TTA Standard

정보통신단체표준(국문표준)

TTAS.KO-12.xxxx 제정일: 2017년 6월 xx일

운영체제별 잡음원 수집 및 응용 지침

Guideline for the Collection and Application of Noise Source on Operating Systems

한국정보통신기술협회
Telecommunications Technology Association

표준초안 검토 위원회 응용보안 및 평가인증 프로젝트그룹(PG504)

표준안 심의 위원회 정보보호 기술위원회(TC5)

	성명	소 속	직위	위원회 및 직위	표준번호
표준(과제) 제안	최희봉	NSR	책임연구원	응용보안 및 평가인증 프로젝트그룹 위원	TTAKS.KO-12.xxxx
표준 초안 작성자	최희봉	NSR	책임연구원	응용보안 및 평가인증 프로젝트그룹 위원	TTAKS.KO-12.xxxx
	김동민	NSR	연구원	-	TTAKS.KO-12.xxxx
	주왕호	NSR	연구원	-	TTAKS.KO-12.xxxx
	서석충	NSR	선임연구원	-	TTAKS.KO-12.xxxx
	장상운	NSR	실장		TTAKS.KO-12.xxxx
사무국 담당	박수정	TTA	책임	-	

본 표준 발간 이전에 접수된 지식재산권 확약서 정보는 본 표준의 '부록(지식재산권 확약서 정보)'에 명시하고 있으며, 이후 접수된 지식재산권 확약서는 TTA 웹사이트에서 확인할 수 있습니다.
본 표준과 관련하여 접수된 확약서 외의 지식재산권이 존재할 수 있습니다.

발행인 : 한국정보통신기술협회 회장
발행처 : 한국정보통신기술협회
13591, 경기도 성남시 분당구 분당로 47
Tel : 031-724-0114, Fax : 031-724-0109
발행일 : 2017.6.

서 문

1 표준의 목적

본 표준에서는 윈도즈 운영체제, 리눅스 운영체제, 안드로이드 운영체제, iOS 등 운영체제에 따라 잡음원을 수집하는 방법 및 전산기의 CPU 칩에서 제공하는 하드웨어 잡음원 생성기로부터 잡음원을 수집하는 방법과 수집된 잡음원을 응용하는 방법에 대한 지침을 제시한다. 운영체제에서 수집할 수 있는 잡음원은 운영체제가 제공하는 함수들을 이용할 수 있으며, 운영체제에서 수집한 잡음원의 엔트로피가 제한적인 경우 운영체제와 별도로 CPU칩에서 제공하는 하드웨어 잡음원 생성기 등 하드웨어로 구현된 잡음원 생성기를 통하여 잡음원을 추가로 수집할 수 있다. 수집된 잡음원은 기밀성, 인증, 접근통제, 부인봉쇄 등 암호의 안전한 사용을 위해서 난수발생기의 씨드 등에 응용될 수 있다. 본 표준은 암호제품을 개발하는 개발자가 운영체제에서 잡음원을 수집하는 경우와 수집된 잡음원을 응용하는 경우에 적용될 수 있다.

2 주요 내용 요약

난수는 기밀성, 인증, 접근통제, 부인봉쇄 등 암호의 안전한 사용을 위해서 꼭 필요한 요소이다. 암호 사용을 위해 난수를 생성하는 난수발생기는 안전하게 설계되어야 한다. 결정론적 난수발생기(DRBG, Deterministic Random Bit Generator)의 논리를 안전한 표준으로 사용한다고 가정하면 결정론적 난수발생기의 안전성은 난수발생기의 씨드로 사용되는 잡음원의 안전성에 있다. 암호응용에서 잡음원의 안전성은 매우 중요한 것을 알 수 있다.

본 표준은 윈도즈 운영체제, 리눅스 운영체제, 안드로이드 운영체제, iOS 등 운영체제에 따라 잡음원을 수집하는 방법 및 전산기의 CPU에서 제공하는 하드웨어 잡음원 생성기로부터 잡음원을 수집하는 방법을 기술한다. 일반적으로 운영체제에서 수집할 수 있는 잡음원은 시스템 이벤트가 발생될 때마다 변경되는 마우스 정보, 키보드 정보, 인터럽트 요청 정보, 디스크 정보, 시간 정보 등이 될 수 있다. 운영체제에서 수집한 잡음원의 엔트로피가 제한적인 경우 하드웨어로 구현된 잡음원 생성기로부터 잡음원을 추가로 수집한다. 하드웨어에서 수집할 수 있는 잡음원은 제너 다이오드의 산탄 잡음, 반도체 회로의 내재적인 열 잡음, 자유 발진하는 링 오실레이터 등으로 하드웨어 잡음원 발생기를 구현하는 방법이 될 수 있고, 물리적 현상 즉 방사선 붕괴, 광전자 효과 등으로 잡음원 발생기를 구현하는 방법이 될 수 있다. 수집된 잡음원은 기밀성, 인증, 접근통제, 부인봉쇄 등 암호의 안전한 사용을 위해서 난수발생기의 씨드 등에 응용될 수 있으며, 본 표준은 수집된 잡음원의 응용방법 및 응용시 주의사항 등을 기술한다.

수집되는 잡음원에 대한 엔트로피 검사는 국내 표준 및 국제 표준에서 정하는 검증방법에 따르며 본 표준의 적용범위에는 해당하지 않는다.

3 인용 표준과의 비교

3.1 인용 표준과의 관련성

- 해당사항 없음

3.2 인용 표준과 본 표준의 비교표

TTAK.KO-12.xxxx			비고

Preface

1 Purpose

This standard proposes the guideline which describes the method to collect noise source on Operating Systems as like Windows OS, Linux OS, Android OS, iOS, etc. and from hardware noise source generator which is provided in CPU chip, and the method to use the collected noise source.

Noise source may be collected by using API functions which are provided on Operating Systems. If the entropy of the noise source is not enough high to use it in the application environment, additionally noise source may be collected from the noise source generator which is implemented with hardware except for Operating System. The noise source is used as the seed of random bit generator etc. which are necessary to the confidentiality, the authentication, the access control, the non-repudiation etc.

