The Journal of Korean
Institute of CALS/EC
Vol. 6, No. 2, August 2001

# 전자상거래를 위한 NPC 암호 알고리즘 설계

김재각*, 전문석**

# The Design of NPC Encryption Algorithms for Electronic Commerce

Chae-Kak Kim , Moon-Seog Jun

## Abstract

EC(Electronic Commerce) is increasing with high speed based on the expansion of Internet. EC which is done through Internet has strong point like independence from time and space. On the contrary, it also has weak point like security problem because anybody can access easily to the system due to open network attribute of Internet. Therefore, we need the solutions that protect the security problem for safe and useful EC activity. One of these solutions is the implementation of strong cipher algorithm.

NPC(Non-Polynomial Complete) cipher algorithm proposed in this paper is good for the security and it overcome the limit of current 64bits cipher algorithm using 256bits key for input, output and encryption key. Moreover, it is designed for the increase of calculation complexity and probability calculation by adapting more complex design for subkey generation regarded as one of important element effected to encryption.

* 김포대학 컴퓨터계열 조교수
** 숭실대학교 컴퓨터학부 부교수

## 1. 서 론

인터넷을 이용한 전자상거래 규모가 급속히 확대 될 것으로 전망됨에 따라 인터넷 접속에 있어서 보안문제는 심각히 고려되어야 할 문제 중의 하나로 떠오르고 있으며, 이에 대한 충분한 신뢰성을 제공하지 못할 때 인터넷을 이용한 상거래의 기반조차 위협할 수 있는 문제로 대두될 수 있을 것으로 보인다. 따라서 본 논문에서는 인터넷상에서 데이터의 안전성과 효율성을 제공 할 수 있는 암호화 알고리즘을 설계하고, 암호 모듈에 대한 안정성을 분석하였으며, 기존의 암호 알고리즘과 성능을 비교 분석하였다.

본 논문의 구성은 다음과 같다. 2장에서는 제안하는 NPC 암호 알고리즘을 설계하고, 3장에서는 제안하는 NPC 암호 알고리즘에 대한 안전성을 분석하고, 4장에서는 제안하는 암호 알고리즘과 기존의 암호 알고리즘을 비교 및 평가 결과를 기술하였다. 그리고 5장에서는 이에 대한 결론을 제시하였다.

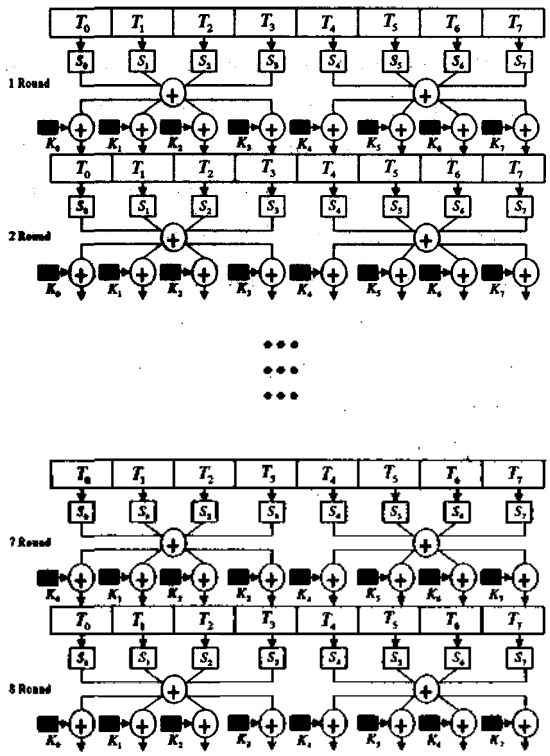
## 2. NPC 암호 알고리즘의 설계

### 2.1 NPC 암호화 알고리즘 구조

제안하는 NPC 암호 알고리즘은 대칭키 암호 알고리즘에 바탕을 둔 개념으로써, 블록 단위로 메시지를 처리하는 블록 암호 알고리즘이다. NPC 암호화 알고리즘에서 블록 크기와 키 길이는 256-비트를 사용하며, 반복적인 라운드 변환 함수는 각각에 대한 변환 변수를 갖고 수행되어진다. 암호화의 비도를 향상시키기 위해 암호화 알고리즘은 선형함수와 비선형 함수로 구성되어져 수행되어진다. 선형함수는 키 합산을 하는데 이용되며, 비선형 함수는 XOR연산과 8번의 반복적인 함수에서 S-Box를 이용하여 8개의 분할된 32-비트 크기의 입력 평문을 갖고 수행된다.

