AES 암호 알고리즘 교육용 학습 프로그램 개발

이동범† · 정명수†† · 곽 진†††

요 약

최근 급변하고 있는 시대에 맞춰 IT 관련 분야에서는 정보보호에 대한 중요성이 강조되고 있다. 정보보호 관련 기관에서는 관련 정책을 통해 개인정보보호 및 보안의 중요성을 강조하고 있지만, 아직까지 일부 업체나 개인 사용자들은 보안 의식이 매우 저조한 수준이다. 이에 본 논문에서는 사용자들의 보안 의식을 제고시키기 위하여 대칭 블록 암호인 AES를 이용한 학습 프로그램을 개발하여 사용자들이 암호 알고리즘에 대해 쉽게 이해할 수 있도록 한다. 즉, AES 암호 알고리즘 교육용 학습 프로그램을 통해 암호화 및 복호화 되는 과정을 직접 확인할 수 있어 AES 암호에 대한 관심을 유발하고 정보보호에 대한 의식을 제고시키고자 한다.

주제어 : AES, 암호, 학습 프로그램, 정보보안

Development of Education Learning Program for AES Cryptography Algorithm

Dongbum Lee† · Myeongsoo Jeong†† · Jin Kwak†††

ABSTRACT

Recently, the importance of information security is emphasized in IT related field. The agency related to information security implements the policies to emphasize the security and protection of the privacy. However, the issue in many companies and users is that awareness of security is still poor. Therefore, in this paper, we develope the learning program for AES(advanced encryption standard) block cipher, to raise the awareness of security. Also, wish to cause interest about AES cipher because user confirms process that is encryption/decryption through program of this paper directly and prove awareness about information security.

Keywords : AES, Cryptography, Learning Program, Information Security

† 준 회 원: 순천향대학교 정보보호학과 박사과정
†† 준 회 원: 순천향대학교 정보보호학과 학사과정
††† 종신회원: 순천향대학교 정보보호학과 교수(교신저자)
논문접수: 2011년 06월 23일, 심사완료: 2011년 07월 19일

1. 서 론

최근 급속도로 진화하는 IT 기술로 인해 사회는 다양한 정보가 오고가는 시대로 변화하고 있다. 이와 동시에 변화하는 시대에 맞춰 IT 관련 분야에서는 정보보호에 대한 중요성이 강조되고 있다. 하지만 아직까지 기업의 고객정보 유출 사례나 개인의 부주의로 인한 정보 유출 등 많은 정보보호 관련 사고가 발생하고 있으며, 이러한 사고로 인한 피해는 단순한 유출 사고로 그치는 것이 아니라 금전적 피해나 정보 도용과 같은 사고로 확대될 가능성이 크다[1].

이에 정보보호 관련 기관에서는 사용자와 인터넷 관련 서비스 제공자에게 정보보호 관련 정책을 통해 개인정보의 중요성을 강조하고 있지만 아직까지 일부 기업이나 다수의 개인 사용자들은 보안의식이 부족한 상태이다. 특히 인터넷 서비스에서 가장 많이 사용되는 패스워드에 대해서도 개인 사용자의 경우에는 많은 기관 및 기업에서 개인정보보호에 대한 피해예방 방법 등을 제정하여 권고하고, 수시로 패스워드 갱신을 통해 피해예방을 유도하여 사용자의 의식을 바꾸려고 노력하고 있지만 실정은 그렇지 못한 상태이다. 또한 정보보호 관련 교육자료 등이 부족한 상황으로 정보보호 교육에 많은 어려움이 따르고 있다[2].

따라서 본 논문에서는 정보보호에 대한 인식을 제고시키기 위해 정보보호의 기초인 암호학 중에서 사용자들이 가장 많이 사용하는 AES 암호 알고리즘에 대한 학습 프로그램을 개발하고자 한다. 블록 암호의 대표 방식인 AES 암호 알고리즘을 활용하여 정보보호 관련 분야의 공부를 하지 않더라도 본 논문의 프로그램을 통해 암호가 만들어지는 과정을 쉽게 파악하고, 실제로 입력한 문장을 암호화 및 복호화를 할 수 있도록 한다. 그리고 여러 사용자에게 평소 알지 못했던 암호 분야에 대한 관심을 유발할 수 있고, 암호학 학습을 하는 학습자들에게는 보다 쉽고 흥미롭게 학습하고, 기존의 책으로만 배우던 암호학을 더욱 효율적으로 학습할 수 있는 학습 프로그램을 개발하고자 한다.

