카오스 암호화 알고리즘을 이용한 웹 보안 시스템 설계 및 구현

이 봉 환[†]·김 칠 민[††]·윤 동 원[†††]·채 용 웅[††††]·김 현 곤[†††††]

요 약

본 논문에서는 카오스 이론에 기초한 카오스(chaos) 암호화 알고리즘을 제안하고 이를 웹 보안 시스템에 적용하여 웹 클라이언트와 서버간의 안전한 통신을 위한 시스템을 설계 및 구현하였다. 웹 보안 시스템은 인증서버, 웹 클라이언트 및 웹 서버로 구성되며, 웹 클라이언트와 웹 서버에는 각각 웹 페이지의 요청 및 응답 페이지의 암호화 및 복호화를 담당하는 프록시 클라이언트와 서버 게이트웨이를 개발하여 탑재하였다. 인증서 형식은 국제표준을 수용하여 X.509 형식에 따라 구현하였으며, 클라이언트와 서버 인증을 위하여 RSA 공개키 알고리즘을 통하여 키 생성 및 분배가 이루어진다. 클라이언트와 서버간에 암호화 채널이 형성되면 카오스, SEED 및 DES 암호화 알고리즘을 통해 데이터의 암·복호화를 수행한다. 카오스 암호화 알고리즘은 기존의 비밀키 암호화 알고리즘들과 비교하여 속도와 비도 면에서 뛰어나다. 따라서 카오스 암호화 알고리즘을 적용한 웹 보안 시스템은 전자상거래, 인터넷 뱅킹 등에 널리 활용될 수 있을 것으로 사료된다.

Design and Implementation of a Web Security System using a Chaos Cipher Algorithm

Bong-Hwan Lee[†]·Chil-Min Kim[††]·Dong-Weon Yoon[†††]·
Yong-Wong Chae[††††]·Hyeun-Gon Kim[†††††]

ABSTRACT

In this paper, a new stream cipher algorithm based on the chaos theory is proposed and is applied to a Web security system. The Web security system is composed of three parts : certificate authority (CA), Web client, and Web server. The Web client and server system include a secure proxy client (SPC) and a secure management server (SMS), respectively, for data encryption and decryption between them. The certificate is implemented based on X.509 and the RSA public key algorithm is utilized for key creation and distribution to certify both the client and server. Once a connection is established between the client and server, outgoing and incoming data are encrypted and decrypted, respectively, using one of the three cipher algorithms : chaos, SEED, and DES. The proposed chaos algorithm outperforms the other two conventional algorithms in processing time and complexity. Thus, the developed Web security system can be widely used in electronic commerce (EC) and Internet banking.

키워드 : 웹보안(Web security), 카오스암호(Chaos), 인증서버(CA), SSL, 소켓(X-509)

1. 서 론

최근 인터넷의 보편화로 그 저변이 확대됨에 따라 웹(World Wide Web)을 이용한 정보시스템의 구축이 활발히 이루어지고 있다. 초기의 ftp, telnet, 전자메일 등의 인터넷 응용서비스는 독립적으로 제공되었으나 넷스케이프, 익스플로러 등의 브라우저가 개발되어 기존의 응용서비스들이 통합되고 편리한 사용자 인터페이스가 제공되면서 인터넷 사용자는 급격히 증가하게 되었다. 인터넷 트래픽은 대략 매 6개월 마다 약 2배 정도로 증가한다고 알려져 있다[1]. 또한, 웹 기술을 이용하여 다양한 응용 서비스가 제공되면서 인터넷을 이용한 전자상거래 시스템이 출현하게 되었다. 그러나 웹은 기본적으로 안전하지 않다는 개념에서 출발한다. 이는 웹 서비스를 지원하는 인터넷 자체가 개방성을 바탕으로 설계된 TCP/IP를 사용하고 있기 때문에 정보를 공유하기 위하여 보안에 대한 문제를 고려하지 않았기 때문이다. 웹이 전자메일, 전자게시판, 전자상거래 등에 광범위하게 사용됨에 따라 개인정보의 불법적인 유출 및 파괴는 엄청난 손실을 유발할 수 있다. 따라서 웹을

※ 이 논문은 한국과학재단의 광인터넷연구센터의 지원에 의한 것임.
† 종신회원 : 대전대학교 컴퓨터정보통신공학부 교수
†† 정 회 원 : 배재대학교 물리학과 교수
††† 정 회 원 : 대전대학교 컴퓨터정보통신공학부 교수
†††† 정 회 원 : 계명대학교 전자공학과 교수
††††† 정 회 원 : ETRI 정보보호기술연구본부 AAA정보보호연구팀장
논문접수 : 2001년 8월 1일, 심사완료 : 2001년 10월 15일

이용한 정보 전송에서 보안문제는 필수요건으로 대두되고 있다.

인터넷 상에서 웹을 이용하여 클라이언트와 서버 사이의 신뢰성과 안전성을 확보하기 위해서는 웹 보안 프로토콜이 필요할 뿐만 아니라 평문(plaintext) 데이터를 암·복호화하기 위한 알고리즘이 필요하다. 웹 보안 프로토콜에는 응용 계층에서 메시지 전체를 암호화하여 전송하는 방식과 메시지 가운데 몸체 부분만 암호화하여 보내는 방식이 있다. 또한 SSL(Secure Socket Layer)[2]과 같이 응용계층과 트랜스포트 계층 사이에 암호화 계층을 별도로 삽입하여 암호화하는 방법이 있으며, 현재 이 방법이 가장 널리 사용되고 있다. 웹 클라이언트와 서버 사이에 신뢰성과 안전성을 확보하기 위해서는 클라이언트와 서버 각각을 인증서버(Certificate Authority : CA)를 통하여 인증을 받는 과정이 필요하다. 이 때 사용되는 데이터의 집합이 인증서이다. 인증서 형식으로 가장 널리 사용되고 있는 표준이 ITU-T의 X.509[3] 인증서 형식이다. 공개키 기반의 웹 보안 시스템에서 클라이언트와 웹 서버의 인증이 종료된 후에 암호화 채널을 형성하여 실질적으로 데이터를 주고받을 때는 비밀키 암호화 알고리즘을 사용한다. 비밀키 암호화 알고리즘으로 가장 널리 사용되고 있는 것으로는 미국에서 개발한 DES[4]가 있고, 국내에서 개발한 것으로는 한국전자통신연구원과 한국정보보호진흥원(KISA)에서 공동 개발한 SEED[5]가 있다. 또한, 대부분의 상용 보안시스템에서는 RSA 공개키 알고리즘을 이용하여 인증과 세션 키 교환을 수행한다.

본 논문에서는 카오스 이론에 기초하여 기존의 DES와 SEED 암호화 알고리즘을 대체할 수 있는 카오스 암호화 알고리즘을 개발하여 이를 웹 보안시스템에 적용하였다. 이는 정보신호에 카오스 신호를 삽입하면 카오스적인 성질을 가지게 되고 수신측이 송신측과 동일한 카오스 시스템을 사용하면 두 시스템간 동기가 가능하여 수신측에서 카오스 신호를 제거하여 정보신호를 복원해낼 수 있다는 원리에 기초한다[6, 7]. 본 논문에서는 기존의 카오스 암호화 알고리즘에 비하여 처리속도 및 비도에서 성능이 우수한 알고리즘을 개발하여 웹 클라이언트와 서버 사이의 데이터 암·복호화에 적용하였다.

2. 카오스 암호화 알고리즘

2.1 개 요

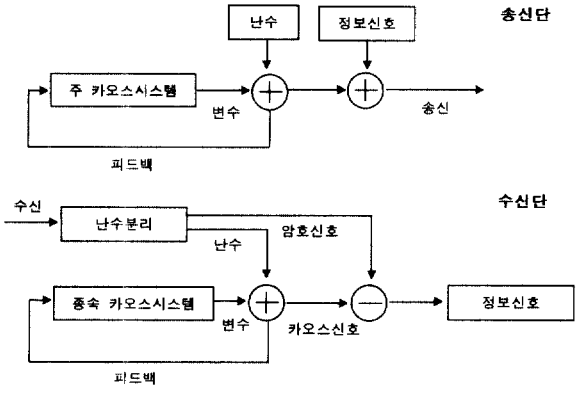
암호 알고리즘은 크게 세 가지 암호화 방법으로 나뉘는데 첫째는 암·복호화하는 과정에서 암·복호화 키가 같은 대칭키 암호알고리즘이고 둘째는 암·복호화 키가 다른 비대칭 즉, 공개키 암호 알고리즘이며, 셋째는 이와는 다른 데이터의 값을 수십 바이트의 데이터로 함축시킨 해쉬함수 알고리즘이다. 이중 공개키 암호 알고리즘은 처음 상대방과 통신을 시작할 때 공개키를 이용하여 상대를 호출하며 이 공개키를 이용하여 대칭키를 암호화시켜 통신을 시작한다. 이때 통신 데이터는 대칭키를 이용하여 송·수신되며 데이터의 끝부분에 데이터의 무결성을 입증하는 해쉬 값을 붙인다. 이것이 현대 암호에서 쓰이는 일반적 방법이다.

