해쉬함수와 스트림 암호기의 개발 및 GSM 보안 시스템에의 적용

김 범 식[†]·신 인 철[††]

요 약

무선 통신 기술의 발달로 인해 이동통신의 사용이 전보다 훨씬 편리해 졌으며 최근 이동 통신 사용자들은 언제, 어디서나 누구와도 통신이 가능하게 되었다. 하지만 이동 통신은 무선채널을 사용함으로서 통신 당사자는 심각한 보안 위협에 노출된다. 안전한 통신채널을 제공하기 위한 방법은 이동통신망의 발전에 필수적이라 하겠다. 본 논문에서는 GSM 망으로의 로밍 서비스를 제공하기 위한 인증 및 메시지 암호 알고리즘에 대하여 논한다. GSM 암호 시스템에 적용 가능한 인증 및 암호 알고리즘을 제안하기 위해 GSM망의 보안구조에 관하여 간략히 소개한다. 사용자 인증 알고리즘은 새로운 해쉬함수를 그리고 메시지 암호화를 위한 스트림 암호기는 LFSR(Linear Feedback Shift Register)을 사용한다. 각 알고리즘은 C언어로 구현하였으며 IBM PC 상에서 시뮬레이션 하였다. 또한 통계적 분석 기법을 사용하여 개발된 알고리즘의 출력 특성을 분석한다.

Development of a Hash Function and a Stream Cipher and Their Applications to the GSM Security System

Bum-Sik Kim[†] · In-Chul Shin[††]

ABSTRACT

With the advance of wireless communications technology, mobile communications have become more convenient than ever. Nowadays, people can communicate with each other on any place at any time. However, because of the openness of wireless communications, the way to protect the privacy between communicating parties is becoming a very important issue.

In this paper, we present a study on the authentication and message encryption algorithm to support roaming service to the GSM network. To propose an authentication and message encryption algorithm applicable to the GSM system, the security architecture of the GSM outlined in the GSM standard is briefly introduced. In the proposed cryptosystems we use a new hash function for user authentication and a stream cipher based on Linear Feedback Shift Register(LFSR) for message encryption and decryption. Moreover, each algorithm is programmed with C language and simulated on IBM-PC system and we analyze the randomness properties of the proposed algorithms by using statistical tests.

1. 서 론

이동통신 기술의 발달로 인하여 이동통신서비스 가입자 수의 증가와 함께 전파를 통신매체로 이용하는 이동 통신의 특성으로 인한 불법적인 서비스 이용이나 도청 또는 추적을 통한 불법적인 행위와 같은 각종 통신 범죄 행위 등도 증가하고 있다. 특히 통화 도용은 이동통신서비스 사업자에게는 요금징수와 관련한 피해를 주며, 가입자에게는 요금체계에 대한 불신감을 주

† 준 회 원 : 단국대학교 대학원 전자·컴퓨터공학과
†† 정 회 원 : 단국대학교 전자·컴퓨터공학과 교수
논문접수 : 2000년 4월 7일, 심사완료 : 2000년 7월 31일

어 이동 통신 발달에 큰 장애요인이 되고 있다. 따라서 이러한 문제점을 방지하기 위한 보호 서비스가 필요한데 이를 신분인증 서비스라 한다[1]. 이동 통신에서의 인증 서비스는 이

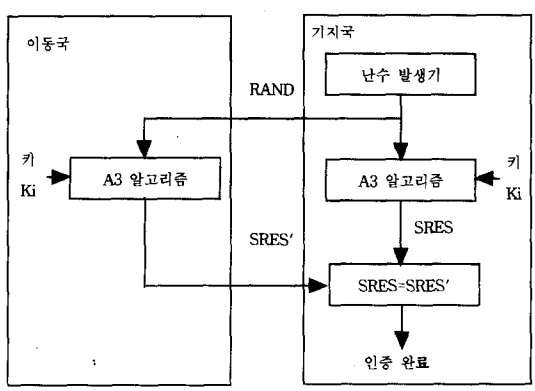
동국과 기지국간의 신분을 확인하기 위하여 서로 관련 정보를 교환하는 것으로 서로 공유한 비밀 데이터가 일치하는지를 확인하는 과정을 의미한다.

