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

실 난수 발생기를 이용한 키 생성에 관한 연구

차재현*, 박중길**, 전문석***

A Study on Key Generation using the Real Random Number Generator

Jae-Hyeon Cha, Joong Gil Park, Moon-Seog Jun

Abstract

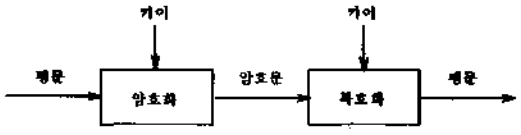
Key is generally formed using the Random Number. How to make the Random Number is to cast coin or dice as classical method, to form the Real Random Number with Hardware and to make the Pseudo Random Number by means of utilizing mathematical algorism. This thesis presented NRNG(New Random Number Generator) which put self-development Hardware to use as Key Generation Method and inspected to compare the Real Random Number with the Pseudo Random Number and special properties which PRNG(Pseudo-Random Number Generator) creates.

Key Word : Key Generator, Real Random Number

* 숭실대학교 컴퓨터학과 박사과정
** 충남대학교 컴퓨터과학과 박사과정
*** 숭실대학교 정보과학대학 부교수

1. 서 론

암/복호화 알고리즘(Cipher/Decipher Algo -rithms)을 사용하여 자료(Data) 또는 메세지(Message)를 교환하는 암호체계(Cryptographic System)에서는 키이를 사용한다. <그림 1>은 일반적인 암호체계이며, 평문(Plaintext)을 해독 불가능한 형태로 변형 조작하는 암호화와 암호문(Ciphertext)을 해독 가능한 형태로 변환하는 복호화, 그리고 암/복호화 알고리즘의 초기 수단이나 열쇠가 되는 문자, 숫자, 부호 또는 이들의 조합인 키이로 구성된다[1, 8].



<그림 1> 암호체계

대부분의 암호 체계에서 안전성은 암호 알고리즘의 강도(Strength)와 키이에 의해 크게 좌우된다[1]. 특히, 사용되는 키이 또는 생성된 수열의 난수 정도가 암호 체계의 안전성에 영향을 크게 미치는 체계(Stream Cipher, Challenge-Response 인증 체계 등)의 경우에는 좋은 키이 또는 난수를 생성하는 것이 매우 중요한 문제가 된다[23]. 따라서 안전한 난수발생기의 사용이 보호제품의 안전성을 보장하기 위한 기본이라 할 수 있다. 암호학적 응용에서 난수가 사용되는 예를 보면 대칭키 암호(Symmetric crypto -system)에서의 비밀키와 같은 각종키나 IV(Initializing Value) 생성, 비대칭 암호(Asymmetric cryptosystem)의 비밀키(Private key)/공개키(Public key) 혹은 공개 파라메터 생성, 인증 메커니즘이나 키관리 프로토콜을 포함하는 각종 암호화 프로토콜에서 세션키나 ramdom challenge 생성 등이 있다. 일반적으로 보안제품의 안전성을 얘기할 때 사용하고 있는 암호알고리즘의 안전성에는 많은 관심을 가지고 있지만 실제로 더 근본적인 안전성의 기저라 할 수 있는 난수발생기(RNG : Random Number Generator)에는 별로 주의를 기울이지 않는 것이 실로 아이러니라 할 수 있다. 실제로 암호제품의 해독을 위해서는 각각의 암호알고리즘을 분석하여 여기에 사용되는 비밀키를 찾아내는 것보다 이 비밀키를 생성하는 난수발생기를 분석하는 것이 훨씬 더 효율적인 공격이 될 수 있다[24].

이러한 키이 발생의 가장 이상적인 방법은 가능한 한 키이 생성 확률이 동등한 완전한 난수로 키이를 생성하는 것이나, 불행하게도 이러한 완전한 난수는 수학적인 알고리즘으로는 생성하기가 불가능하다[11, 12]. 키이의 비 난수성(Non-Randomness)은 적에게 키이 값을 예측 가능하게 할 수 있거나 키이를 찾는 노력을 감소시켜 암호문 해독을 보다 쉽게 할 수 있다. 일반적으로 키에 요구되는 성질은 키이 값의 일양 분포(Uniform Distribution)와 예측할 수 없는 성질(Unpredictability)이다[15].

난수에 대한 정의는 여러 가지가 있지만, 통계학의 확률론에서는 확률변수의 열 {Xn}이 독립이고 같은 분포를 갖을 때, Xn(w) = xn인 실현치의 수열 {x0, x1 , x2·}을 확률론에서 실 난수라고 한다.

