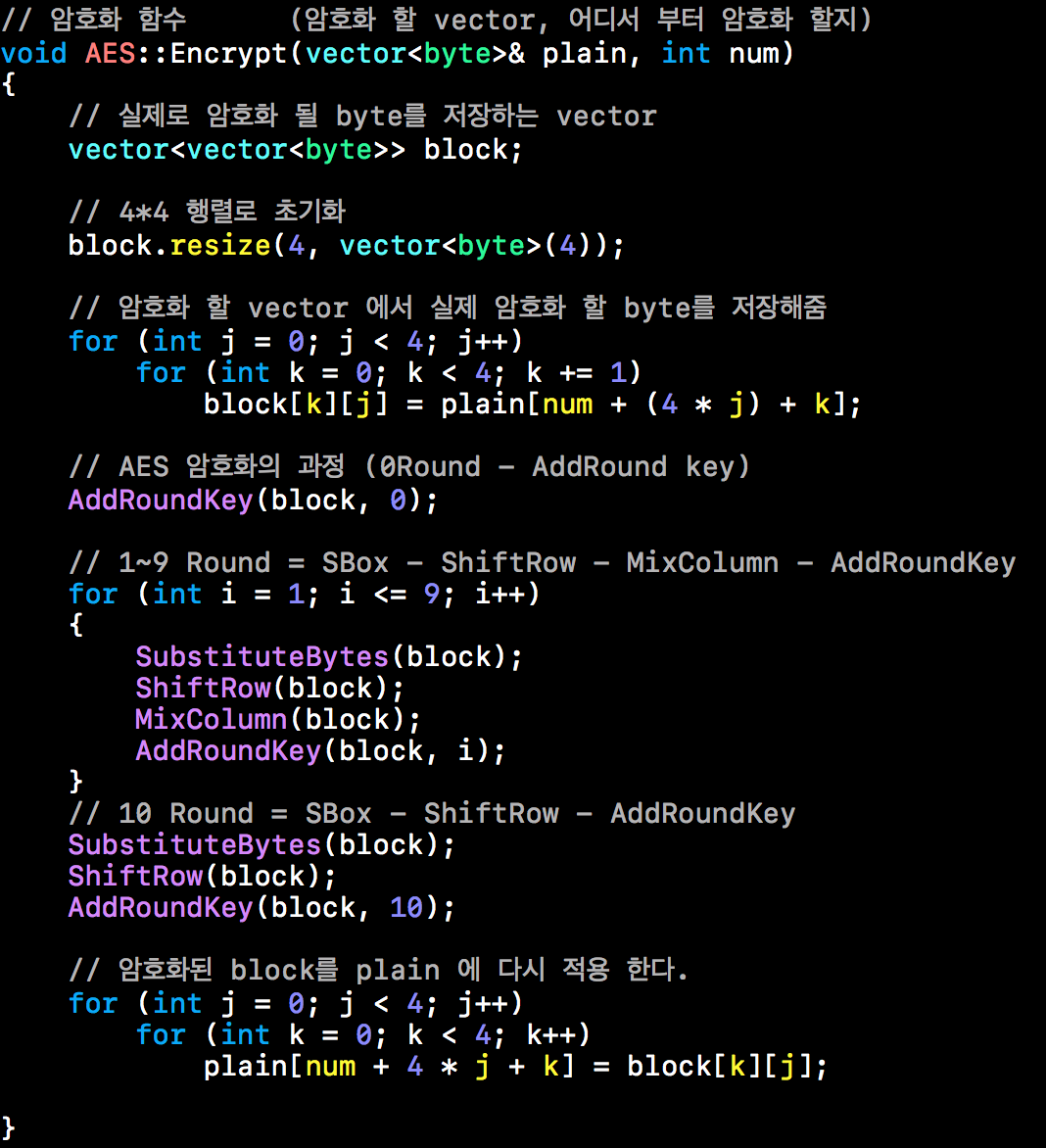
* 1. **main 함수**

파일 입출력을 이용하여 byte단위로 key.bin을 읽어, 상위 16byte는 Key, 하위 16byte를 initial vector로 사용한다. e가 입력되었을때, 5개의 mode로 Encrypt 하여 각 모드에 해당하는 결과를 MOD\_c.bin에 출력한다.

d가 입력되었을 때, 5개의 mode로 Decrypt를 한다. 각 모드에 해당하는 결과를 MOD\_p.bin에 출력한다.