디지털 이동통신 시스템의 대표적인 표준안으로는 IS-95[2], GSM[3] 그리고 PACS[4]등이 있다. ETSI (European Telecommunication Standard Institute)에 의해 제안된 유럽의 TDMA(Time Division Multiple Access) 이동통신망 표준인 GSM(Global System for Mobile Communication)에서는 SIM(Subscriber Identity Module)이라는 스마트 카드를 도입하여 단말기 번호(IMEI, International Mobile Equipment Identity)와 SIM 번호(IMSI, International Mobile Subscriber Identity)를 분리시킴으로서 높은 보안과 개인 이동성, 국제간 로밍 서비스 및 전화 이외의 다른 서비스와의 연계가 가능하도록 하였다. 이러한 장점을 살리기 위해 다양한 사업자들이 SIM을 자신들의 망에 도입하려는 움직임이 활발히 진행 중이다. 그러나 GSM과 호환 가능한 SIM의 도입에 있어서 문제가 되는 것은 GSM에서 제공하는 보안을 위한 암호 알고리즘들이 수출의 제한으로 인하여 공개되지 않으며 각국에서의 정의 사항으로 정해져 있다. GSM에서의 암호 알고리즘으로는 사용자 인증을 위한 A3 알고리즘, 메시지 암호를 위한 A5 알고리즘 그리고 메시지 암호에 사용되어지는 비밀키를 생성하기 위한 A8 알고리즘 등이 있다. 그런데 A3과 A8 알고리즘은 국내 문제로 정의되어있으며 동일한 알고리즘을 사용하도록 권고되고 있다. 반면 A5는 메시지와 음성을 실시간으로 암호화 및 복호화하는 알고리즘이므로 이동국이 방문한 지역과 이동국이 등록된 지역 모두 동일한 알고리즘을 사용하여야만 로밍이 가능하므로 ETSI로부터 제공되어지는 알고리즘을 사용하여야 한다[5].

GSM에서 제공하는 보안을 위한 알고리즘들의 수출(특허) 제한으로 인해 로밍 서비스를 제공하기 위해서는 보안 알고리즘들이 서비스 제공자에 의해 개발, 제공되어져야 한다. 우리나라 역시 IMT2000으로의 진화 과정에서 GSM시스템에 대한 연구가 필요하며 우리나라 고유의 암호 알고리즘들이 개발되어져야 할 것이다. 따라서 본 논문에서는 보안 측면에서 GSM으로의 로밍 서비스를 제공하기 위한 인증 및 암호키 생성알고리즘인 A3/A8 알고리즘과 메시지 암호화를 위한 A5 알고리즘을 개발하였다. 이를 위해 GSM에서의 인증 절차 및 A3, A8, A5 알고리즘에 관하여 알아본다. 제안한 A3/A8 알고리즘은 새로운 전용해쉬함수를 사용하였고 메시지 암호화를 위한 A5알고리즘은 선형귀환 쉬프트 레지스터에 기반한 스트림 암호기를 개발하였다. 개발한 스트림 암호기는 GSM에서 메시지 암호 알고리즘으로 사용되어질 수 있도록 개발하였으나 위에서 언급한 바와 같이 직접 사용되어질 수는 없으며 스트림 암호의 다른 응용분야에서 사용되어질 수 있을 것이다. 본 논문의 결과는 이동통신 시스템의 암호 및 인증 알고리즘의 개발에 기여할 것으로 기대되며 또한 다른 암호 응용분야 즉, 디지털 서명, 워터마킹 등과 같은 분야에 응용되어 질 수 있을 것이다. 제안된 알고리즘은 C언어로 구현하였으며 IBM PC 상에서 시뮬레이션 하였다. 또한 제안된 알고리즘의 출력 수열에 대한 랜덤성 여부를 통계적 테스트를 사용하여 분석하였다. 이를 위해 FIPS140-1에 제안된 빈도 테스트, 시리얼 테스트, 포카 테스트, 런 테스트, 자기상관 테스트를 수행하였다. 실험 결과 제시된 모든 테스트를 통과하였으며 따라서 제안된 알고리즘의 출력 수열이 통계적으로 안전함을 알 수 있었다.

2. GSM 시스템의 보안구조

2.1 인증 절차



(그림 1) 가입자 인증 절차

인증 절차는 (그림 1)과 같은 유일시도응답(Unique Challenge Response) 방식으로 수행되며, 인가 받은 가입자가 주어진 RAND (Random Number)로 SRES

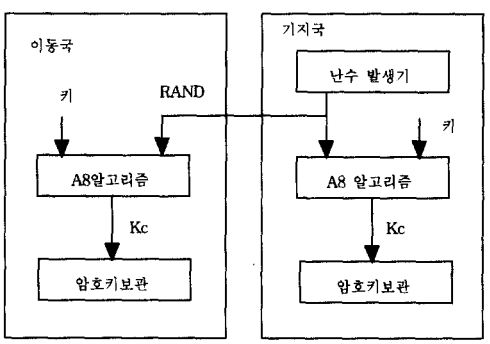
(Signed Response)를 생성해서 응답하게 된다. 가입자 확인은 인증 절차보다 선행되어야 한다. 인증 절차는 기지국에서 먼저 난수를 발생하여 이동국으로 전달함으로써 시작되며, 기지국에서 난수 RAND와 키 Ki를 A3 인증 알고리즘의 입력 데이터로 사용하여 SRES를 계산한다. 이와 마찬가지로 이동국에서는 수신한 난수 RAND와 Ki를 사용하여 SRES'를 생성하고 이를 기지국으로 보낸다. 기지국에서는 생성된 SRES와 SRES'를 비교하여 같을 경우에만 인가된 가입자로 인증한다.