NPC 암호화 알고리즘은 다음과 같은 설계 기준들에 맞게 설계되었다.

- 성격 : 256 비트 블록 암호화 알고리즘이다.
- 구조 : Non-Feistel 구조를 기초로 한다.
- 입·출력문의 크기 : 256-비트를 사용한다.
- 입력키의 크기 : 256-비트 키를 사용한다.




<그림 1> NPC 암호화 알고리즘의 구조도

· 라운드 수 : 8 Round를 사용한다.
· 안전성 : 차분 해독 및 선형 해독에 대하여 안전하게 설계한다.

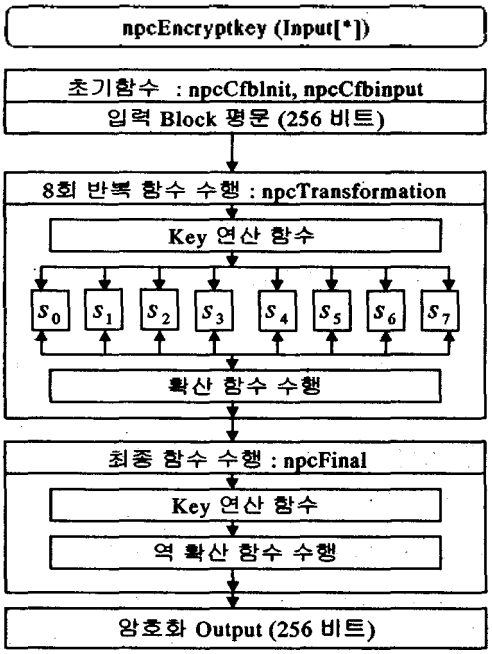
## 2.2 NPC 암호화 및 복호화 알고리즘

### 2.2.1 암호화 알고리즘

INPUT : R = 8회;
입력 블록 length = 256 bit;
256-bit 키값 key
= $K_1, K_2, K_3, \ldots, k_{128} \ldots, K_{256}$;
평문블록(text[0],text[1], ...,text[7]) ←
( $m_1 \ldots m_{32}, m_{33} \ldots m_{64}, \ldots, m_{224} \ldots m_{256}$)
OUTPUT: 256-bit 암호문 output
= output[0], output[1], ...., output[7];
① 초기 Key값 생성: npcCfbinit 함수;
② npcCfbEncrypt: Input[*]
← npcCfbInput;. (초기 입력값 생성,
*: 256-비트크기)
③ While (lenth≥ BLOCKSIZE)
(256-bit 암호화 수행)
(a) data[*]← $npcEncrypt_{RKey}$(Input[*]);
(8개 Substitution-box을 수행)
(b) For 0≤ i < R:
(32-bit 8개 분할단위 수행)
(c) output[i] ←data[i] ⊕ output[i];
(d) output = BlOCKSIZE+16 bytes;
(e) length=BLOCKXSIZE- 16 bytes;
④ 최종 clear 단계: npcCfbFinal;

### 2.2.2 복호화 알고리즘

INPUT : R = 8회;
입력 블록 length = 256 bit;
256-bit키값 key
= $K_1, K_2, K_3 \ldots, K_{128}, \ldots, K_{256}$;
암호문블록(output[0],output[1],..,output[7])
←
( $C_1 \ldots C_{32}, C_{33} \ldots C_{64}, \ldots, C_{224} \ldots C_{256}$)
OUTPUT :
256-bit평문
( $m_1 \ldots m_{32}, m_{33} \ldots m_{64}, \ldots, m_{224} \ldots m_{256}$)
← (output[0],output[1],...,output[7]);
① 초기 Key 값 생성: npcCfbinit 함수;
② npcAlgmEncrypt: Input[*]
← npcAlgmInput;
(초기 입력값 생성, *: 256-비트크기)
③ While (lenth ≥ BLOCKSIZE)
(256-bit 암호화 수행)
(a) data[*] ← $npcEncrypt_{RKey}$(Input[*]);
(8개 Substitution-box을 수행)
(b) For 0≤ i < R:
(32-bit 8개 분할단위 수행)
(c) output[i] ← data[i] ⊕ output[i];
(d) output = BLOCKSIZE + 16 bytes;
(e) length = BLOCKSIZE - 16 bytes;
④ 최종 clear 단계: npcCfbFinal;




<그림 2> 암호모듈의 구현 및 동작 형태

### 2.3 Key 생성/관리

입출력 분할과 키의 분할크기는 32-비트형태로 8회 반복으로 생성된다. Key-Partition은 (t-1)반복에서 키 값을 생성 저장하기 위한 함수이며, S-box를 이용하는 비선형 생성방법을 이용함으로서 공격으로부터 보호하기 위한 함수이다.

