한국학교수학회논문집 제 6 권, 제 2 호
Journal of the Korean School Mathematics Society
Volume 6, Number 2, 101-115, December 2003

DES를 이용한 암호의 이해와 활용 및 DES에서 한글 구현

정상조[1] · 박중수[2]

DES는 수학의 치환을 사용한 암호 알고리즘이다. 현재는 이를 개선한 AES가 표준화되어 사용되고 있다. 본 논문에서는 중 고등학교의 특별활동 자료로 사용할 수 있도록 DES를 소개하고 이의 구현을 시도하였다. 특히 한글의 처리를 할 수 있도록 하였다. 수학의 간단한 이론이 현대의 복잡한 정보화 사회에서 첨단 기술로 사용하고 있음을 알 수 있다.

주요용어 : 치환, 평문, 암호문, 암호화, 복호화, DES, AES

I. 서 론

1. 연구의 필요성

현대는 컴퓨터가 널리 보급되었고 인터넷을 통하여 세계의 모든 컴퓨터가 연결된 상황에서 각 개인의 컴퓨터 또는 기업의 서버 컴퓨터에서 정보의 보호나 보안은 매우 중요하게 되었다. 이중에서 정보의 보호는 수학적인 이론을 이용하면 실현 가능한데 이를 위한 것이 암호이다.

초기의 암호는 군사적 목적을 위하여 제한적으로 사용되었다. 일반인들은 접근이 용이치 않았고 또 그 필요를 별로 느끼지 못하였다. 그러나 전자상거래가 활성화되고 개인이 인터넷을 통하여 비밀번호나 개인의 신상에 관한 파일들을 전송하기 시작하면서 개인들도 암호의 기술을 사용하게 되었다. 따라서 암호는 일상생활과 밀접한 관계를 맺게 되었고 각 개인도 이의 원리를 이해하고 활용하는 것은 필수적이다.

정보화 사회에서 암호의 역할은 매우 중요하다고 본다. 따라서 중 고등학교에서도 암호에 대한 이해와 이의 활용에 대하여 공부할 필요가 있다고 본다.

2. 연구의 목적

암호는 수학적 이론을 사용하므로 중 고등학교에서 암호에 사용되는 이론을 살펴보고 실습함으로써 수학이 응용학문으로서의 역할을 하고 있고 학생들에게 수학학습의 동기 유발을

1) 충남대학교 수학과(math88@dreamwiz,com)
2) 우석대학교 수학과(jspark@woosukac.kr)

시킬 수 있다. 특히 DES는 어려운 수학개념을 많이 사용하지 않고 중 고등학교의 기초수학, 예를 들면 함수와 치환 등 몇 가지의 사실만 가지고도 알고리즘의 전개가 가능하다. 수학의 기초개념이 최첨단 암호이론에 사용됨을 알 수 있다.

DES를 공부함으로써 현재 사용되고 있는 암호에 대한 이해를 높일 수 있고 정보화 사회의 시민으로써 자신의 정보 보호에 대한 경각심을 불러일으킬 수 있다. 현대 정보화 사회의 시민이 가져야 할 소양으로 컴퓨터 언어에 대한 이해는 필수적이라 본다. 컴퓨터 프로그램을 예시함으로써 컴퓨터 언어를 구사하지 못하는 학생이 쉽게 접근할 수 있도록 하였다. 프로그래밍은 논리적 사고의 훈련과 그의 사용에 매우 유리하다고 본다. 학생들이 자연스럽게 컴퓨터언어에 익숙해 질 수 있다.

본 연구는 중 고등학교 단계에서 특별활동으로 암호의 이론을 이해하고 실습 가능한 컴퓨터 언어를 사용하여 제시된 암호 알고리즘을 구현하는데 목적이 있다.

3. 연구의 효과 및 활용

DES를 개량한 블록암호는 현재 가장 널리 사용되고 있는 암호시스템으로 은행의 온라인 업무와 전자상거래의 비밀번호 교환 등에 널리 쓰이고 있으므로 이를 습득하는 것은 정보의 중요성에 대한 이해뿐만 아니라 수학이론의 활용측면에서도 그 의의가 크다고 생각된다.

학생은 정보의 중요성을 인식하고 정보의 보호를 위해서 어떤 일들이 진행되고 있으며 현재 사용 중인 암호화 방법은 무엇인지를 알 수 있다. 또 현재 사용되고 있는 암호화 방법을 습득하고 실습함으로써 암호시스템에 대한 이해를 높일 수 있다.

최근 전자상거래의 활성화로 인터넷 회선 상에서 개인의 정보가 교류되고 있다. 이로 인한 사고를 방지하기 위해서 보다 정밀하게 설계된 암호화 방법은 필수적이다. 선진국에서는 정밀한 암호화 방법의 개발을 위해서 투자를 아끼지 않고 있다. 우리나라도 IT 선진국으로서 암호에 대한 연구가 진행되고 있다. 개인도 그 원리를 앎으로써 안전에 대한 믿음을 가지고 현대 생활을 할 것으로 생각된다. 또 미래의 암호방법의 개발에도 관심을 가질 수 있다.