2.2 인증을 위한 A3 알고리즘

인증 알고리즘은 GSM에서 표준화되어 있지 않은 관계로 통신망 관리자가 적당한 알고리즘을 선택하여야 한다. PLMN(Public Land Mobile Network)의 보안 레벨은 통신망 관리자의 보안 인식에 달려 있으며, 인증 알고리즘은 고비도의 단방향 함수로서 주어진 RAND와 SRES로부터 인증키를 유출하기가 불가능하도록 설계되어야 한다. 알고리즘 A3의 입출력 조건은 RAND가 128비트이고 SRES는 32비트로 표준화되어 있다. 인증 센터는 인증 파라미터 RAND와 SRES를 발생하고 각 가입자 정보를 보관해야 하고 인증 파라미터의 발생은 가입자가 등록된 HPLMN(Home PLMN) 또는 VPLMN(Visited PLMN)에서 이루어진다. 이것은 인증 파라미터가 수시로 타지역 PLMN에 전달되어야 함을 의미한다.

2.3 암호키 생성을 위한 A8 알고리즘

데이터의 암/복호화를 위해 사용되는 암호키 Kc 생성 과정은 (그림 2)와 같다. 암호키는 인증 절차가 완료된 후에 생성되며, 키 변경 주기는 인증 빈도 수에

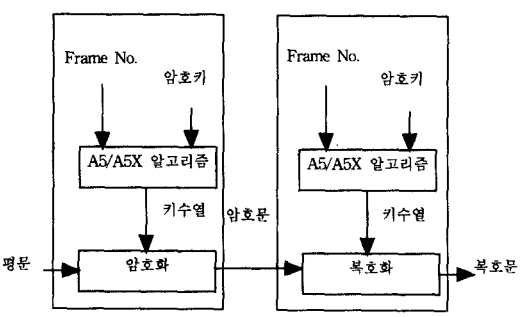


(그림 2) 암호키 생성

따르게 되며 통신망 관리자에 의해 결정된다. A8 알고리즘은 A3 알고리즘과 동일한 입력 파라미터 RAND와 Ki를 사용하여 암호키를 생성한다. 암호키가 무선 채널에 노출되지 않도록 하기 위해 인증 센터와 스마트 카드에 실장된 A8 알고리즘으로 생성하여 사용한다. A8 알고리즘은 GSM에서 표준화되지 않았지만 외부 파라미터가 A3 알고리즘과 동일하므로, 단일 알고리즘으로 구성하도록 권고되고 있다.

2.4 데이터 암호를 위한 A5 알고리즘

데이터의 암/복호화 과정을 살펴보면, 송신 데이터를 PN(Pseudo Noise)비트와 배타적-합(Exclusive-OR)을 하여 암호화한 후에 전송하며, 수신된 데이터는 다시 PN비트와 원래의 신호를 배타적-논리합하여 복호화하게 된다. 암호 알고리즘은 유럽 전역 어디에서나 동일 이동국으로 통화 가능해야 한다는 요구조건에 따라 표준화가 이루어졌다. 암호 알고리즘 입력 파라미터로는 A8 알고리즘으로 생성된 64비트 암호키와 TDMA 프레임 번호가 사용되며, 이 입력 파라메터를 사용하여 A5 알고리즘에서는 4.615msec 마다 114개의 PN비트 수열이 생성되며, TDMA 프레임 번호는 0에서 2715647사이 값을 갖게 되므로 약 209분 이내에 PN수열이 주기적으로 반복된다. 따라서 209분 이내의 일정한 시간마다 키 변경이 연속적으로 이루어져야 된다. 암호화 알고리즘 A5는 상호연동성을 위해 각 나라마다 같아야 하며 엄격한 저작권 관리하에 ETSI로부터 얻을 수 있다.



(그림 3) 암호화 및 복호화 과정

GSM시스템에서는 보안 기능 실현에 <표 1>과 같은 특성을 갖는 3가지 암호 알고리즘을 사용하고 있다.

〈표 1〉 GSM 시스템의 보안구조

종 류	용 도	특 성
A3	가입자 I.D.인증 알고리즘	입력 : RAND 128비트 + K_i 128비트 출력 : SRES 32비트
A5	암호 알고리즘	입력 : 암호키 K_c 64비트 출력 : 114비트
A8	키생성 알고리즘	입력 : A3와 동일 출력 : 암호키(K_c) 64비트

3. 새로운 A3/A8, A5 알고리즘

3.1 해쉬함수의 정의

해쉬함수(hash function), 특히 암호학적 해쉬함수는 임의의 유한 길이 비트열을 고정된 길이의 비트열로 대응시키는 함수이다. 이 고정된 길이의 출력 값을 해쉬값(hash value), 메시지 다이제스트(message digest value)라 한다. 해쉬함수 h와 입력 x가 주어졌을 때, $h(x)$를 계산하는 것이 용이해야 한다. 즉, 암호학적 일방향 해쉬함수는 다음 성질을 만족해야 한다.