이에 본 논문에서는 암호 알고리즘을 쉽게 배울 수 있는 AES 암호 알고리즘 학습 소프트웨어를 개발하여 사용자들에게 AES의 암호 원리를 이해시키고 쉽게 다가갈 수 있도록 한다. 또한 본 논문의 프로그램을 통한 암호학 교육을 통해 사용자들의 보안 의식을 증진시킬 수 있도록 한다.

2. 관련 연구

2.1 AES 알고리즘

AES 암호 알고리즘은 표준 암호 알고리즘으로 사용해오던 DES의 블록 암호 알고리즘에 대한 다양한 공격방법들이 공개되면서 NIST에서 차세대 표준 암호 공모를 통해 선정된 알고리즘이다.

AES 암호 알고리즘은 키와 평문의 길이를 128, 192, 256bit 중에 선택적으로 사용이 가능하며 정식 등록된 알고리즘은 128bit의 평문으로 평문 길이에 따른 AES의 알고리즘 종류는 아래의 <표 1>과 같다.

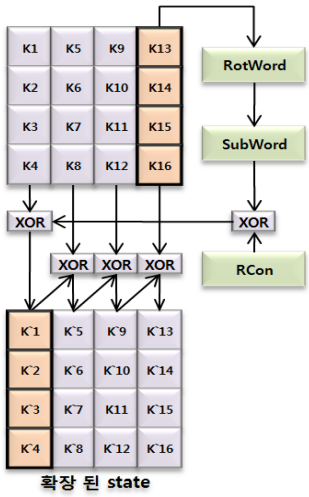
<표 1> 평문 길이에 따른 AES의 종류

구분	키 길이 (Nk)	평문 크기 (Nb)	라운드 수 (Nr)
AES 128bit	4	4	10
AES 192bit	6	4	12
AES 256bit	8	4	14

평문 블록 단위는 4x4 행렬인 state로 구성되었으며 블록으로 구성된 state를 행 단위나 열 단위, byte 단위로 AddRoundKey, SubByte, ShiftRow, MixColumn의 네 개의 연산을 라운드마다 수행하여 암호화를 수행한다[3][4].

2.1.1 KeyExpansion

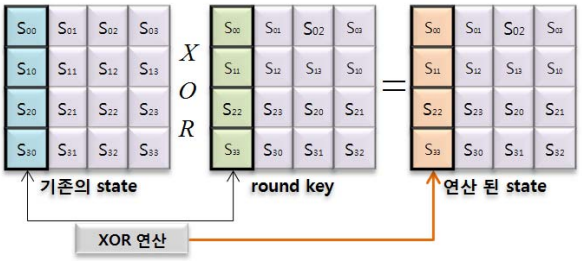
KeyExpansion 연산은 AES의 전체 알고리즘에서 수행해야 하는 라운드 키를 생성하기 위해 입력받은 키를 확장하는 과정이다. 이 과정에서는 키 값을 확장시키기 위해서 워드를 순환 이동시키는 과정인 RotWord, 순환 이동시킨 워드를 S-Box를 통해 치환하는 SubWord, 지정된 행렬 값과 XOR 연산을 수행하는 RCon 연산으로 세 개의 과정을 통해 키를 확장시킨다. 확장되는 키는 $Nb \times Nr + 1$ 만큼 확장하게 된다. 전체적인 수행과정은 아래의 그림과 같다[5][6].



<그림 1> 키 확장 과정

2.1.2 AddRoundKey

AddRoundKey 연산은 각 라운드에서 암호화 과정을 거치는 state와 각 라운드 키를 XOR 연산을 하는 과정이다. AddRoundKey에 사용되는 라운드 키는 KeyExpansion 연산을 통해 확장된 키 중에서 해당 라운드의 키를 사용한다. 평문 state와 해당 라운드 키 state의 동일한 위치의 워드를 각각 XOR 연산하는 과정으로 아래 그림과 같이 수행한다.



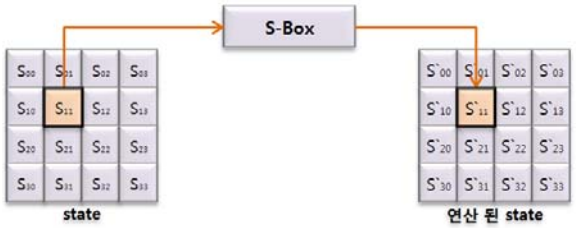
<그림 2> AddRoundKey 연산 과정

AddRoundKey는 전체 알고리즘 과정에서 $Nr+1$번 수행하게 된다. 이는 평문이 본격적인 라운드에 들어가기 전에 기존에 입력받은 키와 XOR 연산을 한 번 수행하기 때문이다[7][8].