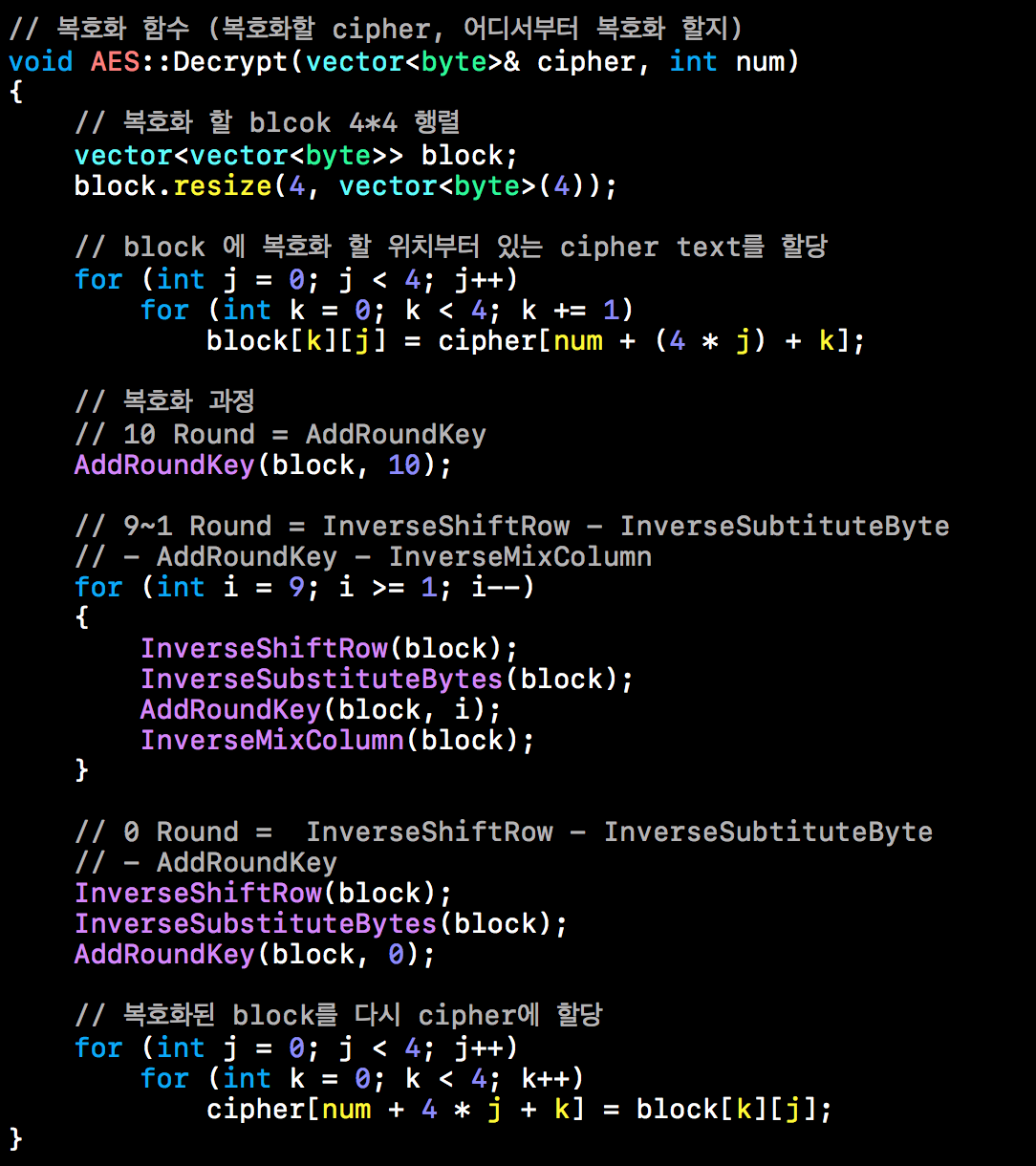
1. **소스코드**

Encrypt (암호화할 vector, 몇번째 block 인지)



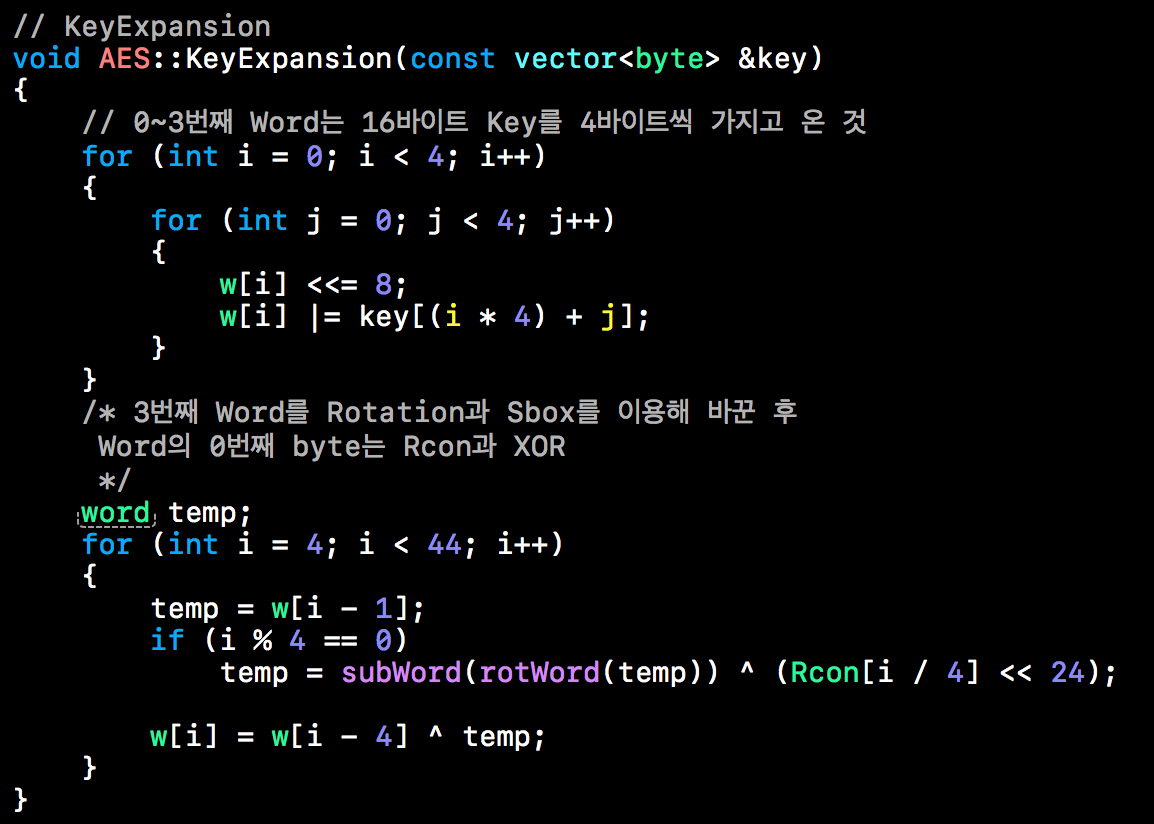
* block vector에 암호화할 vector의 block위치를 찾아 16byte 읽어온다.
* 0 Round : AddRoundKey
* 1~9 Round : SubstituteBytes – ShiftRow – MixColumn – AddRoundKey
* 10 Round : SubstituteByte – ShiftRow – AddRoundKey
* 암호화된 block를 다시 파라미터로 넘겨받은 vector에 적용한다.