This standard is useful for the developer of cryptography products to collect noise source from Operating Systems or hardware and use it to cryptography area.

2 Summary

Random number is necessary in cryptography area using the service of the confidentiality, the authentication, the non-repudiation, etc. The random bit generator which generates random number has to be designed securely to assure that the application using security services should be secure. If it is assumed that the DRBG(Deterministic Random Bit Generator) is implemented according to the standard which supports secure algorithms, the security of DRBG depends on the noise source which is used as the seed of random bit generator. It shows that the noise source is very important.

This standard describes the method to collect noise source on Operating Systems as like Linux OS, Windows OS, Android OS, iOS, etc. and from hardware noise source generator which is provided in CPU chip. Generally noise source may be mouse information, keyboard information, interrupt request information, disk information, time information, etc. changed whenever system events happen. If the noise source which is collected on Operating Systems cannot satisfy entropy criteria required, additionally we can collect noise source from noise source generator which is implemented with hardware. There are the shot noise of Zener diode, the thermic noise of semiconductor circuit, the noise of ring oscillator as a

hardware noise source. This standard describes guideline for the method to utilize the noise source in the application environment and gives application note.

3 Relationship to Reference Standards

- None

목 차

운영체제별 잡음원 수집 및 응용 지침
(Guideline for the Collection and Application of Noise Source on Operating Systems)

1 적용 범위

현대암호는 난수의 안전한 사용을 가정하여 응용되고 있다. 암호 응용에 사용되는 난수의 안전성이 확보되지 않으면 암호 전체가 취약성을 가지게 된다. 난수를 생성하는 난수발생기의 취약성은 암호모듈이나 암호시스템 전체의 안전성에 크게 영향을 미치고 있으며 실제 암호시스템의 공격에서는 암호알고리즘 자체의 취약성보다 유용하게 활용되고 있다. 난수는 기밀성, 인증, 접근통제, 부인봉쇄 등 암호의 안전한 사용을 위해서 꼭 필요한 요소이다. 난수발생기로는 비결정론적 난수발생기(NDRBG, Non-Deterministic Random Bit Generator)와 결정론적 난수발생기(DRBG, Deterministric Random Bit Generator)가 있다. 비결정론적 난수발생기로부터 얻는 출력을 잡음원이라 하며 암호 응용에 직접 사용할 수도 있으나 일반적으로 결정론적 난수발생기(DRBG, Deterministic Random Bit Generator)의 씨드로 사용한다. 결정론적 난수발생기의 논리를 안전한 알고리즘 표준으로 사용한다고 가정하면 결정론적 난수발생기의 안전성은 씨드로 사용되는 잡음원의 안전성에 있다. 암호응용에서 씨드로 사용되는 잡음원의 안전성은 매우 중요하다는 것을 알 수 있다.

현재까지 미국 FIPS, ISO 국제표준 등에서 난수발생기 및 씨드의 안전성 검증기준을 제시하고 있다.

결정론적 난수발생기의 씨드로 사용될 수 있는 잡음원은 소프트웨어 혹은 하드웨어의 운영환경에 따라 수집하는 방법이 다르다. 운영체제에서 제공하는 잡음원이 있을 수 있으며, 추가로 잡음원 생성기를 하드웨어로 구현하여야 하는 경우도 있다.

본 표준은 윈도즈 운영체제, 리눅스 운영체제, 안드로이드 운영체제, iOS 등 운영체제에 따라 잡음원을 수집하는 방법 및 CPU 칩에서 제공하는 하드웨어 잡음원 생성기로부터 잡음원을 수집하는 방법을 기술한다. 일반적으로 운영체제에서 수집할 수 있는 잡음원은 시스템 이벤트가 발생될 때마다 변경되는 마우스 정보, 키보드 정보, 인터럽트 요청 정보, 디스크 정보, 시간 정보 등이 될 수 있다. 운영체제에서 수집한 잡음원의 엔트로피가 제한적인 경우 하드웨어로 구현된 잡음원 생성기로부터 잡음원을 추가로 수집한다. 하드웨어에서 수집할 수 있는 잡음원은 제너 다이오드의 산탄 잡음, 반도체 회로의 내재적인 열 잡음, 자유 발진하는 링 오실레이터 등으로 하드웨어 잡음원 발생기를 구현하는 방법이 될 수 있고, 물리적 현상 즉 방사선 붕괴, 광전자 효과 등으로 잡음원 발생기를 구현하는 방법이 될 수 있다. 수집된 잡음원에 대한 엔트로피 검사는 국내표준 및 국제표준에서 정하는 검사방법에 따르며 본 표준의 적용범위에는 해당하지 않는다.

본 표준은 운영체제 즉 리눅스 운영체제, 윈도즈 운영체제, 안드로이드 운영체제, iOS 등에 따라 잡음원을 수집하는 방법 및 CPU 칩에서 제공하는 하드웨어 잡음원 생성기로부터 잡음원을 수집하는 방법을 제시하고 수집된 잡음원의 응용에 대한 지침을 제시한다. 본 표준은 암호 개발자들이 잡음원을 수집하는데 활용할 수 있고, 수집된 잡음원을 응용하는데 도움을 줄 수 있다.