2.1.3 SubByte

AddRoundKey 과정을 거친 state는 SubByte 연산과정을 수행하게 된다. SubByte 연산은 state의 각 byte를 서로 독립적인 비선형성을 갖는 치환 테이블인 S-Box를 적용하여 치환하는 과정이다. 먼저 각 byte를 유한체 $GF(2^8)$ 위의 다항식으로 표현하여 $\bmod x^8+x^4+x^3+x+1$상에서 역수를 구하는 것으로 유클리드 호제법을 이용한다. 그리고 $GF(2)$상에서 Affine 변환을 적용하는 과정으로 이 과정을 정리한 표가 S-Box이다.

S-Box를 이용하는 방법은 128bit 블록의 state를 1byte씩 치환한다. 상위 4bit는 S-Box상에서 행의 값을 선택하고, 하위 4bit는 열의 값을 선택하여 행과 열에 해당하는 값을 기존의 state에 대입한다. 이 과정을 정리한 그림은 아래 그림과 같다.

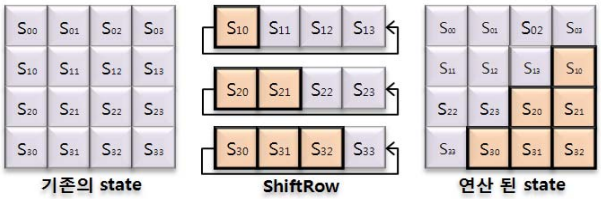


<그림 3> S-Box를 이용한 SubByte 과정

위와 같은 과정을 모든 byte에 적용하여 기존의 state를 비선형성을 갖는 새로운 state로 변환한다[9][10].

2.1.4 ShiftRow

ShiftRow 연산은 SubByte 연산을 마친 state를 행 단위로 순환 시프트 연산을 수행하는 과정이다. 아래의 그림과 같이 byte를 좌측으로 순환 시프트를 수행한다. 여기서 1행은 이동하지 않고, 2행은 1번, 3행은 2번, 4행은 3번 수행함으로써 n행에 대하여 $n-1$만큼 왼쪽으로 이동하는 연산을 수행하게 된다[11][12].



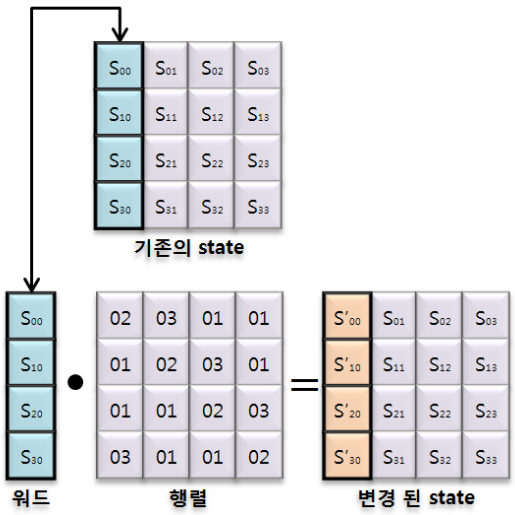
<그림 4> ShiftRow 연산과정

2.1.5 MixColumn

MixColumn 연산은 이전 과정을 거친 state를 다항식 $a(x)=03x^3+01x^2+01+02$와 연산하는 과정이다.

$GF(2^8)$상에서 두 개의 3차 다항식인 $a=a_3x^3+a_2x^2+a_1x+a_0$와 $b=b_3x^3+b_2x^2+b_1x+b_0$를 곱한 결과인 $c(x)$를 $\bmod x^4+1$를 계산하는 것이다.

위의 연산이 정리된 행렬을 이용하여 state를 열 단위로 연산을 수행하여 새로운 값을 갖는 state로 변환한다.



<그림 5> 행렬을 이용한 MixColumn 과정

위 그림과 같은 연산을 모든 워드에 수행하여 state의 모든 값이 변경된다. 그리고 이 과정은 전체 알고리즘에서 마지막 라운드에는 수행하지 않기 때문에 총 $Nr-1$만큼 수행하게 된다[13][14].