난수 발생 방법에는 동전 혹은 주사위를

던져서 난수를 발생하는 고전적 방법[15]과 수학적 알고리즘을 이용하여 구성하는 PRNG [3, 10, 13], 그리고 하드웨어로 구성한 실 난수 발생기가 있다. 난수 발생기를 하드웨어로 구성하는 경우에는 하드웨어적인 잡음원(Thermal Noise or White Gaussian Noise)을 증폭시켜 샘플링함으로써 난수를 획득하도록 구성한다.

본 논문에서는 하드웨어의 특성을 이용하여 개발한 실 난수 발생기인 NRNG의 특징을 기술하고, 발생 난수의 통계적 특성을 알아보기 위해 PRNG가 생성하는 의사 난수와 함께 통계적 검정을 수행하여 비교 검토하였다. 2장에서는 난수 발생 방법 고찰 및 새로운 실 난수 발생원에 대해 기술하고, 3장에서는 발생한 난수의 통계적 검정, 4장에서는 검정 결과 및 결론을 맺는다.

2. 난수 발생 방법 고찰 및 제안

2.1 기존의 난수 발생 방법

2.1.1 고전적 방법

동전을 던지거나 주사위를 던져서 난수를 발생시키는 것으로 적은 수의 키이를 생성하는 데는 적절하나, 많은 수의 키이를 생성할 때에는 과다한 노동력을 필요로 하여 실제 사용에 부적합하다[15].

2.1.2 PRNG(Pseudo-Random Number Generator)

PRNG는 대체적으로 수학적 알고리즘을 이용하여 구성하며, LFSR (Linear Feedback Shift Register), LCG (Linear Congruential Generator), Multiplication 시스템 등이 있다[10, 13, 19]. 이 PRNG에 의해 출력되는 난수를 의사 난수라 하며, 의사 난수는 통계적 특성은 우수하지만 제한된 주기를 갖는다[15, 8]. 본 논문에서는 위 세 가지 방법에 대해 시뮬레이션 프로그램을 작성하고 의사 난수 데이타를 추출하여 통계적 검정에 사용하였다.

PRNG는 크게 각종 잡음원(noise source)로부터 충분한 엔트로피를 갖는 무작위 잡음(random noise)을 수집하는 잡음 수집 과정, 수집된 무작위 잡음을 이용하여 PRNG의 내부상태변수(internal state, random pool)에 난수성(randomness)을 추가하는 잡음 추가 과정, 그리고 현재의 내부상태변수로부터 원하는 길이의 의사난수를 발생시키는 난수생성 과정으로 나누어 생각할 수 있다. 보다 정교한 PRNG의 경우는 수집된 잡음 정보(noise sample)들에 대한 엔트로피(entropy)를 계산(추정)하여 이를 바탕으로 언제 잡음 추가 과정을 수행시켜야 할 지를 결정하는 콘트롤을 두는 경우도 있다[25].

PRNG에 가능한 공격방법들은 다음과 같이 나누어 생각해 볼 수 있다[24].

● Seed/State에 대한 전수조사 (Exhaustive Seed/State Search):

PRNG의 안전성은 거의 전적으로 seed에 대한 예측 불가능성(unpredictability)과 PRNG의 내부 상태변수의 비밀성에 의존한다. PRNG의 초기내부 상태변수는 다양한 잡음원을 이용하여 초기화되는데, 흔히 이용되는 clock이나 keystroke, system/network statistics 등은 쉽게 예측 가능하거나 네트워크 연결을 통한 관찰, seeding 중 다른 프로

세스에 의한 관찰, 그리고 시스템 리부팅에 의한 초기화 등에 대해 취약성을 보인다. 따라서 가능한 다양한 잡음원으로부터 얻은 정보를 축적하여 충분히 높은 엔트로피를 갖도록 초기 내부상태변수를 만들어야 한다. 반면 seeding 과정에서 충분한 엔트로피가 도입되었다면 그 후의 내부상태변수에 대한 전수검사는 내부상태변수가 충분히 길다면(예를 들어 64비트 이상) 현실적으로 불가능하다. 그러나 이론적으로는 이 PRNG의 안전도, 즉 내부상태변수의 길이는 이 PRNG를 이용하는 암호시스템의 안전도에 대한 상한이 될 수 있으므로 충분히 길게 잡아야 한다. 예를 들면, AES(Rijndale)의 256-비트 키에 상응하는 안전도를 주기 위해서는 PRNG의 내부상태변수도 256-비트 이상이 되어야 한다.