2 인용 표준

- 해당사항 없음

3 용어 정의

3.1 난수발생기(RBG, Random Bit Generator)
암호 응용을 위해 사용되는 난수발생기는 일반적으로 0과 1의 비트열을 생성하며, 이 수열은 난수 블록으로 결합될 수 있음

3.2 씨드(Seed)
암호함수 또는 암호연산의 초기화를 위해 사용되는 값으로 엔트로피 소스라 하기도 함

3.3. 결정론적 난수발생기(DRBG, Deterministic Random Bit Generator)
씨드(seed)라고 부르는 초기값으로부터 비트 열을 생성하는 알고리즘으로 구성

3.4. 비결정론적 난수발생기(NDRBG, Non-Deterministic Random Bit Generator)
예측 불가능한 물리적 소스에 의존하는 출력을 생성함

3.5. 잡음원(NS, Noise Source)
예측 불가능한 물리적 소스

3.6. 엔트로피(Entropy)
잡음원에서 발생한 난수의 수준을 평가하기 위해 사용하는 척도

3.7. CPU 칩(Chip)
전산기의 중앙처리장치를 구성하는 집적 회로 모임

4 약어 및 기호

4.1 약어

AES	Advanced Encryption Standard
API	Application Programming Interface
LRNG	Linux Random Number Generator
SP	Special Publication
NIST	National Institute of Standards and Technology
GPU	Graphics Processing Unit
ID	Identification
VPN	Virtual Private Network
WRNG	Windows Random Number Generator
PKI	Public Key Infrastructure
CPU	Central Processing Unit
PRNG	Pseudo Random Number Generator
VCO	voltage controlled oscillator

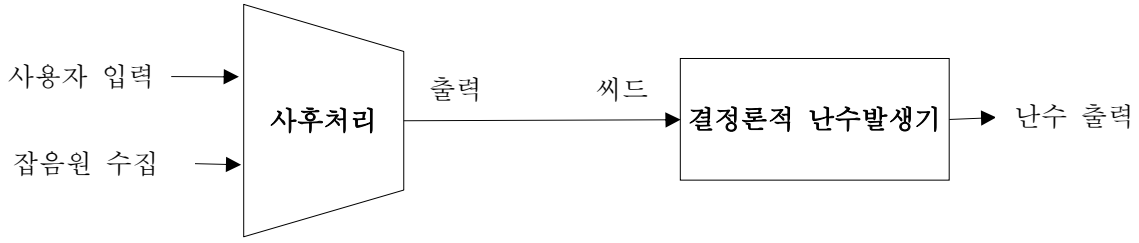
5 운영체제별 잡음원 수집방법 및 응용방법

5.1 잡음원의 중요성

암호학적 난수를 생성하고, 안전하게 응용하기 위해서는 잡음원 수집(씨드 생성)부터 응용까지의 각 단계에 대한 이해가 필수적이며, 다음과 같이 크게 3단계로 구분할 수 있다.

- 잡음원 수집(씨드 생성)
- 사후처리 (post processing, 엔트로피 증폭)
- 결정론적 난수발생기 (암호학적 난수 출력)

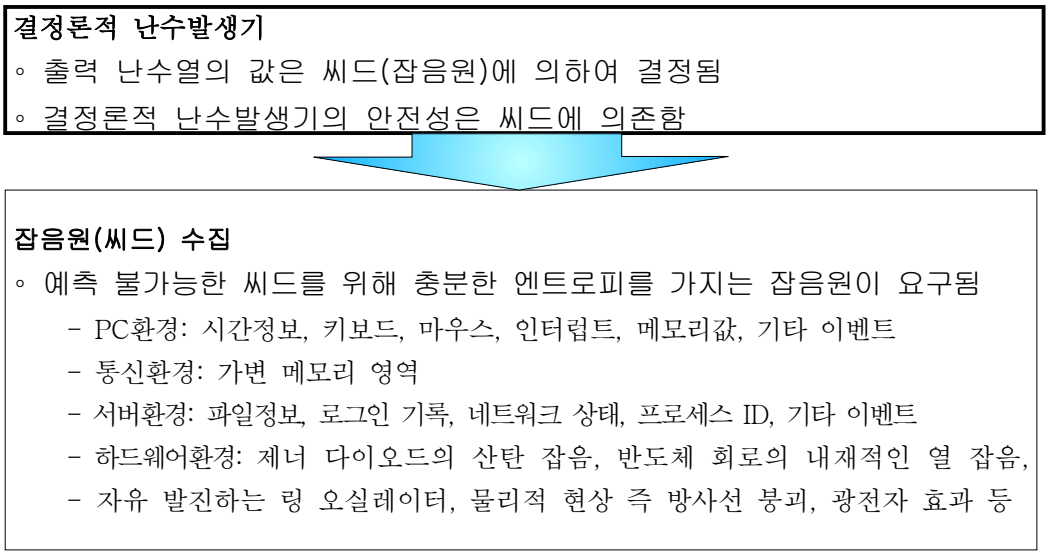
(그림 5-1)은 난수를 생성하는 3단계를 나타내고 있다.



(그림 5-1) 난수 생성단계

(그림 5-1)의 난수 생성단계 중 결정론적 난수발생기와 사후처리(post processing)에서는 안전성을 정량적으로 확인할 수 있으며, 관련된 기법들이 널리 알려져 있다. 결정론적 난수발생기의 출력과 암호문의 통계적 특성을 분석하기 위한 정량적인 기준 연구는 매우 활발히 전개되어 왔으며, 20여 가지의 정량적인 평가방법이 알려져 있다. 특히 미국 NIST의 SP 800-22는 미국표준 블록암호 AES의 선정에 활용되면서 암호학적 난수가 만족해야 할 중요한 기준으로 인정받고 있다. 그리고 하드웨어 잡음원과 씨드에 대한 평가기준은 독일 BSI의 AIS.31[1] 및 NIST의 SP 800-90B[2]에서 다양한 평가기법을 제시하고 있다.

대부분의 암호모듈에서는 결정론적 난수생성을 위해 표준난수발생기를 사용하고 있어 난수발생기의 출력에 대한 통계적 난수성은 확보된다. 통계적 난수성은 위에 언급된 다양한 평가방법을 통해 확인할 수 있다. 씨드로 사용된 잡음원의 엔트로피가 제한적이라면, 출력되는 난수는 통계적 특성은 우수하지만 예측 가능한 수열이 된다. 즉 씨드로 사용된 잡음원의 엔트로피 이하의 비도를 갖는다. 따라서 잡음원의 엔트로피는 난수발생기의 안전성에 영향을 크게 미친다.



(그림 5-2) 잡음원(씨드)의 중요성 및 종류

(그림 5-2)는 운영환경에 따라 달리 수집되는 잡음원이 결정론적 난수발생기의 안전성에 있어서 매우 중요함을 나타내고 있다.

