<AES 구현 과제>

12161595 배재경

128 bit의 AES 암호 알고리즘을 구현하는 프로그램을 만들었다. 구현 환경은 Visual Studio 2017, 구현 언어는 C++이다.

우선 평문과 key를 각각 plain.bin과 key.bin 파일에서 읽어들여오고 배열에 저장한다. 먼저 KeyExpansion 함수를 통해 키를 확장한다. 각 라운드의 키를 추가하기 위한 키 확장 함수로, 먼저 초기 라운드 키 배열에 키 값을 저장한다. 그리고 Word 단위로 순환 이동을 수행하여 B0, B1, B2, B3의 순서를 B1, B2, B3, B0으로 만든다. 그 다음 4 바이트를 Sbox를 통해서 치환한다. temp[0]은 라운드 상수 Rcon과 XOR 한다. 라운드키에 temp 배열과 라운드 키를 각각 XOR 연산하여 저장한다.

RCon 함수를 사용하여 라운드 상수를 생성하는데, 2^0부터 2^9까지 각각 irreducible polynomials(기약다항식)와의 modular 값을 구한다. 128bit의 AES이므로 크기가 128을 넘기면 기약다항식 값에서 Rcon값을 뺀다. 기약다항식은 변경된 값이고, 2를 대입해보면 0x0169가 나온다.

Sbox 함수는 sbox를 생성하는 함수로, sbox는 암호화에 사용되는 0부터 255까지의 8 bit 값의 순열을 포함하고 있는 16*16 바이트 배열이다. 먼저 0부터 255까지의 각각의 값의 GF(2^8)에서의 곱셈에 대한 역원을 구하는데, 이때 갈루아 필드를 구현해야 한다. 두 개의 배열 GF와 InvGF을 사용하였다. 우선 a의 값을 InvGF 배열에 저장하고, a와 0x80을 AND 연산하여 w에 저장한다. w가 128이면 0x69와 XOR 연산하여 a에 값을 저장한다. 0x69는 기약다항식의 값에서 0xFF와 AND 연산하면 구할 수 있다. GF 배열의 인덱스는 구한 InvGF의 인덱스가 e인 값이고, 배열에 e를 저장한다.

역원은 InvGF에서 바이트의 전체 크기-1(인덱스가 0부터 시작하므로)에서 GF를 뺀 값이 된다. 역원을 구할 때 기약다항식이 변경되었으므로 역원 값도 달라진다. 이로서 1부터 255까지의 역원을 구할 수 있다. 구한 역원은 2진수로 b7부터 b0까지로 표현하여 고정된 다항식의 행렬과 곱한다. b0부터 행렬곱한 값마다 XOR 연산을 한다. 2진수로 나뉘어 b7부터 저장된 0x15의 b0와 XOR하면 b0의 변환이 완료된다. b0부터 b7까지 순서대로 구한다. 2진수로 표현되어 있으므로 다시 16진수로 만들어 sbox에 저장한다. 이때 0은 예외적으로 역원이 존재하지 않으므로, 0x15로 값을 지정해준다. Sbox 함수를 호출할 때마다 그 위치의 sbox 배열 값을 리턴한다.

InvSbox 함수는 inverse sbox를 생성하는 함수로, inverse sbox는 복호화에 사용되는 역 sbox이다. Sbox를 만들 때의 순서를 반대로 하면 된다. 먼저 2진수로 만들어 저장하고, sbox의 고정된 행렬의 역행렬과 행렬곱하여 XOR하고 0xC7과 XOR 한 뒤 16진수로 만들어 역원을 구하고 invsbox 배열에 저장한다. InvSbox 함수를 호출할 때마다 그 위치의 invsbox 배열 값을 리턴한다.

Cipher 함수에서 암호화 과정을 수행한다. 암호화 과정 동안의 상태를 저장하는 2차원 state 배

열을 생성하여 열 단위로 plain 배열의 평문을 저장한다. 그리고 AddRoundKey 함수로 초기 라운드 키를 추가한다. AddRoundKey에서는 상태 배열과 라운드 키를 bit 단위의 XOR 연산을 수행하여 라운드 키 값을 추가한다. 그 다음 1라운드부터 9라운드까지 SubBytes 함수-ShiftRows 함수-MixColumns 함수-AddRoundkey 함수 과정을 반복한다. SubBytes 함수는 상태 배열의 각 바이트 값을 Sbox를 통해서 치환하는 함수이다. ShiftRows 함수는 bit 단위로 순환시키는 함수이다. 첫 번째 행, 두 번째 행, 세 번째 행을 각각 기준으로 하여 왼쪽으로 순환시킨다. MixColumns 함수는 4 bit 단위로 열을 혼합시키는 함수이다. 상태 배열의 각 열과 고정된 다항식 행렬을 곱하고, xtime을 통해 열 혼합이 끝난 tm과 0xff를 AND 연산하여 다시 상태배열의 같은 위치에 저장한다. 이후 AddRoundKey 함수를 수행하고 다음 라운드로 넘어간다. 마지막 10라운드에서는 SubBytes-ShiftRows-AddRoundKey(마지막 키)를 수행하고 최종 암호화 상태 배열을 cipher 배열에 저장하여 암호문이 완성된다.

InvCipher 함수에서는 복호화 과정이 수행된다. Cipher 배열의 암호문을 상태 배열에 열 단위로 저장하고, 마지막 라운드 키를 AddRoundKey로 초기 라운드 키로 추가한다. 그리고 InvShiftRows-InvSubBytes-AddRoundKey-InvMixColumns 과정을 1라운드부터 9라운드까지 반복한다.

InvSubBytes 함수는 상태 배열의 각 바이트를 Inverse Sbox로 치환한다. Inverse Sbox는 Sbox의 값을 특정 행렬의 역행렬과 곱하여 XOR하고 다시 곱셈에 대한 역원을 구하여 만든다. InvSubBytes는 첫 번째, 두 번째, 세 번째 행을 각각 기준으로 오른쪽으로 순환한다. InvMixColumns는 Multiply로 상태 배열의 열을 GF(2^8) 상에서 고정된 다항식과 곱셈을 연산하고 XOR 연산을 수행한다. 마지막 10 라운드에서는 InvShiftRows-InvSubBytes-AddRoundKey(첫번째 키)를 수행하고 최종 복호화 상태 배열을 decrypted 배열에 저장하여 복호문을 완성시킨다.

생성된 암호문과 복호문을 각각 이진 파일 cipher.bin과 decrypted.bin에 작성하여 저장하고 프로그램이 종료된다.