키 분할 함수 합산은 선형 생성 방법으로 Bitwise 덧셈을 수행한다.

#### 2.3.1 키생성 구성도

For $0 \le t \le 7$

/* 8회 반복 */

$K_0^t$

$= K_0^{t-1} \oplus \phi(K_3^{t-1}, 8) \oplus offset[t-1]$

$K_1^t = K_1^{t-1} \oplus K_0^t$

: :

$K_7^t = K_7^{t-1} \oplus K_0^t$

Key-partition( $K^{t-1}$ )

end-For

#### 2.3.2 키 생성 순서

각 라운드에 사용되는 키는 다음과 같은 방식으로 생성되어진다.

① 256 비트 입력키를 32 비트 8개의 부분으로 나눈다.

② t를 하나씩 증가시키며,

(256-비트 값 key= $K_1, K_2, K_3, \ldots, K_{128}, \ldots, K_{256}$생성)

③ RKeys_e[t][0]( $K_0^t$) = RKeys_e[t-1][0] ^F_ROTL (RKeys_e[t-1][7],8) ^offset[t-1]

④ RKeys_e[t][1] = RKeys_e[t-1][1] ^RKeys_e[t][0]

⑤ RKeys_e[t][2] = RKeys_e[t-1][2] ^RKeys_e[t][1]

⑥ RKeys_e[t][3] = RKeys_e[t-1][3] ^RKeys_e[t][2]

⑦ RKeys_e[t][4] = RKeys_e[t-1][4] ^RKeys_e[t][3]

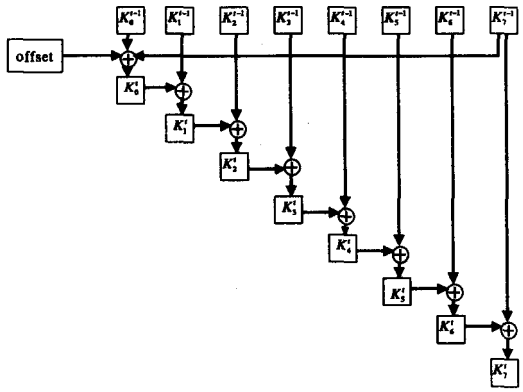
⑧ RKeys_e[t][5] =RKeys_e[t-1][5] ^RKeys_e[t][4]

⑨ RKeys_e[t][6] = RKeys_e[t-1][6] ^RKeys_e[t][5]

⑩ RKeys_e[t][7] = RKeys_e[t-1][7] ^RKeys_e[t][6]

(라운드 키 생성)

⑪ ②에서 ⑩번 까지를 8번 반복 수행한다.




<그림 3> Key 생성 알고리즘 구현 Diagram

## 3. NPC 암호 알고리즘의 안전성 분석

### 3.1 암호 알고리즘의 안전성 분석

본 논문에서 제안하는 NPC 암호 알고리즘에 임의의 데이터 비트 및 필드 값을 입력하였을 때, 계산에 소요되는 시간을 측정하기 위한 분석을 실시하였다. 우선, 1MIPS 장치는 초당 $4\times10^4$번의 NPC 덧셈을 수행할 수 있다고 가정한다. 이 가정은, Field $F_{2^{163}}$상의 NPC 그룹연산을 수행하도록 설계되었으며, 40Mhz 클럭 속도를 갖는 ASIC (application-specific integrated circuit)이 초당 40,000번의 NPC 덧셈을 수행하고, 또한 25 MIPS의 SPARC IPC에 구현된 소프트웨어가 초당 2,000번의 NPC 덧셈을 수행하는 것으로 보아 타당성이 있다고 보여진다. 따라서, 1 MIPS의 장치가 1년 동안 수행할 수 있는 NPC 덧셈은 $(4\times 10^4)\cdot(60\times60\times24\times365)\approx 2^{40}$ 정도이다. [표 1]은 다양한 정수 n에 대해, Pollard rho-method를 사용하여 하나의 이산 대수를 계산하기 위해 필요한 계산능력(computing power)을 나타낸다.