이 중 대칭키 암호는 블록 단위로 데이터를 처리하는 블록 암호 알고리즘과 계속 발생되는 난수에 데이터를 처리하는 스트림 암호 알고리즘으로 나뉜다. 전자는 정해진 비트 수만큼 데이터를 블록으로 나누어 각각의 블록을 전치, 도치, 변환 등의 과정을 거쳐 암호화시키는 방법이고, 후자는 발생되는 난수에 차례대로 데이터를 처리하는 방법이다. 따라서 블록 암호 알고리즘의 암호키는 데이터의 암호화 과정에 들어가고 스트림 암호 알고리즘은 난수의 초기 값 결정에 주로 쓰인다. 이 결과 블록 암호는 한 블록 단위씩 데이터를 처리하기 때문에 중간에 데이터의 손실이 생겨도 나머지 부분을 해독할 수 있으나 스트림 암호는 중간에 한 비트의 오차가 생겨도 그 이후의 데이터는 복원할 수 없는 문제점이 있다. 스트림 암호 알고리즘이 블록 암호 알고리즘 보다 안정성 면에서 뛰어난 반면 데이터의 에러 발생의 문제로 인해 데이터 송·수신에 문제가 발생할 가능성이 있으며 블록 암호는 데이터 에러는 예방할 수 있으나 안정성을 높이기 위해 블록의 크기 및 라운드 회수가 커야하는 문제점이 있다. 그럼에도 불구하고 블록 암호의 데이터 송·수신의 장점으로 인해 블록 암호를 선호하는 편이다.

개발한 카오스암호화 알고리즘(이하 K-7)은 대칭키 암호로서 블록 단위로 데이터를 처리하는 블록 암호의 일종이나 카오스 특성을 이용하기 때문에 스트림 암호의 특성도 동시에 가지고 있다. 암호 수행시 스트림 암호와 비슷하게 처리되어 안정성에서 뛰어나고 데이터의 복원 과정과 에러에 의한 데이터의 손실 정도는 블록 암호 알고리즘의 특성을 가지고 있으므로 두 알고리즘의 장점을 모두 갖추고 있는 이상적인 형태의 암호 알고리즘이다. 블록 암호 알고리즘과 스트림 암호 알고리즘을 서로 비교한 후 K-7 암호 알고리즘을 살펴보면 K-7 암호 알고리즘의 특성을 쉽게 알 수 있다.

1990년 Pecora와 Carroll은 동일한 두 개의 카오스시스템 가운데 주카오스 시스템의 신호를 종속카오스시스템에 넣어 피드백시키면 종속 카오스시스템의 신호가 주 카오스시스템의 신호와 일치하게 되는 카오스의 동기화 현상이 생김을 발견하였다[7]. 이 현상을 이용하여 1993년 Coumo와 Oppenheim은 전자회로에서 두 카오스시스템을 이용하여 아날로그 신호를 암호화시킬 수 있음을 밝혔다[8]. 이것은 주 카오스시스템에서 나오는 카오스 신호 속에 음성신호를 섞어 송신하면 카오스 신호는 잡음과 같이 무질서하기 때문에 외부에서 그 신호를 찾아낼 수 없다는 가정에서 출발

했다. 동시에 종속 카오스시스템에서는 이 암호화된 신호를 받아 피드백시키면 주 카오스시스템과 같은 신호가 복원되므로 이 복원된 신호와 수신된 신호의 차이를 구하여 음성 신호를 복원할 수 있음을 보였다. 일반적인 카오스 시스템의 개념도를 나타내면 (그림 1)과 같다. 최근의 카오스 시스템과 관련하여 낮은 차원의 카오스시스템을 이용한 암호화 방법은 해독이 가능하다는 여러 이론들이 나오고 있다. 이것은 카오스 신호는 잡음과 달리 프랙탈 구조라는 특이한 규칙성이 있기 때문이다. 본 연구에서 개발한 암호화 시스템에서는 고차원의 카오스 시스템을 사용하며, 카오스 신호의 규칙성을 난수를 이용한 카오스의 동기화 방법으로 완전히 없앰으로 난수 속에 데이터 파일을 감추는 형태로 바뀌어 해독이 불가능한 카오스를 이용한 암호화 방법이다. 특히, 본 연구에서는 기존의 아날로그 암호화 시스템이 아니라 차분 방정식을 이용한 디지털 데이터 암호화 시스템을 개발하였다.



(그림 1) 카오스 암호화 및 복호화 개념도

2.2 K-7의 설계기준

K-7 암호 시스템은 난수에 의한 카오스 시스템의 동기화를 이용한 암호화 시스템이다. 암호화는 난수에 의해 섭동받는 카오스 신호 속에 데이터 파일을 bit 단위 연산을 통하여 암호화시키고, 복호화는 동일한 난수를 수신자의 카오스시스템에 피드백시켜 송신 카오스시스템과 동기화시키고 수신 카오스 신호의 비트 스트림과 암호문의 비트 스트림을 bit 단위의 역연산을 통하여 데이터 파일을 복원한다. 그러므로 암호화를 위하여 카오스를 발생시키는 고차원의 카오스 시스템이 있어야 하고 카오스가 생성되는 계수 값의 범위를 결정하여야 하며, 동기화를 위한 난수 발생장치가 필요하다. 또한, 동일한 두 카오스 시스템을 동기화 시킬 수 있는 난수와 변수를 결합시키는 결합 상수 값의 범위를 결정하여야 한다. 여기서 key의 길이는 계수의 수와 그 길이에 의해 결정되는데 계수의 수가 많으면 그만큼 key의 길이가 길어진다. 그리고 암호화 데이터 블럭의 길이는 카오스 신호의 비트 스트림 길이에 비례한다. 예를 들어, 비트 스트림을 64 bit로 사용할 경우 8차원 카오스 시스템의 경우 8개의 변수에 데이터 파일을 암호화시키므로 컴퓨터 기종에 따른 round off 에러없이 한 변수 당 4 byte씩 암호화가 가능하므로 32 byte의 암호화가 한번에 이루어진다. 키의 길이는 8차원 카오스 시스템의 경우 128 bit까지 안정적으로 사용할 수 있다. 그러므로 카오스를 이용한 암·복호화의 특성상 K-7 암호는 다음과 같은 기준을 가진다.

- 전체 구조
  - 데이터 처리 단위 : 32 byte(카오스시스템의 종류에 따라 길이 변환이 가능)
  - 암·복호화 방식 : 스트림 암호 방식으로 데이터 복원은 블록 암호의 특성이 있음
  - 입력문의 크기 : 256 bits
  - 출력문의 크기 : 256 bits
  - 안정성 : 알려진 공격법에 대하여 안전
  - 효율성 : triple DES 및 SEED의 속도 이상
  - 구 조 : 카오스 시스템을 이용하므로 스트림 암호구조와 유사
  - 내부함수 : 8차원 카오스 시스템 이용
  - 라운드 수 : 1 라운드로 안전성이 충분함
  - 키 생성 알고리즘 : 키를 여러 bit로 나눈 후 실수로 변환하여 각 계수에 대입
  - 키 길이 : 8차원 카오스 알고리즘으로 128 bits
- 안정성에 대한 설계 기준
  - 1 라운드를 기본으로 하나 보안 강도 증강시 2 라운드를 채택
  - 차분해독법, 선형 해독법, Related key attack 및 기타 공격 방식이 적용되지 않음
  - known text attack, chosen text attack에 안전
- K-7의 일반적 특성
  - 암호 블록 길이의 변환이 쉬우며 블록 길이는 변수의 수에 비례
  - 키와 암호문과는 상관 관계가 없음
  - DES의 ECB 모드와 동일한 방법으로도 사용 가능
  - 암호문은 stream 암호처럼 동일한 평문도 항상 다른 암호문으로 생성
  - 카오스 시스템의 소수점 이하 실수만을 사용
  - co-processor 채택시 속도가 배가됨
  - 역계산이 불가능하고 알려진 공격법으로는 공격되지 않음
  - 안정성에 대한 수학적 계산이 가능하다
  - 카오스 신호는 난수 신호와 동일한 특성을 가진다
  - 소프트웨어적인 처리속도는 SEED의 25배

2.3 K-7 카오스 시스템

K-7 암호 알고리즘은 난수에 의한 카오스의 동기화를 이용한 방법이다. 이 방법은 카오스의 프랙탈 구조를 변형시켜 보다 안전한 암호시스템을 만들기 위해 개발되었다. 그러므로 난수에 의해 어떻게 두 카오스 시스템이 동기화 되는지 살펴본다. 잡음에 의한 카오스의 동기화 방법은 그 가능성이 1995년 Maritan과 Banavar에 의해 처음 제안되었다[9]. 그들은 동일한 두 카오스 시스템에 동일한 잡음을 동시에 가하면 두 카오스 시스템이 동기화된다는 사실을 발표하였다. 그러나 이것은 컴퓨터 계산시 발생하는 round off error에 의한 것임을 Pikovsky [10] 등이 밝혔는데 이 모든 현상을 다 고려한다고 하여도 동기화되는 영역이 존재함을 Kim[11] 등은 밝혔다. 본 동기화 방법은 이 방법을 이용하여 카오스 시스템을 동기화시키는 방법이다. 그 이론을 가장 간단한 카오스시스템의 하나인 logistic map을 이용하여 설명하기로 한다. logistic map은 식 (1)과 같이 주어진다.

$$x_{n+1} = \lambda x_n (1 - x_n) \tag{1}$$

이때 종속 카오스시스템의 값을 $x'_n = \lambda x'_n (1 - x'_n)$으로 두고 난수로서 두 카오스시스템을 동기화 시키는 방법을 생각하기로 하자. 먼저 난수 r_n을 외부에서 피드백시키는 경우를 생각하자. 그러면 두 식은 다음과 같이 주어진다.