3. 프로그램 설계

본 논문에서 구현한 프로그램은 실제 암·복호화를 수행하면서 각 암·복호화 과정에 따른 설명을 포함한 인터페이스를 구현하였다. 알고리즘을 구현하기 위해 C++ 언어를 사용했으며 인터페이스 구현을 위해서 C++ 기반의 MFC가 사용되었다.

3.1 프로그램 설계

학습 프로그램에 기본 암·복호화 기능을 구현하기 위해 기본적인 AES 알고리즘을 따로 구현하였다. 이 기본 알고리즘 모듈은 아래 <그림 6>과 같은 인터페이스로 사용자가 암호화나 복호화를 선택하고 128bit 이하의 키를 이용하여 이에 따른 결과를 확인할 수 있다.

<그림 6> MFC로 구현한 기본 모듈

3.1.1 연산과정 설계

AES 암호 알고리즘에 사용되는 주요 연산과정에는 KeyExpansion, AddRoundKey, SubByte, ShiftRow, MixColumn으로 5가지 연산과정이 있다. 본 절에서는 AES에 사용되는 주요 연산과정을 C++를 통해 구현한 핵심적인 소스코드에 대해 설명한다.

1) KeyExpansion

텍스트로 입력받은 키 값을 확장하는 KeyExpansion 연산에서는 확장시킨 키를 AddRoundKey에서 각 라운드에 필요하게 된다. 총 Nr+1만큼의 키로 늘리기 위해 128bit의 키 state를 AES 알고리즘의 키 확장 스케줄에 따라 확장한다. 그리고 여기에 사용되는 SubWord와 RotWord를 함수로 구현하여 KeyExpansion 함수 내부에서 동작할 수 있도록 구현하였다.

```
void KeyExpansion(BYTE* key, WORD* W)
{
	WORD temp;
	int i=0;
	while(i<Nk)
	{
		W[i] = wCha(key[4*i], key[4*i+1],
			key[4*i+2], key[4*i+3]);
	}
	i=Nk;//i=4
	while(i<(Nb*(Nr+1)))
	{
		temp=W[i-1];
		if(i%Nk==0)
			temp = Sub(Rot(temp))^Rcon[i/Nk-1];
		else if((Nk>6) && (i%Nk==4))
			temp=Sub(temp);

		W[i]=W[i-Nk]^temp;
		i+=1;
	}
}
```

위 소스코드에서는 입력받은 128bit의 키를 4개의 32bit 씩 워드로 나누어 계산을 수행한다. 그리고 이 행을 다시 state에 넣어 배열로 구성하여 새로운 확장된 키를 생성한다.

2) AddRoundKey

이 과정에서는 KeyExpansion에서 확장된 키에서 해당 라운드에 포함되는 라운드 키와 평문 state를 XOR한다. state를 2차원 배열로 구현하여 해당하는 열을 마스킹기법을 이용하여 8bit씩 XOR 연산을 수행하고 있다. 평문의 한 워드와 키의 한 워드를 뽑아내 각각 XOR 연산을 수행하도록 연산과정을 구현하였다.

```
void AddRoundKey(BYTE state[][4], WORD* rKey)
{
	int i, j;
	WORD mask, shift;

	for(i=0;i<4;i++)
	{
		shift=24;
		mask=0xFF000000;

		for(j=0;j<4;j++)
		{
			state[j][i]=((rKey[i]&mask)>>shift)^state[j][i];
			mask>>=8;
			shift-=8;
		}
	}
}
```

3) SubByte

SubByte 연산에서는 S-Box를 이용하여 state의 한 byte를 치환하는 과정이다. 먼저, S-Box에 해당하는 데이터를 2차원 배열에 대입하였다. 그리고 state에 있는 값의 상위 값과 하위 값을 구분하기 위해 8bit의 byte를 상위 4bit, 하위 4bit로 나누어 해당하는 bit의 값을 이용하여 S-Box 배열에서 값을 찾아 해당하는 값으로 치환한다.

```
#define HIGH(x) (x>>4)
#define LOW(x) (x&0x0F)

void SubBytes(BYTE state[][4])
{
	int i, j;
	for(i=0;i<4;j++)
	{
		for(j=0;j<4;j++)
		{
			state[i][j]=
			S_box[HIGH(state[i][j])][LOW(state[i][j])];
		}
	}
}
```

위와 같은 과정을 state의 16byte 모두 수행하여 state의 모든 값이 치환된다.