중 고등학교의 특별활동 프로젝트로 활용할 수 있다[3]. 또 하나의 언어를 선택하여 학생이 직접 코딩하면 컴퓨터 언어에도 익숙해 질 것으로 생각된다. 컴퓨터 언어는 철저하게 계산된 논리위에 건설되었으므로 학생이 컴퓨터 언어를 이용하여 프로그래밍 실습을 하면 논리적 사고의 중요성과 그의 의미를 실감할 것으로 생각된다.

II. 본 론

1. 암호의 이해

누구에게나 지키고 싶은 비밀이 있다. 개인과 기업, 국가에서 정보의 중요성은 다시 말할 필요가 없을 것이다. 이 소중한 정보를 어떻게 타인이나 타 기관으로부터 지킬 것인가? 첫째 방법은 정보에 대한 침입을 막기 위해서 보안을 철저히 하는 것이고 둘째로는 보안이 뚫

3) http://math88.com.ne.kr/crypto.htm

렸다고 하더라도 탈취된 정보가 암호화되어 있어서 상대방이 알 수 없게 되어야 한다. 보안은 어느 정도 하드웨어적인 기술이 필요하지만 정보의 보호는 암호화 알고리즘이라는 수학적 방법으로 해결할 수 있다.

암호는 그 의미를 바로 알 수 없게 의도적으로 만든 기호나 문자의 나열을 말한다. 원래의 문장을 평문(plain text)이라 하며 어떤 과정, 즉 암호화 과정을 거쳐 의미를 알 수 없는 문장이 된 것을 암호문(cipher text)이라 한다. 이 과정을 암호화(encryption)라 하고 다시 암호문을 평문으로 고치는 과정을 복호화(decryption)라 한다. 여기서 암호화키(encryption key)와 복호화키(decryption key)가 사용된다.

역사적으로 암호의 사용은 여러 곳에서 발견되고 개인 용도의 암호까지 생각하면 다양한 종류의 암호가 존재 할 것이다. 이와 같이 많은 암호 방법들은 그 특징에 따라 다양하게 구별될 수 있고 다음 구별법들도 그것들 중의 하나이다(장은성, 1999).

(1) 입력크기에 의한 분류: 블록 암호시스템은 고정된 크기의 평문을 고정된 크기의 암호문으로 변형하는 경우를 말하고 스트림 암호시스템은 처리 단위가 한 글자씩인 것을 말한다.

(2) 키에 의한 분류: 사용하는 암호화키와 복호화키가 같으면 대칭키 암호시스템이라 하고 다르면 비대칭키 암호시스템이라 한다. 비 대칭키 암호시스템에는 현대 암호에서 매우 중요한 개념인 공개키 암호시스템이 포함된다.

(3) 시대적 분류: 암호는 발생시기에 따라 고전암호와 현대암호로 나눈다. 그러나 이 구별은 큰 의미는 없다. 단지 고전암호에 비해 현대암호는 현대적인 알고리즘과 암호체계를 가지고 있다고 생각하는 것이다. 다시 말하면 현대암호는 고전암호에 비하여 정밀한 암호 알고리즘을 사용하고 알고리즘과 암호화키의 공개를 지향한다고 볼 수 있다. 이를 정리하면 다음 그림 1과 같다.

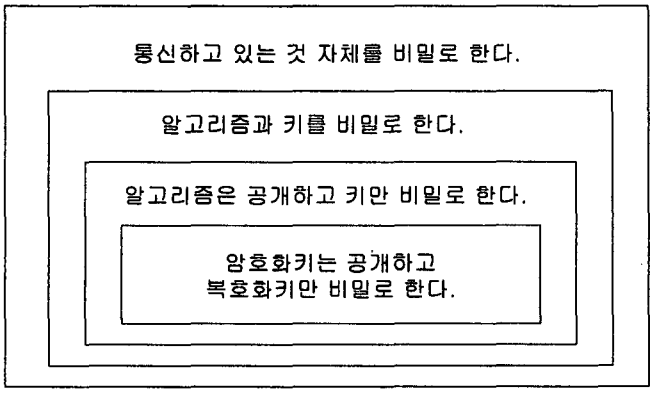

그림 1. 암호의 발전과정과 비밀부분의 축소

2. DES의 이해

DES는 시대적으로는 현대암호에 속하고, 입력크기에 의한 분류로는 블록암호에 속하며 키에 의한 분류로는 대칭키 암호에 속한다. DES 알고리즘은 공개되었으므로 누구나 로열티

를 내지 않고 DES를 이용하여 암호화와 복호화가 가능한 프로그램을 만들 수 있다. DES는 안전성과 속도가 검증된 최초의 표준화된 암호화 시스템이다. 그러나 암호화키와 복호화키가 동일하므로 키의 관리가 매우 중요하다. 따라서 직접 만나서 키를 교환하든지 아니면 공개된 회선 상에서 키를 교환하려면 공개키 암호를 사용하여야 한다.

1) 암호 알고리즘의 표준화는 왜 필요한가?

DES는 최초의 표준화된 알고리즘이다. 다른 분야에서의 표준화(standardization)와 마찬가지로 암호화에서도 표준화가 필요하다. 컴퓨터와 통신기술의 발달은 어떤 개인이나 단체가 개별적으로 만든 암호 알고리즘을 사용하여 많은 자료를 처리하고 저장하기에는 어렵게 되었다. 이에 미국 상무성의 국립표준국(NBS, National Bureau of Standards)은 1973년 5월 컴퓨터의 자료와 통신정보를 보호할 목적으로 저장과 전송에 사용될 수 있는 효과적인 암호 알고리즘을 공모하였으며 1975년 IBM의 연구원 W. Tuchman과 C. Meyer는 DES(Data Encryption Standard)를 제안하였고 안전성과 속도를 검증받아 1977년 1월 미국표준암호알고리즘으로 채택되었다(NBS, 1997).

