2000 년 한국정보처리학회 추계 학술발표논문집 제 7 권 제 2 호

# 정보보호를 위한 암호시스템에 관한 연구

서장원*, 전문석*
*숭실대학교 컴퓨터학부
e-mail : jwsuh@duck.snut.ac.kr

# A Study on Cipher System for Information Security

Jang-Won Suh*, Moon-Seog Jun*
*School of Computing, Soong-Sil University

요 약

최근, 네트워크의 발달로 인해 인터넷을 기반으로 하는 전자상거래가 활성화되고 있는 추세이다. 이러한 전자상거래는 여러 가지 장점에도 불구하고 개방형 네트워크의 특성상 거래 전반에 걸친 정보들이 자칫 노출될 수 있다는 문제점을 지니고 있다. 따라서, 이와 관련하여 정보보호 문제를 해결할 수 있는 안전한 장치가 필요한데, 이런 장치들 중에 하나로 제시되고 있는 것이 암호 알고리즘을 이용한 암호시스템이다.

본 논문에서 제안한 NC 암호 알고리즘은 이러한 전자상거래 상의 정보보호에 적합한 암호시스템으로서, 입/출력 및 암호키의 크기를 각각 128 비트로 구성하였고 F 함수 내부의 구조가 64 비트의 서브키와 2 개의 S-Box, 그리고 16 라운드로 전개되도록 설계하였다. 또한, 암호화에 민감한 영향을 미치는 키 스케쥴링 알고리즘을 복잡하게 설계함으로써 계산 복잡도의 증가를 도모하였다.

## 1. 서론

최근 들어, 네트워크와 컴퓨터 통신이 발전하면서 일상적으로 행하던 일부 거래가 컴퓨터 네트워크를 이용한 전자 공간에서 이루어지고 있다. 특히, 인터넷의 급속한 확산에 따라 전자상거래도 역시 급속히 확산되고 있는 실정이다. 이에 따라, 전자상거래를 보다 활성화하기 위해서는 전자상거래의 신뢰성을 확보할 수 있는 보안 기술이 필수 선결 요소라 하겠다.

이에 본 논문에서는 안전하고 효율적인 NC 암호 알고리즘을 제안함으로써 정보보호 상의 문제점을 해결하여 전자상거래 상의 거래 정보나 거래 내용들의 안전성과 신뢰성을 도모하도록 하였다.

본 논문에서 제안한 NC 암호 알고리즘은 비선형 변환의 적절한 조합에 의해 설계됐으며, 전체 구조는 데이터 블록의 좌· 우측에 교대로 비선형 변환을 적용시키는 전형적인 Feistel 구조를 적용하여 설계하였다[3]. 이를 바탕으로 암호문의 생성 속도가 빠르게 진행되고, 해킹이나 침입 등의 외부 공격에도 대처할 수 있도록 설계하였다.

## 2. 전자상거래 상의 정보보호 문제

전자상거래라 함은 통합적으로 자동화된 정보체계 환경 하에서 거래 당사자간의 정보교환, 구매, 대금지불, 전달, 서비스 등의 제반 비즈니스를 네트워크를 통해 전자적으로 행하는 것으로 정의할 수 있다[1].

인터넷을 기반으로 하는 전자상거래는 각종 정보들이 인터넷을 통해 손쉽게 상대방에게 전달됨으로써 다수의 정보들을 제 3 자가 인터넷을 통해 손쉽게 접근할 수 있게 되었다. 더욱이 현재의 웹 기술의 발달에 따라 일반 사용자들도 정보 접근이 용이해짐으로써 인증이나 개인정보 보호 등의 전자상거래 보안 문제는 더욱 중요한 과제로 대두되고 있는 실정이다.

따라서, 인터넷상의 정보보호 문제를 포함한 보안 문제를 해결하기 위한 제반 장치가 마련되지 않는다면 전자상거래의 안전성과 신뢰성을 기대할 수 없다. 이러한 정보보호 문제를 해결하기 위한 방법 중의 하나가 네트워크 상에서 송/수신 메시지나 거래 내용에 관한 정보를 암호화하여 사용하는 암호 기술이다[2].