4) ShiftRow

ShiftRow 연산에서는 state에 있는 값을 행 단위로 시프트 연산을 수행하기 위해 행에 해당하는 값을 인자로 전달하여 수행해야 하는 시프트 연산만큼 수행한다. state의 배열을 이용하여 각 행마다 $n-1$번 씩 배열이 좌측으로 이동하도록 함수를 구현하였다.

```
void ShiftRows(BYTE state[][4])
{
	int i, j;

	for(i=1;i<4;i++)
		for(j=0;j<i;j++)
			sShiftRows(state[i]);
}

void sShiftRows(BYTE* mrow)
{
	BYTE temp=mrow[0];

	mrow[0]=mrow[1];
	mrow[1]=mrow[2];
	mrow[2]=mrow[3];
	mrow[3]=temp;
}
```

5) MixColumn

MixColumn 연산에서는 state의 byte 블록끼리 유한체 상에서의 변환 행렬과 곱해짐으로써 state의 값이 변경된다. 기약다항식 $GF(2^8)$행렬과 state의 각 열을 XOR하여 연산 된 행을 다시 state에 넣는 과정이다. 이를 위해 다항식의 곱을 행렬 형태의 배열로 연산과정을 구현하였다. 그리고 state의 워드를 왼쪽부터 순차적으로 뽑아내 입력해놓은 행렬값과 연산이 이루어지도록 구현하였다.

```
void MixColumns(BYTE state[][4])
{
	int i, j, k;
	BYTE a[4][4]={
			0x02, 0x03, 0x01, 0x01,
			0x01, 0x02, 0x03, 0x01,
			0x01, 0x01, 0x02, 0x03,
			0x03, 0x01, 0x01, 0x02
	};
	for(i=0;i<4;i++)
	{
		BYTE temp[4]={0,};
		for(j=0;j<4;j++)
			for(k=0;k<4;k++)
				temp[j]^=x_time(state[k][i], a[j][k]);

		state[0][i]=temp[0];
		state[1][i]=temp[1];
		state[2][i]=temp[2];
		state[3][i]=temp[3];
	}
}
```

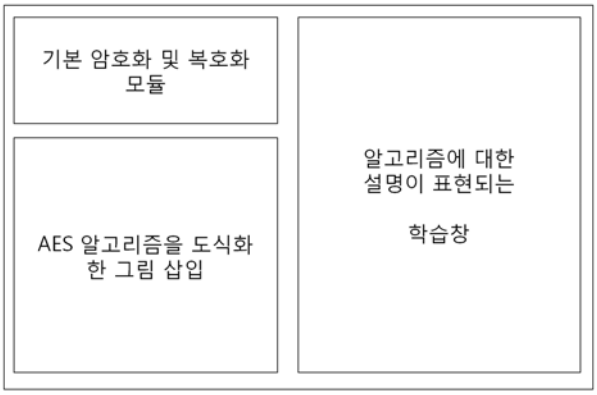
3.1.2 학습기능 설계

AES 암호 알고리즘 학습 프로그램에 필요한 기능을 아래와 같이 정리하였다.

- o 사용자가 직접 암호화 및 복호화 수행 가능
- o 알고리즘 전체적인 구조 설명
- o 연산과정별 동작과정 구현
- o 연산과정에 대한 구체적 설명
- o 이해를 돕기 위한 그래픽 환경

본 논문에서 구현하는 AES 암호 알고리즘 학습 프로그램에서는 사용자가 직접 암·복호화 과정을 직접 살펴볼 수 있는 기능을 제공해야 한다. 그리고 GUI 환경을 중심으로 state가 변화되는 과정을 살펴보면서 복잡한 알고리즘을 설명해야 한다.

위와 같은 요구조건을 만족시키기 위해서는 AES 암호 알고리즘의 이론적인 내용을 쉽게 이해할 수 있도록 과정에 따른 설명이 부가적으로 보일 수 있도록 구현해야 한다. AES 암호 학습 프로그램에 필요한 인터페이스를 아래 <그림 7>에 표시하였다.



<그림 7> 프로그램 기본 설계

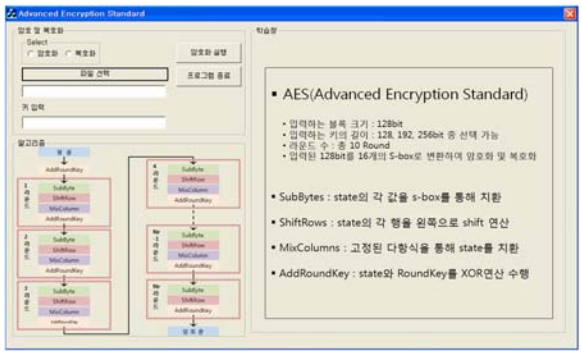
위의 그림에 보이는 바와 같이 기본적인 암·복호화 기능과 전체적인 알고리즘을 그림으로 도식화하여 나타내고 암호화 과정을 순차적으로 보면서 각 연산과정에 따른 설명이 나오도록 구현하였다.