- 원상 저항 (preimage resistance)

해쉬함수 h와 해쉬값 y가 주어졌을 때, $h(x)=y$을 만족하는 입력 x을 찾는 것이 계산상 실행 불가능하여야 한다.

- 이차-원상 저항 (second preimage resistance)

해쉬함수 h와 입력 x에 대응하는 해쉬값 y가 주어졌을 때 $h(x)=h(x')$을 만족하는 입력 x'을 찾는 것이 계산상 실행 불가능해야 한다.

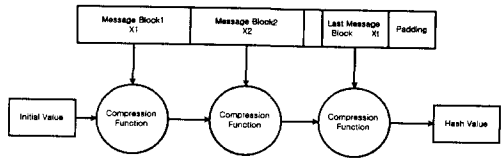
- 충돌 저항 (collision resistance)

해쉬함수 h가 주어졌을 때 $h(x)=h(x')$을 만족하는 입력 x와 x'을 찾는 것이 계산상 실행 불가능해야 한다.

거의 모든 해쉬함수의 처리 과정은 입력을 연속적인 고정된 블록들로 나누어 처리함으로서 임의의 길이 입력을 해쉬하는 반복적인 처리과정이다. 먼저 입력 X는 블록 길이의 배수가 되도록 패딩(padding)되고 t개의 블록 X_1에서 X_t로 나누어진다. 해쉬함수 h는 다음과 같이 기술된다.

$$H_0 = IV$$
$$H_i = f(H_{i-1}, X_i), \quad 1 \le i \le t;$$
$$h(X) = H_t$$

여기서 f는 압축함수 (compress function)이며, H_i는 단계 $i-1$과 단계 i 사이의 연쇄 변수 (chaining variable)이고, IV는 초기값 (initial value)이다. 압축함수를 사용한 반복적인 해쉬함수의 일반적인 구조는 (그림 4)와 같다.



(그림 4) 반복적인 해쉬함수의 일반적인 구조

해쉬값의 계산은 연쇄 변수에 의존한다. 해쉬 계산을 시작할 때, 이 연쇄 변수는 알고리즘의 일부로 명시된 고정된 초기값을 가진다. 압축 함수는 해쉬 되어질 메시지 블록을 입력으로 받아 이 연쇄 변수의 값을 갱신한다. 이 과정이 모든 메시지 블록에 대해 순환적으로 반복되고, 연쇄 변수의 마지막 값이 그 메시지에 대한 해쉬값으로 출력된다[6, 7]. 해쉬함수는 내부 압축함수로 어떤 구조를 사용하느냐에 따라 3가지로 분류된다.

① 블록 암호 기반 해쉬함수
② 모듈러 연산 기반 해쉬함수
③ 전용해쉬함수

전용 해쉬함수는 빠른 처리 속도를 가지고 다른 시스템 서브 요소에 무관하도록 해쉬을 위해 특별히 설계된 함수들이다. 현재까지 제안된 전용해쉬함수는 대부분 1990년에 Rivest에 의해 설계된 MD4[8]에 기반한 구조를 가진다. 현재 널리 사용되는 MD계열 해쉬함수로는 MD5[9], SHA-1[10], RIPEMD-160[11], HAVAL[12]등이 있다.

특정 해쉬함수가 주어지면, 안전한 해쉬함수의 입증을 위해 해쉬함수를 공격하는 복잡도에 관한 하한을 증명할 수 있는 것이 바람직하지만 실제 그러한 방법은 거의 알려져 있지 않고 대부분의 경우에 적용 가능한 알려진 공격의 복잡도가 해쉬함수의 안전성으로 고려되어진다. 해쉬값이 균등한 확률 변수라고 가정하면, 다음은 잘 알려진 사실이다[6].

- n비트 해쉬함수 h에 대해, 2^n 연산으로 원상(pre-image)과 2nd-원상(second preimage)를 발견하기 위한

추측 공격(guessing attack)을 기대할 수 있다.

● 메시지를 선택할 수 있는 공격자에 대해, 생일 공격(birthday attack)은 약 $2^{n/2}$연산으로 $hash(M) = hash(M')$인 충돌 메시지 쌍 M, M'을 발견할 수 있다.

n 비트 해쉬함수가 다음 두 성질을 만족한다면, 이상적인 안전성을 가진다.

① 해쉬값이 주어지면, 원상(preimage)와 이차-원상(second preimage)발견은 약 2^n연산을 요구한다.

② 충돌발견은 약 $2^{n/2}$ 연산을 요구한다[6].