$$x_{n+1} = \lambda x_n (1 - x_n) + r_n \tag{2}$$

$$x'_{n+1} = \lambda x'_n (1 - x'_n) + r_n \tag{3}$$

실제 Maritan Banavar 등은 난수의 폭을 임의로 정했다. 그러므로 난수 ξ_n은 0과 1 사이의 값이고 난수 r_n은 0과 β 사이의 값으로 주어진다고 하면 두 카오스시스템에 대입되어 피드백되는 난수 r_n은 $r_n = \beta \xi_n$이다. 이 두 식은 서로 동기화되는데 두 식이 동기화 되는 것을 보이기 위해 homomorphism을 이용하여 $x_n - r_{n-1} = \alpha y_n$으로 두고 $x_{n+1} - r_n = \alpha y_{n+1}$으로 두면 $x_n = r_{n-1} + \alpha y_n$ $x_{n+1} = r_n + \alpha y_{n+1}$가 되어 다음과 같이 된다.

$$\alpha y_{n+1} + r_n = \lambda (\alpha y_n + r_{n-1})(1 - (\alpha y_n + r_{n-1})) + r_n \tag{4}$$

$$\alpha y'_{n+1} + r_n = \lambda (\alpha y'_n + r_{n-1})(1 - (\alpha y'_n + r_{n-1})) + r_n \tag{5}$$

이 식에서 양변의 r_n을 모두 소거한 다음 α로 양변을 모두 나누어주고 $r_n = \beta \xi_n$으로 대치시켜 $\lambda / \alpha = \mu$로 대치시킨 후 이 식이 발산하는 것을 막기 위하여 절대값을 취하고 1 이하의 값을 유지하기 위하여 modulus 1을 택하면 식은 다음과 같이 된다.

$$y_{n+1} = |\mu (\alpha y_n + \beta \xi_{n-1})(1 - (\alpha y_n + \beta \xi_{n-1}))| \bmod 1 \tag{6}$$

$$y'_{n+1} = |\mu (\alpha y'_n + \beta \xi_{n-1})(1 - (\alpha y'_n + \beta \xi_{n-1}))| \bmod 1 \tag{7}$$

여기서 ξ_{n-1}은 임의의 난수이므로 ξ_n으로 둘 수 있다.

위 두 식이 동기화되는 것을 알기 위해서는 두 식의 변수의 차를 $y_n - y'_n = z_n$이라 두면 두 식의 차는 식 (8)과 같이 된다.

$$z_{n+1} = \mu \alpha (1 - 2\beta \xi_n - 2\alpha y_n) z_n + \mu \alpha^2 z_n^2 \tag{8}$$

식 (8)은 새로운 비선형 차분 방정식의 꼴이 된다. 그런데 이 식을 보면 먼저 z_n 앞의 매개변수로서 y_n과 ξ_n으로 변조되는 값이 있다. 이 식의 의미는 주 카오스 시스템의 변수와 난수로 매개변수가 변조되는 새로운 식이 된 것이다. 여기서 z_n 값 앞에 붙어 있는 모든 값들을 매개변수로 볼 수 있는데 이렇게 카오스신호나 잡음신호로 다른 비선형시스템을 변조시키는 방법들은 blow-out bifurcation을 하는 것으로 알려져 있다. 그런데 이런 카오스 신호나 잡음신호로 비선형시스템의 매개변수를 변조시키면 그 카오스 시스템은 매우 복잡한 양상을 지니는데 각 매개변수의 조건에 따라 이 카오스시스템은 카오스 신호와 0의 값에 매우 가까운 값 사이를 불규칙적으로 왕복하기도 하고 0의 값으로 수렴할 때도 있고 때로는 카오스를 보이기도 한다. 카오스와 0의 값에 매우 가깝게 왕복하는 것을 on-off 간헐성이라고 하는데 이때 0의 값으로 위 식이 수렴하게 되는 임계조건이 생기게 된다. 이 임계조건을 넘어서면 이 두 동일한 카오스시스템의 변수 차이로 만드는 새로운 카오스 시스템은 곧바로 0으로 수렴하게 된다. 따라서 그 변수의 차가 0이 되면 두 카오스시스템의 궤적차가 없으므로 두 카오스시스템의 궤적은 서로 같아지게 되어 곧 동기화가 되는 것이다. 이 현상이 생기는 조건이 바로 on-off 간헐성의 임계조건에서 평균 laminar 길이가 무한히 길어져 두 변수의 차가 0으로 수렴하는 α, β의 조건이 되고 이 조건은 바로 두 카오스시스템이 서로 동기화 되는 조건인 것이다.

암호화 방법은 이런 이론적 얼개에서 주어지는 두 개의 동일한 차분 방정식을 하나는 송신기로 두고 다른 하나를 수신기로 두어 다음과 같이 주어지게 된다. 먼저 난수 ξ_n으로 송신 카오스시스템을 섭동시키면 송신 카오스시스템의 변수인 x_n의 신호는 매우 불규칙한데 이 신호는 섭동을 위한 잡음과는 거의 무관하게 된다. 이 난수로 초기 값이 다른 수신 카오스 시스템을 섭동시키면 수신 카오스 시스템은 처음에는 송신 카오스 시스템과 다른 신호를 보이다가 곧 송신 카오스 시스템과 동기화 된다. 그러므로 암호화 방법은 처음에는 두 카오스시스템이 동기화 될 때까지 난수만 송신한다. 그리고 두 카오스 시스템이 동기화되는 시간이 지나면 (동기화되는 시간은 구할 수 있다) 송신 카오스 시스템의 신호에 데이터를 XOR하여 송신한다. 이

때 두 카오스 시스템이 동기화 되는 것은 두 카오스 시스템의 계수와 난수의 시간적 파형에만 의존하므로 송신 카오스 시스템의 초기 값이 무엇이든 아무런 관계가 없다. 즉, 카오스 시스템의 초기 값을 제 삼자든 수신자든 아무도 모른다는 것이다. 그러나 수신 카오스 시스템은 일정한 동기화를 위한 시간의 경과 후 송신 카오스 시스템의 계수 값과 난수만 일치하면 송신 카오스 시스템과 동일한 신호를 만들게 되므로 데이터를 복원할 수 있게되는 것이다.

이러한 방법으로 key의 길이와 암호 블록의 크기를 고려하여 역함수 계산이 불가능한 최적의 카오스 시스템을 만들고 그 후 두 카오스 시스템을 동기화 시킬 수 있는 α, β의 값들을 정한 다음 암호화 알고리즘과 key 생성 알고리즘을 만들면 하나의 암호화 시스템이 되는 것이다. K-7 암호화 방법은 난수를 이용하여 처음부터 카오스신호의 프래탈 구조를 파괴하여 그 구조를 파악할 수 없는 신호로 만들고 또 난수를 이용하여 동기화까지 시킴으로 완벽한 암호 시스템을 만드는 것이다. 특히, 이 방법은 아날로그 신호의 암호화를 위하여 개발된 것을 디지털 신호의 암호화로 확장한 디지털 데이터의 암호화 방법이다.

K-7의 암호화 방법과 안정성에 대해서 10차원 카오스 시스템을 예로 들어 설명하고자 한다. 일반적 블록 암호나 스트림 암호의 F-함수에 해당되는 카오스 시스템은 식 (9)와 같이 주어진다.

$$
\begin{aligned}
x_{n+1} &= |a_1 x_n(1-x_n w_n)+b_1 w_n(1-x_n)+c_1 v_n w_n| \bmod 1 \quad (9)\\
y_{n+1} &= |a_2 y_n(1-v_n)+b_2 p_n(1-y_n)+c_2 x_n p_n| \bmod 1\\
z_{n+1} &= |a_3 z_n(1-z_n)+b_3 q_n(1-z_n)+c_3 u_n r_n y_n| \bmod 1\\
u_{n+1} &= |a_4 u_n(1+u_n)+b_4 r_n(1-u_n)+c_4 p_n q_n| \bmod 1\\
v_{n+1} &= |a_5 v_n(1-z_n)p_n r_n+b_5 s_n(1-v_n)z_n+c_5 u_n s_n| \bmod 1\\
w_{n+1} &= |a_6 w_n(1-w_n)+{}_6 x_n(1-r_n z_n)+c_6 u_n q_n| \bmod 1\\
y_{n+1} &= |a_7 p_n(1+p_n)+b_7 y_n(1-p_n)+c_7 u_n p_n| \bmod 1\\
q_{n+1} &= |a_8 q_n(1-q_n)q_n+b_8 z_n(1-p_n)+c_8 y_n p_n| \bmod 1
\end{aligned}
$$

이 10차원 시스템을 동기화 시키기 위하여 각각의 변수를 다른 난수로 섭동시키므로 각 변수가 난수로 섭동되면 각 변수는 다음과 같이 둘 수 있다.

$$
\begin{aligned}
X_n &= \alpha x_n-\beta\xi_n-\gamma, \quad & Y_n &= \alpha y_n-\beta\xi_{n-1}-\gamma,\\
Z_n &= \alpha z_n-\beta\xi_{n-2}-\gamma, \quad & U_n &= \alpha u_n-\beta\xi_{n-3}-\gamma,\\
V_n &= \alpha v_n-\beta\xi_{n-4}-\gamma, \quad & W_n &= \alpha w_n-\beta\xi_{n-5}-\gamma,\\
P_n &= \alpha p_n-\beta\xi_{n-6}-\gamma, \quad & Q_n &= \alpha q_n-\beta\xi_{n-7}-\gamma.
\end{aligned}
$$