암호에 응용되는 씨드(잡음원)의 수집방법은 운영환경에 따라 다르다. 잡음원은 운영체제에서 제공하는 함수에 의해서 수집될 수 있으며, 일반적으로 운영체제에서 발생하는 이벤트에 의해 수집되는 잡음원의 엔트로피는 충분하지 않으며, 하드웨어로 구현된 잡음원 생성기를 추가로 사용해야 되는 경우가 대부분이다.

5.2. 리눅스 운영체제의 잡음원 수집방법 및 응용방법

5.2.1. 잡음원 수집방법

리눅스 운영체제에서 수집할 수 있는 잡음원은 다음과 같다.

가. 리눅스 난수발생기(LRNG)의 출력

리눅스 운영체제에서 리눅스 난수발생기의 출력을 잡음원으로 수집할 수 있다. 리눅스 난수발생기(LRNG)의 씨드는 키보드, 마우스, 디스크, 인터럽트 등 이벤트 발생시 리눅스 운영체제에서 제공되는 함수를 통해 수집된 잡음원을 사용한다. 리눅스 운영체제에서 수집할 수 있는 잡음원은 /dev/urandom 함수 또는 /dev/random 함수를 호출하여 그 출력을 사용하면 된다.

리눅스 난수발생기의 씨드로 사용되는 키보드, 마우스, 디스크 이벤트에 대한 설명은 다음과 같다.

◦ 키보드

키보드 이벤트에서 두 번째 워드에 저장되는 값은 0에서 255까지의 값을 갖는다.

◦ 마우스

마우스 이벤트는 타입(type), 코드(code), 값(value) 의 3가지 정보를 추출한다. 타입은 버튼이 눌려졌는지 버튼을 놓았는지, 움직이기 시작하는지 멈추는지를 나타내며, 코드는 마우스 버튼, 휠 스크롤, 움직이는 축 등을 의미하며, 값은 버튼이 눌렸는지 놓였는지, 휠의 움직이는 방향, 마우스의 움직이는 양 등을 뜻한다. 마우스로부터 얻어지는 정보의 양은 움직이는 양 등의 정보에서 10비트 엔트로피, 버튼정보 등에서 2비트 엔트로피를 넘지 못해 총 12비트 엔트로피 이하가 된다.

◦ 디스크

디스크 이벤트에서는 IDE(Integrated Drive Electronics), SCSI(Small Computer System Interface), 플로피 등의 입출력 이벤트가 종료되면 운영체제가 제공하는 정보를 메이저(major)와 마이너(minor)로 나누어 잡음원을 수집한다. IDE 디스크의 경우 메이저는 디스크의 번호를 의미하며(첫 IDE의 경우 3), 마이너 정보는 각 파티션 정보를 반환한다. 8개 이하의 디스크를 사용하는 시스템의 경우 최대 3비트의 엔트로피를 제공한다.

◦ 인터럽트

사용된 인터럽트의 번호를 반환한다. 인터럽트(IRQ)의 번호는 0과 15사이에서 정해진다. 따라서 인터럽트는 최대 4비트 엔트로피를 제공한다.

나. 리눅스 운영체제에서 잡음원을 제공하는 함수 및 정보

리눅스 운영체제에서 리눅스 난수발생기 /dev/urandom 함수 출력 및 리눅스 난수발생기 /dev/random 함수 외 추가 잡음원을 제공하는 함수 및 정보는 <표 5-1>과 같다. <표 5-1>에 나타난 함수 및 정보를 모두 이용하여 잡음원을 수집하면 함수 호출하는데 시간이 많이 소모될 수 있다. 따라서 잡음원을 응용하는 암호제품에 따라 호출함수를 선택할 수 있다. 또한 운영체제 버전에 따라 제공하는 함수를 선별해서 사용한다.

<표 5-1> 리눅스 운영체제의 잡음원 제공 함수 및 정보

수집정보	함수명
리눅스 난수발생기 (Blocking 없이 사용 가능)	/dev/urandom
리눅스 난수발생기	/dev/random
현재 프로세스 ID	getpid

현재 프로세스의 부모 ID	getppid
그룹 ID	getgid
시스템의 현재 시간	gettimeofday
프로세스의 자원 사용량	getrusage
사용된 디스크의 크기 정보	/bin/df -a 2> /dev/null
수행중인 모든 프로세스의 정보	/bin/ps -elf 2> /dev/null
이더넷(ethernet) 정보	/sbin/ifconfig -a 2> /dev/null
메모리 사용량 정보	/bin/cat /proc/meminfo 2> /dev/null
장치에서 발생된 인터럽트(IRQ) 정보	/bin/cat /proc/interrupts 2> /dev/null
시스템 자원의 평균 부하량	/bin/cat /proc/loadavg 2> /dev/null
시스템의 어라이브(alive) 시간	/bin/cat /proc/uptime 2> /dev/null
시스템 사용 통계	/bin/cat /proc/stat 2> /dev/null
저장장치 IO 통계 현황	/bin/cat /proc/diskstats 2> /dev/null
가상메모리 통계	/bin/cat /proc/vmstat 2> /dev/null
시스템 디바이스 정보	Number of Disks, Number of Floppies, Number of CD Roms, Number of Tapes, Number of Serial Ports, Number of Parallel Ports
시스템 프로세스 전력 정보	Current Frequency, Thermal Limit Frequency, Constant Throttle Frequency, Degraded Throttle Frequency, Last Busy Frequency, etc
Local Date & Time	localtime()
Traffic 정보	/proc/net/dev
가상메모리 정보	/proc/vmstat
시스템 자원의 평균 부하량	/proc/loadavg
인텔칩 하드웨어 난수발생기	RDRAND
인텔칩 하드웨어 난수발생기	RDSEED

◦ 리눅스 난수발생기 /dev/urandom 함수 분석
- Blocking 없이 난수 생성
- 짧은 시간내에 난수를 많이 사용할 시 엔트로피를 보장하지 못함

◦ 리눅스 난수발생기 /dev/random 분석
- Blocking을 사용한 난수 생성방법
- 프로세스가 달라도 Blocking 카운터
- 난수를 생성할 때 리씨드 하므로 엔트로피를 보장함

◦ 리눅스 getpid, 프로세스 id, 부모 id 난수 분석
- 프로세스가 생성될 때마다 난수성을 갖는 프로세스 id가 할당된다.
- 이 때 난수는 엔트로피를 갖는 잡음원이 아니라 PRNG의 출력을 주로 사용하고 있어 엔트로피가 낮다.