2) DES의 이론적 배경 및 문제점

C. E. Shannon은 논문에서 확산과 혼동이라는 개념으로 실용적인 암호시스템을 만들 수 있다고 주장했다(Shannon, 1948; Shannon, 1949)). 여기서 확산은 평문에 사용되는 문자들을 암호문에 고루 분산시키고 혼동은 암호문과 평문의 대응을 어렵게 만드는 것이다. DES는 샤논의 개념을 적용하여 64비트의 평문을 32비트씩 나누어 전치변환과 48비트의 암호키로 환자변환을 16단계(round)나 반복한다.

DES는 초기에 대략 5년마다 공개적인 검토를 거쳐 재인증하는 과정을 거침으로서 암호학이 공개적인 영역으로 출현되게 하여 공개적으로 암호학을 연구하고 토론하여 오늘날과 같이 암호학을 발전시키는 계기가 되는 최초의 암호가 되었으며 1983년 첫 검토를 한 이래 1994년 1월 재승인 되었다.

현재까지 DES는 가장 널리 사용되고 신뢰를 얻고 있는 대표적인 대칭암호계로 미국 외에서도 널리 사용되어 왔다. 그러나 열쇠의 길이가 짧고, 컴퓨터 속도의 개선과 암호해독기술의 발전으로 오늘날 더 이상 DES를 안전하다고 만 생각하지 않게 되었다. 최근 DES를 보완하는 끊임없는 연구로 많은 블록암호들이 개발되어 공개되어 왔으며, 최근 2000년 10월 2일에 벨기에에서 만든 Rijndael(Rijmen & Daemen가 개발)이 AES로 채택되어, DES암호의 뒤를 이은 블록암호 시스템으로 활용되고 있다[4].

3) DES의 이해

DES는 블록암호로 평문을 64비트 단위로 끊어서 암호화한다. 키는 64비트-실제로는 56비트이고 8비트는 검사용으로 사용-를 사용한다. 따라서 입력 64비트에 대하여 64비트의 키를 사용하여 출력 64비트를 얻는다. 영문인 경우 영문 1자당 2진수 8비트씩 할당되므로 8글자씩, 한글인 경우 한글 1자당 2진수 16비트씩 할당되므로 4글자씩 블록으로 나누어 암호화가 이루어진다.

---

4) http://www.nist.gov/aes

Ⅲ. 실 습

1. DES 알고리즘

우선 DES의 전 과정을 살펴보기 위하여 DES를 크게 세 부분으로 나누어 보자. 이를 그림으로 나타내면 아래의 그림 2와 같다.

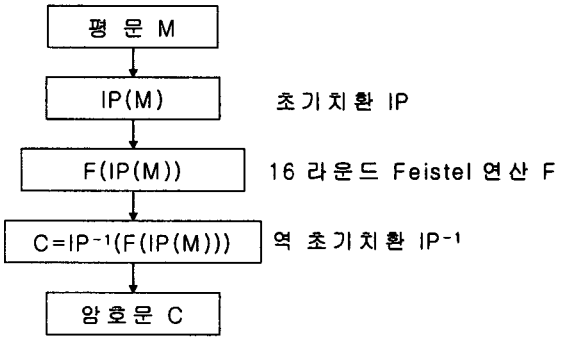

그림 2. DES의 알고리즘

먼저 그림 2의 과정에 대해 알아보면

① 64비트 길이의 평문 M에 초기치환 IP(initial permutation)을 적용하여 64비트 길이의 치환문 IP (M)을 얻는다.

② 16 라운드의 Feistel 연산 F을 적용하여 64비트 길이의 치환문 F (IP (M))을 얻는다. 여기서 16개의 48비트 크기의 키가 필요하다.

③ 64비트 길이의 치환문 F (IP (M))에 역 초기치환(inverse initial permutation) IP^{-1}를 적용하여 암호문 $C=IP^{-1}$ (F (IP (M))) 얻는다.

그림 2의 DES 구조를 좀더 상세히 설명하면 그림 3과 같다. 그림 3에서 사용된 함수를 간단히 설명하면 다음과 같고 자세한 정의는 참고문헌을 참조하면 된다(이민섭, 2000; NBS, 1997).

① 괄호 안의 숫자는 비트수를 나타낸다.

② IP는 초기치환(initial permutation) 함수이다.

③ E는 32비트를 48비트로 바꾸는 함수이다.

④ ⊕은 배타적 합(exclusive or, xor)이다.

⑤ S는 8개의 S-박스 (S_1,S_2, ... ,S_8)로 되어있다. S함수는 48비트를 입력받아 6비트씩 8개의 블록으로 나눈 후 각각의 블록을 8개의 부분 Si-박스를 통과시켜 4비트씩을 얻는다. 따라서 S함수는 48비트를 32비트로 바꾼다.

⑥ P는 32비트를 32비트로 바꾸는 치환함수이다.

⑦ IP^{-1}은 역 초기치환(inverse initial permutation) 함수이다.

⑧ PC-1은 64비트를 56비트로 바꾸는 함수이다.