여기서 α와 β는 동기화를 위하여 필요한 상수로 고정된다. 이 상수의 고정은 어떤 초기 값에서도 두 카오스 시스템이 동기화되는 시간을 미리 알기 위해서 필요하다. 그래야만 그 시간 뒤에는 암호화된 신호를 송신하여도 에러없이 데이터를 복원할 수 있다. 또 외부에서 Maritan과 Banavar의 방법으로 난수로 각 식을 섭동시키면 앞의 식은 다음과 같이된다.

$$
\begin{aligned}
x_{n+1} &= |a_1 X_n(1-X_n W_n)+b_1 W_n(1-X_n)+c_1 V_n W_n+\delta_1\xi_{n-7}| \bmod 1 \quad (10)\\
y_{n+1} &= |a_2 Y_n(1-V_n)+b_2 P_n(1-Y_n)+c_2 X_n P_n+\delta_2\xi_{n-5}| \bmod 1\\
z_{n+1} &= |a_3 Z_n(1-Z_n)+b_3 Q_n(1-Z_n)+c_3 U_n R_n Y_n+\delta_3\xi_{n-2}| \bmod 1\\
u_{n+1} &= |a_4 U_n(1+U_n)+b_4 R_n(1-U_n)+c_4 P_n Q_n+\delta_4\xi_n| \bmod 1\\
v_{n+1} &= |a_5 V_n(1-Z_n)P_n R_n+b_5 S_n(1-V_n)Z_n+c_5 U_n S_n+\delta_5\xi_{n-4}| \bmod 1\\
w_{n+1} &= |a_6 W_n(1-W_n)+b_6 X_n(1-R_n Z_n)+c_6 U_n Q_n+\delta_6\xi_{n-1}| \bmod 1\\
p_{n+1} &= |a_7 P_n(1+P_n)+b_7 Y_n(1-P_n)+c_7 U_n P_n+\delta_7\xi_{n-6}| \bmod 1\\
q_{n+1} &= |a_8 Q_n(1-Q_n)Q_n+b_8 Z_n(1-P_n)+c_8 Y_n P_n+\delta_8\xi_{n-3}| \bmod 1
\end{aligned}
$$

그러면 동일한 두 카오스 시스템은 초기 값에 관계없이 동기화 된다. 그리고 이 식이 카오스를 이용한 암호시스템인 K-7 암호 알고리즘의 F 함수에 해당된다. 이 식에서 송·수신을 위하여 쓰이는 변수는 x_n, y_n, z_n, u_n, v_n, w_n, p_n, q_n의 8개의 변수이다. 이들을 통해 본 함수는 known text attack과 chosen text attack에 대해서 안전해지며 전수조사 공격 때 전수 조사횟수의 감소를 막기 위해 상위 6-bit는 감춘다. 그 다음 상위 32-bit만 암호화에 사용하고, 나머지 bit는 컴퓨터 시스템 마다 다른 반올림 처리를 위해 버린다. 8개의 변수가 암호화 송·수신에 쓰이게 되고 암호 블록의 길이는 이 변수의 개수에 의해서 결정된다.

그리고 식 (8)에서 a_i, b_i, c_i, δ_i의 각각 8개의 계수가 생기는데 이 계수는 K-7 암호 시스템의 key로 쓰이게 된다. 그 이유는 두 카오스 시스템이 동기화되는 것은 두 카오스 시스템의 계수가 일치하여야만 가능하기 때문이다. 그러나 카오스의 동기화 이론에서 두 카오스 시스템의 계수가 조금이라도 틀릴 경우 두 카오스 시스템은 attractor bubbling에 의해 on-off intermittency(간헐성)이 생기게 되며, 이것은 고유한 임계지수를 가지고 있다. 이 현상은 전수 공격 때에 처음의 key와 우연히 차이가 작아지면 공격자는 key의 공격 횟수를 줄일 수 있는 근거를 주게 된다. 이 현상을 막기 위하여 key 생성 방법은 일반 암호 시스템과는 달라야 한다. key 생성 방법은 계수 값이 일치하면 동기화가 생겨 복호화가 가능하게 되므로 계수값이 key가 된다. 여기서 128-bit 길이를 송·수신단 모두 동일한 방법으로 임의로 ${}_{128}C_{32}$을 취하여 32개의 32-bit 스트림으로 만든 후 각 시스템에서 이것을 각각 실수로 바꾸어 계수로 주게 된다. 본 시스템에서는 128-bit를 4-bit씩 shift시켜 상위 32-bit씩을 구하여 이것을 실수로 바꾼 것을 시스템의 key로 사용하였다.

2.4 암호화 방법

암호화는 부동소수점 비트 스트림에 정보신호의 비트 스

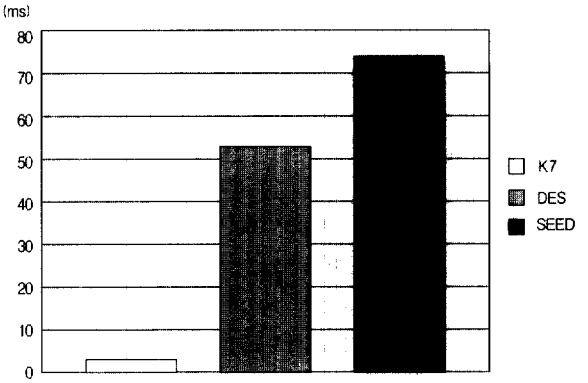
트림에 XOR 등 연산을 이용하여 암호화시키고 송신은 난수 값과 암호문을 더하여 송신하게 된다. 이 경우 외부에서는 난수값과 암호문을 알 수 있으나 감춘 상위 6-bit의 값, modulus와 절대 값을 취하는데 필요한 3 bit의 값을 알 수 없으므로 키 값인 계수를 구할 수 없다. 따라서 K-7은 안전한 암호기법이 된다. 카오스 수식을 이용한 데이터의 암호화의 방법은 구체적으로 다음과 같다.

변수의 비트 스트림은 double 형인 경우 64bit, real 형인 경우 32bit로 계산한다. 여기서 double 형인 경우 첫 번째 bit는 parity bit이다. 뒤의 12bit는 지수 bit 이고, real 형인 경우는 첫 번째 bit는 parity bit 이며, 뒤의 8bit는 지수 bit 이다. 그러나 암호화에 쓰이는 bit는 소수점 이하의 실수이므로 parity와 지수 부분은 쓰이지 않는다. 또한 카오스 시스템의 계산에서 쓰이는 수가 실수이므로 데이터의 끝 부분은 반올림처리를 하여야 하는데 컴퓨터의 종류에 따라 round-off 방법이 다르다. 이것에 의한 오차도 없애야하므로 마지막 byte는 암호 시스템에서 강제적 반올림 처리를 하여야한다. 따라서 real 형인 경우 한 변수당 수행할 수 있는 데이터의 암호는 2byte이고 double 형인 경우 5 byte가 된다. double 형인 경우를 고려하면 각 변수 당 40bit의 소수점 이하 비트 스트림이 남게 되는데 이 비트 스트림에 상위 6bit는 attractor bubbling 문제를 피하며 역계산이 되지 않도록 감추며 나머지 32bit에 데이터 stream을 bit 단위로 XOR등의 연산을 이용하여 데이터를 암호화시킨다. 여기서 변수 하나로 암호화 할 수 있는 bit 수는 32bit이므로 통신 변수의 개수가 위의 예처럼 8개이면 256bit를 한꺼번에 처리할 수 있다. 또 통신 변수의 개수가 4개이면 한번에 암호화시킬 수 있는 데이터의 bit 수는 128bit가 된다. 결국 한번에 처리되는 암호화 bit 수는 변수의 개수와 부동소수점 계산때 나오는 변수의 길이에 비례된다. 그러므로 한번에 처리되는 데이터의 수는 얼마든지 그 크기를 변화시킬 수 있는 장점이 있다. 그리고 데이터의 암호화는 위의 카오스식을 한번 계산할 때마다 수행한다.

2.5 K-7 암호의 특성

- 카오스 시스템을 사용하여 암호화시킨다.
- 카오스는 난수와 같이 불규칙하고, 예측 불가능하며 비가역적이기 때문에 역계산이 불가능하다. 그러므로 해독이 불가능하다(unbreakable).
- 암호는 스트림 암호화 블록 암호의 특성을 동시에 지닌다.
  - 블록 암호의 특성
    카오스 신호의 부동소수점 값에 블록 bit 수와 같은 데이터 bit 수를 더하여 암호화시킨다.
    복호화시 error가 발생하여도 곧 카오스신호가 복원되므로 블록암호 특성을 지닌다.
  - 스트림 암호의 특성
    불규칙한 카오스신호에 데이터를 더하여 암호화시키므로 스트림 암호의 특성을 지닌다.
- 카오스 시스템은 차분 방정식에 의해 수행된다.
- 연산 속도가 빨라 SEED에 비해 소프트웨어로 구현시 25배 정도 빠르다.
- key는 카오스시스템의 계수를 이용하고 데이터 처리는 변수에 의존하므로 key와 변수 사이 상관관계가 없다.
- 카오스의 특성상 암호시스템의 크기가 아주 작다.