예를 들어, 각각이 1,000 MIPS인 10,000개의 컴퓨터로 $n\approx 2^{160}$인 경우의 NPC 이산대수를 계산하기 위해서는 96,000년이 소요된다[3].

<표 1> NPC 대수 계산에 요구되어지는 계산량

| [illegible] | [illegible] | [illegible] | [illegible] |
|---|---|---|---|
| 163 | 160 | $2^{80}$ | $9.6\times10^{11}$ |
| 191 | 186 | $2^{93}$ | $7.9\times10^{15}$ |
| 239 | 234 | $2^{117}$ | $1.6\times10^{23}$ |
| 359 | 354 | $2^{177}$ | $1.5\times10^{41}$ |
| 431 | 426 | $2^{213}$ | $1.0\times10^{52}$ |

NPC 암호 알고리즘에 대한 좀 더 가능성 있는 공격방법은, Pollard rho-method를 이용하여 병렬검색을 수행하는 특별한 하드웨어를 고안하는 것이다. Van Oorschot와 Wiener은 이러한 가능성에 대해 상세한 연구결과를 제공한다. 1994년의 연구결과에서, $n\approx 10^{36}\approx 2^{120}$의 경우, 325,000개의 프로세서를 가진 장치에서 하나의 이산 대수를 계산하는데 약 35일이 걸릴 것으로 예측했다. 하지만, $n> 2^{160}$인 경우에는 이러한 방법도 실행 불가능하다[5].

## 4. NPC와 DES 알고리즘의 비교평가

개발된 NPC 암호화 알고리즘은 이산수학 문제로 키를 생성하기 때문에 평문을 암호화 할 때 걸리는 시간도 기존의 소인수분해 형태로 암호화를 생성했을 때보다 시간을 단축할 수 있어 네트워크의 부담을 줄일 수 있다. 제안된 NPC 암호화 알고리즘은 다양한 입력 데이터를 갖고 순환되는 결과 없이 통계적으로 추적 불가능한 암호화 단계와 복호화 단계를 거쳐서 완성된 알고리즘이다. 임의의 키 값을 넣고 프로그램을 실행시켰을 경우 NPC 알고리즘에 의해서 서로 다른 키 값에 의해 유효한 암호문을 생성한다. 또한, 임의의 평문을 넣고 프로그램을 실행시켰을 경우 NPC 알고리즘에 의해서 같은 키 값에 대해 다른 암호문을 생성한다.

NPC 암호화 알고리즘의 입출력 변화 공격법(DC: Diferential Cryptanlysis)과 선형 근사 공격법(LC: Linear Cryptanlysis)은 DES 알고리즘과 비교해서 비도(security level)와 시간 복잡도가 복잡한 결과를 시험 결과를 통해서 얻을 수 있다[6] <표 2>.

<표 2> DES와 NPC 알고리즘과의 DC/LC 비교

| | | | | | |
|---|---|---|---|---|---|
| 4 | $2^{-9.6}$ | $2^{-6}$ | 2 | $2^{-246}$ | $2^{-246}$ |
| 8 | $2^{-30.5}$ | $2^{-19.5}$ | 4 | $2^{-492}$ | $2^{-492}$ |
| 12 | $2^{-46.2}$ | $2^{-31.5}$ | 8 | $2^{-984}$ | $2^{-984}$ |
| 48 | $2^{-128}$ | $2^{-148}$ | 16 | $2^{-1968}$ | $2^{-1968}$ |

따라서 NPC 암호 알고리즘은 입출력 변화 공격과 선형 근사 공격에 안전하며, 암호화 키 크기나, 라운드의 수에서 64 비트 크기 4개로 구성된 256 비트 단위로 유연성 있는 차세대 암호화 알고리즘으로 이용 가능한 구조를 보여주고 있다.

DES는 기본적으로 16라운드로 구성되며, 암호화는 동일한 동작 과정의 반복으로 이루어지나, 제안하는 NPC 암호 알고리즘은 이산로그 문제(Discrete log problem)에 초점을 두고 있다[4].

그리고 무엇보다도 NPC 암호 알고리즘은 기존 다른 암호 알고리즘에 비해 보다 작은 키를 사용하면서 거의 비슷한 수준의 보안을 보장해 주고 있다. 수행속도에서 보면, 연산 시 곱셈을 연속적으로 사용하는 알고리즘 유형들이 수행시간이 길어지는 것에 반해, NPC 암호 알고리즘에서는 주요 연산이 덧셈이기 때문에 수행 시간을 많이 절약할 수 있다.