Decrypt(복호화할 vector, 몇번째 block 인지)



* block vector에 복호화할 vector의 block위치를 찾아 16byte 읽어온다.
* 10 Round : AddRoundKey
* 9~1 Round : InverseShiftRow - InverseSubstituteBytes – InverseMixColumn – AddRoundKey
* 0 Round : InverseShiftRow – InverseSubstituteByte - AddRoundKey
* 복호화 할 block를 다시 파라미터로 넘겨받은 vector에 적용한다.

KeyExpansion(확장할 key)



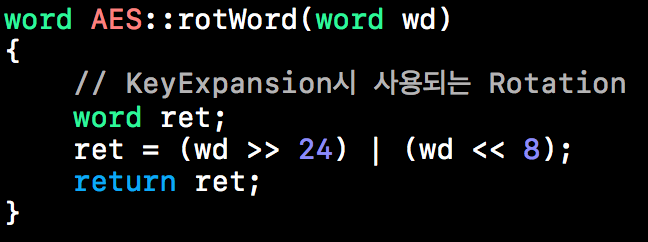
- P.177의 의사코드를 그대로 나타낸 함수이다

16byte의 key를 4byte씩 word[0~3]에 할당해준다.

- i%4 ==0 일때 i-1번째의 word를 Rotation과 S-Box를 이용하여 word를 4바이트로 나눠 치환, 전치 암호화를 이용해 암호화해준다

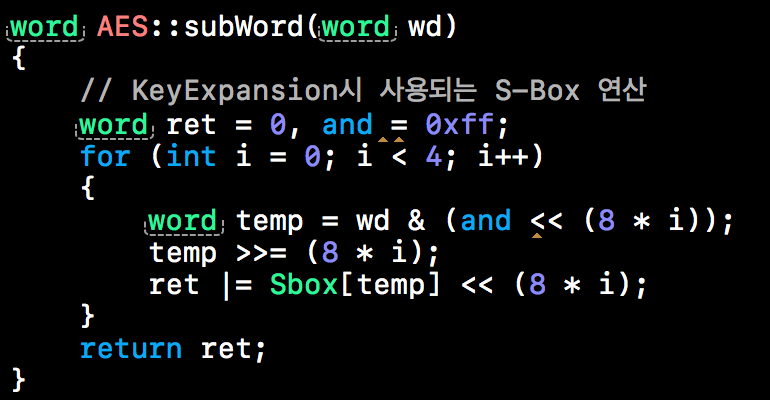
- w[i] = w[i-4] ^ 암호화한w[i-1]

rotWord



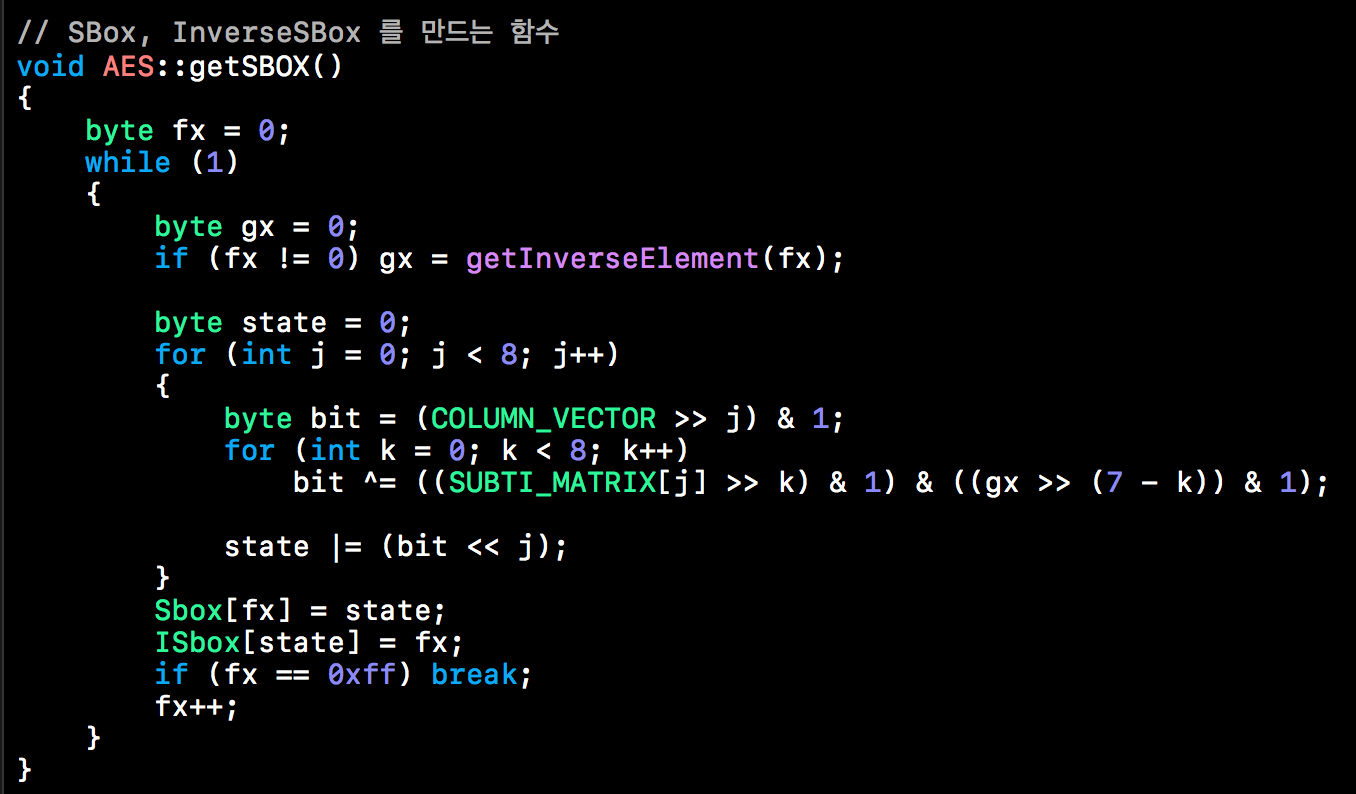
* KeyExpansion시 사용되는 Rotation이다.
* Rotate하기 위해 왼쪽으로 8번 쉬프트한 결과(b1, b2, b3)와 오른쪽으로 24번 쉬프트한 결과(b0)를 OR 하게 되면 (b1, b2, b3, b0)변환된다