본 연구에서 제안한 카오스 암호화 알고리즘과 다른 알고리즘과의 암호화 속도를 비교하면 (그림 2)와 같다. (그림 2)의 암호화 속도는 각 알고리즘 별 128 비트 한 블록을 암호화하는데 걸리는 시간을 20회 측정하여 평균한 값이다. 그림에서 보는 바와 같이 제안한 카오스 암호화 알고리즘은 SEED에 비하여 약 25배 빠르며, DES 보다는 약 18배 빠르다.



(그림 2) 암호화 알고리즘 속도 비교

3. 웹보안 프로토콜

웹 보안기술은 크게 나누어 내용기반방식(content-based), 메시지 기반방식(message-based) 및 채널기반방식(channel-based) 등으로 구분할 수 있다. HTTP 프로토콜 상위계층에서 메시지 전체를 암호화하여 전송하는 방식이 내용기반방식이며, Kerberos와 PGP(Pretty Good Privacy)등이 대표적인 예이다. 이는 HTTP 상에서 외부 응용에 의해 암호화된 문서 형태로 안전하게 데이터를 전송하는 접근 방법으로 대부분 MIME 형태로 구현된다. 또한 전자우편에서 암호화 및 전자서명에 많이 사용되고 있는 PGP와 웹의 CCI (Common Client Interface) 기능을 이용한 방법도 있다. 이 방식의 기본 개념은 요구되는 모든 보안 서비스 기능을 처음부터 웹 브라우저 응용으로 구현하기 보다는 암호화, 전자서명, 검증 및 키 관리를 처리하기 위하여 PGP를 사용한다는 것으로 이를 위해서는 HTTP 확장, 새로운 HTML 앵

커 속성, 그리고 CCI 기능이 확장되어야 한다.

한편, HTTP 메시지 가운데 내용 부분만을 암호화하여 전송하는 방식이 메시지 기반 방식이다. 이 방식은 채널기반 방식이 HTTP 등과 같은 응용 프로토콜의 하위에서 독립적인 채널을 동작시키는데 비해 동등 응용 프로토콜로서 다른 응용 프로토콜에 의해 생성되는 메시지를 암호화하는 방식이며, SHTTP(Secure HTTP)[12]가 이 범주에 속한다. SHTTP는 1994년 EIT 사에서 개발한 보안 프로토콜로서 범용으로 사용될 수 있도록 설계되었으며, 기밀성, 인증, 무결성 등의 보안 요구사항을 지원한다. RSA 공개키 암호알고리즘을 이용하여 클라이언트와 서버 사이의 비밀키 등을 암호화하여 전송한다. SHTTP는 HTML을 수정하지 않고 사용할 수 있다는 장점이 있으나 기존의 HTTP와 호환성이 없기 때문에 SHTTP를 지원하는 웹 서버와 브라우저를 새로 개발해야 한다는 단점을 가지고 있으므로 현재 거의 사용되고 있지 않고 있다.

또한, HTTP 계층과 TCP/IP 계층 사이에 보안 계층을 삽입한 형태를 채널기반 방식이라고 하며, 대표적인 프로토콜로 SSL(Secure Socket Layer)[2] 및 TLS(Transport Layer Security)[13] 등이 있다. 현재 웹보안 프로토콜로 가장 널리 사용되고 있는 SSL은 Netscape사에서 개발한 채널기반의 프로토콜로서 응용계층과 TCP/IP 계층 사이에 보안 계층을 추가한 형태를 취하고 있다. 따라서 SSL은 웹과 같은 특정 응용뿐 아니라 FTP, TELNET 등의 일반적인 인터넷 보안 프로토콜로도 사용될 수 있다는 장점이 있다. SSL은 메시지 내용의 암호화, 서버의 인증 및 무결성 등을 제공한다. SSL은 SSL Record와 SSL Handshake의 두 개의 부계층으로 이루어져 있다. 클라이언트와 서버는 SSL Handshake 프로토콜을 이용하여 한 세션 동안 사용할 암호 알고리즘, 세션키, 인증서 등의 암호 파라미터를 공유하게 되며 이러한 세션 정보는 SSL Record 프로토콜에서 보안 서비스를 제공할 때 사용된다.

SSL 이외에 웹 보안 기술로 제시되고 있는 방법 중에 한 방법으로 웹 브라우저와 웹 서버 외부에 프록시 서버와 서버 게이트웨이를 각각 설치하여 브라우저와 서버가 주고받는 모든 데이터를 암호화하고 복호화하는 방법이다. 이 방법은 기존의 HTTP 서버와 브라우저를 그대로 사용하면서 웹 보안을 제공해준다는 커다란 이점을 가지고 있다. 즉, HTTP request 및 response 메시지가 브라우저와 서버를 떠나 전송되기 직전에 외부 프로그램에 보내지고 암호화된 메시지들은 다시 서버와 브라우저로 보내져서 인터넷을 통해 보내지게 된다. 본 연구에서는 이 방법을 사용하여 웹 보안시스템을 구현하였으며, 프록시 서버(SPC : Secure Proxy Server)와 서버 게이트웨이(SMS : Secure Management Server) 사이에 암호화 채널이 형성되면 본 연구팀에서 개발한 카오스 암호화 알고리즘을 통하여 클라이언트와 서버 사이의 데이터를 암·복호화 한다.

4. 웹 보안시스템 설계

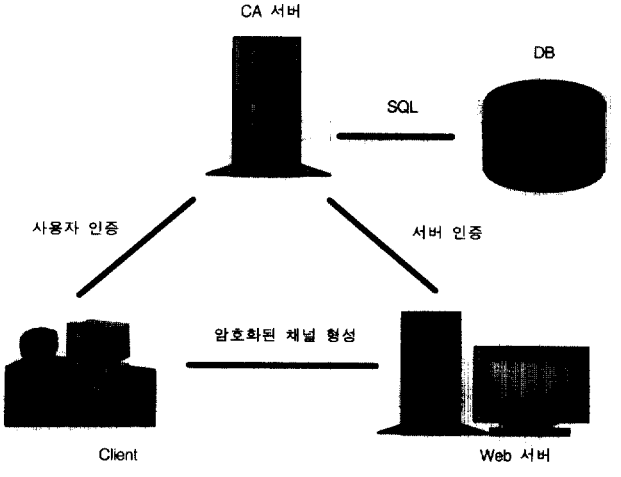
4.1 시스템 설계 기준

본 연구에서 구현하고자 하는 웹 보안 시스템은 다음과 같은 설계기준을 갖는다.

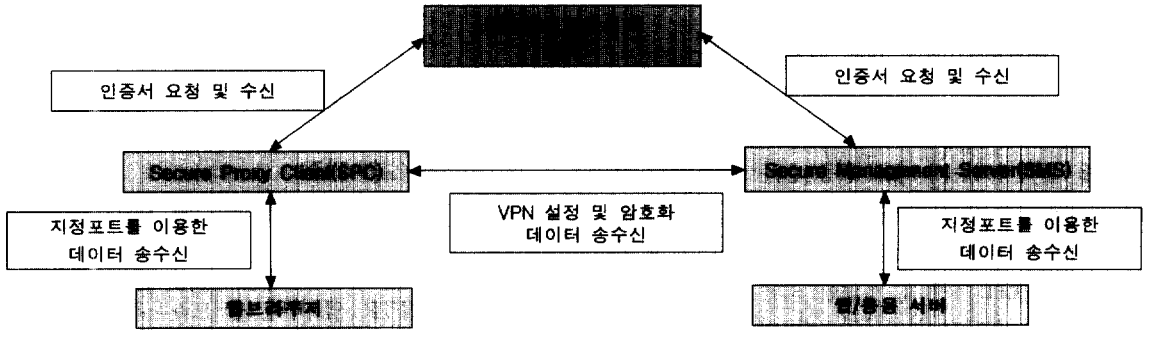
- 기존의 HTTP, 웹 서버, 웹 브라우저에 대한 변경 없이 사용 가능해야 한다.
- 중소규모 기관에서 독자적으로 사용 가능해야 한다.
- 공개키 기반의 웹 보안을 제공한다.
- MS 윈도 계열의 운영체제를 사용한다.
- CA 서버는 별도로 구현하지 않고 MS 윈도 계열과 호환 가능한 CA 서버를 채택하여 본 연구에서 개발하는 시스템과의 인터페이스를 개발한다.
- 모든 형식은 국제 표준을 따른다.

4.2 시스템 구성

이상의 설계기준에 따라 구현하고자 하는 웹 보안 시스템은 인증서버(CA : Certificate Authority), 클라이언트 시스템 및 Web 서버 시스템으로 이루어져 있다. CA 서버는 윈도2000에 MS Access를 탑재한 인증서버이며 웹 서버로는 IIS 서버를 사용하였다. 클라이언트는 Win 95/98이 탑재된 PC이다. 웹 서비스를 제공하는 웹 서버나 웹 서비스를 제공받으려는 웹 브라우저 모두 MS2000 인증서버에 인증서를 요청하여 인증서를 받아야 한다. 웹 서버 시스템에는 SMS가 탑재되고 웹 클라이언트시스템에는 SPC가 설치된다. 각각의 SMS와 SPC가 설치된 웹 서버와 웹 클라이언트 사이에 VPN이 설정되어 암호화된 데이터 송·수신을 제공한다. (그림 3)은 웹 보안 시스템의 개략적인 구성도를 나타내고 있다.