또한 NPC 암호시스템은 다른 공개키 암호시스템에 비해 높은 암호키-비트-당-보안수준(strength-per-key-bit)을 가능하게 한다. 본 암호화 모듈에서 사용한 키는 256 비트로써 1024 비트의 DES 보다 높은 수준의 보안레벨을 제공한다. 이렇게 암호키 크기가 작으면서 동등한 수준의 보안레벨을 제공함으로써, 더 작은 크기의 시스템변수 및 공개키 인증서를 사용할 수 있으며, 네트워크 대역폭을 절약할 수 있다. 또한 실행시간을 절약할 수 있으며, 더 작은 하드웨어 프로세서를 구현가능 하게 한다.

## 5. 결 론

인터넷을 이용한 전자상거래 서비스가 급속히 확대 될 것으로 전망됨에 따라 인터넷 접속에 있어서 보안문제는 시급히 개선되어야 할 부분으로 떠오르고 있다. 원활한 전자상거래 서비스의 확대를 위해서는 이를 뒷받침 해줄 수 있는 성능과 안전성이 고려된 효율적인 암호 알고리즘의 필요성이 대두된다. 본 논문에서 제안한 NPC 암호화 알고리즘은 이러한 점을 감안하여 기존의 암호화 알고리즘에 비해 충분한 안전성을 제공하면서 더 빠른 성능을 제공할 수 있도록 설계하였다. 제안된 NPC 암호화 알고리즘은 암호화 키 크기나, 라운드의 수에서 64비트 크기 4개로 구성된 256 비트 단위로 유연성 있는 차세대 암호화 알고리즘으로 이용 가능한 구조를 보여주고 있으며, 인터넷을 사용하는 전자상거래 상에서의 안전성과 신뢰성을 제공하기 때문에 개인 및 기업의 중요정보 보호나 전자우편 시스템에서의 메시지 암호화, 그리고 전자화폐나 전자지불 시스템에 유용하게 사용될 수 있을 것으로 보인다[2].

## 참고문헌


[1] B.A. LaMacchia and A.M. Odlyzko. "Computation of discrete logarithms in prime fields", Designs, Codes and Cryptography, volume 1, pages 47-62, 1991

[2] C.P. Schnorr, "Efficient signature generation by smart cards", Journal of Cryptology, volume 4, pages 161-174, 1991.

[3] G. Agnew, R Mullin and S. Vanstone, "An implementation of elliptic curve cryptosystems over $F_{2^{155}}$", IEEE Journal on selected Areas in communications, volume 11, pages 804-813, 1993.

[4] K. Nyberg and R. Rueppel, "Messages recovery for signature schemes based on the discrete logaritm problem:, Designs, Codes and Cryptography, volumn 7, pages 61-81, 1996

[5] M. Gardner, "A new kind of cipher that would take millions of years to break", Scientific American, volume 237, pages 120-124, August 1997.

[6] M. Mastsui, "Linear Cryptoanalysis Method for DES Cipher", Advances in Cryptology : Proc. of Eurocrypt '93, Springer-Verlag, pages 386-394, 1996

[7] S. Pohlig and M. Hellman, " An improved algorithm for computing logarithms over GF(p) and its cryptographic significances", IEEE Transactions on Information Theory, volumn 24, pages 106-924, 1978

[8] S. Lucks, "Attacking Seven Rounds of Rijndael under 192-bit and 256-bit keys", Third AES Candidate Conference, Apr 2000.

## 저자 소개

**김 계 각**(ckkim@kimpo.ac.kr)

1981년 숭실대학교 전자계산학과(공학사)
1985년 연세대학교 산업대학원 전산전공(공학석사)
1998년~ 숭실대학교 대학원 박사과정
1985년~1994년 LG전자, 삼보컴퓨터 근무
1996년~현재 김포대학 컴퓨터계열 조교수
관심분야 : 인터넷 보안, 암호학, OFDM,

**전 문 석**(mjun@computing.soogsil.ac.kr)

1981년 숭실대학교 전자계산학과(공학사)
1986년 University of Maryland. Computer Science(석사)
1988년 University of Maryland. Computer Science(박사)
1989년 Morgan State Univ. 부설 Physical Science Lab. 책임연구원
1991년~현재 숭실대학교 컴퓨터학부 부교수
관심 분야: 인터넷 보안, 침입 차단 시스템, 암호학, 침입 탐지 시스템, OFDM,