NC 암호시스템을 이용한 웹 보안에 관한 연구

*서 장 원, **전 문 석
*서울산업대학교 공동실험실습관, **숭실대학교 컴퓨터학부
전화 : 02-970-6751 / 핸드폰 : 016-217-5304

A Study on Web Security using NC Cipher System

*Jang Won Suh, **Moon Seog Jun
*E&P Center, SNUT, **School of Computing, Soongsil University
E-mail : jwsuh@plaza1.snut.ac.kr

Abstract

EC, which is done the virtual space through Web, has weakly like security problem because anybody can easily access to the system due to open network attribute of Web. Therefore, we need the solutions that protect the Web security for safe and useful EC. One of these solutions is the implementation of a strong cipher system.

NC(Nonpolynomial Complete) cipher system proposed in this paper is advantage for the Web security and it overcomes the limit of the 64 bits cipher system using 128 bits key length for input, output, encryption key and 16 rounds. Moreover, it is designed for the increase of time complexity by adapted more complex design for key scheduling regarded as one of the important element effected to encryption.

I. 서론

최근 들어, 통신과 컴퓨터 기술이 비약적으로 발전하면서 일상적으로 행하던 일부 거래가 웹을 이용한 전자 공간에서 이루어지고 있다. 이러한 전자 거래는 웹이라는 제 3 의 가상 공간을 통해 이루어지므로, 시간과 장소에 구애받지 않고 상거래를 할 수 있다는 장점이 있다. 반대로, 누구든지 쉽게 접근할 수 있는 개방형 네트워크의 특성에 따라 웹 보안상의 문제점이 대두될 수 있다는 단점도 갖고 있다. 따라서, 안전하고 효율적인 전자상거래의 구현을 위해서는 웹 보안 문제를 해결하기 위한 정보보호 기술이 필수적이다[1].

이러한 정보보호 기술을 바탕으로 본 논문에서는 효율적인 암호시스템인 NC(Nonpolynomial Complete) 암호 알고리즘을 제안함으로써 웹 상의 보안 문제점을 해결하여 전자상거래의 거래 정보나 거래 내용들의 안전성과 신뢰성을 도모하도록 하였다.

II. 전자상거래에서의 웹 보안 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[2].

최근, 네트워크의 발달에 따라 각 개인이나 기업 또는 정부의 각종 정보들이 웹을 통해 손쉽게 상대방에게 전달되고 있는데, 이것은 이러한 다수의 정보들을 다른 사람이 웹을 통하여 손쉽게 접근할 수 있다는 것을 의미한다. 더욱이 웹 기술의 발달에 따라 일반 사용자들도 정보들에 접근이 용이해짐으로써 인증이나 개인의 사생활 및 개인정보 보호 등의 전자상거래 보안 문제는 더욱 중요한 과제로 대두되고 있는 실정이다. 따라서, 전자상거래에서도 웹 보안 문제를 반드시 고려하여야 하며 웹 상의 정보보호 문제를 포함한 보안 문제를 해결하기 위한 제반 장치가 마련되지 않는다면 전자상거래의 안전성을 기대할 수 없다. 이러한 정보보호 문제를 해결하기 위한 방법 중의 하나가 네트워크 상에서 송·수신 메시지나 거래 내용에 관한 정보를 암호화하여 사용하는 암호 기술이다.

암호 기술은 암호키의 운용에 따라 대칭키 암호시스템과 공개키 암호시스템으로 크게 구별된다. 대칭키 암호 시스템에서는 송·수신자가 동일한 비밀키를 공유해야 한다. 이것은 메시지 처리 형식에 따라 다시 블록 암호 알고리즘과 스트림 암호 알고리즘으로 나누

어진다. 이에 비해 공개키 암호시스템은 키 분배(공유) 문제에 근거하여 공개키 분배 알고리즘과 공개키 관리 알고리즘으로 나누어질 수 있다. 이것은 암·복호화 키가 서로 다르며, 어느 한 키를 공개하더라도 대응되는 다른 키를 유도해 내는 것은 계산상 불가능하도록 설계되어진다[3].