(그림 3) 웹 보안 시스템의 구성도



(그림 4) 웹 보안시스템 데이터 흐름도

4.2.1 CA 서버

인증서는 인터넷처럼 보안성이 떨어지는 네트워크에서 개인 또는 기관의 인증 및 안전한 정보 교환을 위해 사용되는 데이터 집합이다. 인증서는 공개키를 해당 개인키를 가진 엔터티와 안전하게 연결한다. 인증기관이 인증서에 디지털 서명을 하며 사용자, 컴퓨터 또는 서비스에 대해 인증서를 관리할 수 있다. 가장 널리 사용되는 인증서 형식은 ITU-T X.509 국제 표준에 의해 정의된 X.509 v3이다. 1995년에 수정된 X.509 버전에서 최초로 선보인 이 인증서는 Windows 2000 인증서 기반 프로세스에서 사용하는 표준 인증서 형식이다. X.509 인증서에는 인증서가 발급되는 개인이나 엔티티에 대한 정보, 인증서에 대한 정보, 인증서를 발급하는 인증기관(CA)에 대한 선택적 정보가 포함된다. 주제 정보에는 엔티티 이름, 공개키, 공개키 알고리즘, 고유 ID 등이 포함된다. 각 항목들에 대한 정보는 인증서 구조의 특정 필드들에 해당된다. 인증서들은 우편 주소, 전자 우편 주소, 국가, 연령, 성별 등과 같은 다른 정보를 포함할 수도 있다. 이런 추가적인 데이터는 옵션으로 인증서의 특정 형식에 의존한다. <표 1>은 본 연구에서 사용된 인증서 필드의 예를 나타낸다.

〈표 1〉 인증서 형식

버 전	V3
일련 번호	212E 44EE 0000 017C
알고리즘 서명	MD5, RSA
발행자	Certificate Server
발행 날짜	2001년 7월 26일 목요일 오후 5 : 15 : 53
만료 날짜	2003년 11월 17일 월요일 오후 9 : 14 : 01
소유자 이름	홍길동
소유자의 공개키	3048 0241 00C4 3795 650C 95DE A788 …
발행자의 서명	ECE4 1677 068C B05B 4224 8ABC 6469 …

<표 1>은 인증서가 가지고 있어야 할 기본적인 구조를 나타내고 있다. 인증서는 그 자체로서 개인의 신분을 확인할 수 없다. 모든 사람이 위에서 설명된 표준 포맷을 따르는 인증서를 발행할 수 있다. 이 때 필요한 사항은 인증서를 발행한 시스템이 몇 가지의 매우 엄격한 표준들을 따라야 한다는 사실이다. 그렇게 해야만 인증서들이 우리가 원하는 보안 기능을 제공할 수 있게 된다. 한편, 인증서를 관리하는 인증기관은 인증주체로서 사용자나 다른 인증기관에 속한 공개키의 신뢰성을 입증하고 보증하는 책임 기관이다. 인증기관은 서명된 인증서를 통해 공개키를 전체 이름에 연결하고 인증서 일련 번호와 인증서 해지 등의 관리 기능을 가지고 있다. (그림 4)의 MS2000 인증서버(CA)는 클라이언트 및 서버로부터 인증 요청이 오면 인증서들을 발행, 취소, 재발급하고, 인증서들을 저장하기 위한 디렉토리를 제공한다. CA는 인증서를 발행하는 개인과 기관들을 인증하는데 있어서 엄격한 절차들을 따라야 한다. 인증은 CA 안에서 신뢰하고 있는 모든 사람에게 키 매칭 과정이 유효하다는 것을 보장하게 되고, 그리하여 CA가 인터넷의 공증인으로서 기능을 담당한다고 할 수 있다.

인증서 포맷중 CA와 연관된 두 개의 필드를 살펴보도록 하자. 발행자와 발행자의 서명 필드들은 그 인증서 안에 있는 공개키가 인증서의 소유자에게 해당된다는 것을 CA가 보장함을 의미한다. 두 사용자나 두 기관이 인증서들을 교환할 때 그 둘이 인증서들을 발행했던 CA를 신뢰한다면 그들은 상대방의 신분을 확인할 수 있다. 각 인증서는 소유자의 공개키를 포함한다. 그러므로 그 공개키는 인증서의 소유자에게 전달될 데이터를 암호화하는데 사용될 수 있다. 또한, 인증서는 그것을 발행했던 CA의 디지털 서명도 포함한다. 그럼으로써 어떤 사람도 그 인증서를 수정하지 않았다는 것과 그 안에 저장되어 있는 정보가 올바르다는 것이 보증되는 것이다.

4.2.2 SMS

SMS는 서버 응용프로그램이 데이터를 주고받기 전에 데이터의 암·복호화 기능 외에 암호화 알고리즘 선택, 프로토콜 선택, 로그정보 저장 등 다양한 관리 기능을 제공하여 안전한 데이터 전송이 가능하도록 해준다.

- 암호화 및 복호화 데이터 전송기능

클라이언트로부터 전송된 데이터를 복호화하여 서버에

전송하고, 서버로부터 전송된 데이터를 암호화하여 클라이언트에 전송한다.

- 다양한 암호화 알고리즘 선택 기능
  K7, DES, SEED와 같은 대칭키 암호 알고리즘을 선택하여 원하는 알고리즘에 의한 암·복호 기능을 수행한다.
- 프로토콜 선택 기능(HTTP, TELNET, FTP 등)
  다중 프로토콜을 선택하고 지원하여 각각의 접속된 클라이언트에 대한 프로토콜 정보를 기억하고, 그에 따른 서비스를 제공한다.
- 서버 관리 기능
  DBMS와 연동하여 서버 관리자 입장에서 DBMS를 고려하지 않고, 응용 프로그램의 GUI를 통하여 처리할 수 있다.
- 전송 데이터에 대한 로그 정보 저장 및 검색 기능
  선택적인 로그정보 저장모듈을 통하여, 로그정보를 저장하고 검색할 수 있다.

4.2.3 SPC

SPC는 클라이언트 응용프로그램이 데이터를 주고받기 전에 데이터를 암·복호화하여 응용프로그램이 안전한 서비스를 제공받도록 해준다.

- 암호화 및 복호화 데이터 전송기능
  클라이언트로부터 전송된 데이터를 암호화하여 서버에 전송하고, 서버로부터 전송된 데이터를 복호화하여 클라이언트에 전송한다.
- 프로토콜 선택 기능(HTTP, TELNET, FTP 등)
  원하는 프로토콜을 선택하여 그에 따른 서비스를 제공받는다.

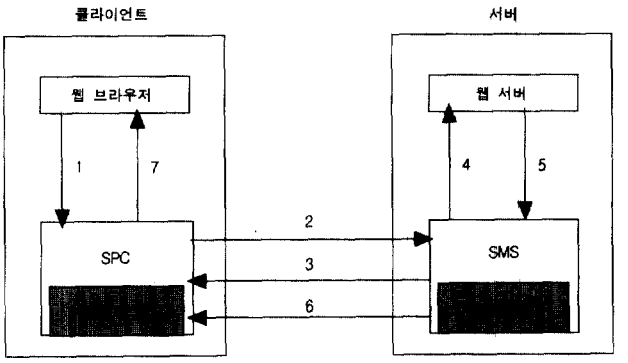
4.3 시스템 동작과정

4.3.1 인증서 발급신청

웹 클라이언트와 서버는 상호 간의 암호화 데이터를 주고받기 전에 CA 서버에 접속하여 인증서를 발급 받아야 한다. 인증서 발급에 관한 일련의 절차는 다음과 같다.

1) CA서버에 접속
2) CA서버에 인증서 신청 요구
3) CA서버는 인증 요청서를 웹 서버/웹브라우저에 전송
4) 웹 서버/웹브라우저는 자신의 키쌍을 생성하고, 인증 요청서를 작성
5) 웹 서버/웹브라우저에서 CA서버로 자신의 공개키와 인증 요청서를 전송
6) CA서버는 수신된 인증 요청서를 확인하여 인증서 발급(공개키 포함)
7) CA서버는 웹 서버/웹브라우저의 인증 요청서 정보와 인증서를 DB에 저장
8) CA서버는 웹 서버/웹브라우저의 인증서를 웹 서버/웹브라우저에 전송
9) 웹 서버/웹브라우저는 CA서버로부터 수신된 인증서를 자신의 비밀키와 함께 저장

인증서 발급이 끝나면 통신하고자 하는 응용 실체 사이에 특별히 설계된 소켓 루틴들을 사용하여 먼저 안전한 통신채널을 확립한 다음, 안전한 통신 채널을 통하여 정보를 교환할 수 있게 된다. 안전한 채널을 확립하기 위해 소켓 루틴은 RSA 공개키 암호화 알고리즘을 이용하여 인증과 세션키 교환 과정을 수행하며, 키 교환과정을 통해 공유되는 세션키를 이용하여 K7을 비롯한 대칭키 암호 통신을 하게 한다. (그림 5)는 본 시스템의 동작 과정을 나타낸다.



(그림 5) 웹 보안 시스템 동작 과정

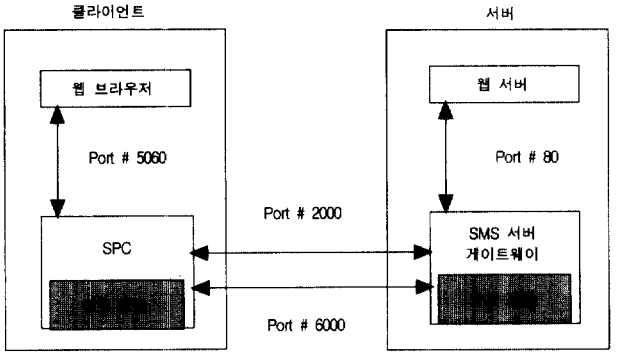
(그림 5)의 시스템 동작과정을 순차적으로 나타내면 다음과 같다.