다. 리눅스 운영체제 외 하드웨어 잡음원 수집

리눅스 운영체제에서 함수들이 제공하는 잡음원의 엔트로피는 암호에 응용할 만큼 충분하지 못하다. 따라서 리눅스 운영체제외 하드웨어 구현에 의한 잡음원 수집방법들이 필요하며 다음과 같은 수집방법이 있다.

◦ CPU 칩에서 제공하는 하드웨어 잡음원 생성기로부터 잡음원을 수집하는 방법
- 인텔 칩을 사용하는 경우 RDRAND 및 RDSEED를 사용하여 잡음원을 수집할 수 있다.
- RDRAND는 하드웨어 잡음원이 PRNG의 씨드로 사용되고 PRNG의 출력이 난수가 되므로, PRNG의 리씨드를 수행하도록 512비트를 128번 수집하면 128 비트 엔트로피를 얻을 수 있다.
- RDSEED는 수집하는 비트 수의 90% 이상이 엔트로피로 평가된다 (예: 100비트를 수집하여 SP 800-90B에 의한 평가방법으로 엔트로피를 평가하면 90비트 이상의 엔트로피 결과를 얻을 수 있다.)
- RDRAND는 CPU 버전 Ivy Bridge부터 지원되며, RDSEED는 Broadwell부터 지원된다.
- 다음 <표 5-2>는 CPU 하드웨어 잡음원 지시문 호출 시 각 명령어에 대한 지원 여부가 저장되는 CPU 레지스터 정보이다.

<표 5-2> 리눅스 운영체제에서 하드웨어 잡음원 지원 CPU 레지스터 정보

Leaf	레지스터	Bit	Mnemonic	설명
1	ECX	30	RDRAND	해당 비트가 1일 경우 RDRAND 지원됨
7	EBX	18	RDSEED	해당 비트가 1일 경우 RDSEED 지원됨

◦ GPU의 온도측정 함수를 이용한 잡음원 생성[3]
- 온도측정함수의 수행시간이 일정하지 않음을 이용함

◦ 공유 메모리의 경쟁상태(Race Condition)를 이용한 잡음원 생성[3]
- 공유 메모리를 여러 개의 코어 프로세서가 동시에 접근하려고 할 때 수행순서와 수행시간을 예측할 수 없다는 특성을 이용함

◦ 잡음원 생성기가 구현된 하드웨어 모듈로부터 잡음원을 수집함(부록 II. 하드웨어 잡음원 수집방법 및 응용방법을 참조)

5.2.2. 응용방법

리눅스 운영체제에서 수집된 잡음원의 응용 분야는 다음과 같다.

◦ 결정론적 난수발생기의 씨드
◦ 비밀키 및 공개키 암호 알고리즘의 키 값 초기화
◦ 암호프로토콜에서 난스(nonce) 값
◦ 랜덤한 패딩 값

◦ 블록 암호에서 랜덤한 입력값
◦ 인증을 위한 일회용 랜덤 값

잡음원의 응용 시 주의사항은 다음과 같다.

◦ 잡음원은 이벤트 발생에 의하여 수집되는 정보를 사용해야 한다. 즉 이벤트 발생에 의해 변경되는 정보를 잡음원로 사용해야 한다.
◦ 리눅스 운영체제에서 잡음원을 수집하기 위해 리눅스 운영체제에서 제공하는 모든 함수를 호출할 경우 수행시간이 많이 소모될 수 있으므로 잡음원이 응용되는 암호제품에 따라 호출함수를 선택해야 한다.
◦ 잡음원은 결정론적 난수발생기의 씨드로 사용하기 위하여 저장될 경우 안전하게 보호하여야 한다.
◦ 리눅스 운영체제에서 잡음원을 수집할 때마다 잡음원에 대한 온라인 검사(III.1절 참조)를 수행하여야 하며 실패시 잡음원을 재수집해야 한다. 온라인 검사기능이 암호제품에 탑재되기 위해서는 빠르게 동작해야 하며, 짧은 코드길이를 가져야 한다.
◦ 결정론적 난수발생기의 씨드로 사용할 경우 잡음원을 정기적으로 재수집하는 것이 필요하다.
◦ 리눅스 운영체제에서 소프트웨어적인 함수만을 호출하여 잡음원을 수집하는 경우 잡음원의 엔트로피가 충분하지 못하므로 하드웨어 잡음원을 사용하거나 또는 추가로 잡음원을 외부에서 입력받을 것을 권고한다.
◦ 수집된 잡음원이 씨드의 길이를 초과할 때 해시 또는 MAC으로 축소하여 사용한다.
◦ 수집된 잡음원에 대한 오프라인 검사(III-2 절 참조)를 수행하여 잡음원의 엔트로피를 검사할 수 있다.

5.3. 윈도즈 운영체제의 잡음원 수집방법 및 응용방법

5.3.1. 잡음원 수집방법

윈도즈 운영체제에서 수집할 수 있는 잡음원은 리눅스 운영체제에서 수집할 수 있는 잡음원과 유사하다.

가. 윈도즈 난수발생기(WRNG)의 출력

윈도즈 운영체제에서 윈도즈 난수발생기의 출력을 잡음원으로 수집할 수 있다. 윈도즈 난수발생기(WRNG)의 씨드는 키보드, 마우스, 디스크 이벤트, 인터럽트 등 이벤트가 발생시 윈도즈 운영체제에서 제공되는 함수를 통해 수집된 잡음원을 사용한다. 윈도즈 운영체제가 제공하는 잡음원은 윈도즈 난수발생기 CryptGenRandom() 함수를 호출하면 된다.