대칭키 암호시스템에서 블록 암호 알고리즘은 고정된 크기의 입력 블록이 암호화에 사용된 임의의 키에 의해 고정된 크기의 출력 블록으로 변형되는 암호 알고리즘으로써, 출력 비트의 각 비트는 입력 블록과 암호화에 사용된 키의 모든 비트에 영향을 받아 결정된다. 반면, 스트림 암호 알고리즘은 선형 쉬프트 레지스터를 이용한 이진 수열 발생기를 사용하는 암호 알고리즘의 형태로서 평문을 이진 수열로 부호화 한 뒤 이를 이진 수열 발생기에서 생성된 이진 수열과 XOR하여 이진 수열로 된 암호문을 생성하는 방식이다[4].

III. 웹에 기반한 NC 암호시스템

본 논문에서 제안한 NC 암호 알고리즘은 대칭키 암호 알고리즘에 바탕을 둔 128비트 블록 암호 알고리즘으로, Feistel 구조를 사용하여 설계하였다. 또한, 입·출력문과 암호키의 크기가 128비트이고 64비트의 서브키와 16라운드를 수행하며, 내부 함수 F 에서는 2개의 S-Box S_1과 S_2를 사용하였다.

3.1 NC 암호시스템의 설계

3.1.1 전체 구조

본 논문에서 제안한 NC 암호시스템의 전체 구조는 그림 1과 같다.

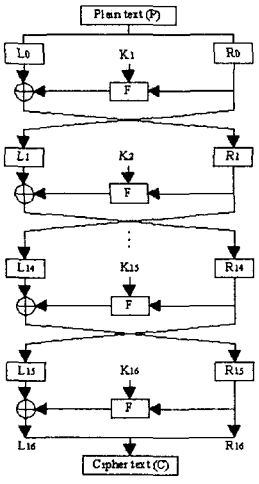


그림 1 NC 암호시스템의 전체 구조

3.1.2 키 생성 스케쥴링

내부 함수 F에서 사용되는 서브키를 생성하기 위한 키 생성 스케쥴링의 가장 중요한 목표는 안전성을 보장하는 것이다. 다음의 그림 2는 NC 암호시스템에 대한 키 생성 스케쥴링의 구조를 나타낸 것이다.

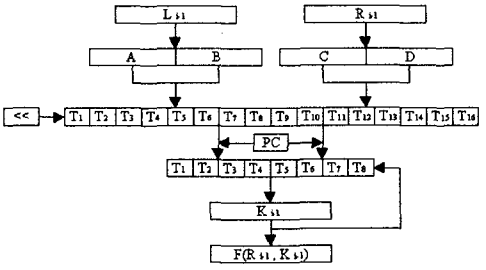


그림 2 키 생성 스케쥴링의 구조

각 라운드에서 내부 함수 F의 수열과 밀접한 연관이 있는 2^{128}의 하나에 따라 선택되는 각 라운드의 서브키를 생성하기 위한 키 스케쥴링은 임의의 128비트 입력키를 2개의 64비트 블록 (I_L, I_R)로 분할하고, 이를 다시 좌·우측 각 2개의 32비트 블록 (A, B, C, D)로 분할하여 PC 테이블과 각 라운드별로 left shift를 수행하여 각 라운드의 서브키를 생성한다. 이 때 n 라운드에서의 서브키는 n-1 라운드에서 생성된 서브키에 영향을 받도록 설계했다. 즉, n-1 라운드에서 생성된 서브키를 1~3 비트씩 회전이동 한 후, n 라운드의 서브키로 적용한다. 1~3 비트씩 회전이동을 시키는 이유는 이전 라운드의 키 비트 값에 따라 몇 라운드 후에 키 비트 값이 선형 변환이 될 수도 있으므로 이를 방지하기 위해 이와 같이 회전이동 시킨다.

표 1 PC 테이블

31	1	23	9	63	33	55	41
29	3	21	11	61	35	53	43
27	5	19	13	59	37	51	45
25	7	17	15	57	39	49	47
95	65	87	73	127	97	119	105
93	67	85	75	125	99	117	107
91	69	83	77	123	101	115	109
89	71	81	79	121	103	113	111

NC 암호시스템의 F 함수 연산을 위해 각 라운드에서 사용되는 서브키는 다음과 같은 방식으로 생성된다.

- 임의의 128비트 암호키 K를 2개의 블록 (I_L, I_R)로 분할한다.
- 이 암호키 K를 다시 4개의 블록 (A, B, C, D)로 분할한다.