● Known/Chosen Input Attacks:

Seeding 과정에 공격자가 예측하기 쉬운 잡음원을 이용하거나 이 잡음원을 부분적으로라도 제어할 수 있다면 (예를 들어 컴퓨터의 타이머를 조작하거나 시스템 리부팅 등을 통한 각종 카운트의 초기화) seed에 대한 전수검사가 훨씬 쉬어진다. 스마트카드나 다른 휴대용 보안토컨 등에서 사용자 패스워드를 seed의 일부로 사용하는 경우도 마찬가지로 쉽게 예측할 수 있는 경우가 많다. 따라서 seed는 가능한 다양한 잡음원으로부터 충분히 많은 잡음 정보들을 추출하여 생성하여야 한다. 한편 입력 값을 알 수는 없더라도 반복적인 입력의 조작을 통해 내부 동작이 같은 상태가 반복되게 하거나 발생될 수 있는 난수의 주기를 짧게 하는 등의 공격(replay attack, cycle shortening attacks)도 가능하므로 잡음수집 과정을 통해 얻은 입력은 항상 해쉬함수와 같은 일방향 함수를 통과한 후 사용되도록 해야 한다.

● Known Output Attacks:

PRNG의 출력은 다른 암호 시스템의 비밀키를 생성하는데도 쓰이지만, 블록암호의 IV나 각종 인증/키관리 프로토콜에서 평문으로 전송되는 random challenge, 혹은 이산대수 문제(DLP:Discrete Logarithm Problem)에 바탕을 둔 비대칭 암호시스템의 공개 파라메터(도메인 변수인 소수 p, q 및 primitive element g) 생성 등과 같이 그 출력이 그대로 노출되는 경우도 많다. 따라서 충분히 긴 PRNG 출력이 알려진다 해도 이로부터 내부상태변수를 알아내거나 그 이전 혹은 이후에 생성되는 출력에 대한 정보를 얻을 수 없도록 해야 한다. 통상 출력을 해쉬함수 같은 일방향함수를 통해 생성함으로써 막을 수 있다.

● State Compromise Forward/Backward Tracking:

PRNG가 안전한 메모리 영역(Tamper protection area)에서 수행되지 않는 한 내부상태변수에 대한 정보를 완벽히 보호한다는 것은 근본적으로 불가능하다(Virus, memory scanning 등). 따라서 PRNG의 설계시 비록 PRNG의 내부 정보가 일부 누출되더라도 가능한 빠른 시간내에 원래의 안전한 상태로 복귀되도록 함으로써 피해를 최소화시켜야 한다(즉 주기적으로 잡음 추가 과정을 시행하여 난수성을 추가시켜야 한다). 또한 내부상태변수의 사용 후에는 항상 해쉬함수 같은 일방향 함수를 통해 갱신시켜 주는 것이 바람직하다.

● Implementation Errors:

복잡한 PRNG는 그 구현 과정에서 사소한 에러에 의해서도 쉽게 정상적인 동작을 하지 못하게 되므로 가능한 간단 명료하게 구현이 가능하도록 해야 한다. 또한 만일의 사태에 대비하여 비대칭키 암호에서의 비밀키와 같은 중요한 키의 생성에 사용될 때는 시간이 걸리더라도 가능한 많은 잡음원을 이용하며(사용자로부터 외부입력을 받는 것이 좋음), 일단 키 생성이 끝나면 PRNG의 내부 정보를 재생성(refresh)하여 더 이상 생성된 키에 대한 정보가 PRNG에 남지 않도록 하는 것이 중요하다. 일반적으로 PRNG가 난수 출력을 낸 후에는 내부상태변수를 일방향 함수를 통해 섞어 주어 이전의 난수 생성에 사용된 정보가 더 이상 남지 않도록 하는 것이 좋다[25].

이상에서 보는 바와 같이 암호학적으로 매우 중요한 난수 생성의 방법을 보안성이 중요하지 않는 곳에서는 비용이 상대적으로 저렴한 PRNG의 사용이 권장될 수 있지만, 보안성이 중요한 군사/외교 장비나 PKI내의 CA 발급 등에서는 실 난수 발생기를 사용하여 충분한 안전성을 확보하여야 한다.

2.2 제안한 난수 발생 방법(NRNG:New Random Number Generator)

2.2.1 실 난수 발생원

전기 혹은 전자 회로에는 Thermal 노이즈, Flicker 노이즈, Shot 노이즈, Avalanche 노이즈 등 많은 종류의 노이즈(Noise)가 존재하고 있다[21, 22]. 이러한 노이즈는 완전한 불규칙 신호이며, 신호의 전압 크기(Amplitude)와 신호 위상(Phase)은 불규칙한 주파수 성분을 갖게 된다. 이런 신호에서 긴 시간 동안의 RMS(Root-Mean-Square) 값을 얻어낼 수 있다 할지라도, 어느 한 순간의 정확한 신호의 크기를 예측하는 것은 불가능하다[21, 22]. 그러므로 이런 노이즈를 이용하여 완전한 난수를 얻어낼 수 있다.