암호 기술은 암호키의 운용에 따라 대칭키 암호시스템과 공개키 암호시스템으로 크게 구별된다. 대칭키 암호시스템에서는 송/수신자가 동일한 비밀키를 공유해야 한다. 이에 비해 공개키 암호시스템은 암/복호화키가 서로 다르며, 어느 한 키를 공개하더라도 대응되는 다른 키를 유도해 내는 것은 계산상 불가능하도록 설계되어진다[4].

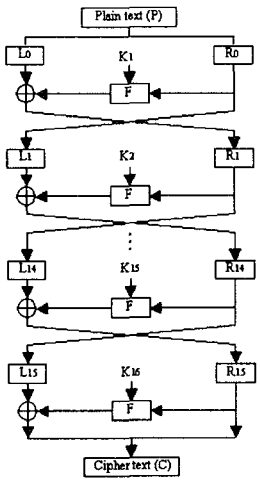
이 중 대칭키 암호시스템은 메시지 처리 형식에 따라 블록 암호 알고리즘과 스트림 암호 알고리즘으로 나누어진다. 블록 암호 알고리즘은 고정된 크기의 입력 블록이 암호키에 의해 고정된 크기의 출력 블록으로 변형되는 암호 알고리즘으로써, 출력 비트의 각 비트는 입력 블록과 암호키의 모든 비트에 영향을 받아 결정된다[5]. 이에 비해 스트림 암호 알고리즘은 선형 쉬프트 레지스터를 이용한 이진 수열 발생기를 사용하는 암호 알고리즘의 형태로서, 최초 평문을 이진 수열로 부호화 한 뒤, 이를 이진 수열 발생기에서 생성된 이진 수열과 XOR 하여 이진 수열로 된 암호문을 생성하는 방식이다.

## 3. NC 암호 알고리즘의 설계

### 3.1 전체 구성

본 논문에서 제안한 NC 암호 알고리즘은 대칭키 암호 알고리즘에 바탕을 둔 128 비트 블록 암호 알고리즘으로, Feistel 구조를 사용하여 설계되었다. 또한, 입· 출력과 암호키의 크기가 128 비트이고 내부 함수 F 내에서는 64 비트의 서브키와 16 라운드를 수행하며, 2 개의 S-Box $S_1$ 과 $S_2$ 를 사용하였다.

NC 암호 알고리즘의 전체 구조는 (그림 1)과 같다.




(그림 1) NC 암호 알고리즘의 전체 구조

이런 구조는 메시지 처리 방식에 있어 128 비트의 평문 블록 단위 당 128 비트 암호키로부터 치환되어 생성된 64 비트 서브키(16 개)를 입력으로 받아 16 라운드를 수행한 후 128 비트의 암호문 블록을 출력한다.

### 3.2 키 스케쥴링 알고리즘

암호 알고리즘의 내부 함수 F 에서 사용되는 서브키를 생성하기 위한 키 스케쥴링 알고리즘의 가장 중요한 설계 목표는 안전성을 보장하는 것이다.

각 라운드에서 F 함수 내부의 라운드 서브키를 생성하기 위한 키 스케쥴링 알고리즘은 128 비트 평문 블록을 두 개의 64 비트 블록으로 분할하고, 이를 다시 좌/우측 각 2 개의 32 비트 키 값 A, B, C, D 로 분할하고 이를 이용하여 서브키를 생성한다. 이를 위해 임의의 128 비트 키를 재배열하기 위해 <표 1>의 PC1 테이블을 이용한다.