기본 암·복호화 모듈은 단순한 암·복호화 기능만 수행하는 것이 아니라 이 때 사용자가 입력한 평문과 암호화 키는 '학습창'에서 각 연산과정을 설명하는데 사용된다. 사용자가 입력한 값들을 이용하여 각 연산과정의 설명될 부분을 직접 입력한 값들을 이용하여 출력되도록 구현 하였다.

3.2 프로그램 구현

본 논문에서 구현한 AES 암호 알고리즘 학습 프로그램는 크게 세 부분으로 나눌 수 있다. 먼저 기본 설정 및 평문, 키 입력 부분인 '암호 및 복호화'부분과 알고리즘을 도식화하여 나타낸 '알고리즘' 부분, 알고리즘에 설명이 출력되는 '학습창'으로 세 가지로 구성되어 있다.

3.2.1 기본 인터페이스



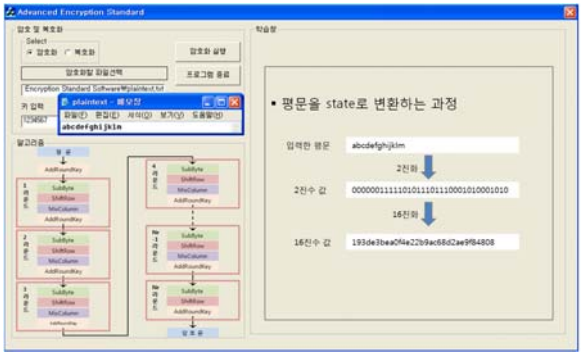
<그림 8> 구현된 기본 인터페이스

프로그램을 실행시킨 초기화면은 위의 그림과 같이 구성되었다. 기본적으로 암·복호화 기능을 가지고 있으며 사용자가 파일을 선택하여 키를 입력하여 암·복호화 하는 방식이다. 그리고 왼쪽 하단에 AES 암호 알고리즘의 전체적인 흐름을 그림으로 구성하여 나타내었다. 사용자가 알고리즘에 대해 쉽게 알아볼 수 있도록 연산과정을 색상으로 구분하여 나타냈다. 오른쪽에는 각 과정마다 설명을 볼 수 있는 '학습창'이 있으며, 위의 그림은 초기 실행화면으로 AES에 대한 간략한 설명과 연산과정에 따른 설명이 간단히 나타나 있다.

알고리즘 도식화 그림에 표시되어 있는 연산과정의 박스를 클릭하면, 해당 부분 과정에 대한 설명이 오른쪽 '학습창'에 출력된다.

오른쪽에 설명 부분에서는 가장 먼저 해당 연산과정에 대한 기본적인 설명이 나오도록 프로그램이 구성되어 있으며, 이 후에 사용자가 직접 입력한 값을 이용하여 각 연산과정에 대한 설명이 그림과 함께 출력된다.

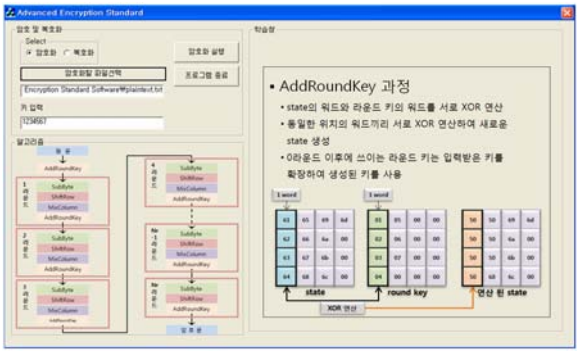
3.2.2 평문 변환 과정



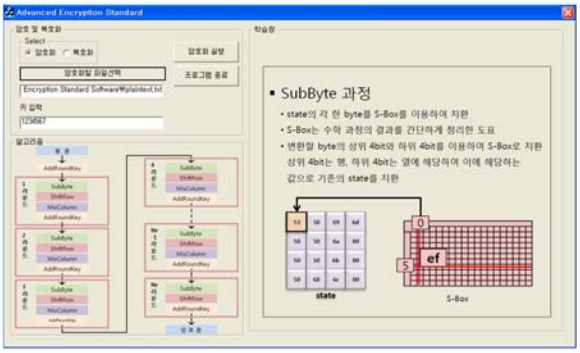
<그림 9> 초기 state 설정 화면

처음 암호화를 실행하기 위해서 왼쪽 상단부분에서 사용자가 평문과 키를 입력한 뒤, 왼쪽 하단에 위치한 알고리즘 부분의 "평문"을 클릭하면 오른쪽에 있는 '학습창'에 AES 암호 알고리즘 상에서 평문을 state로 변환하는 과정이 나타난다.