실 난수 발생을 위한 어떤 입력 요소들은 바로 엔트로피의 요소가 된다. 공격자로부터 미지의 입력 요소를 찾기 위한 여러 가지 것들 중에서도 더 좋은 것이 있을 수 있으며, 각각의 나름대로의 엔트로피를 갖고 있다. 다음에서는 여러 가지 실 난수 발생을 위한 입력 요소들을 구하는 방법 몇 가지를 기술한다[26].

● Radioactive source :

RS232 출력이 있는 방사 모니터를 이용한다.

● Quantum Effect in a semiconductor (e.g. noise diode) :

Noise diode나 저항을 이용한 방식으로써 흔히 사용되는 방법이고, 가격적인 면에서도 유리하다.

● Photon polarization detection 45?out of phase :

빛의 성질 중 랜덤하게 편광되는 특성을 이용한다.

● Air turbulence within a sealed disk drive, dedicated to this task :

disk drive의 회전 속도, disk 공간, 열전도에 의한 공기 흐름, 헤드와 지지대의 움직임 등으로 인한 드라이버 내부의 공기 대류

현상에서 엔트로피를 추출한다.

● Stereo microphones, subtracted :
동일한 방안에서 stereo 마이크간의 신호 차를 한 개의 마이크에서 나온 신호로 재구성하기가 매우 어렵다.

● Microphone :
평범한 공간에 놓여진 보통 mono 마이크에서도 필요한 엔트로피를 얻을 수 있다.

다음은 엔트로피의 요소로써 흔히 사용되는 것이지만, 이것들은 예측될 수 있거나 관찰의 대상이 될 수도 있으며, 적으로부터 영향을 받을 수 있는 중대한 허점을 가질 수 있다.

● Network statistics

● Process statistics

● I/O completion timing and statistics

다음은 엔트로피의 요소로써 거의 가치는 없는 것들이지만, 일부는 편리한 이유로 사용되는 경우도 있다.

● TV or radio broadcasts :
대부분의 신호가 적에게 유용하다 할 지라도, 어떠한 부분을 취한 전기적 잡음 신호가 효과적인 엔트로피 신호를 가질 수 있다.

● Published information on a CD of tap or in newspapers, magazines of library books :
적이 같은 출판물에 접근한다고 가정하면, 엔트로피로서의 가치가 없다고 본다.

● System date and time
이것은 극히 낮은 엔트로피를 갖는다.

● Multiple, free running ring oscillators :
부분적으로는 랜덤한 것처럼 보이지만 주기적 수열을 갖는 기본적인 PRNG의 하드웨어 수준이다.

실 난수 발생을 위한 여러 가지 방법이 있지만 다음은 가능한 피해야 할 방법이다.

● Chaos equation
이것은 복잡하게 보이는 것처럼 많은 혼동을 준다.

● Math library ranno generators
결코 암호학적으로 강하게 설계되지 않았다.

● CD ROMs, audio CDs or tapes
기록 매체에는 많은 비트들을 가지고 있지만, 이 많은 양의 비트들이 랜덤성을 갖고 있다고 혼동하기 쉽다. 출간된 모든 기록물들에는 색인을 만들어 놓았기 때문에 어떤 공격자가 많은 시험을 통해 유추해 내기에는 충분히 작다고 볼 수 있다.

본 논문에서는 Zener Diode에서 발생한 Avalanche 노이즈를 사용하였다. 이 노이즈는 접합 다이오드(PN Junction) 내의 Zener 혹은 Avalanche Breakdown에 의해 생기는 신호이다. 역전압이 걸린 접합 다이오드 내의 전자와 정공은 충분한 에너지(Energy)를 가지고 있어서 실리콘(Silicon) 원자와 충돌하여 전자-정공 쌍을 만들 수 있다. 이 과정이 반복되어 일어남으로써 불규칙한 일련의 커다란 노이즈 스파이크(Spike)를 발생시킨다. 이러한 노이즈는 일반적으로 같은 전류의 Shot 노이즈, Thermal 노이즈보다 훨씬 크기 때문에 실 난수 발생원으로 사용하였다.

2.2.2 하드웨어 설계

NRNG 시스템 블럭도는 <그림 2>와 같다.