subWord



* KeyExpansion에서 쓰이는 word를 4개의 byte로 나누며 S-Box를 이용해 치환하는 함수이다. word를 치환한 후 return 해준다

getSBox

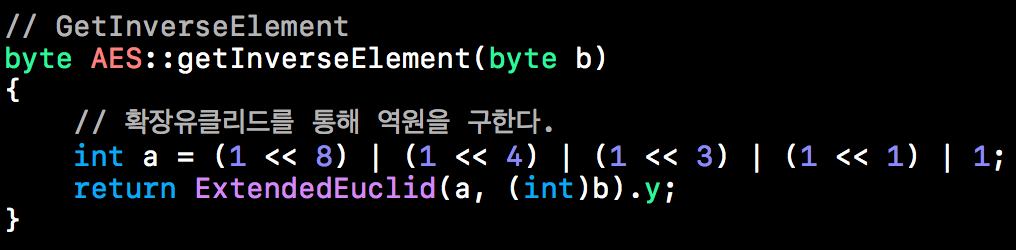


- 확장유클리드를 통해 역원을 구한다.

- 역원과 SUBTI\_MATRIX와 행렬 연산을 하고 COLUMN\_VECTOR와 XOR을 하여 S-Box를 구한다.

- 이 결과를 반대로 저장하면 ISBox를 구성할 수 있다.

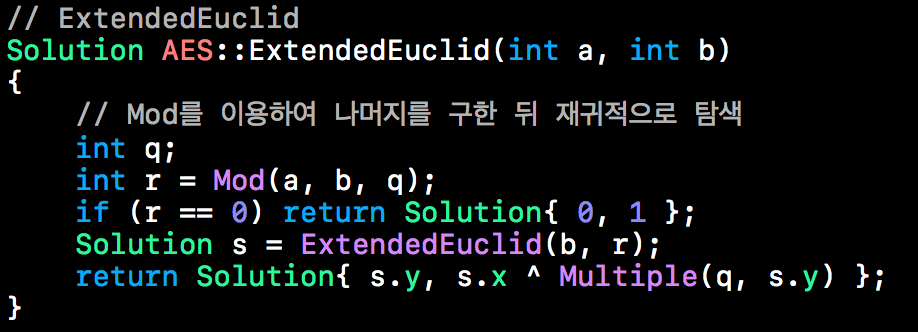
getInverseElement



- 역원을 구하기 위하여 확장 유클리드 함수에 역원을 구할 수와 를 넣으면 그 수에 대한 역원을 구할 수 있다.

- 이렇게 구한 역원을 return 한다.

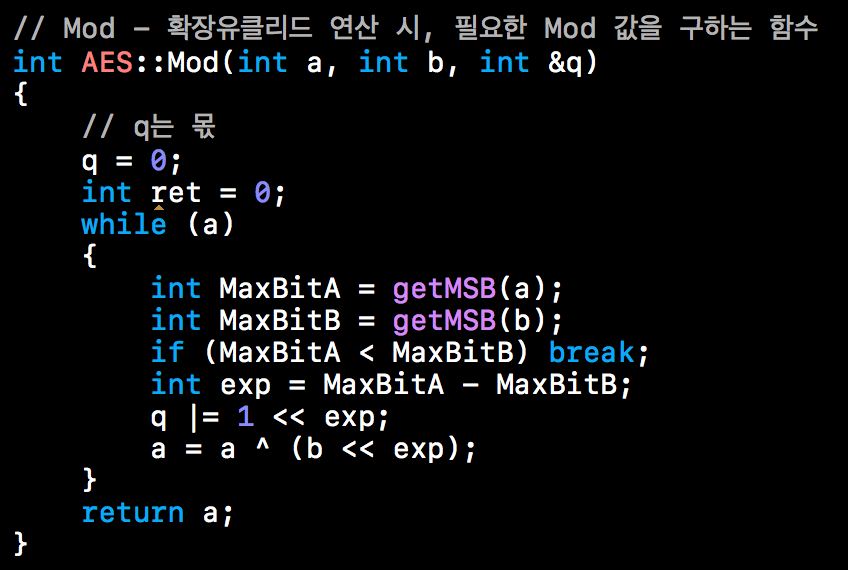
ExtendedEuclid



- Galois Field상에서 ax + by = 1을 구하는 확장 유클리드 함수이다.