3.2 새로운 A3/A8 알고리즘

본 논문에서 제안한 전용해쉬함수는 32비트 프로세서에서 최적의 성능을 갖도록 설계되었다. 빠른 연산을 위해 CPU 기본 연산인 덧셈, 뺄셈, 곱셈, 배타적 논리합 연산을 사용하였다. MD계열 전용해쉬함수의 경우 비선형성을 높이기 위하여 부울함수를 사용하였으나 본 논문에서는 x^{-1} 연산을 사용하였다. 일반적으로 역원을 구하는 연산은 시간이 많이 걸리나 GF(2^8)에서의 역원이므로 미리 연산하여 참조 테이블을 구성하였다. 모든 연산은 32비트 단위로 이루어지며 6개의 32비트 레지스터 a, b, c, d, e, f는 연쇄변수로서 최종 해쉬값을 갖는다. 이 레지스터들은 h_0 값으로 초기화되는데 그 값은 다음과 같다.

```
a = 0x01234567; b = 0xefcdab89; c = 0x98badcef;
d = 0x10325476; e = 0xc3d2e1f0; f = 0x5a3cf01d;
```

256 비트 메시지 블록은 32비트 단위로 나뉘어져 x0, x1, x2 ... x7 레지스터의 초기값으로 사용되어지고 이어지는 연산에 의해 h_i는 h_{i+1}로 갱신된다. 따라서 256비트 메시지는 A3/A8 알고리즘의 입력에 해당된다. 본 알고리즘은 3번의 패스로 이루어지며 그 사이에 키 스케줄링이 이루어진다. 최종적으로 feedforward 단계에서는 레지스터 a, b, c, d, e, f의 현재값과 그들의 초기값에 대한 연산에 의해 최종 해쉬값 h_{i+1}을 만들어낸다. 알고리즘을 설명하기 위한 식들은 C 프로그래밍 언어의 표현방식을 사용하였다.

3.2.1 알고리즘의 구조

```
save_abcdef
pass(a,b,c,d,e,f, 3)
key_schedule
pass(a,b,c,d,e,f, 5)
key_schedule
pass(a,b,c,d,e,f, 7)
feedforward
```

① save_abcdef는 feedforward 단계에서 사용할 h_i 초기 값을 저장한다.

```
aa = a; bb = b; cc = c; dd = d; ee = e; ff = f;
```

② pass(a, b, c, d, e, f, mul)은 다음과 같이 구성된다.

```
round(a, b, c, d, e, f, x0, mul);
round(b, c, d, e, f, a, x1, mul);
round(c, d, e, f, a, b, x2 ,mul);
round(d, e, f, a, b, c, x3, mul);
round(e, f, a, b, c, d, x4, mul);
round(f, a, b, c, d, e, x5, mul);
round(a, c, e, b, d, f, x6, mul);
round(f, b, d, a, c, e, x7, mul);
```

round(a, b, c, d, e, f, X, mul)은 다음과 같다

```
f ^= X;
a -= Gen_32(f, f, f, f);
f ^= a;
b += Gen_32(f, f, f, f);
b *= mul;
f ^= b;
c += Gen_32(f, f, f, f);
c *= mul;
f ^= c;
d += Gen_32(f, f, f, f);
d *= mul;
f ^= d;
e += Gen_32(f, f, f, f);
e *= mul;
```

여기서 Gen_32( ) 함수는 32비트 레지스터 4개를 입력으로 받아 각 레지스터의 첫 번째, 두 번째, 세 번째, 네 번째 8비트를 S-box의 입력으로 사용하고 이에 대응하는 S-box출력을 가지고 32비트 값을 만드는 함수이다.

③ key_schedule은

```
x0 -= x7 ^ 0xA5A5A5A5;
x1 ^= x0;
x2 += x1;
x3 -= x2 ^ ((~x1) << 7);
x4 ^= x3;
x5 += x4;
x6 -= x5 ^ ((~x4) >> 23);
x7 ^= x6;
x0 += x7;
x1 -= x0 ^ ((~x7) << 7);
x2 ^= x1;
x3 += x2;
```

```
x4 -= x3 ^ ((~x2) >> 23);
x5 ^= x4;
x6 += x5;
x7 ^= x6 ^ 0x01234567;
```

다음과 같다. 여기서 >>, <<은 좌, 우 논리적 쉬프트 연산자이다.

④ feedforward 은

```
a ^= aa; b -= bb; c += cc;
d ^= dd; e -= ee; f += ff;
```

이다. 여기서 a, b, c, d, e, f레지스터는 192비트의 중간 해쉬 값인 h_{i+1}이며 알고리즘의 종료 후 최종 해쉬값이 된다. 따라서 A3/A8 알고리즘의 출력인 32비트 SRES와 암호키 64비트 K_c는 다음 식에 의해 최종 생성된다.

$SRES$ = a^b^c^d;

K_c = ef;

3.2.2 S-box

S-box는 비선형성을 높이기 위하여 x^{-1}연산을 사용하였으며 일반적으로 역원을 구하는 연산은 많은 연산 시간을 요구한다. 그러나 GF(2^8)에서의 역원이므로 미리 연산하여 참조 테이블을 구성하였다. 0의 역원은 존재하지 않으므로 0의 값으로 대응되는데 이는 암호학적으로 안전하지 않으므로 역원의 각 값에 0xa5값을 배타적-합(exclusive-OR)하여 256개의 8비트 값을 갖는 테이블을 구성하였다. S-box테이블은 <표 2>에 나타내었다.