<표 1> PC1 테이블

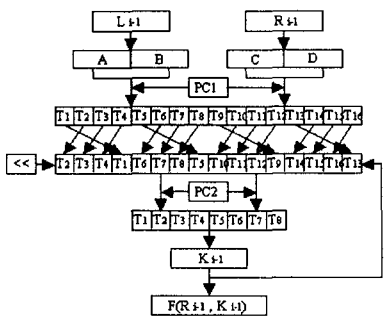
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 121 | 113 | 105 | 97 | 89 | 81 | 73 | 65 |
| 57 | 49 | 41 | 33 | 25 | 17 | 9 | 1 |
| 66 | 122 | 114 | 106 | 98 | 90 | 82 | 74 |
| 2 | 58 | 50 | 42 | 34 | 26 | 18 | 10 |
| 75 | 67 | 123 | 115 | 107 | 99 | 91 | 83 |
| 11 | 3 | 59 | 51 | 43 | 35 | 27 | 19 |
| 84 | 76 | 68 | 124 | 116 | 108 | 100 | 92 |
| 20 | 12 | 4 | 60 | 52 | 44 | 36 | 28 |
| 125 | 120 | 112 | 104 | 96 | 88 | 80 | 72 |
| 64 | 56 | 48 | 40 | 32 | 24 | 16 | 8 |
| 71 | 127 | 119 | 111 | 103 | 95 | 87 | 79 |
| 7 | 63 | 55 | 47 | 39 | 31 | 23 | 15 |
| 78 | 70 | 126 | 118 | 110 | 102 | 94 | 86 |
| 14 | 6 | 62 | 54 | 46 | 38 | 30 | 22 |
| 85 | 77 | 69 | 125 | 117 | 109 | 101 | 93 |
| 21 | 13 | 5 | 61 | 53 | 45 | 37 | 29 |

여기서, $n$ 라운드에서의 서브키는 $n$-1 라운드에서 생성된 서브키에 영향을 받도록 설계했다. 이를 위해 $n$-1 라운드에서 생성된 서브키를 1 비트(1,2,7,8,9,10,15,16 라운드) 또는 3 비트(3,4,5,6,11,12,13,14 라운드)씩 각 라운드별로 좌측 회전이동 한 후, $n$ 라운드의 서브키로 적용하였다. 그런 후에, 128 비트 키를 64 비트로 치환하여 산출하기 위해 <표 2>와 같은 PC2 테이블을 이용한다.

<표 2> PC2 테이블

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 31 | 1 | 23 | 9 | 63 | 33 | 66 | 41 |
| 29 | 3 | 21 | 11 | 61 | 35 | 63 | 43 |
| 27 | 6 | 19 | 13 | 69 | 37 | 61 | 45 |
| 26 | 7 | 17 | 16 | 67 | 39 | 49 | 47 |
| 96 | 86 | 87 | 73 | 127 | 97 | 119 | 106 |
| 93 | 87 | 86 | 76 | 126 | 99 | 117 | 107 |
| 91 | 89 | 83 | 77 | 123 | 101 | 116 | 109 |
| 89 | 71 | 81 | 79 | 121 | 103 | 113 | 111 |

다음의 (그림 2)는 NC 암호 알고리즘에 대한 키 스케쥴링 알고리즘의 구조를 나타낸 것이다.




(그림 2) 키 스케쥴링 알고리즘의 구조

(그림 2)에서, 평문의 우측 키 값과 함께 내부 함수 F 의 입력 키 값으로 사용되는 64 비트 서브키 생성을 위한 키 스케쥴링 알고리즘 구조는 기존의 Feistel 구조와는 달리 내부 함수 F 의 출력 결과 값이 다음 라

운드에 영향을 미치면서 동시에 입력 블록을 변화시키기 때문에 데이터의 복잡도를 빠르게 증가시킬 수 있고, 또한 암/복호화가 빠르게 처리된다는 관점에서 좋은 키 스케쥴링 구조라 할 수 있다.