- K를 PC 테이블에 적용하여 64비트로 치환한다(A_0, B_0, C_0, D_0). (PC_Table[8*8] ← K)
- A_0, B_0, C_0, D_0를 1비트 left shift 시켜 각각 A_1, B_1, C_1, D_1을 산출한다. ($A_1B_1C_1D_1$ ← S_K <<= 1)
- 이 결과를 K_1 서브키라 한다. (K_1 ← Shift_Table)
- A_1, B_1, C_1, D_1을 1비트 left shift 시켜 각각 A_2, B_2, C_2, D_2를 산출한다. ($A_2B_2C_2D_2$ ← S_K <<= 1)
- 이 결과를 K_2 서브키라 한다. (K_2 ← Shift_Table)
- A_2, B_2, C_2, D_2를 3비트 left shift 시켜 각각 A_3, B_3, C_3, D_3를 산출한다. ($A_3B_3C_3D_3$ ← S_K <<= 3)
- 이 결과를 K_3 서브키라 한다. (K_3 ← Shift_Table)
- 동일한 방법으로 반복 수행하여 K_{16} 서브키를 산출한다.

3.1.3 S-Box 적용

대개의 블록 암호 알고리즘에서 서브키의 입·출력과 관련하여 내부 함수 F에서 비선형 대입 연산을 수행하는데 사용되는 S-Box(Substitution-Box)는 비선형 함수로서 입력 크기와 출력 크기를 변경할 수 있다. 이러한 S-Box의 내용은 암호화에 민감한 영향을 미치므로 S-Box의 구성을 어떻게 하느냐에 따라 견고한 암호시스템을 구축할 수 있다. 결국, 암호시스템의 핵심은 견고한 S-Box의 새로운 설계나 또는 실험을 통해 검증된 S-Box를 적용하는 것이다.

제안한 NC 암호시스템에서는 2개의 8×8비트 S-Box S_1과 S_2를 사용하였으며, 여기서 각각의 S-Box는 NC 알고리즘의 키 크기에 의존한다. 즉, 암호화 과정을 어렵게 하기 위해 각 라운드간에 키-의존 관계를 갖도록 설계되어 입증된 2개의 S-Box를 사용하였다[5].

3.1.4 F 함수 설계

제안한 NC 암호시스템에서 128비트 키를 통한 암·복호화를 위한 근본적인 블록 구축은 평문의 우측 비트 값 64비트와 키 생성 스케쥴링에 의해 생성되는 각 라운드의 서브키 64비트를 입력 비트 값으로 하여 그것들의 비트 값을 XOR 함으로서 출력 비트 값을 산출하는 내부 함수 F 내에서 이루어진다.

이런 F 함수의 구조는 입력 문자열과 출력 문자열이 사상되는 키-의존 관계를 갖고 있다. 다시 말해서, 키 생성 스케쥴링(P 테이블과 left shift 연산)에 의해 생성된 64비트의 서브키와 2개의 각 8비트 S-Box를 이용하여 데이터의 우측면에서 수행된다. F 함수는 일반적으로 n개의 S-Box, 비트 수열, 수학적 연산 그리고 XOR에 근거하고 있다. NC 암호시스템에서는 S-Box 2개만을 이용했다.

여기서, 각 라운드에서의 원본 블록은 F 함수로의 입력이고 F 함수의 출력은 그것들의 2개 블록이 다음 라운드를 위해 교체된 후에 목표 블록과 XOR 된다. 그러므로, 전형적인 Feistel 구조를 갖는 NC 암호시스템은 F 함수의 특성에 따라 입·출력을 구분할 수 있다.

다음의 그림 3은 NC 암호시스템에서의 내부 함수 F의 구조를 나타낸 것이다.

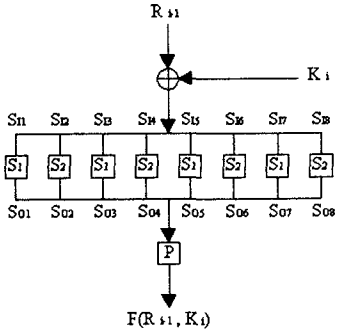


그림 3 F 함수의 구조

다음의 과정은 그림 3에서 도식화한 F 함수 내부에서의 흐름을 단계별로 설명한 것이다.

- NC 암호시스템에서 F 함수는 첫 번째로 우측 평문 값 64비트 블록을 입력으로 받아 키 생성 스케쥴링에 의해 생성된 서브키 K_i를 XOR 한다.
- XOR 이후에 산출된 각각의 8비트 블록 B_i = $b_1b_2b_3b_4b_5b_6b_7b_8$는 2개의 S-Box S_1과 S_2에 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 8비트 블록 $S_i(B_i)$로 출력되고, 모든 블록들은 P 테이블에 의해 치환된다.