나. 윈도즈 운영체제에서 잡음원을 제공하는 함수 및 정보

윈도즈 운영체제에서 윈도즈 난수발생(WRNG)의 출력 외에 추가 잡음원을 제공하는 함수 및 정보는 <표 5-3>과 같다. <표 5-3>에 나타난 함수 및 정보를 모두 이용하여 잡음원을 수집하면 함수 호출하는데 시간이 많이 소요될 수 있다. 따라서 잡음원을 응용하는 암호제품에 따라 호출함수를 적절하게 선택할 수 있다. 또한 운영체제 버전에 따라 제공하는 함수를 선별해서 사용한다.

<표 5-3> 윈도즈 운영체제의 잡음원 제공 함수 및 정보

수집정보	함수명
windows 난수발생기 (난수 또는 엔트로피 소스 설정 가능)	CryptGenRandom()
시스템의 구성 정보	GetSystemInfo
운영체제 버전 정보	GetVersionEx
현재 프로세스 ID	GetCurrentProcessId
현재 스레드 ID	GetCurrentThreadId
메모리의 상태 정보	GlobalMemoryStatusEx
마우스 커서 위치	GetCursorPos
부팅한 이후 누적된 CPU의 클럭	QueryPerformanceCounter
1초 동안 발생한 CPU의 클럭	QueryPerformanceFrequency
프로세스 힙의 핸들	GetProcessHeap
부팅한 이후 경과된 시간	GetTickCount
운영체제의 현재 시간 정보	GetSystemTime
GUID	CoCreateGuid
운영체제의 메트릭 정보	GetSystemMetrics
힙 리스트의 힙 엔트리 정보	heapListFirst, heapListNext, heapFirst, heapNext
시스템에서 실행되는 프로세스 리스트의 엔트리 정보	processFirst, processNext
시스템에서 실행되는 스레드 리스트의 엔트리 정보	threadFirst, threadNext
프로세스에 속한 모듈의 엔트리 정보	moudleFirst, moudleNext
특정 워크스테이션에 대한 통계정보	netstatget(“LadymanWorkstation”)
특정 서버에 대한 통계정보	netstatget(“LadymanServer”)
남은 디스크 공간정보	GetFreeSpace
시스템 디바이스 정보	Number of Disks, Number of Floppies, Number of CD Roms, Number of Tapes, Number of Serial Ports, Number of Parallel Ports
현재 프로세스 환경 블록의 해쉬값	-
특정 하드웨어 CPU 사이클 카운터	-
시스템 파일 캐쉬 정보	Current Size, Peak Size, Page Fault Count, Minimum Working Set, Maximum Working Set, etc
시스템 프로세스 전력정보	Current Frequency, Thermal Limit

	Frequency, Constant Throttle Frequency, Degraded Throttle Frequency, Last Busy Frequency, etc
시스템 프로세스 휴면 정보	-
시스템 프로세스 성능 정보	Idle Process Time, Read Transfer Count, Io Write Transfer Count, Io Read Operation Count, Io Write Operation Count, Available Pages, Committed Pages, etc
시스템 제외 정보	Alignment Fix up Count, Exception Dispatch Count, Floating Emulation Count, Byte Word Emulation Count
시스템 프로세스 성능 정보	Idle Time, Kernel Time, User Time, Deferred Process Call Time, Interrupt Time Interrup Count
시스템 인터럽트 정보	context switches, deferred procedure call count, deferred procedure call rate, etc
인텔칩 하드웨어 난수발생기	RDRAND
인텔칩 하드웨어 난수발생기	RDSEED

◦ CryptGenRandom()의 난수 분석
- 윈도즈 운영체제에서 수집된 잡음원은 PRNG의 씨드로 사용되며 PRNG의 출력이 CryptGenRandom()이 생성하는 난수가 된다. 따라서 난수는 PRNG가 리씨드를 하지 않으면 엔트로피가 낮다.
- 난수로 사용할 경우 CryptGenRandom()의 출력을 사용하면 된다.
- 엔트로피 소스로 사용할 경우 CryptGenRandom()에 포함되는 PRNG를 리씨드할 수 있도록 128 x 128바이트를 생성하여 사용한다.

◦ GUID, 프로세스 ID, 쓰레드 ID의 난수 분석
- GUID는 프로세스가 생성될 때마다 난수가 ID로 할당된다. 이 때 난수는 PRNG의 출력을 사용하므로 엔트로피가 낮다.
- ID에 할당되는 난수는 PRNG를 사용함으로써 난수성이 좋으나 대부분 엔트로피가 낮음에 유의해야 한다.

다. 윈도즈 운영체제 외 하드웨어 잡음원 수집

리눅스 운영체제와 유사하게 윈도즈 운영체제에서 함수들이 제공하는 잡음원의 엔트로피는 암호에 응용할 만큼 충분하지 못하다. 따라서 윈도즈 운영체제 외 하드웨어 구현에 의한 잡음원 수집방법들이 필요하며 다음과 같은 수집방법이 있다.

◦ 인텔 CPU 칩에서 제공하는 하드웨어 잡음원 생성기로부터 잡음원 수집하는 방법
- RDRAND 및 RDSEED를 사용하여 잡음원 수집할 수 있다.
- RDRAND는 하드웨어 잡음원이 PRNG의 씨드로 사용되고 PRNG의 출력이 난수가 되므로, PRNG의 리씨드를 수행하도록 512비트를 128번 수집하면 128 비트 엔트로피를 얻을 수 있다.

- RDSEED는 수집하는 비트 수의 90% 이상이 엔트로피로 평가된다 (예: 100비트를 수집하여 SP 800-90B에 의한 평가방법으로 엔트로피를 평가하면 90비트 이상의 엔트로피 결과를 얻을 수 있다.)
- RDRAND는 CPU 버전 Ivy Bridge부터 지원되며, RDSEED는 Broadwell부터 지원된다.
- 다음 <표 5-4>는 CPU 하드웨어 잡음원 호출 시 각 명령어에 대한 지원 여부가 저장되는 CPU 레지스터 정보이다.