<그림 2> NRNG 시스템 블록도

NRNG 시스템 블록도의 첫번째 단에서 발생된 하드웨어 난수원인 Avalanche 노이즈는 두번째 단의 필터링을 거쳐 세번째 단의 증폭기에서 22배로 증폭된다. 네번째 단의 비교기는 임계값(Threshold) 0V와 약간의 Hysteresis를 가지며 +12V 또는 -12V의 신호를 출력한다.

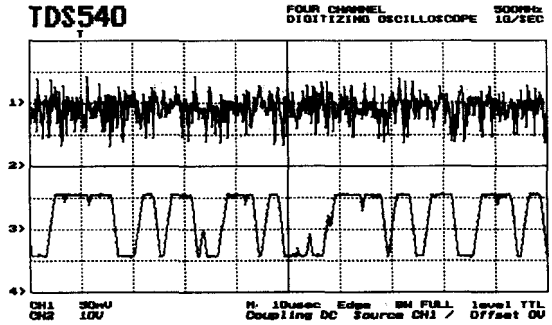
PC(Personal Computer) 인터페이스는 RS-232C Port를 이용하였으며 <표 1>과 같이 연결하였다. 회로를 동작시키기 위한 전원은 RTS(+12V)와 DTR(-12V) 핀을 이용하였으며 각각 1과 0으로 초기화 시켰다. 회로의 출력은 입력 핀인 CTS 핀을 통해 RS-232C 신호의 형태로 PC에서 읽는다.

<표 1> RS-232C 인터페이스

회로	RS-232C Port (핀 번호)
+VCC	RTS(4)
GND	GND(7)
-VCC	DTR(20)
출력	CTS(5)

2.2.3 실 난수 획득 방법

네번째 단 OP-AMP 전, 후의 Avalanche 노이즈 신호는 <그림 3>과 같다. 이 Avalanche 노이즈의 신호를 일정 구간(Interval)에서 읽어 난수를 추출한 결과는 Avalanche 노이즈 신호의 특성으로 인해 비 난수성을 갖는다. 본 논문에서는 임의의 시점에서 전이(Transition)가 일어날 때까지의 샘플 수를 조사하여, 이들의 분포를 분석하였다.



<그림 3> Avalanche 노이즈 신호

153,600 비트(Bit)를 추출할 때, <그림 4>는 전이 2, 4, 그리고 6 구간에서 샘플 수의 분포를 나타내며, <그림 5>는 전이 10, 20 구간에서의 샘플 수의 분포를 나타낸다. 이 분포들로 부터 전이 구간이 증가함에 따라 샘플 수의 분포는 Gaussion이 됨을 알 수 있다. PC 속도와 난수성 검정의 결과를 토대로 20을 전이 구간으로 선택하였다. 그리고 Gaussion 분포인 샘플 수에서 0과 1의 일양 분포를 추출하는 방법으로 샘플 수의 LSB (Least Significance Bit)를 선택하였다.

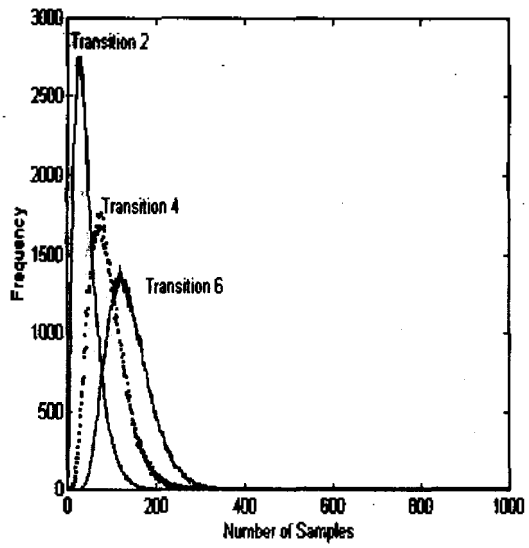
NRNG로부터 1 비트 난수를 읽어 내는 알고리즘은 다음과 같다.