- Solution 구조체의 x와 y 에는 결과값이 저장된다.

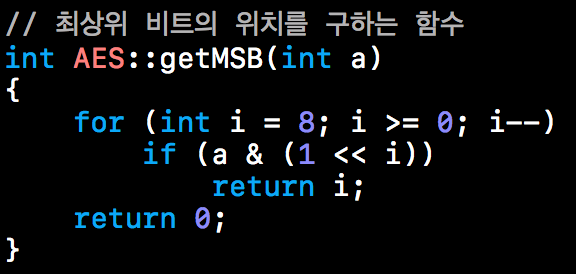
Mod



- S-Box 연산시 역원을 구하기 위해 Galois Field상에서 몫과 나머지를 구하는 함수이다.

- 나머지를 return 하고, 함수의 세번째 인자가 몫의 결과를 가진다.

getMSB

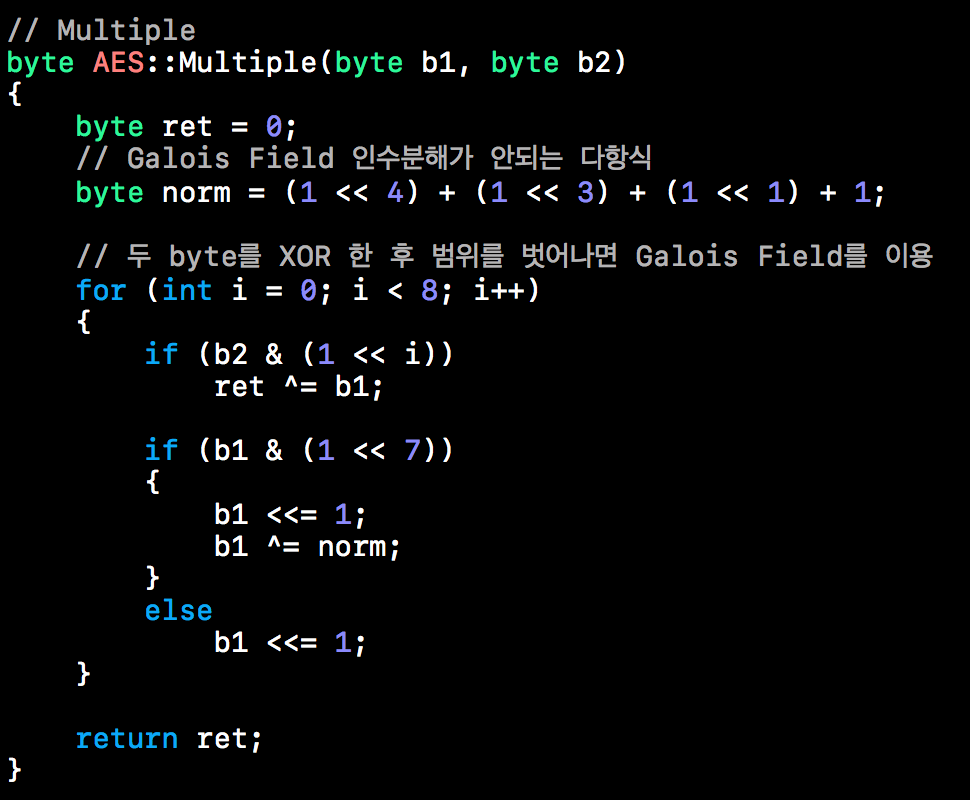


- 최상위 비트의 위치를 구하는 함수이다.

- Galois Field 상에서의 나누기를 하기 위해서는 최고차항의 계수를 알아야한다

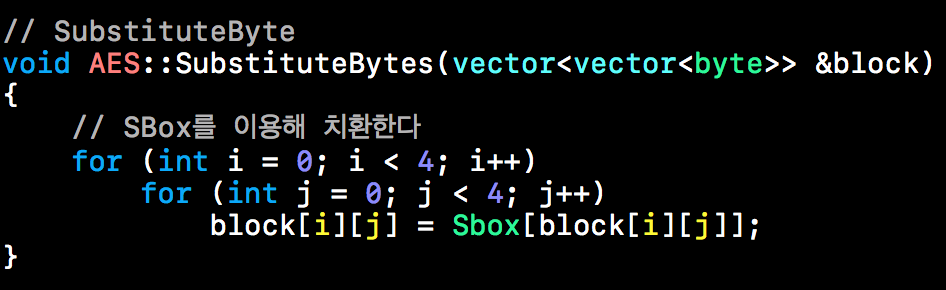
- 방정식을 비트열로 나타내었는데 가장 큰 비트 MSB가 최고차항의 계수가 된다.