⑨ SH는 좌측 쉬프트(left shift) 함수이다.

⑩ PC-2은 56비트를 48비트로 바꾸는 함수이다.

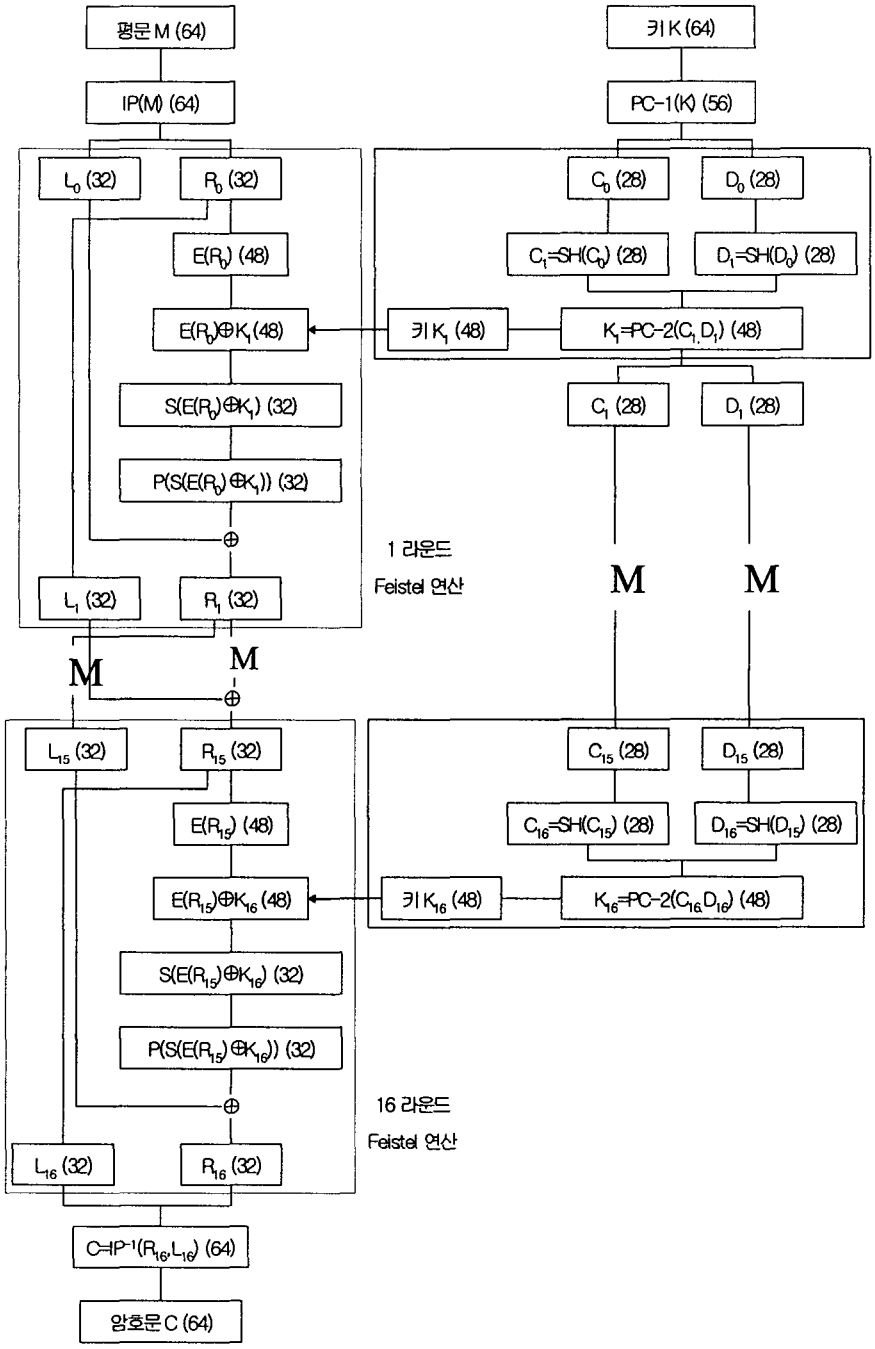

그림 3. DES의 상세 알고리즘 [단, ( )은 비트수]

단, DES암호의 복호화과정은 암호화의 역순으로 이루어진다. 위 그림에서 각 과정을 표

를 이용하여 나타내면 다음 표 1~4와 같다.

표 1. DES암호 암호화 알고리즘

단계	암호화 과정		비트수
평문	$M = m_1 m_2 \cdots m_{64}$		64
초기치환	$IP(M) = p_1 \cdots p_{32} p_{33} \cdots p_{64}$		64
나눔	$L_0 = p_1 \cdots p_{32}$	$R_0 = p_{33} \cdots p_{64}$	각각 32
1라운드	$L_1 = R_0$	$R_1 = L_0 \oplus P[F(E(R_0) \oplus K_1)]$	각각 32
2라운드	$L_2 = R_1$	$R_2 = L_1 \oplus P[F(E(R_1) \oplus K_2)]$	각각 32
⋮	⋮	⋮	⋮
15라운드	$L_{15} = R_{14}$	$R_{15} = L_{14} \oplus P[F(E(R_{14}) \oplus K_{15})]$	각각 32
16라운드	$L_{16} = R_{15}$	$R_{16} = L_{15} \oplus P[F(E(R_{15}) \oplus K_{16})]$	각각 32
바꿔 합침	$R_{16} L_{16}$		64
역치환	$IP^{-1}(R_{16} L_{16})$		64
암호문	$C = IP^{-1}(R_{16} L_{16})$		64