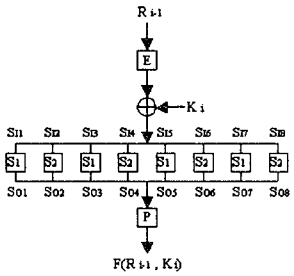
### 3.3 S-Box 적용

대개의 블록 암호 알고리즘에서 서브키의 입/출력과 관련하여 내부 함수 F 내에서 비선형 대입 연산을 수행하는데 사용되는 S-Box 는 비선형 함수로서 입력 크기와 출력 크기를 변경할 수 있다. 이러한 S-Box 는 암호화에 민감한 영향을 미치므로 S-Box 의 구성을 어떻게 하느냐에 따라 견고한 암호 알고리즘을 구축할 수 있다. 결국, 암호화의 핵심은 견고한 S-Box 의 새로운 설계나 또는 실험을 통해 검증된 S-Box 를 적용하는 것이다.

제안한 NC 알고리즘에서는 2 개의 8× 8 비트 S-Box $S_1$ 과 $S_2$ 를 사용하였으며, 각각의 S-Box 는 NC 알고리즘의 키 크기에 의존한다. 즉, 암/복호화 과정을 어렵게 하기 위해 각 라운드간에 키-의존 관계를 갖도록 설계되어 입증된 2 개의 S-Box 를 사용하였다[6].

### 3.4 F 함수 구축

제안한 NC 암호 알고리즘에서 128 비트 키를 통한 암/복호화를 위한 근본적인 블록 구축은 평문의 우측 비트 값 64 비트와 키 생성 알고리즘에 의해 생성되는 각 라운드의 서브키 64 비트를 입력 비트 값으로 하여 그것들의 비트 값을 XOR 함으로서 출력 비트 값을 산출하는 내부 함수 F 내에서 이루어진다. 다음의 (그림 3)은 NC 알고리즘에서의 내부 함수 F 의 구조를 나타낸 것이다.




(그림 3) F 함수의 구조

이런 F 함수의 구조는 입력 문자열과 출력 문자열이 사상되는 키-의존 관계를 갖고 있다. 다시 말해서, 키 스케쥴링 알고리즘에 의해 생성된 64 비트의 서브키와 2 개의 각 8 비트 S-Box $S_1$, $S_2$를 이용하여 데이터 우측면에서 수행된다. 내부 함수 F 는 2 개의 S-Box, 비트 수열, 수학적 연산 그리고 XOR 에 근거하여 수행된다.

$$F: \{0,1\}^{n/2} * \{0,1\}^{N} \rightarrow \{0,1\}^{n/2}$$

여기서 $n$ 은 NC 구조의 블록 크기이고 F 는 입력처럼 블록의 $n/2$ 비트와 키의 $N$ 비트를 취하고 $n/2$ 비트 길이의 출력을 산출하는 내부 함수이다. 각 라운드에서 원본 블록은 내부 함수 F 로의 입력이고 내부 함수 F 의 출력은 그것들의 2 개 블록이 다음 라운드를 위해 교체된 후에 목표 블록과 XOR 된다. 따라서, 전형적인 Feistel 구조를 갖는 NC 암호 알고리즘은 내부 함수 F 의 특성에 따라 입/출력을 구분할 수 있다.

내부 함수 F 내에서의 처리 과정을 보다 세부적으로 서술하면 다음과 같다.

- 내부 함수 F 는 첫 번째로 우측 평문 값 64 비트 블록을 입력으로 받아 외부 공격에 따른 암호 해독을 복잡하게 하기 위해 블록 내의 위치를 재정렬하는 <표 3>의 E 테이블을 통해 치환된 64 비트와 <표 1>과 <표 2>의 PC1 테이블, PC2 테이블에 의해 생성된 서브키 $Ki$ 를 XOR 한다.

$$E(R_{i-1}) \text{ XOR } b_1b_2b_3b_4b_5b_6b_7b_8$$

<표 3> E 테이블

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 7 | 5 | 3 | 1 | 2 | 4 | 6 | 8 |
| 15 | 13 | 11 | 9 | 10 | 12 | 14 | 16 |
| 23 | 21 | 19 | 17 | 18 | 20 | 22 | 24 |
| 31 | 29 | 27 | 25 | 26 | 28 | 30 | 32 |
| 39 | 37 | 35 | 33 | 34 | 36 | 38 | 40 |
| 47 | 45 | 43 | 41 | 42 | 44 | 46 | 48 |
| 55 | 53 | 51 | 49 | 50 | 52 | 54 | 56 |
| 63 | 61 | 59 | 57 | 58 | 60 | 62 | 64 |