1) 웹 브라우저는 SPC에 서비스 요청
2) SPC는 SMS에 연결요청
3) SMS의 암호 모듈은 키를 생성하여 SPC에 전송
4) SMS는 웹 서버에 서비스 요청
5) 웹 서버는 요청한 서비스에 해당하는 페이지 제공
6) SMS의 암호 모듈은 해당 페이지를 암호화하여 전송
7) SPC는 송신한 데이터를 복호화하여 웹 브라우저에 전송

4.3.2 시스템 설정 및 포트할당

Explorer 웹 브라우저에서 도구 메뉴의 [인터넷 옵션] 항목을 선택하여 [연결]메뉴의 [LAN 설정]을 선택한 다음 [프록시 서버] 항목을 선택한다. 프록시 서버의 주소와 포트번호에 127.0.0.1과 임의로 정한 포트번호(5060)를 입력한다. 위의 과정을 마치면 클라이언트 시스템의 암호모듈과 웹 브라우저간에 일반적인 HTTP 메시지 전송이 이루어지게 된다. 서버의 경우 SMS를 구동시키면 웹 서버와 SMS 간에 일반적인 HTTP 메시지 전송이 이루어지게 된다. 이때, 기본 HTTP 프로토콜의 포트번호인 80번을 사용하므

로 추가적인 설정은 하지 않아도 된다. 웹 브라우저의 웹 서비스 요청시 구동된 두 데몬 프로그램에 의해 포트번호 2000번의 암호화된 채널이 형성되는데 이 채널을 통하여 제어 메시지를 전송하고 실질적으로 데이터는 별도의 포트 6000번을 통하여 안전한 웹 서비스를 제공하고 제공받게 된다. (그림 6)은 본 시스템의 모듈간의 포트번호를 나타내고 있다.



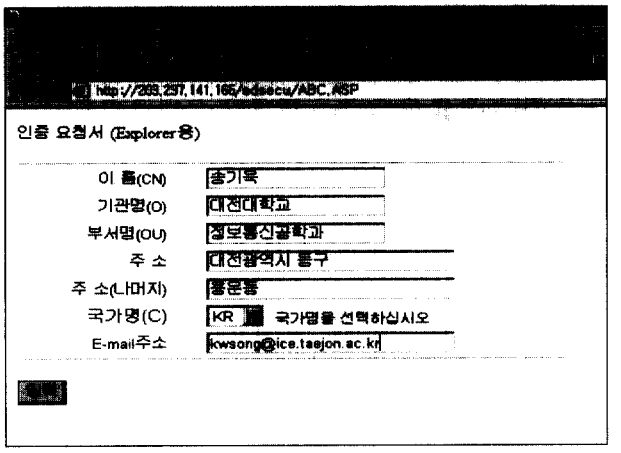
(그림 6) 웹 보안 시스템 모듈간 연결 포트번호

SPC의 설계시 웹 브라우저와의 연결을 위한 포트번호 5060번을 할당하고, 또한 웹 서버에 요청하기 위해서 SMS와의 통신을 위한 포트번호 6000번을 할당하여, 두 개의 포트로 웹 브라우저에서 오는 포트번호 5060번의 메시지를 암호화하여 포트번호 6000번으로 SMS에게 전송하고, SMS로부터 포트번호 6000번으로 수신된 암호화된 메시지를 복호화하여 포트번호 5060번으로 웹 브라우저에 전송한다. SMS 설계도 프록시 서버의 설계와 마찬가지로 두 개의 포트를 할당하여 프록시 서버로부터 포트번호 6000번으로 수신된 암호화된 메시지를 복호화하여 포트번호 80번으로 웹 서버에 전송하고, 포트번호 80번으로 웹 서버로부터 수신된 메시지를 암호화하여 포트번호 6000번으로 SPC에 전송한다. 또한, 포트번호 2000번을 사용하여 세션키 분배, 연결설정 등과 같은 관리채널로 사용한다.

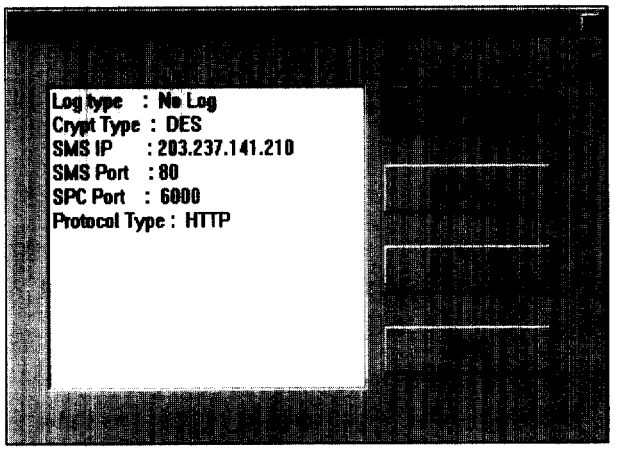
5. 시스템 구현결과

(그림 7)은 클라이언트가 인증을 받기 위하여 인증서버에 전송하는 인증요청서 형식을 나타낸 것이다. 이 인증서 형식은 X.509 표준을 수용하여 작성되었으며 사용자 이름, 기관명, 주소, e-mail 주소 등을 포함하고 있다.

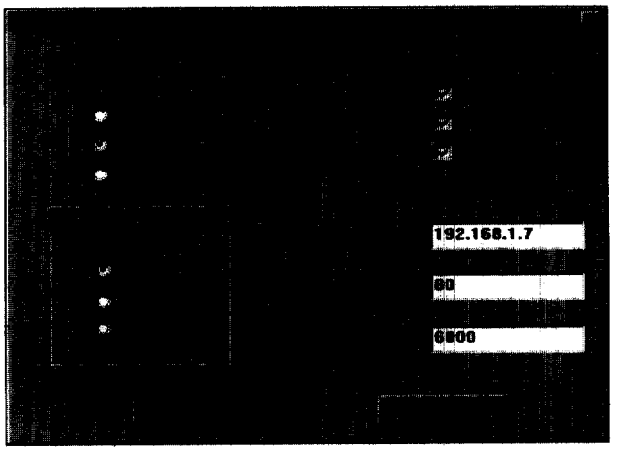
SMS를 실행하여 사용자명과 패스워드를 입력하여 로그온하면 (그림 8)과 같은 메인 메뉴가 나타난다. 메인 메뉴의 리스트박스는 현재 설정되어 있는 로그 형태, 암호화 알고리즘, SMS IP 주소, 포트번호, 프로토콜 등의 환경설정 값을 보여준다. 환경설정 버튼을 누르면 (그림 9)와 같은 환경설정 대화상자가 나타나고, 사용자가 원하는 환경을 설정할 수 있다.



(그림 7) 인증요청서 포맷



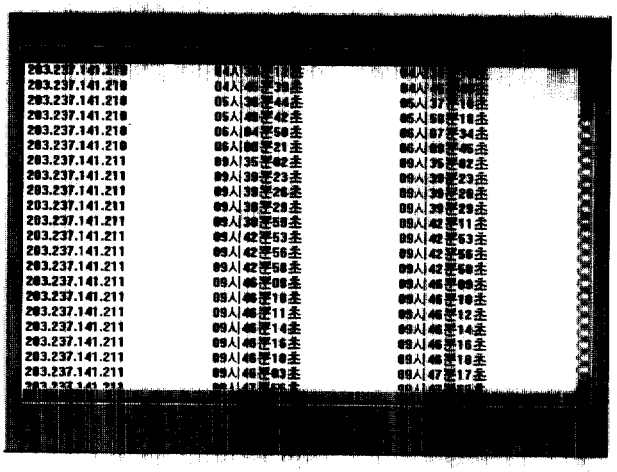
(그림 8) SMS 메인 메뉴



(그림 9) SMS 환경설정

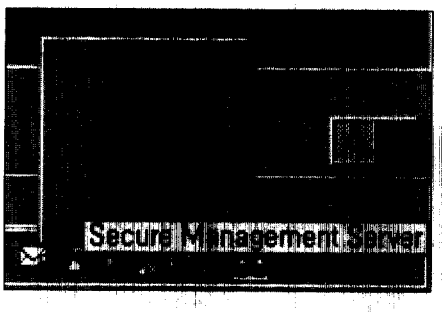
환경 설정에서 로그형태는 서버로 수신되는 전송 데이터의 기록시 로그 상황을 나타내고 암호 알고리즘은 클라이언트와 서버가 세션을 설정하여 데이터를 전송할 때 사용하고자 하는 비밀키 암호화 알고리즘을 나타낸다. 프로토콜은 제공하기를 원하는 프로토콜들을 나타내며, Dest. Server IP는 웹 서버의 IP 주소를 나타내며 Dest. Server Port

와 SMS Port는 각각 웹 서버의 포트번호와 SPC와 통신할 때 사용하는 포트 번호를 나타낸다. (그림 10)은SMS에 접속한 SPC에 대한 로그정보를 나타낸 것이다.



(그림 10) SMS 로그정보

(그림 8)의 SMS 메인 메뉴에서 시작 버튼을 누르면 대화상자는 없어지고 SMS 프로그램은 트레이 아이콘을 생성하고, SPC의 요청을 기다린다. (그림 11)은 트레이 아이콘 및 마우스 우측 버튼을 클릭했을 때 나타나는 세부 메뉴를 나타낸 것이다.