-- Sum_num: Computer가 회로출력을 읽는 횟수

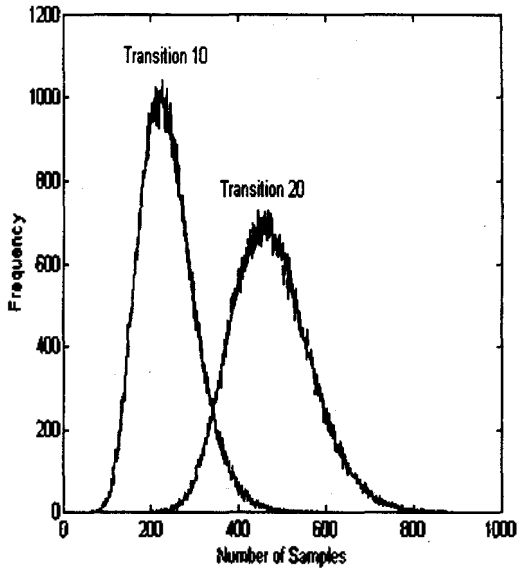
-- Trans : 회로의 출력이 0 에서 1 또는, 1 에서 0으로 전이하는 횟수

```
-- Before   : 앞의 출력 상태
-- Current  : 현재의 출력상태

BEGIN
   RS-232C PORT 설정;
   Trans <- 0;
   Sum_num <- 0;
   Before <- 0;
   Current <- READ( RS-232C PORT);
     WHILE ( Trans not equal 20 )
       IF Before not equal Current
       THEN  Trans <- Trans+1;
         Sum_num <- Sum_num + 1;
         Before <- Current;
         Current <- READ (RS-232C PORT);
     END
         RESULT <- Sum_num mod 2;
END
```



<그림 4> 2, 4, 6 길이의 전이 구간에서 샘플 수



<그림 5> 10, 20 길이의 전이 구간에서 샘플 수

3. 발생한 난수의 통계적 검정

통계적 검정의 원리로부터 한 난수열이 어떤 통계적 검정을 통과하였다 함은 난수성을 갖지 않는다고 말 할 수 없다는 소극적인 긍정(필요 조건)이며, 그 수열이 난수성을 갖는다고 단정하는 적극적인 긍정(충분 조건)은 아니다[12].

한 수열이 통계적 검정 T_1, T_2, · · ·, T_n 을 통과하였다고 하여 다른 통계적 검정 T_{n+1} 을 통과한다는 보장은 없으며 T_{n+1} 까지 통과할 경우에 그 수열의 난수성에 더 많은 신뢰성을 부여할 수 있다는 것을 의미한다. 확률 및 통계의 이론으로부터 난수성에 대한 많은 검정방법들이 알려져 있으나 그 중에서 암호학에 유용하고 또한 사용하기에

편리한 방법들에 대해 기술한다[1, 2, 4, 11].

실제로 난수열에 대한 난수성 측정은 사용할 난수열 전체 주기에 걸쳐 검토되어야 하나 난수 발생기들에서 발생되는 난수열 주기가 매우 크기 때문에 사실상 불가능하다. 따라서 전체 난수열의 일부인 길이 153,600 비트를 선택하여 검정을 시행하였다. 이 검정을 국부 난수성(Local Randomness) 검정이라고도 한다[1].

3.1 Frequency 검정

난수열에서 0과 1의 빈도를 측정하여 χ^2 (chi-square) 검정을 수행한다.

가. 검정통계량

$$T = \frac{(n_0 - n_1)^2}{n}$$

여기서

n : 난수열의 총 비트수

n_0 : 난수열에서 "0"의 총비트수

n_1 : 난수열에서 "1"의 총비트수

나. 통계량의 분포와 기각역

분포 : T ~ 자유도 1인 χ^2 분포

유의수준 α의 기각역 : $T > \chi^2(1, 0.05)$

3.2 Serial 검정

난수열에서 "0"에서 "0" 또는 "1", "1"에서 "0" 또는 "1"로 전이되어 가는 과정이 독립인가를 검정한다.

가. 검정통계량

$$T = \frac{4}{(n-1)}\sum_{i=0}^{1}\sum_{j=0}^{1}(n_{ij})^2 - \frac{2}{n}\sum_{i=0}^{1}(n_i)^2 + 1$$

여기서

n_{00} : 00 전이의 빈도

n_{01} : 01 전이의 빈도

n_{10} : 10 전이의 빈도

n_{11} : 11 전이의 빈도

나. 통계량의 분포와 기각역

분포 : T ~ 자유도 2인 χ^2 분포

유의수준α의 기각역 : $T > \chi^2(2, 0.05)$

3.3 t-serial 검정

난수열을 t 비트와 $t-1$ 비트로 분할하여 서로 다른 2^t 종류의 r-sequence와 2^{t-1} 종류의 s-sequence의 차이를 검정한다. 즉, 위의 Serial 검정의 일반화한 검정이되며, 여기서는 $t = 3,4,5$ 를 택하여 검정을 수행한다.