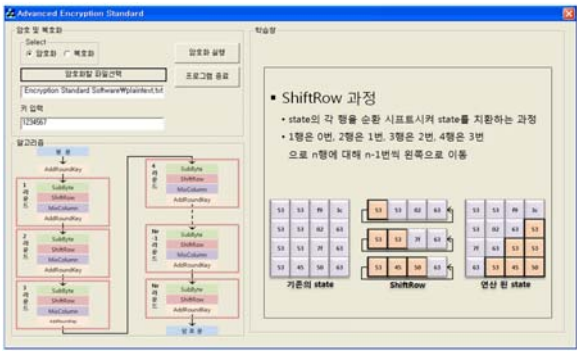
3.2.3 각 연산과정



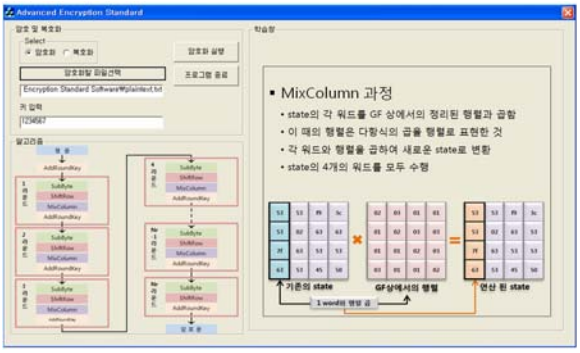
<그림 10> AddRoundKey 과정



<그림 11> SubByte 과정



<그림 12> ShiftRow 과정



<그림 13> MixColumn 과정

왼쪽 하단의 '알고리즘'에서 AddRoundKey, SubByte, ShiftRow, MixColumn과 같은 연산과정을 선택하면 해당 연산과정이 '학습창'에 출력된다. 연산과정에 대한 기본적인 설명과 각 연산과정에 대한 설명을 그림으로 표현된 설명이 나오게 되며, 이 때 그림으로 출력되는 state와 사용되는 라운드 키는 모두 사용자가 직접 입력한 평문과 키에서 변환되는 과정의 state와 입력된 키를 실제로 변환되고 있는 state와 실제로 사용되는 라운드 키를 이용하여 출력된다.

4. 결 론

본 논문에서는 사용자들의 보안 인식 제고를 위해 가장 많이 사용하고 있는 AES 암호 알고리즘을 사용하여 사용자들이 보안에 대해 좀 더 쉽게 다가갈 수 있도록 학습 프로그램을 개발하였다. 많은 사용자들은 인식하지 못하는 사이에 암호를 자주 사용하지만, 실제로 암호로 변화되는 과정을 배우기는 쉽지 않다. 또한 암호학을 학습하는 학습자의 경우에도 암호학 관련 서적을 참고해야 하며, AES 암호 알고리즘 또한 수학적 원리를 이용한 암호이기 때문에 쉽게 접근하기 어렵다.

따라서 본 논문에서 구현한 AES 암호 알고리즘 학습 프로그램은 AES 암호 알고리즘에 대한 설명을 단순한 글이나 그림으로 이루어진 설명이 아닌, 사용자가 직접 암·복호화를 수행하면서 AES 암호 알고리즘에 대한 지식을 습득할 수 있다. 또한 암호화 과정에 필요한 키 확장 및 평문화 되는 과정을 사용자가 직접 입력한 값을 바탕으로 설명이 나오기 때문에 관련 서적으로 학습하는 것과 달리 이해하는데 있어 더 수월하도록 설계되어 있다. 알고리즘에 대한 설명 부분에서도 사용자가 직접 입력한 평문과 키를 이용하여 설명하기 때문에 암호 알고리즘에 대한 전반적인 이해가 쉽도록 구현하였다.

본 논문에서 구현한 프로그램을 통하여 여러 사용자에게 암호학에 대한 교육 및 흥미를 유발하고 정보보호가 어려운 이론이 아닌, 쉽게 다가갈 수 있는 학문으로 인식 될 수 있을 것으로 기대된다.