단, $E(R_i)$ 와 암호키 K_{i+1} 은 48 비트

표 2. DES암호키 생성 알고리즘

단계		단계별 암호키 생성과정		비트수
준비	암호키	$K = k_1 k_2 \cdots k_{64}$		64
	치환PC1	$\mathrm{PC1}(K) = r_1 \cdots r_{28} r_{29} \cdots r_{56}$		56
	나눔	$C_0 = r_1 \cdots r_{28}$	$D_0 = r_{29} \cdots r_{56}$	각각 28
1라운드	좌측이동T	$C_1 = T(C_0)$	$D_1 = T(D_0)$	각각 28
	치환PC2	$K_1 = \mathrm{PC2}(C_1 D_1)$		48
2라운드	좌측이동T	$C_2 = T(C_1)$	$D_2 = T(D_1)$	각각 28
	치환PC2	$K_2 = \mathrm{PC2}(C_2 D_2)$		48
⋮	⋮	⋮	⋮	⋮
15라운드	좌측이동T	$C_{15} = T(C_{14})$	$D_{15} = T(D_{14})$	각각 28
	치환PC2	$K_{15} = \mathrm{PC2}(C_{15} D_{15})$		48
16라운드	좌측이동T	$C_{16} = T(C_{15})$	$D_{16} = T(D_{15})$	각각 28
	치환PC2	$K_{16} = \mathrm{PC2}(C_{16} D_{16})$		48

표 3. DES암호 복호화 알고리즘

단계	복호화 과정		비트수
암호문	$C = c_1 c_2 \cdots c_{64}$		64
초기치환	$IP(C) = q_1 \cdots q_{32} q_{33} \cdots q_{64}$		64
나눔	$R_{16} = q_1 \cdots q_{32}$	$L_{16} = q_{33} \cdots q_{64}$	각각 32
1라운드	$R_{15} = L_{16}$	$L_{15} = R_{16} \oplus P[F(E(L_{16}) \oplus K_{16})]$	각각 32
2라운드	$R_{14} = L_{15}$	$L_{14} = R_{15} \oplus P[F(E(L_{15}) \oplus K_{15})]$	각각 32
⋮	⋮	⋮	⋮
15라운드	$R_1 = L_2$	$L_1 = R_2 \oplus P[F(E(L_2) \oplus K_2)]$	각각 32
16라운드	$R_0 = L_1$	$L_0 = R_1 \oplus P[F(E(L_1) \oplus K_1)]$	각각 32
바꿔 합침	$L_0 R_0$		64
역치환	$IP^{-1}(L_0 R_0)$		64
평문	$M = IP^{-1}(L_0 R_0)$		64

표 4. DES복호키 생성 알고리즘

단계		단계별 복호키 생성과정		비트수
준비	암호키	$K = k_1 k_2 \cdots k_{64}$		64
	치환PC1	$\mathrm{PC1}(K) = r_1 \cdots r_{28} r_{29} \cdots r_{56}$		56
	나눔	$C_0 = r_1 \cdots r_{28}$	$D_0 = r_{29} \cdots r_{56}$	각각 28
1라운드	우측이동T	$C_1 = T(C_0)$	$D_1 = T(D_0)$	각각 28
	치환PC2	$K_{16} = \mathrm{PC2}(C_1 D_1)$		48
2라운드	우측이동T	$C_2 = T(C_1)$	$D_2 = T(D_1)$	각각 28
	치환PC2	$K_{15} = \mathrm{PC2}(C_2 D_2)$		48
⋮	⋮	⋮	⋮	⋮
15라운드	우측이동T	$C_{15} = T(C_{14})$	$D_{15} = T(D_{14})$	각각 28
	치환PC2	$K_2 = \mathrm{PC2}(C_{15} D_{15})$		48
16라운드	우측이동T	$C_{16} = T(C_{15})$	$D_{16} = T(D_{15})$	각각 28
	치환PC2	$K_1 = \mathrm{PC2}(C_{16} D_{16})$		48