가. 검정통계량

$$T = \frac{2^t}{n}\sum_{r=0}^{2^t-1}\left(n_r - \frac{n}{2^t}\right)^2 - \frac{2^{t-1}}{n}\sum_{s=0}^{2^{t-1}-1}\left(n_s - \frac{n}{2^{t-1}}\right)^2$$

여기서 n_r : t 비트의 r 패턴(Pattern) 빈도

n_s : $t-1$ 비트의 s 패턴 빈도

나. 통계량의 분포와 기각역

분포 : T ~ 자유도 2^{t-1}인 χ^2

유의수준 α의 기각역 : $T > \chi^2(2^{t-1}, 0.05)$

3.4 Poker 검정

난수열을 m 비트로 분할하여 전에 기술한 Frequency 검정을 적용한 것으로 Partition 검정이라고도 한다. 난수열을 m 비트로 분할할 때 0 에서 2^m-1 인 패턴이 있으며 각 패턴의 발생 빈도를 f_i 라 한다. 여기서는 m = 3,4,5를 택하여 검정을 수행한다.

가. 검정통계량

$$T=\frac{2^m}{F}\sum_{i=0}^{2^m-1}(f_i)^2-F$$

여기서

F : $\left[\frac{n}{m}\right]-\sum_{i=0}^{2^m-1}(f_i)$, $\left[\frac{n}{m}\right]$: $\frac{n}{m}$ 보다 크지 않은 최대 정수

나. 통계량의 분포와 기각역

분포 : $T\sim$ 자유도 2^m-1인 χ^2 분포 유의수준 α의 기각역 : $T > \chi^2(2^m-1, 0.05)$

4. 검정 결과 및 결론

<표 2>는 LFSR, LCG, Multiplication 시스템 그리고 NRNG에서 추출된 각 153,600 비트를 추출하여 Frequency, Serial, t-serial, Poker 검정을 수행한 결과이다. 검정 결과 LCG와 Multiplication 시스템은 난수성 검정을 통과하지 못한 반면에, LFSR은 가장 좋은 통계적 특성을 가지고 있음을 나타내고, 또한 NRNG에 의한 실 난수도 상당히 좋은 통계적 특성을 갖음을 나타낸다. 그러나, n 단(Stage) LFSR로 생성된 난수열은 $2n$ 개의 연속된 비트를 조사하여 LFSR의 주기인 2^n-1 비트 전체를 알 수 있으므로 난수 생성에 사용하기는 부적합하다[8, 13]. 따라서, NRNG는 주기가 무한대이며, 통계적 특성이 좋은 난수를 발생시키므로 난수 생성시 사용하기에 가장 적합하다.

<표 2> 난수성 검정 결과

검정법 / 키어 발생기	Frequency	Serial	t-serial			Poker		
			t = 3	t = 4	t = 5	m = 3	m = 4	m = 5
LFSR (기각역)	통 과 (0.496) <3.841>	통 과 (3.204) <5.991>	통 과 (4.115) <9.488>	통 과 (6.162) <15.50>	통 과 (13.507) <26.29>	통 과 (5.113) <14.06>	통 과 (13.608) <24.99>	통 과 (15.619) <44.65>
LCG	통 과 (2.160)	기 각 (6431.43)	기 각 (6728.6)	기 각 (7026.9)	기 각 (7380.4)	기 각 (4367.9)	기 각 (4999.9)	기 각 (5575.1)
Multi.	기 각 (41141.25)	기 각 (52304.14)	기 각 (67668)	기 각 (86729)	기 각 (112944)	기 각 (53722)	기 각 (61354)	기 각 (72159)
NRNG	통 과 (1.641)	통 과 (2.764)	통 과 (4.007)	통 과 (8.416)	통 과 (22.530)	통 과 (6.931)	통 과 (16.221)	통 과 (23.598)

(참고)

LCG :

$X_{i+1} = 0$ if $X_{i+1} < (m/2)$

$X_{i+1} = 0$ if $X_{i+1} < (m/2)$

LFSR : $f(x) = x^{32} + x^{28} + x^{27} + x^1 + 1$,

초기값 = 55555555H

Multi :

LFSR1: F(x)=

$x^{33}+x^{13}+1$, 초기값 $=555555555H$

LFSR2 : F(x)=

$x^{35}+x^2+1$, 초기값 = 555555555H

참고문헌

[1] Henry Beker and Fred Piper, Cipher Systems, Northwood Publications,1982, pp. 170-174.

[2] Donald E.Knuth, The Art of Computer Programming, Addison Wesley, Vol 2, 1980, pp. 1-113.

[3] Solomon W. Golomb, Shift Register Sequences, Aegean Park Press, 1982, pp. 1-74.

[4] M.Kimberley, "Comparison of two statistical tests for keystream sequences," Elec. Lett., Vol.23, no.8, 9th April 1987, pp. 365-366.