〈표 2〉 S-box 테이블

0xa5	0xa4	0x33	0x41	0xee	0xf9	0xd7	0x60	0x16	0xe6	0x8b	0x58	0x9c	0x99	0x51	0x91
0x6a	0x52	0x12	0x97	0xb2	0x4c	0x4d	0xb1	0x2f	0x83	0xbb	0x7f	0xdf	0x15	0xbf	0x7e
0x54	0xec	0x48	0xce	0x68	0x1f	0xbc	0x2e	0x38	0x5b	0x47	0x5c	0xd1	0x08	0xaf	0x59
0xe0	0x44	0xb6	0x13	0xaa	0x50	0xc8	0x04	0x98	0xa9	0xfd	0x20	0xa8	0x9d	0x5e	0x19
0x4b	0x2a	0x17	0xac	0x45	0x95	0x06	0x56	0x55	0x84	0xf8	0xa1	0x3f	0x3d	0x76	0x0e
0x7d	0x6c	0xda	0xfa	0xd4	0x2c	0x4f	0xd8	0x9f	0x21	0x65	0xc3	0xa0	0xef	0xdb	0xf6
0x11	0x6f	0x43	0x79	0x3a	0x67	0xfe	0x64	0x34	0xdc	0x49	0x86	0x05	0x93	0x63	0x28
0x2d	0xf1	0xa3	0x61	0x89	0x09	0x71	0x25	0x35	0xcc	0xb9	0x14	0x4e	0xf2	0xfb	0xf7
0xd2	0x70	0x74	0x7a	0xfc	0x9e	0x37	0x73	0xd5	0xf0	0xbd	0x82	0x62	0xca	0x4a	0xe4
0xdd	0xcd	0x23	0x72	0x1d	0x02	0xa7	0x40	0xe8	0x3e	0xe9	0x3c	0x5a	0x8d	0x66	0xc1
0xc9	0x92	0x57	0xe3	0x0c	0x0a	0x1c	0x30	0x0b	0x01	0x77	0xea	0xd0	0x88	0x0d	0x00
0xb8	0xde	0xe7	0xad	0xc5	0x6e	0x96	0xb7	0x31	0x03	0x80	0x69	0x9a	0x5f	0x1a	0x1b
0xff	0xc2	0xc0	0x3b	0xd6	0xa2	0xcb	0x29	0x7c	0xf4	0xc4	0x10	0x1e	0x81	0x53	0xb5
0x7b	0x27	0x0f	0xeb	0xd3	0x24	0x22	0x36	0xf5	0x6d	0xbe	0xba	0xc6	0x42	0x75	0x26
0xe1	0x94	0x8f	0x5d	0xa6	0x32	0xc7	0x78	0xb3	0xb0	0xf3	0xd9	0xcf	0x87	0xe5	0x2b
0xed	0x85	0x07	0xe2	0xab	0x90	0x6b	0xb4	0x46	0x8e	0x18	0x9b	0x8a	0xae	0x8c	0x39

3.3 새로운 스트림 암호 (A5)

잘 설계된 시스템은 기지 평문 공격에 대해서 안전해야 하므로 LFSR 그 자체만으로는 키 스트림 생성기를 만드는 데에 사용될 수 없다. 따라서 LFSR를 비선형적으로 구성하기 위한 방법은 다음과 같다[6, 13].

① 여러 개의 LFSR의 출력 값들에 비선형 함수를 사용하여 설계.

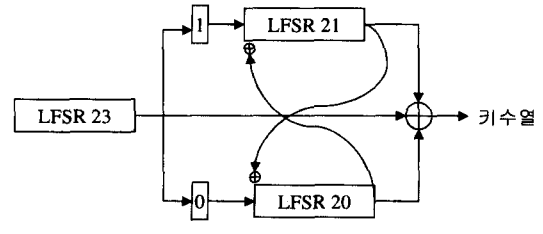
② 하나의 LFSR의 값을 이용한 비선형 filtering 함수를 사용하여 설계.

③ 하나 (또는 둘 이상)의 출력이 하나 (또는 둘 이상)의 클럭을 제어하도록 설계.

LFSR을 기초로 한 키 스트림 생성기는 모든 가능한 비밀키에 대하여 출력 수열이 다음 세가지 성질 즉, 큰 주기, 큰 선형복잡도, 좋은 통계적 특성을 만족해야 한다. 본 논문에서는 자체 클럭에 의한 제어를 함으로서 추가의 비선형 함수를 사용하지 않아도 되며 이러한 이유로 구현이 용이하고 속도가 빠른 클럭 제어 방식의 alternating 생성기[6]를 개선하여 스트림 암호기를 제안하였다.

3.3.1 스트림 암호기의 구조

새로 개발된 스트림 암호 알고리즘은 3개의 LFSR로 구성되어 있으며 64비트 키를 입력으로 23단, 21단, 20단인 3개의 LFSR을 초기화하고 이를 이용하여 키 수열을 생성한다. 전체적인 구조는 다음의 (그림 5)와 같다.