2. DES 알고리즘 실습프로그램

1) 암호화하기

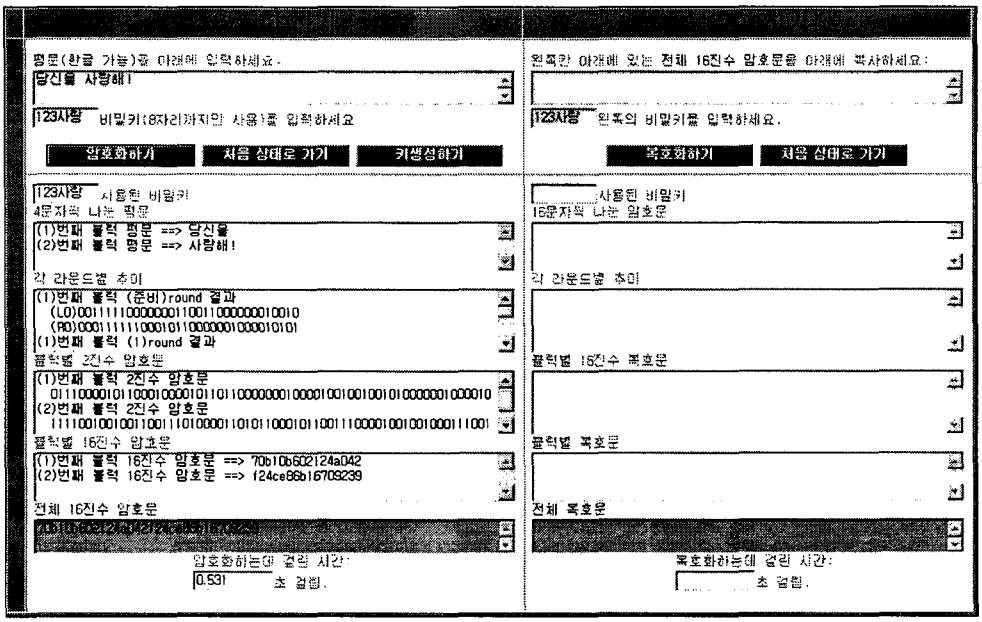

그림 4. 암호화하기

DES암호를 이용한 한글 암호화와 복호화를 위한 프로그램은 누구나 인터넷이 연결되어 있으면 연습을 할 수 있도록, 저자의 홈페이지 (http://math88.com.ne.kr/crypto.htm)에 올려 놓았다. 이 홈페이지에는 여러 가지 암호를 연습할 수 있는 프로그램이 있는데 그 중 **DES 암호(한글 가능)**을 클릭하면 위와 같은 모양의 그림 4가 나온다.

사용방법은 왼쪽 위의 노란 칸에 한글 또는 영문이 포함된 평문을 입력하고 그 밑에 비밀키를 입력한 후 **암호화하기** 키를 누르면 암호화하기 과정이 위 그림 4와 같이 출력된다. 한글은 한 글자당 16비트에 해당하므로 DES암호는 64비트씩 나누는 블록암호이므로 한글을 암호화하기 위해서는 4글자(16비트×4 = 64비트)씩 블록을 나누어 각 블록마다 암호화 과정을 거친다. 각 단계마다 생성되는 $(L_0, R_0), \cdot\cdot\cdot, (L_{16}, R_{16})$을 출력하였고, 각 블록마다 2진수 암호문과 16진수 암호문을 출력하였다. 암호문은 2진수로 하면 길이가 매우 길어지므로 16진수 변환하여 암호문을 출력하도록 하였다.

그리고 가장 밑에는 암호화하는데 걸린 시간이 출력되도록 하여 다른 암호화 프로그램과 암호화하는데 걸린 시간을 비교할 수 있도록 하였다.

처음 상태로 가기를 누르면 처음상태로 가도록 하였다. **키생성하기**를 누르면 16단계의 키가 다음 그림 5와 같이 경고문 형태로 출력된다.

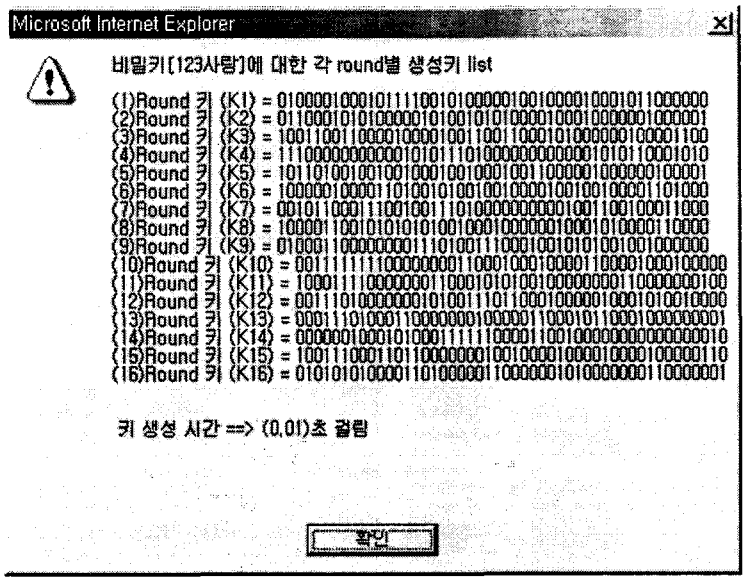

그림 5. 비밀키 생성하기

위 전 과정을 자세히 알고 싶으면 홈페이지(http://math88.com.ne.kr/crypto.htm)의 **DES 암호(전과정을 자세히 알수 있음)**를 클릭하면 아래와 같은 그림 6이 출력된다. 여기에서 **전과정을 자세히 알 수 있는 암호화 하기** 키를 누르면 암호화하기 전과정을 자세히 알 수 있다.