- XOR 이후에 산출된 8 비트 블록 $B_i = b_1b_2b_3b_4b_5b_6b_7b_8$ 는 2 개의 S-Box $S_1$ 과 $S_2$ 에 적용되어 $(S_1(B_1)\ S_2(B_2)S_1(B_3)S_2(B_4)S_1(B_5)S_2(B_6)S_1(B_7)S_2(B_8))$ 순서로 8 비트 블록 $Si(Bi)$로 출력되고, 모든 블록들은 계산 복잡도를 증가시키기 위해 블록의 위치를 재정렬하는 <표 4>의 P 테이블에 의해 치환된다.

$$P(S_1(B_1)S_2(B_2)S_1(B_3)S_2(B_4)S_1(B_5)S_2(B_6)S_1(B_7)S_2(B_8))$$

<표 4> P 테이블

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 30 | 5 | 35 | 63 | 52 | 13 | 41 | 17 |
| 1 | 16 | 29 | 40 | 42 | 54 | 22 | 58 |
| 47 | 24 | 26 | 61 | 9 | 3 | 33 | 55 |
| 49 | 60 | 27 | 19 | 10 | 44 | 38 | 8 |
| 14 | 64 | 53 | 32 | 23 | 43 | 6 | 37 |
| 15 | 62 | 25 | 45 | 4 | 21 | 39 | 50 |
| 2 | 12 | 57 | 34 | 20 | 51 | 28 | 45 |
| 59 | 31 | 7 | 55 | 18 | 48 | 36 | 11 |

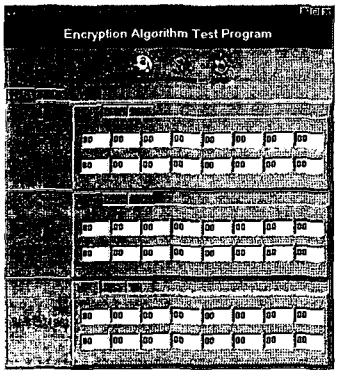
- 위의 절차에 따라 64 비트 블록 $F(R_{i-1}, K_i)$가 출력되며, 최종적으로 $L_{i-1}$ XOR $F(R_{i-1}, Ki)$는 다음 라운드의 우측 입력 비트로 이동하고, 이런 과정을 $i$ 라운드 만큼 수행한 후에 최종 암호문 쌍을 출력한다.

## 4. NC 암호 알고리즘의 구현

### 4.1 구현 환경 및 초기화면 구성

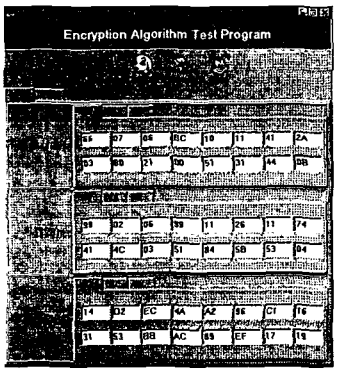
NC 암호 알고리즘 구현을 위한 환경은 다음과 같다.

- 시스템 : Pentium Pro 200MHz
- 메모리 : 64MB
- OS : Windows 98
- 컴파일러 : Borland C++ Builder 4.0

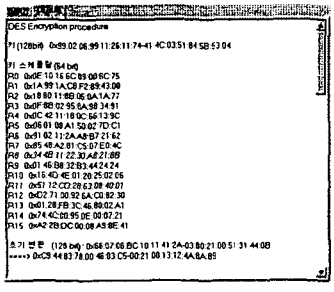



### 4.2 암호화 과정

NC 알고리즘에서 암호화 과정은 프로그램 초기화 면에서 "평문"을 클릭하여 임의의 128 비트 평문을 생성하고, "키"를 클릭하여 임의의 128 비트 암호키를 생성한다. 생성된 128 비트 암호키는 본 논문에서 제안한 키 스케쥴링 알고리즘의 과정을 거쳐 16 개의 64 비트 서브키를 산출한다. 그런 후에, "Encoding" 버튼을 클릭하여 최종적으로 다음과 같은 128 비트 암호문을 산출한다.