<표 5-4> 윈도즈 운영체제에서 하드웨어 잡음원 지원 CPU 레지스터 정보

Leaf	레지스터	Bit	Mnemonic	설명
1	ECX	30	RDRAND	해당 비트가 1일 경우 RDRAND 지원됨
7	EBX	18	RDSEED	해당 비트가 1일 경우 RDSEED 지원됨

◦ GPU의 온도측정 함수를 이용한 잡음원 생성[3]
  - 온도측정함수의 수행시간이 일정하지 않음을 이용함
◦ 공유 메모리의 경쟁상태(Race Condition)를 이용한 잡음원 생성[3]
  - 공유 메모리를 여러 개의 코어 프로세서가 동시에 접근하려고 할 때 수행순서와 수행시간을 예측할 수 없다는 특성을 이용함
◦ 잡음원 생성기가 구현된 하드웨어 모듈로부터 잡음원을 수집함(부록 II. 하드웨어 잡음원 수집방법 및 응용방법 참조)

5.3.2. 응용방법

윈도즈 운영체제에서 수집된 잡음원의 응용 분야는 다음과 같다.

◦ 결정론적 난수발생기의 씨드
◦ 비밀키 및 공개키 암호 알고리즘의 키 값 초기화
◦ 암호프로토콜에서 난스(nonce) 값
◦ 랜덤한 패딩 값
◦ 블록 암호에서 랜덤한 입력값
◦ 인증을 위한 일회용 랜덤 값

잡음원의 응용 시 주의사항은 다음과 같다.

◦ 윈도즈 운영체제의 잡음원은 이벤트 발생에 의하여 수집되는 정보를 사용해야 한다. 즉 이벤트 발생에 의해 변경되는 정보를 잡음원로 사용해야 한다.
◦ 윈도즈 운영체제에서 잡음원을 수집하기 위해 윈도즈 운영체제에서 제공하는 모든 함수를 호출할 경우 수행시간이 많이 소요될 수 있으므로 잡음원이 응용되는 암호제품에 따라 호출함수를 선택해야 한다.
◦ 잡음원은 결정론적 난수발생기의 씨드로 사용하기 위하여 저장될 경우 안전하게 보

호하여야 한다.

◦ 윈도즈 운영체제에서 잡음원을 수집할 때마다 잡음원에 대한 온라인 검사(III-1 절 참조)를 수행하여야 하며 실패시 잡음원을 재수집해야 한다. 온라인 검사기능이 암호 제품에 탑재되기 위해서는 빠르게 동작해야 하며, 짧은 코드길이를 가져야 한다.

◦ 잡음원이 결정론적 난수발생기의 씨드로 사용될 경우 정기적으로 재수집하는 것이 필요하다.

◦ 윈도즈 운영체제에서 소프트웨어적인 함수 호출에 의해 잡음원을 수집하는 경우 잡음원의 엔트로피가 충분하지 못하므로 하드웨어 잡음원을 사용하거나 또는 추가로 잡음원을 외부에서 입력받을 것을 권고한다.

◦ 수집된 잡음원이 씨드의 길이를 초과할 때 해시 또는 MAC으로 축소하여 사용한다.

◦ 수집된 잡음원에 대한 오프라인 검사(III-2 절 참조)를 수행하여 잡음원의 엔트로피를 검사할 수 있다.

5.4. 안드로이드 운영체제의 잡음원 수집방법 및 응용방법

5.4.1. 잡음원 수집방법

안드로이드 운영체제에서 수집할 수 있는 잡음원은 다음과 같다.

가. 안드로이드 난수발생기의 출력

안드로이드 운영체제에서 리눅스 난수발생기와 동일한 안드로이드 난수발생기의 출력을 잡음원으로 수집할 수 있다. 안드로이드 난수발생기의 씨드는 키보드, 디스크 이벤트, 인터럽트 등 이벤트 발생시 안드로이드 운영체제에서 제공되는 함수를 통해 수집된 잡음원을 사용한다. 안드로이드 운영체제가 제공하는 잡음원은 /dev/urandom 함수 및 /dev/random 함수를 호출하면 된다.

나. 안드로이드 운영체제에서 잡음원을 제공하는 함수 및 정보

안드로이드 운영체제에서 안드로이드 난수발생기 출력 외 추가 잡음원을 제공하는 함수 및 정보는 <표 5-5>과 같다. 운영체제 버전에 따라 제공하는 함수를 선별해서 사용한다.

<표 5-5> 안드로이드 운영체제의 잡음원 제공 함수 및 정보

수집정보	함수명
안드로이드 난수발생기 (Blocking 없이 사용 가능)	/dev/urandom
안드로이드 난수발생기	/dev/random
현재 프로세스의 ID	getpid

현재 프로세스의 부모의 ID	getppid
그룹의 ID	getgid
시스템의 현재 시간	gettimeofday
현재 프로세스의 자원 사용량	getrusage
시스템에서 실행중인 모든 프로세스와 쓰레드의 정보	ps -p -t
커널 상태에 대한 정보	dmesg
메모리 사용량 정보	cat /proc/meminfo
장치에서 발생된 인터럽트(IRQ) 정보	cat /proc/interrupts
시스템 자원의 평균 부하량	cat /proc/loadavg
시스템의 어라이브(alive) 시간	cat /proc/uptime
시스템 사용 통계	cat /proc/stat
저장장치 I/O 통계 현황	cat /proc/diskstats
가상 메모리의 사용 통계	cat /proc/vmstat
사용된 프로세스 시간	clock
수행중인 모든 프로세스의 정보	ps -elf
유저 ID	getuid

◦ 프로세스 ID, 그룹 ID의 난수 분석
  - ID에 할당되는 난수는 PRNG를 사용함으로써 난수성이 좋으나 대부분 엔트로피가 낮음에 유의해야 한다.

다. 안드로이드 운영체제 외 하드웨어 잡음원 수집

안드로이드 운영체제에서 함수들이 제공하는 잡음원의 엔트로피는 암호에 응용할 만큼 충분하지 못하다. 따라서 안드로이드 운영체제 외 하드웨어 구현에 의한 잡음원 수집방법들이 필요하며 다음과 같은 수집방법이 있다.