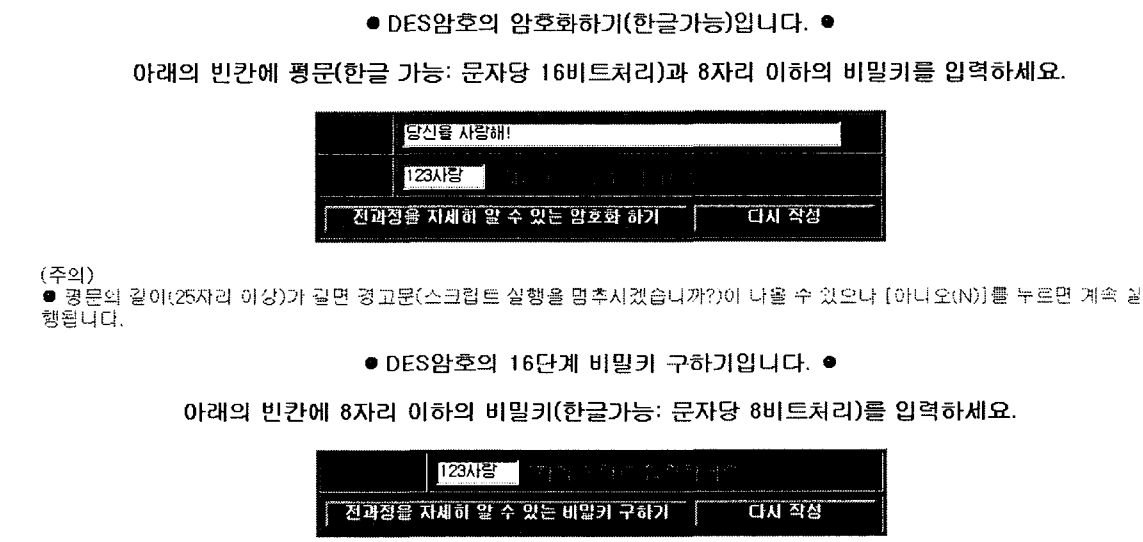

그림 6. 전 과정을 알 수 있는 암호화하기

마찬가지로 암호화와 복호화에 사용되는 16단계의 키생성 전 과정을 자세히 알고 싶으면 그림 6에 있는 **전과정을 자세히 알 수 있는 비밀키 구하기** 키를 누르면 된다.

2) 복호화하기

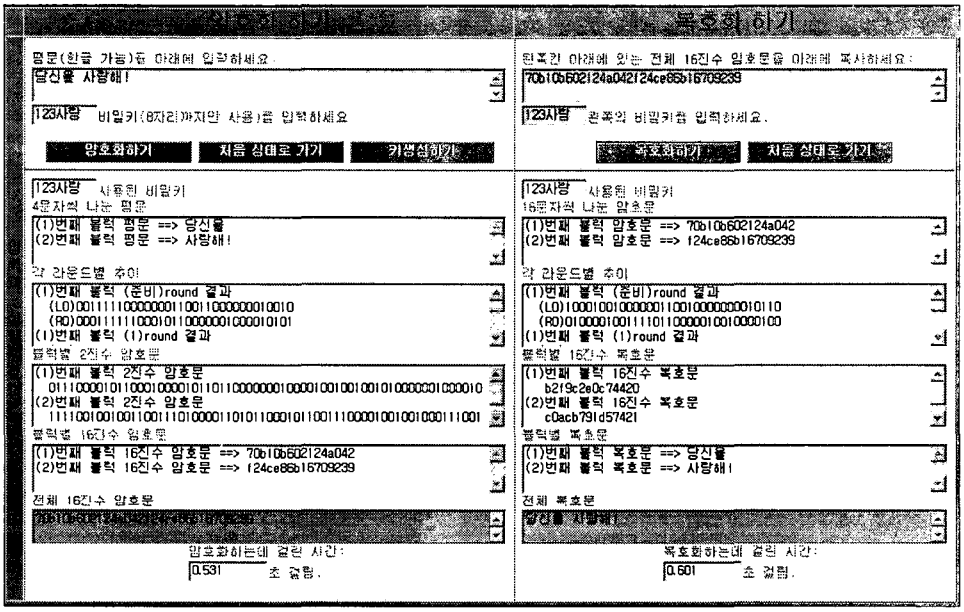

그림 7. 복호화하기

복호화하기 위해서는 DES암호화가 된 상태에서 16진수 암호문을 복사하여 그림 7의 오른쪽 위에 있는 노란 칸에 16진수 암호문을 복사하여 붙인 다음 **복호화하기** 를 누르면 암호화의 반대 과정인 복호화가 오른쪽 아래에 출력이 된다.