(그림 5) 제안한 스트림 암호 알고리즘의 구조

위의 (그림 5)의 알고리즘은 다음의 순서로 동작한다.

① 23단 LFSR이 동작한다.

② 결과 값이 1이면 21단 LFSR만이, 0이면 20단 LFSR만이 동작된다.

③ 각 LFSR의 결과값을 XOR하여 최종 키 수열로 출력한다.

④ ②에서 동작한 LFSR의 결과값이 대칭되는 LFSR의 최하위 비트에 XOR된다.

3.3.2 알고리즘의 주기 및 선형 복잡도

각 LFSR의 연결 다항식은 최대 주기의 수열을 생성하기 위해 원시 다항식이 사용되는데 23단 LFSR에 사용되는 연결 다항식은 $x^{23}+x^5+1$이고 21단 LFSR에 사용되는 연결 다항식은

$x^{21}+x^2+1$이며 20단 LFSR에 사용되는 연결 다항식은 $x^{20}+x^3+1$이다. 따라서 각 LFSR은 $2^{23}-1$, $2^{21}-1$, $2^{20}-1$의 주기를 갖는다. 또한 전체 알고리즘의 주기는 하한을 제시한 것이며 $2^{23}\cdot(2^{21}-1)\cdot(2^{20}-1)\approx 2^{64}$이다.

스트림 암호 알고리즘에서 생성되는 키 수열은 선형 복잡도가 커야 한다. 선형 복잡도란 주어진 키 수열을 생성할 수 있는 LFSR의 최소 길이이다. 따라서 주어진 수열의 선형 복잡도가 작으면 Berlekamp-Massey 알고리즘에 의해 LFSR을 구성하여 주어진 수열을 생성할 수 있게 된다. 선형 복잡도는 일반적으로 키 수열의 길이에 비례하며 키 수열의 주기에 이르면 더 이상 증가하지 않고 고정된 상수값이 된다. 이 값은 대략 주기의 1/2 값이 된다. 새로 개발된 알고리즘의 선형 복잡도는 하한을 제시한 것이며 약 $(21+20)\cdot 2^{23}\approx 2^{31.7}$의 값을 갖는다[6].

4. 알고리즘의 통계적 특성 및 결과

4.1 통계테스트

암호 알고리즘에 의해 생성되는 키 수열은 통계적 테스트를 통과하여야 한다. 통계적 테스트의 통과란 암호 알고리즘에 의해 생성되는 키 수열이 난수열에 가깝다는 것을 의미하는 것으로 암호 알고리즘의 키 수열이 갖추어야 할 기본적인 사항이다.

FIPS (Federal Information Processing Standards) 140-1에 제안된 빈도 테스트(Frequency test), 시리얼 테스트(Serial test), 포카 테스트(Poker test), 런 테스트(Run test), 자기상관 테스트(Autocorrelation test)에 관한 5가지의 실험 결과를 제시한다. 각 테스트에 사용된 표본의 크기는 FIPS 140-1에서 제시한 20000비트를 사용하였다. 표본은 임의로 주어진 입력을 이용하여 알고리즘을 동작시킨 후 생성된 키 수열 20000비트이다[6, 10, 13].

아래 <표 3, 4, 5>는 통계 테스트별 임계값과 제안된 알고리즘들의 출력 비트열에 대한 테스트 결과이다. 결과에서 알 수 있듯이 제안된 알고리즘의 출력 비트열에 대한 통계적 특성이 모두 만족함을 알 수 있다.

〈표 3〉 빈도, 시리얼 그리고 포카 테스트 결과

Test	FIPS140-1	New A3/A8	New A5
Freq. test	$9654<X<10346$	9998	10003
Serial test	5.9915	2.698	1.822
Poker test	$1.03<X<57.4$	26.099	8.243

〈표 4〉 런 테스트 결과

Length of run	FIPS 140-1	New A3/A8	New A5
1	2267 - 2733	2543	2549
2	1079 - 1421	1221	1268
3	502 - 748	658	613
4	223 - 402	307	322
5	90 - 223	150	145
6	90 - 223	151	147

〈표 5〉 자기상관 테스트 결과

d	Threshold value	significance level	New A3/A8	New A5
4	-1.96~1.96	5%	-0.50915	0.63646
8	-1.96~1.96	5%	1.51351	0.50921
16	-1.96~1.96	5%	-0.99034	0.45272
32	-1.96~1.96	5%	-0.12738	-0.11322

4.2 SAC(Strict Avalanche Criteria)

SAC란 알고리즘의 입력 중 한 비트가 변할 경우 출력의 각 비트가 변할 확률이 1/2이 되는지를 조사하는 것으로 A3/A8알고리즘에 대하여 수행하였다. 표본은 임의로 주어진 입력 1000개의 메시지를 이용하여 알고리즘을 동작시킨 후 생성된 결과이다. 본 테스트에서는 x^2(chi-square distribution) 검정을 이용하였다. 256×192=49152의 경우 중 46004번이 유의수준 5% 이내임을 알 수 있었다.