[5] 김우철 외, 현대통계학, 영지문화사, 1984, pp. 140-157

[6] G.B.Agnew, "Random Sources for Cryptographic Systems," Advances in Cryptology EUROCRYPT '87 proceedings, David Chaum Wyn L. Price (Eds), Springer-Verlag, 1987, pp. 77-81.

[7] 소영일 외, SPSSx를 활용한 비모수통계학, 법문사, 1987, pp. 130-136

[8] 한국전자통신연구소, 현대 암호학, 1991, pp. 1-102

[9] V.N.Yarmolik and S.N.Demidenko, Generation and Application of Pseudorandom Sequences for Random Testing, John Wiley, 1988, pp. 1-94.

[10] Birger Jansson, Random Number Generators, Victor Pettersons Bokindustri Aktiebolag, 1966, pp. 22-74.

[11] 최봉대, "Randomness 특성 분석에 관한 연구," 데이타 보호의 기반 기술 연구(II), 전자통신연구소, 1991, pp. 681-698.

[12] 최봉대, "이진 수열의 Randomness 검정법과 그의 Package 개발에 관한 연구," 데이타 보 호의 기반 기술 연구(I), 전자통신연구소, 1992, pp. 300-305.

[13] Terry Ritter, "The Efficient Generation of Cryptographic Confusion Sequences," CRYTOLOGIA, Vol 15, no.2, April 1991, pp. 81-131.

[14] P.Christoffersson et al, Crypto User's Handbook, North-Holland, 1988, pp. 23-38.

[15] D.W.Davies and W.L.Price, Security for Computer Networks: An Introduction to Data Security in Teleprocessing and Electronic Funds Transfer, John Wiley, 1989, pp. 133-168.

[16] Tony Patti, "Galois Field Cryptosystems," Cryptosystems Journal, Vol 2, no.1, December 1989, pp. 24-54.

[17] Tony Patti, "The Summit Cryptosystem," Cryptosystems Journal, Vol 2, no.2, June 1992, pp. 37-44.

[18] G.Marsaglia and T.A.Bray, "One-Line Random Number Generators and Their in Combinations," Communication of ACM, Vol 11, no.11, November 1968, pp. 757-759.

[19] M.Blum and S.Micali, "How to Generate Cryptographically Strong Sequences of Pseudo-Random Bits", SIAM J. COMPUT., Vol 13, no.4, November 1984, pp. 850-864.

[20] R.A.Rueppel, Analysis and Design of Stream Ciphers, Springer-Verlag, 1986, pp. 5-16.

[21] C.D.Motchenbache and F.C.Fitchen, Low-Noise Electronic Design, Jonh Wiley, 1973, pp. 7-19.
[22] P.R.Gray and R.G.Meyer, Analysis and Design of Analog Integrated Circuits, Jon Wiley, 1977, pp. 635-650.
[23] Bruce Schneier, Applied Cryptography, John Wiley & Sons Inc, 1994, pp.129-151
[24] J.Kelsey, B.Schneier, D.Wagner, and C.Hall, Cryptanalytic attacks on Pseudorandom Number Generators, Fast software encryption, LNCS 1372, Springer-Verlag, 1988, pp.168-188
[25] 임채훈, 황효선, 강명희, 소프트웨어 의사난수 발생기의 설계 및 구현
[26] 백창현, 김용, 김춘수, 완전 난수 발생 방법을 위한 고찰, WISC97 논문집, 1997, pp.63-68

저자소개

차재현
1995년 : 단국대학교 전자공학과 석사
1982년~현재 산업자원부 기술표준원 보안담당관
ISO/IEC JTC1/SC27(정보기술보안) 대한민국 간사
1999년~2000년 동서울대학 전자계산학과 강사 역임
1997년~현재 숭실대학교 컴퓨터학과 박사과정중
관심분야 : 정보보안, 암호이론, 표준화, 인식기술

박중길
1986년 : 동국대학교 전자계산학과 졸업
1988년 : 서강대학교 전자계산학과 석사
1988년~2000년 국방과학연구소 선임연구원
2001년~현재 국가보안기술연구소 선임연구원
2001년~현재 충남대학교 컴퓨터과학과 박사과정중
관심분야 : 컴퓨터통신보안, 접근통제, 암호이론

전문석
1980년 : 숭실대학교 전자계산학과 졸업(학사)
1996년 : University of Maryland 전산과 졸업(석사)
1989년 : University of Maryland 전산과 졸업(박사)
1989년 : Morgan State University 전산수학과 조교수
1989년~1991년 : New Mexico State University 부설 Physical Science Lab. 책임연구원
1991년~현재 숭실대학교 정보과학대학 부교수
<관심분야> 컴퓨터 알고리즘, 병렬처리, VLSI 설계, 암호학