Multiple

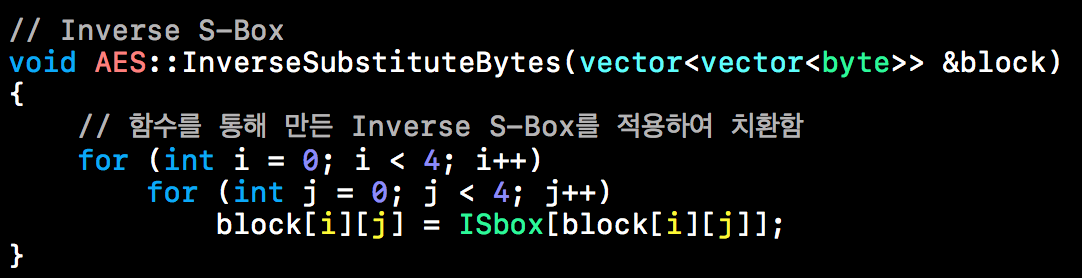


* Galois Field 상에서의 곱셈을 하는 함수이다.
* Shift하며 연산을 한다. bit OverFlow가 발생하면 normal과 XOR연산을 한다. 연산을 한 값을 모두 XOR시킨 결과값을 return 한다.

SubstituteBytes

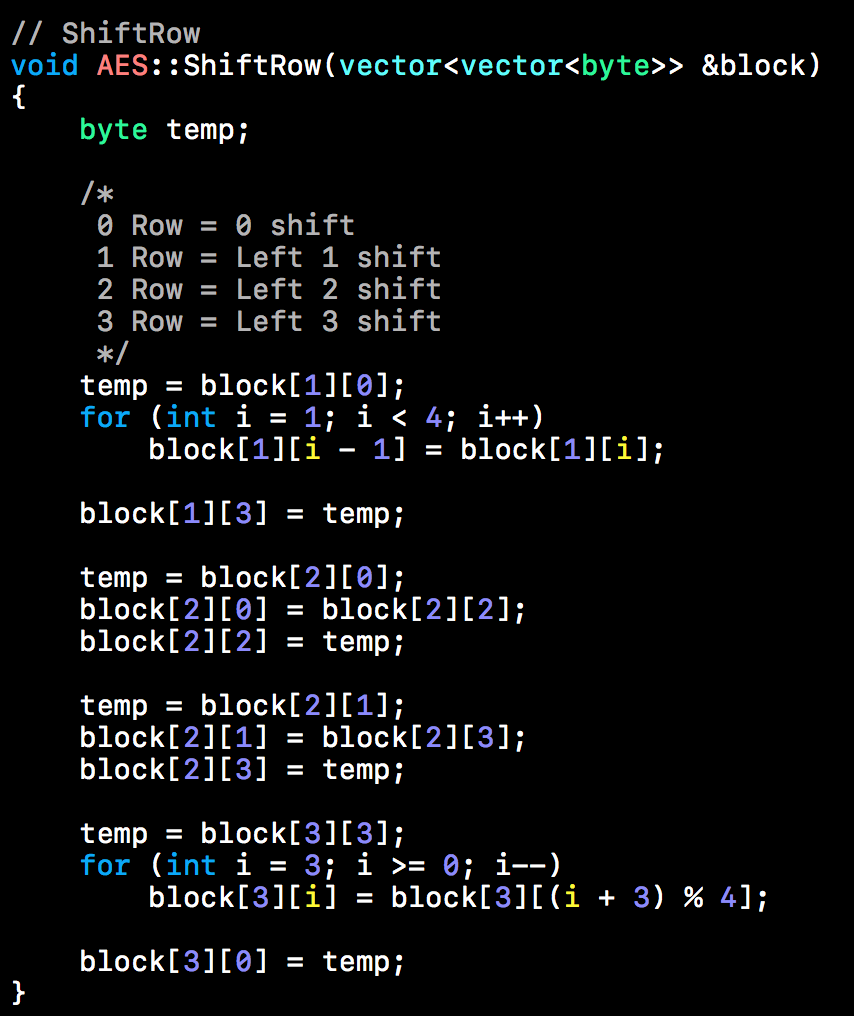


InverseSubstituteBytes