5. 결 론

본 논문에서는 GSM망으로의 로밍 서비스를 위한 인증 및 세션키 생성 알고리즘을 개발하였다. 제안한 A3/A8 알고리즘은 새로운 전용해쉬함수를 사용하였다. 메시지 암호화를 위한 A5알고리즘은 선형 귀환 쉬프트 레지스터에 기반한 스트림 암호기를 개발하였다. 개발한 스트림 암호기는 GSM에서 메시지 암호 알고리즘으로 사용되어질 수 있도록 개발하였으나 수출제한(특허)의 문제로 직접 사용되어질 수는 없으며 스트림 암호의 다른 응용분야에서 사용되어질 수 있을 것이다. 제안된 알고리즘은 C언어로 구현하였으며 IBM PC 상에서 시뮬레이션 하였다. 또한 제안된 알고리즘의 출력 수열에 대한 랜덤성 여부를 통계적 테스트를 사용하여 분석하였다. 이를 위해 FIPS140-1에 제안된 빈도 테스트, 시리얼 테스트, 포카 테스트, 런 테스트, 자기상관 테스트를 수행하였다. 실험 결과 제시된 모든 테스트를 통과하였으며 따라서 제안된 알고리즘의 출력 수열이 통계적으로 안전함을 알 수 있었다. 그러나 이는 암호학적으로 안전하기 위한 필요조건일 뿐이다. 알고리즘의 안전도에 대한 수학적인 증명은 아직 없기 때문에 충분히 많은 공격자의 공격을 견딜 때라야만 계산적으로 안전하다고 말할 수 있다. 본 논문의 결과는 이동통신 시스템의 암호 및 인증 알고리즘의 개발에 기여할 것으로 기대되며 또한 다른 암호 응용분야 즉, 디지털 서명, 워터마킹 등과 같은 분야에 응용되어 질 수 있을 것이다.

참 고 문 헌

[1] ISO/IEC 7498-2. *Information Processing-OSI Basic Reference Model-Part2 : Security Architecture*, 1989

[2] TIA/EIA IS-95A, *Mobile-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System*, July 1993

[3] ETSI, *European Digital Cellular Telecommunication System(phase 2)-Security Related Network Functions*, July 1993

[4] JTC, *Personal Communications Services PACS Air Interface Specification*, Jan. 1995

[5] Asha Mehrotra, Leonard s. Golding, "Mobility and security management in the GSM system and some proposed future improvements," *Proceedings of the IEEE*, Vol.86, No.7 , July 1998.

[6] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997

[7] B. Preneel, "*Analysis and design of cryptographic hash functions*," Doctoral Dissertation, Katholieke Universiteit Leuven, 1993

[8] R. L. Rivest, "The MD4 message-digest algorithm," *Advances in Cryptology-Crypto'90, Lecture Notes in Computer Science*, Vol.537, pp. 303-311, 1991

[9] R. L. Rivest, "The MD5 message-digest algorithm," Request For Comment(RFC) 1320, Internet Activities Board, Internet Privacy Task Force, April 1992

[10] FIPS 180-1, "*Secure hash standard*," Federal Information Processing Standards Publication 180-1, U.S. Department of Commerce/Nist, 1995

[11] H. Dobbertin, A. Bosselaers, B. Prennel, "RIPEMD-160 : A strengthened version of RIPEMD," *Fast Software Encryption-Cambridge Workshop, Lecture Notes in Computer Science*, Vol.1039, Springer-Verlag, pp.71-82, 1996

[12] Y. Zheng, J. Pieprzyk, J. Sebery, "Haval-a one-way hashing algorithm with variable length of output," *Advances in Cryptology-AUSCRYPT '92, LNCS*, Vol.718, pp.83-104, 1993

[13] 박홍근, 남길현, "스트림 암호 시스템의 설계 및 비도 분석에 관한 연구", CISC'95 Proceedings, Vol. 5, No.1, pp.141-150, 1995

김 범 식

e-mail : kbs87@dankook.ac.kr

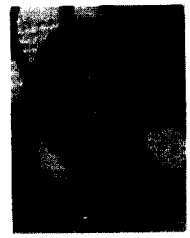
1993년 단국대학교 전자공학과 졸업(학사)

1995년 단국대학교 대학원 전자공학과 졸업(공학석사)

1999년 단국대학교 대학원 전자공학과 수료

현재 (주)지텍인터내셔널 스마트카드개발 팀장

관심분야 : 정보보안, 이동통신, 스마트카드, 전자상거래



신 인 철

e-mail : char@dankook.ac.kr

1973년 고려대학교 전자공학과 졸업(학사)

1978년 고려대학교 대학원 전자공학과 졸업(공학석사)

1986년 고려대학교 대학원 전자공학과 졸업(공학박사)

1979년~현재 단국대학교 전자·컴퓨터공학과 교수

관심분야 : 병렬처리, 정보보안, 스마트카드, 전자상거래