향후 연구 방향으로는 사용자들에게 정보보호에 대한 중요성과 보안 의식을 향상시킬 수 있는 다양한 보안 프로그램에 대한 연구가 필요하다. 단순한 보안에 대한 이론을 전달하는 것이 아닌, 사용자들로 하여금 흥미를 가질 수 있고 이에 대해 쉽게 이해할 수 있는 방안을 모색하는 연구가 필요하다.

참 고 문 헌

[1] 엄명용, 박진희, 김미량 (2002). 사이버 일탈 행위 예방을 위한 개인 정보보호 노력의 영향요인 연구. **한국컴퓨터교육학회논문지, 5**(2), 1-11.

[2] 장윤재, 김동형, 김한성, 이원규, 김현철 (2011). 정보보호 교육을 위한 언플러그드 활동의 개발 및 유용성 평가. **한국컴퓨터교육학회논문지, 14**(1), 55-64

[3] Lee, A. (1999). *Guideline for Implementing Cryptography in the Federal Government*. Nist SP 800-21. 112

[4] Daemen, J., &Rijmen, V. (2002). *The Design of Rijndael*. Berlin: Springer
[5] NIST (2001). *Specification for the Advanced Encryption Standard(AES).* National Institute of Standards and Technology. IEEE 2, 481-483.
[6] Daemen, J., &Rijmen, V. (2003). *AES Proposal: Rijndael Block Cipher*. NIST Documnet ver.2.
[7] 구본석, 유권호, 양상운, 장태주, 이상진 (2006). RFID 태그를 위한 초소형 AES 연산기의 구현. **정보보호학회논문지, 16**(5), 67-76
[8] Gueron, S. (2008). *Advanced Encryption Standard (AES) Instructions Set*. White paper of Intel.
[9] Carter, B., Kassin, A., &Magoc, T. (2007). *Advanced Encryption Standard*. Penn State University.
[10] 노진수, 이강현 (2006). 합성체를 이용한 유한체의 역원 계산 알고리즘 구현. **전자공학회논문지, 43**(3), 77-78
[11] Selent, D. (2010). Advanced Encryption Standard. *Rivier Academic Journal*. Volume 6.
[12] 안하기, 신경욱 (2002). AES Rijndael 블록 암호 알고리듬의 효율적인 하드웨어 구현. **정보호호학회논문지, 12**(2), 53-63
[13] Dawood, M. Z., & Khan, A. R. (2006). *Advanced encryption standard*. National Computer Conference. King Faisal University.
[14] 최병윤 (2001). AES Rijndael 알고리즘용 암호 프로세서의 설계. **한국통신학회논문지, 26**(10), 1491-1500

이 동 범

2008 순천향대학교 정보보호학과 (정보보호학 학사)
2010 순천향대학교 정보보호학과 (정보보호학 석사)
2010～현재 순천향대학교 정보보호학과 박사과정
관심분야: 컴퓨터교육, 정보보호, 보안성 평가
E-Mail: dblee@sch.ac.kr

정 명 수

2006～현재 순천향대학교 정보보호학과(학사과정)
관심분야: 컴퓨터교육, 정보보호, 암호 프로토콜 등
E-Mail: msjeong@sch.ac.kr

곽 진

1994～2006 성균관대학교(공학사, 공학석사, 공학박사)
2006～2006 일본 큐슈대학교 방문연구원
2006～2006 일본 큐슈시스템 정보기술연구소 특별연구원
2006～2007 정보통신부 개인정보보호기획단 개인정보보호팀 통신사무관
2007～2009 정보통신연구진흥원 집필위원
2007～현재 순천향대학교 정보보호학과 교수
2009～2009 순천향대학교 공과대학 교학부장
2009～2010 순천향대학교 정보보호학과 학과장
2010～2010 교육과학기술부 국가기술수준평가 전문위원
현재: 정보통신산업진흥원 기술평가위원, 사)국제정보능력평가원 쇼핑몰 플래너 자격 검정 출제 및 채점위원, 한국과학기술정보연구원 충남 과학기술 정보협의회 전문위원, 지식경제부 지식경제기술혁신평가단 평가위원, 순천향BIT 창업보육센터 센터장, 순천향대학교 중소기업산학협력센터 센터장
관심분야: 암호프로토콜, 응용시스템보안, 개인정보보호, 정보보호제품평가, 클라우드 컴퓨팅보안 등
E-Mail: jkwak@sch.ac.kr