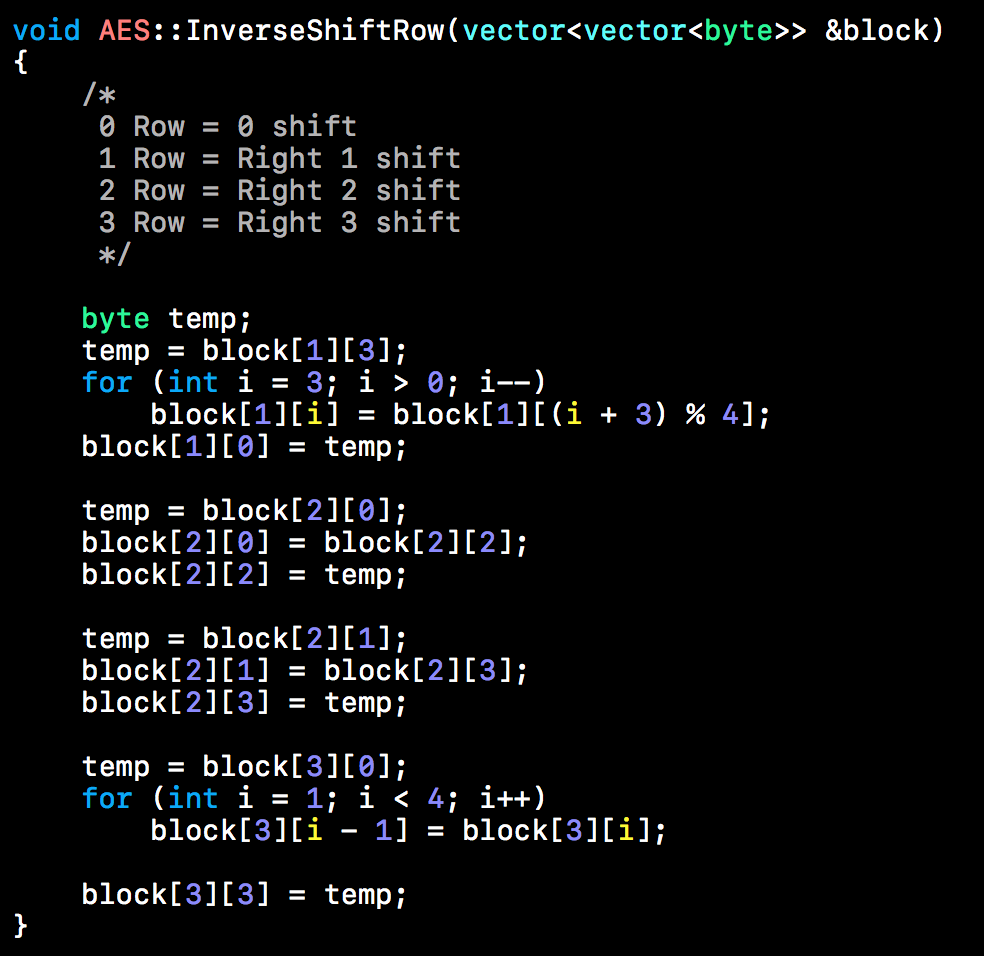


- getSBox() 함수를 이용하여 만든 S-Box와 ISBox를 이용하여 block를 치환해준다.

ShiftRow



InverseShiftRow



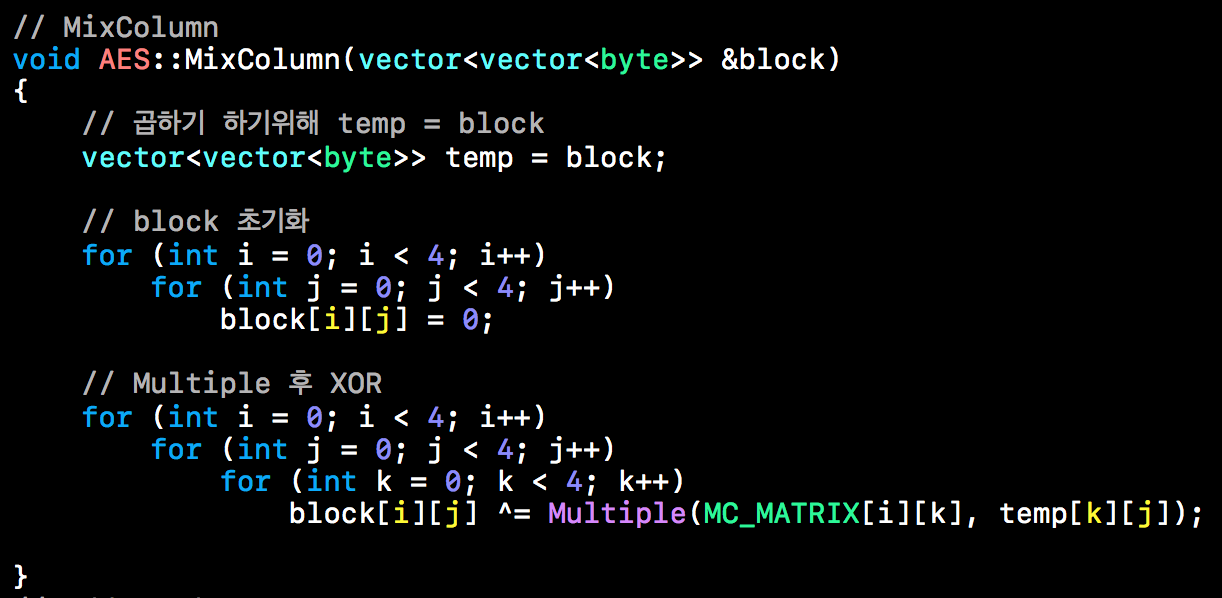
- 각 행을 ShiftRow 시키는 함수이다.

0번째 행은 그대로, 1번째 행은 왼쪽으로 1칸, 2 번째 행은 왼쪽으로 2칸, 3번째 행은 왼쪽으로 3칸 Shift 한다.