(그림 11) SMS 트레이 아이콘 메뉴

한편 클라이언트 시스템에 설치되어 있는 SPC를 실행하면 SMS와 같은 인증 절차를 거쳐 사용자가 원하는 환경으로 설정할 수 있다. 즉, 사용하고자 하는 브라우저, SMS IP 주소, 포트번호, 프로토콜 등을 선택할 수 있다. 이와 같이 SMS와 SPC의 환경설정이 완료되면 웹 브라우저로부터 SPC를 거쳐 SMS, 그리고 웹 서버로 통하는 암호화 채널이 설정되어 웹 브라우저와 웹 서버간에 주고받는 데이터는 설정된 암호화 알고리즘에 따라 암호화되어 안전한 통신을 제공하게 된다.

6. 결 론

인터넷 사용 저변의 확대와 멀티미디어 정보 처리 기술의 발전 및 웹 기술의 보급으로 다양한 응용서비스가 제공되고 있다. 그러나 인터넷이 전자상거래 등에 응용되면서 웹 기술은 많은 보안상의 문제점들을 내재하고 있다. 본 논문에서는 카오스 이론에 기초하여 기존의 DES와 SEED 암호화 알고리즘을 대체할 수 있는 카오스 암호화 알고리즘을 개발하여 이를 웹 보안시스템에 적용하였다. 이는 정보신호에 카오스 신호를 삽입하면 카오스적인 성질을 가지게 되고 수신측이 송신측과 동일한 카오스 시스템을 사용하면 두 시스템간 동기가 가능하여 수신측에서 카오스 신호를 제거하여 정보신호를 복원해낼 수 있다는 원리에 기초한다. 개발한 카오스 암호화 알고리즘은 기존의 카오스 암호화 알고리즘에 비하여 처리속도 및 비도에서 성능이 우수하다.

본 논문에서 구현한 웹 보안시스템은 윈도 2000 환경의 인증서버시스템, 클라이언트시스템 및 서버 시스템으로 구성된다. 인증서버시스템은 X.509 형식에 따라 사용자를 인증하고 인증서를 발급하며 인증서가 발급되면 클라이언트 시스템은 공개키와 비밀키 쌍을 생성하여 레지스트리에 저장한다. 클라이언트 시스템의 프락시 클라이언트(SPC)는 서버시스템의 관리서버(SMS)와 연결 설정하여 자신의 공개키를 전송하고 SMS는 암호화 채널에 사용할 세션키를 생성하여 이를 암호화하여 SPC에게 전송한다. 이외에 SMS는 암호화 알고리즘, 전송프로토콜 및 LOG 선택기능을 제공하도록 구현하였다. SPC와 SMS 간에 암호화 채널이 형성되면 웹 브라우저가 요청하는 데이터는 사전에 할당된 로컬 포트를 통하여 SPC로 전달되고 SPC는 이를 암호화하여 할당된 포트를 사용하여 SMS로 전송하며, SMS는 이를 복호화하여 웹 서버로 전송한다. 한편, 웹 서버로부터의 응답은 SMS에서 암호화되어 SPC로 전송되고 SPC는 이를 복호화하여 웹브라우저로 전송한다. 따라서 클라이언트와 웹 서버는 설정된 암호화 채널을 통하여 안전한 데이터를 전송할 수 있게 된다. 특히, 본 연구에서 구현한 웹 보안시스템은 기존의 HTTP 뿐 아니라 FTP, TELNET 등의 프로토콜을 선택하여 암호화 데이터 전송이 가능하다. 또한, 암호화 채널로 데이터를 전송할 때 사전에 SMS의 환경 설정에서 설정한 비밀키 암호화 알고리즘에 따라 암호화 기능을 제공하며 비밀키 암호화 알고리즘으로는 Chaos, SEED, DES 등의 알고리즘 선택이 가능하다. SMS는 다중의 웹 서버로의 연결요청을 처리할 수 있도록 구현하였으며 각 연결설정에 대한 세션별 관리가 가능하도록 하였다.

본 연구에서 구현한 웹보안 시스템은 윈도 2000에서 제공하는 인증서버를 이용한 사용자 인증을 통하여 PC 클라이언트와 NT 기반의 웹 서버간의 전송 데이터에 대한 VPN 채널 설정을 통한 안전한 데이터 전송을 보장한다. 개발한 웹 보안시스템은 웹 서버의 환경설정을 통하여 기관내의 등록된 사용자들만이 웹 서버에 접근할 수 있도록 하여 인트라넷에서의 웹 보안시스템으로 활용할 수 있다. 또한, 웹 서버뿐 아니라 웹 환경에서 사용되고 있는 ERP 시스템 및

그룹웨어 시스템의 보안에도 널리 활용될 수 있을 것으로 사료된다.

참 고 문 헌

[1] J. Anderson et al., "Protocol and architecture for IP optical networking," Bell Labs Tech. J., Feb.-Mar. 1999.

[2] A. O. Freier, P. Karlton, and P. C. Kocher, "The SSL Protocol Version 3.0," www.netscape.com/eng/ssl3, Nov. 1996.

[3] ITU-T Rec. X.509, Information technology-Open Systems Interconnection-The Directory : Public-key and attribute certificate frameworks, March. 2000.

[4] National Bureau of Standards, "Data Encryption Standard," FIPS Pub., 46, 1977.

[5] KISA, 128비트 블록 암호화 알고리즘(SEED) 개발 및 분석보고서, 1998.

[6] K. M. Coumo and A. V. Oppenheim, " Robustness and Signal Recovery in a Synchronized Chaotic System," Int. J. Bifurcation and Chaos, Vol.3, pp.1629-1638, 1993.

[7] L. Pecora and T. Carroll, "Synchronization in Chaotic Systems," Phys. Rev. Lett, Vol.64, pp.821-823, 1990.

[8] K. M. Coumo and A. V. Oppenheim, "Circuit Implementation of Synchronized Chaos with Applications to Communications," Phys. Rev. Lett, Vol.71, pp.65-68, 1993.

[9] A. Maritan and J. Banavar, "Chaos, Noise, and Synchronization : Reply," Phys. Rev. Lett, Vol.73, No.21, pp.2932, 1994.

[10] A. S. Pikovsky, "Chaos, Noise, and Synchronization : Comment," Phys. Rev. Lett, Vol.73, No.21, pp.2931, 1994.

[11] R. Rim, D. U. Hwang, I. Kim and C. M. Kim, "Chaotic transition of random dynamical systems and chaos synchronization by common noises," Phys. Rev. Lett, Vol. 85, pp.2304-2307, 2000.

[12] E. Rescorla and A. Schiffman, The Secure HyperText Transfer Protocol, RFC 2659, Aug. 1999.

[13] T. Dierks and C. Allen, "The TLS Protocol Version 1.0," IETF RFC 2246, jAN. 1999.

이 봉 환

e-mail : blee@dragon.taejon.ac.kr

1985년 서강대학교 전자공학과 졸업(학사)

1987년 연세대학교 대학원 전자공학과 졸업(공학석사)

1993년 텍사스 A&M 대학교 전기공학과 졸업(공학박사)

1987년~1995년 한국통신 연구원

1995년~현재 대전대학교 컴퓨터정보통신공학부 조교수

관심분야 : 컴퓨터네트워크, 광인터넷, 네트워크보안, GRID 보안

김 철 민

e-mail : chmkim@mail.paichai.ac.kr

1982년 서강대학교 물리학과(학사)

1983년 서강대학교 대학원 물리학과 (이학석사)

1985년 서강대학교 대학원 물리학과 (이학박사)

1986년~현재 배재대학교 물리학과 교수

1991년~1992년 CREOL, University of Central Florida, 방문교수

1998년~현재 과기부 창의적 연구진흥과제, 광혼돈제어연구단 단장

관심분야 : 광학, 혼돈 이론, 레이저, 혼돈계의 제어, 전자회로에서의 혼돈 현상, 혼돈을 이용한 암호시스템 등

윤 동 원

e-mail : dwyoon@dragon.taejon.ac.kr

1989년 한양대학교 전자통신공학과 졸업 (공학사)

1992년 한양대학교 전자통신공학과 졸업 (공학석사)

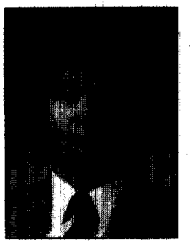
1995년 한양대학교 전자통신공학과 졸업 (공학박사)

1995년~1997년 동서대학교 정보통신공학과 전임강사

1997년~현재 대전대학교 컴퓨터정보통신공학부 조교수

2001년~현재 펜실베니아주립대 연구교수

관심분야 : 무선통신, 이동통신, 디지털통신 시스템



채 용 웅

e-mail : yychai@kmu.ac.kr

1985년 서강대학교 전자공학과 졸업 (학사)

1991년 Oklahoma State Univ. 대학원 Electrical Engr.(공학석사)

1994년 Oklahoma State Univ. 대학원 Electrical Engr.(공학박사)

1995년~1997년 삼성전자

1998년~현재 계명대학교 전자공학과 부교수

관심분야 : 집적회로설계 등

김 현 곤

e-mail : hyungon@etri.re.kr

1992년 금오공과대학교 전자공학과 학사

1994년 금오공과대학교 전자공학과 석사

1999년 충남대학교 전자공학과 박사과정 수료

1994년~현재 한국전자통신연구원 정보보호본부 AAA정보보호연구팀장

관심분야 : 이동통신에서의 정보보호, Mobile IP, AAA