이 결과로 64비트 블록 $F(R_{i-1},K_i)$가 출력되며, 최종적으로 $L_{i-1}\oplus F(R_{i-1},K_i)$는 다음 라운드의 우측 입력 비트로 이동하고, 이런 과정을 i 라운드만큼 수행한 후에 최종 암호문을 출력한다. 다음의 표 2는 P 테이블의 목록을 나타낸 것이다.

표 2 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	56
49	60	27	19	10	44	38	8
14	64	53	32	23	43	6	37
15	62	25	46	4	21	39	50
2	12	57	34	20	51	28	45
59	31	7	55	18	48	36	11

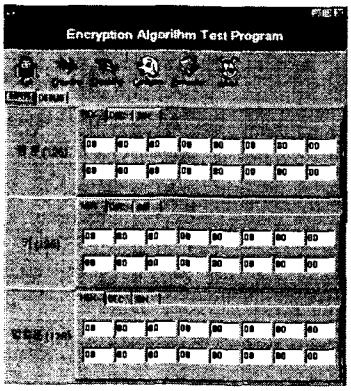
위의 표 2에서 각 라인의 숫자는 각 비트의 번지, 즉 치환되는 각 비트의 위치를 나타낸다.

3.2 NC 암호시스템의 구현

3.2.1 구현 환경 및 초기화면 구성

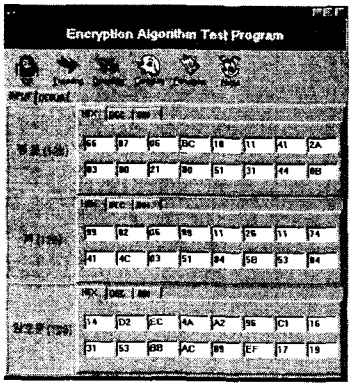
NC 암호시스템 구현을 위한 환경은 다음과 같다.

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

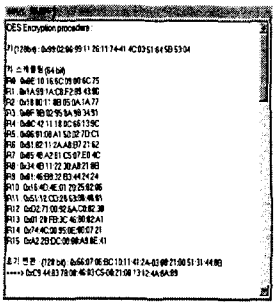


3.2.2 암호화 과정

NC 암호시스템에서 암호화 과정은 프로그램 초기화면에서 "평문"을 클릭하여 임의의 128비트 평문을 생성하고, "키"를 클릭하여 임의의 128비트 암호키를 생성한다. 생성된 128비트 암호키는 본 논문에서 제안한 키 생성 스케줄링의 과정을 거쳐 16개의 64비트 서브키를 산출한다. 그런 후에, "Encoding" 버튼을 클릭하여 다음과 같은 128비트 암호문을 산출한다.



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



Ⅳ. 결론

본 논문에서 제안한 NC 암호시스템은 웹 상의 암호체계에 적합하도록 안전성과 효율성을 고려하여 설계하였다. 또한, 일반적으로 암호화에 있어서 영향을 미치는 부분이 바로 암호키의 연산에 따라 이에 대응되는 키 비트 값이 결정되는 S-Box와 서브키를 생성하기 위한 키 생성 스케줄링인데, 이를 위해 검증된 두 개의 S-Box S_1과 S_2를 사용하였고, 외부 공격에 의해 암호키가 쉽게 발견되지 않도록 키 생성 스케줄링을 설계하였다.

제안한 NC 암호시스템은 전자상거래에서의 신뢰성을 보장하기 위한 목적으로 개발했으며, 개인 및 기업의 중요 정보보호나 메시지 암호화, 그리고 웹을 이용한 전자상거래 등에 적합하게 사용될 수 있다.

향후, NC 암호시스템의 보안 레벨을 좀 더 향상시키기 위해 키 생성 스케줄링을 보완하고, F 함수 내에 새로운 함수를 추가하여 계산 복잡도를 증가시킴으로서 더욱 견고한 암호시스템이 구축되리라고 생각된다.

참고문헌

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

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

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

[4] J.Daemen, L.Knudsen, V.Rijmen, "The Block Cipher Square", Software Encryption, 4th Int. Workshop Proceedings, Springer-Verlag, 1997.

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