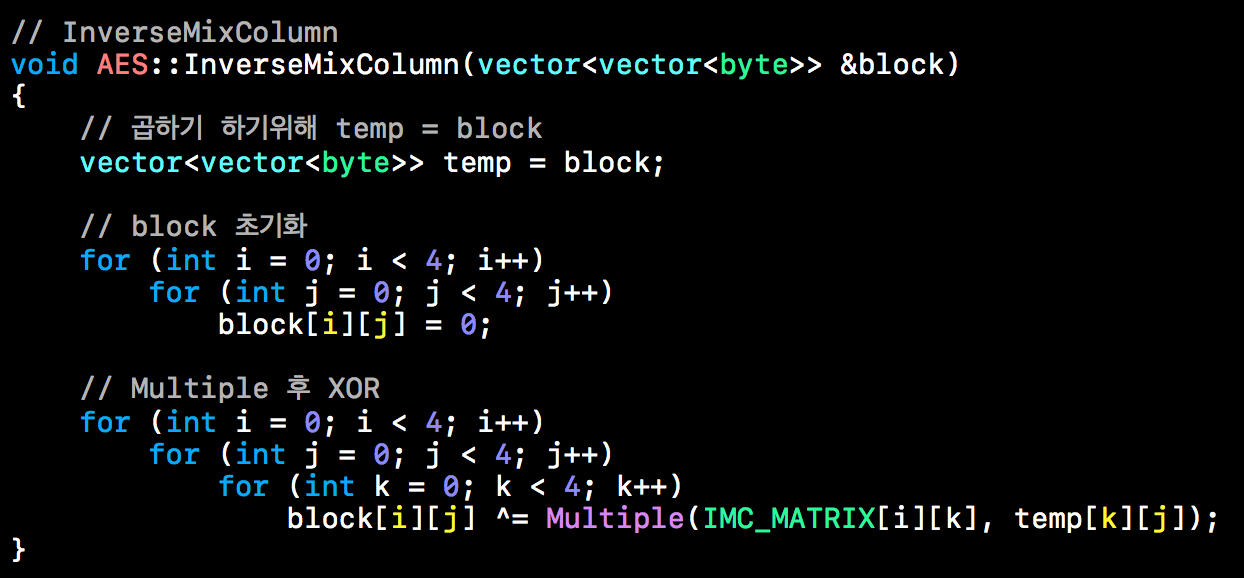
- 각 행을 InverseShiftRow 하는 함수이다.

0번째 행은 그대로, 1번째 행은 오른쪽으로 1칸, 2 번째 행은 오른쪽으로 2칸, 3번째 행은 오른쪽으로 3칸 Shift 한다.

MixColumn



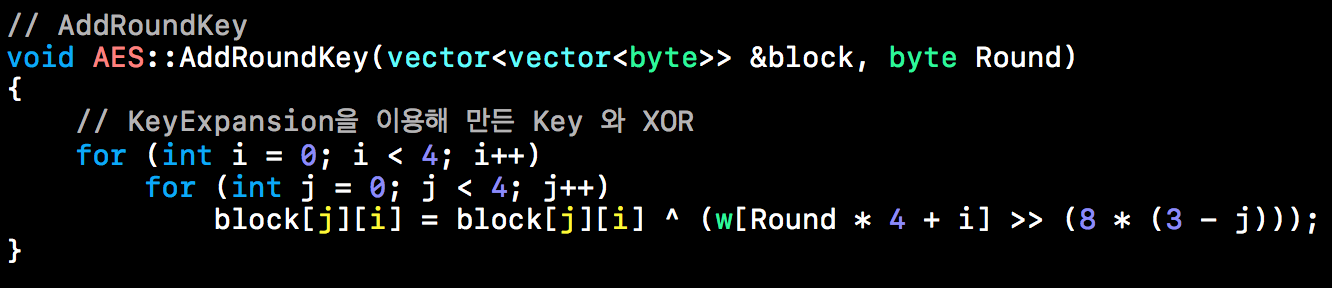
InverseMixColumn



- (Inverse)MixColumn을 하기위해 temp에 block 을 저장해 두고 block를 초기화한다.

- Multiple 함수를 이용하여 temp와 (I)MC\_MATRIX와 저장해둔 block를 곱한 후, XOR 해준다.

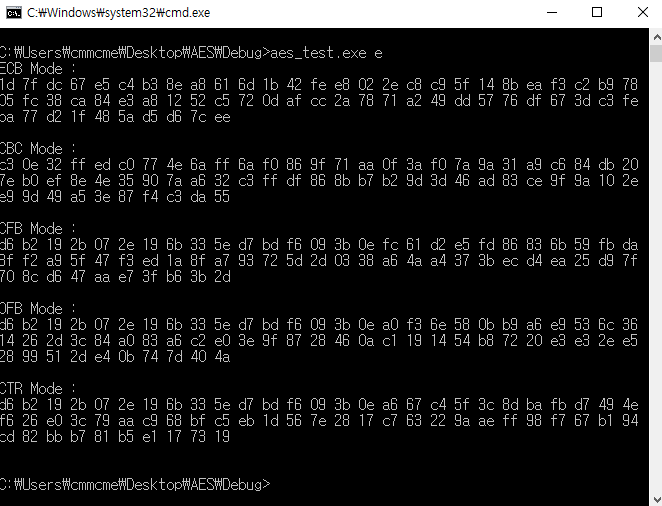
AddRoundKey



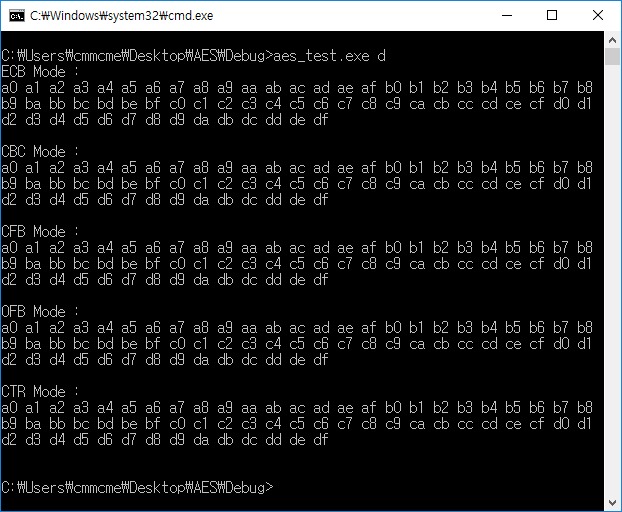
- KeyExpansion을 이용하여 만든 Key를 각 Round에 맞는 Key와 block을 한 byte씩 XOR 한다.

1. **실행 화면**

**Encrypt**

****

**Decrypt**

****