3. DES 코드

위의 알고리즘을 이용하여 코드를 작성한다. 알고리즘이 크게 세 부분으로 되어 있으므로 객체지향 프로그래밍(OOP)을 위해서 세 부분으로 나누어 코딩하면 편리하다고 생각된다. 이것은 전체를 관장하는 클래스, 즉 메인함수를 포함하는 클래스-이것을 DESCipher 클래스라 했다. Feistel 연산을 위한 클래스-이것을 Feistel 클래스라 했다. 키 생성을 위한 클래스-이것을 KeyMaker 클래스라 했다. 여기서는 KeyMaker 클래스의 코드를 살펴보자.

```
public class KeyMaker {
    static int[] PC1 = {57, 49, 41, 33, 25, 17,  9,
                         1, 58, 50, 42, 34, 26, 18,
                        10,  2, 59, 51, 43, 35, 27,
                        19, 11,  3, 60, 52, 44, 36,
                        63. 55. 47. 39. 31. 23. 15.
```

```
                        7, 62, 54, 46, 38, 30, 22,
                       14,  6, 61, 53, 45, 37, 29,
                       21, 13,  5, 28, 20, 12,  4}
static int[] PC2 = {14, 17, 11, 24,  1,  5,
                        3, 28, 15,  6, 21, 10,
                       23, 19, 12,  4, 26,  8,
                       16,  7, 27, 20, 13,  2,
                       41, 52, 31, 37, 47, 55,
                       30, 40, 51, 45, 33, 48,
                       44, 49, 39, 56, 34, 53,
                       46, 42, 50, 36, 29, 32};
static byte[] key1 = new byte[64];
static byte[] key2 = new byte[56];
static byte[] block = new byte[56];
static byte[] temp2 = new byte[48];
static byte[] rightKey = new byte[28];
static byte[] leftKey = new byte[28];
static byte[][] keyList = new byte[16][48];
static byte tmp2 = 0;
static byte tmp1 = 0;
static int i, j, v, k, l, m=0;
static int digit[] = {128,64,32,16,8,4,2,1};
static byte[] pc1 (String key) {
    key1=Feistel.toBinary(key);
  for(i=0;i<56;i++)
  key2[i] = key1[PC1[i]-1];
  return key2; }
static byte[] shift (byte x[], int j) {
  byte[] tmp = new byte[2];
  tmp[0] = x[0];
  tmp[1] = x[1];
  if(j==1||j==2||j==9||j==16) v=1;
    else v=2;
  for(i=0;i<28;i++) {
      if(i+v<28) {
          x[i]=x[i+v];
          x[27]=tmp[v-1]; } }
  if(v==2) x[26]=tmp[0];
  return x; }
static byte[][] makeKey (byte a[]) {
  for(i=0;i<28;i++) {
```

```
        leftKey[i] = a[i];
    rightKey[i] = a[i+28]; }
for(k=0;k<16;k++) {
        leftKey = shift(leftKey, k+1);
    rightKey = shift(rightKey, k+1);
    for(l=0;l<28;l++) {
        block[l] = leftKey[l];
        block[l+28] = rightKey[l]; }
    for(m=0;m<48;m++) {
        temp2[m] = block[PC2[m]-1];
        keyList[k][m]=temp2[m]; } }
return keyList; } }
```

Ⅳ. 결론 및 향후 연구 방향

지금까지 DES의 알고리즘과 그의 구현에 대하여 살펴보았다. DES는 최초의 표준화된 암호시스템으로 최근까지 사용되어 왔다. DES 알고리즘은 기초적인 수학 개념인 치환(permutation)과 단사함수(one-to-one function) 만을 사용하여 고도의 복잡하고 안전한 암호방법을 제시한다. 암호화 알고리즘에는 여기서 언급한 DES 외에 다수가 있다. 이러한 알고리즘들도 대부분 쉬운 수학적 사실들에 바탕을 둔 것들이 많으므로 공부하고 구현해 보면 도움이 될 것으로 생각된다[5].

오늘날 정보의 중요성은 아무리 강조해도 지나치지 않다. 이를 위하여 암호화 방법 또한 현재 매우 발전하고 있는 분야 중 하나이다. 우리나라도 IT 선진국으로서 그 위상을 유지하기 위해서는 안전한 암호법의 개발 및 그의 활용이 전제가 되어야 한다. 수학은 기초학문으로써 뿐만 아니라 응용학문으로써 정보 분야 발전의 밑거름이 되어야 한다.

오늘날 하드웨어의 발전으로 계산속도가 빨라짐에 따라 64비트의 키를 사용하는 DES도 더 이상 안전하지 않게 되었다. DES의 키 문제를 해결하기 위하여 2000년 10월 2일에 128비트의 키를 사용하는 AES가 발표되었다[6]. AES의 공모 과정에서 많은 알고리즘들이 제안되었고 그중에서 우리나라 연구진이 제안한 알고리즘도 끝까지 선전하였다. 향후 연구로는 AES의 알고리즘과 이의 구현이다. DES에서와 마찬가지로 AES에서도 한글의 구현이 목표이다.

참 고 문 헌

서광석 외 3인 (1998). 수론과 암호학, 경문사..
이민섭 (2000). 현대암호학, 교우사.
장은성 (1999). 네트워크 사회의 에티켓 암호학, 전파과학사.
정상조. http://math88.com.ne.kr/crypto.htm.

---

5) http://math88.com.ne.kr/crypto.htm
6) http://www.nist.gov/aes

AES. http://www.nist.gov/aes.

KISA. http://www.kisa.or.kr.

National Bureau of Standards (NBS), Data Encryption Standard (DES), Federal Information Processing Standards (FIPS) Publication 46. (1997).

RSA. http://www.rsasecurity.com.

C. E. Shannon (1948). A mathematical theory of communication, Bell Systems Technical Journal, 27, 379-423, 623-656.

C. E. Shannon (1949). Communication theory of secrecy systems, Bell Systems Technical Journal, 28, 656-715.

D. R. Stinson (1995). Cryptography-Theory and Practice, CRC Press.

DES Algorithm and its Implementation in School Mathematics Education

Chung, Sang-Cho[1)] · Park, Joong-Soo[2)]

Abstract

DES is a very simple crytosystem that uses only permutation in mathematics. Recently AES is standardized based on DES. In this paper we introduce DES and its implementation. In particular, we tried to process Hangul in DES. This paper may be used in school mathematics education.

Key words : Permutation, Plain text, Cipher text, Encryption, Decryption, DES, AES

1) Department of Mathematics, Chungnam Univeristy, math88@dreamwiz,com
2) Department of Mathematics, Woosuk Univeristy, jspark@woosukac.kr