또한, 구현한 프로그램에서 임의의 128 비트 입력키에 대해 키 스케쥴링 알고리즘에 의해 생성된 16 개 64 비트 서브키의 목록에 관한 사항은 "DEBUG" 탭을 이용하여 확인할 수가 있다.




그리고, 수행되는 라운드 수의 조정과 변환 기능, 그리고 설정된 최종 라운드에서의 swap 여부 등을 컨트롤할 수 있게 구현하였다. 이것은 프로그램에 의해 산출된 128 비트의 암호문을 해독하는 과정에 있어서 전수조사 공격(exhaustive attack)이나 차분 공격(differential attack) 등으로 암호를 해독할 때, 수작업으로 3 라운드 정도는 확인해 볼 수 있다.

본 논문에서는 복호화 과정에 대해서는 생략하는데, 복호화 과정 역시 암호화 과정에서 산출된 암호문과 동일한 암호키를 사용하여 "Decoding"을 클릭하면 평문을 산출할 수 있다.

## 5. 결론

본 논문에서 제안한 NC 암호 알고리즘은 전자상거래 상의 정보보호에 적합하도록 안전성과 효율성을 고려하여 설계하였다. 또한, 일반적으로 암호화에 있어서 영향을 미치는 부분이 바로 암호키의 연산에 따라 이에 대응되는 키 비트 값이 결정되는 S-Box 와 서브키를 생성하기 위한 키 스케쥴링인데, 이를 위해 검증된 두 개의 S-Box $S_1$ 과 $S_2$ 를 사용하였고, 외부 공격에 의해 암호키가 쉽게 발견되지 않도록 키 스케쥴링 알고리즘을 설계하였다.

NC 암호 알고리즘은 민간분야의 전자상거래 상에서의 안전성과 신뢰성을 보장하기 위한 목적으로 개발했으며, 개인 및 기업의 중요 정보보호나 전자우편 시스템에서의 메시지 암호화, 그리고 인터넷을 이용한 전자지불 시스템 등에 적합하게 사용될 수 있다. 아울러, 가상교육 시스템이나 한정 수신 시스템(CAS) 등에 유용하게 사용될 수 있을 것이라 예측된다.

향후, NC 알고리즘의 보안 레벨을 좀 더 향상시키기 위한 방법의 하나로 키 스케쥴링 알고리즘을 좀 더 보완하고, 동일한 구조로 설계된 다른 형태의 S-Box 를 적용시켜 봄으로서 그것을 분석하고, 내부 함수 F 내부에 새로운 함수를 하나 더 추가하여 계산 복잡도를 증가시킴으로써 더욱 견고한 암호 알고리즘이 구축되리라고 생각된다.

### 참고문헌

[1] A.Froomkin, "The Essential Role of Trusted Third Parties in Electronic Commerce", Ver. 1.02 Oct. 1996.

[2] H.Sun, "Computer and Network Security", Lecture by Rivest of MIT, http://theory.lcs.mit.edu

[3] H.Feistel, "Cryptography and Computer Privacy", Scientific American, V. 228, N. 5, May 1973.

[4] W.Maga, "A High Performance Encryption Algorithm", Computer Security:A Global Challenge, Elsevier Science Publishers, 1984.

[5] J.Daemen, L.Knudsen, V.Rijmen, "The Block Cipher Square", Software Encrytion, 4th International Workshop Procdddings, Springer-Verlag, 1997.

[6] NIST, "Announcing Development of a Federal Information Standard for Advanced Encryption Standard", Federal Register, Vol. 62, No.1, Jan. 1997.