◦ 최근 안드로이드 운영체제를 탑재하는 CPU 칩에는 하드웨어 잡음원 생성기가 구현되어 있으며 개발자 권한으로 잡음원 수집할 수 있다.
◦ 잡음원 생성기가 구현된 하드웨어 모듈로부터 잡음원을 수집함(부록 II. 하드웨어 잡음원 수집방법 및 응용방법 참조)

5.4.2. 잡음원 응용방법

안드로이드 운영체제에서 잡음원의 응용 및 응용시 주의사항은 리눅스 운영체제와 동일하다.

5.5. iOS 운영체제의 잡음원 수집방법 및 응용방법

5.5.1. 잡음원 수집방법

iOS에서 수집할 수 있는 잡음원은 다음과 같다.

가. iOS 난수발생기의 출력

iOS 운영체제에서 리눅스 난수발생기와 동일한 iOS 난수발생기의 출력을 잡음원으로 수집할 수 있다. iOS 난수발생기의 씨드는 키보드, 디스크, 인터럽트 등 이벤트 발생시 iOS 운영체제에서 제공되는 함수를 통해 수집된 잡음원을 사용한다. iOS 운영체제가 제공하는 잡음원은 /dev/urandom 함수 혹은 응용 API에서 사용되는 /dev/urandom 반환함수 SecURandomCopyBytes 함수 및 /dev/random 함수 혹은 응용 API에서 사용되는 /dev/random 반환함수 SecRandomCopyBytes 함수를 호출하면 된다.

나. iOS 운영체제에서 잡음원을 제공하는 함수 및 정보

iOS 운영체제에서 iOS 난수발생기 출력 외 추가 잡음원을 제공하는 함수 및 정보는 <표 5-6>와 같다. 운영체제 버전에 따라 제공하는 함수를 선별해서 사용한다.

<표 5-6> iOS 운영체제의 잡음원 제공 함수 및 정보

수집정보	함수명
iOS 난수발생기	SecURandomCopyBytes
iOS 난수발생기	SecRandomCopyBytes
현재 프로세스 ID	getpid
현재 프로세스의 부모 ID	getppid
그룹 ID	getgid
시스템의 현재 시간	gettimeofday
프로세스의 자원 사용량	getrusage
시스템에 마운트된 저장장치의 정보	getmntinfo
네트워크 인터페이스와 IP정보	getifaddrs
메모리 사용량 정보	host_statistics(HOST_VM_INFO)
시스템의 어라이브(alive) 시간	sysctl(CTL_KERN, KERN_BOOTTIME)
수행중인 모든 프로세스의 정보	sysctl(CTL_KERN, KERN_PROC, KERN_PROC_ALL)
사용된 프로세스 시간	clock
유저 ID	getuid

◦ 프로세스 ID, 그룹 ID의 난수 분석

- ID에 할당되는 난수는 PRNG를 사용함으로써 난수성이 좋으나 대부분 엔트로피가

낮음에 유의해야 한다.

다. iOS 운영체제 외 하드웨어 잡음원 수집

iOS 운영체제에서 함수들이 제공하는 잡음원의 엔트로피는 암호에 응용할 만큼 충분하지 못하다. 따라서 하드웨어 구현에 의한 잡음원 수집방법들이 필요하며 다음과 같은 수집방법이 있다.

◦ 최근 iOS 운영체제를 탑재하는 CPU 칩에는 하드웨어 잡음원 생성기가 구현되어 있으며 개발자 권한으로 잡음원 수집할 수 있다.
◦ 잡음원 생성기가 구현된 하드웨어 모듈로부터 잡음원을 수집함(부록 Ⅱ. 하드웨어 잡음원 수집방법 및 응용방법 참조)

5.5.2. 잡음원 응용방법

iOS 운영체제에서 잡음원의 응용 및 응용시 주의사항은 리눅스 운영체제와 동일하다.

부 록 Ⅰ-1

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

지식재산권 확약서 정보

Ⅰ-1.1 지식재산권 확약서(1) : 해당 사항 없음

- 발명의 명칭
- 권리자의 성명
- 등록(출원) 번호
- 등록(출원) 연월일
- 실시조건
- 확약서 접수일

Ⅰ-1.2 지식재산권 확약서(2) : 해당 사항 없음

- 발명의 명칭
- 권리자의 성명
- 등록(출원) 번호
- 등록(출원) 연월일
- 실시조건
- 확약서 접수일

부 록 Ⅰ-2

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

시험인증 관련 사항

Ⅰ-2.1 시험인증 대상 여부 : 해당 사항 없음

Ⅰ-2.2 시험표준 제정 현황 : 해당 사항 없음

부 록 Ⅰ-3

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

본 표준의 연계(family) 표준

해당 사항 없음

부 록 Ⅰ-4

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

참고 문헌

[1] W. Killmann, W. Schindler, A Proposal for: Functionality classes and evaluation methodology for true (physical) random number generators version 3.1, BIS 2001

[2] E. Barker, J. Kelsey, Recommendation for the Entropy Sources Used for Random Bit Generation, NIST Second Draft Special Publication 800-90B, 2016.

[3] J. Chan, B. Sharma, J. Lv, G. Thomas, R. Thulasiram, P. Thulasirman, “True random number generator using GPUs and Histogram equalization techniques”, IEEE HPCC, 2011.

[4] B. Jun and P. Kocher, “The Intel random number generator,” white paper prepared for Inter Corporation, Apr. 1999.

[5] V. Fischer and M. Drutarovsky, “True random number generator embedded in reconfigurable hardware,” in Proc. CHES 2002, LNCS, vol. 2523, pp. 415-430.

[6] M. Epstein, L. Hars, R. Krasinski, M. Rosner, and H. Zheng, “Design and implementation of a true random number generator based on digital circucit artifacts,” in Proc. CHES 2003, LNCS, vol. 2779, pp. 152-165, 2003.

[7] NIST, Security Requirements for Cryptographic Modules, Federal Information Processing Standards Publication 140-2, 2002.

[8] W. Schindler, "Efficient online tests for true random number generators," in Proc. CHES 2001, LNCS, vol. 2162, pp. 103-117, 2001.

[9] A. Renyi, "On measures of entropy and information," in Proc. 4th Berkeley Symp. Mathematical Statistics and Probability, vol. 1, 1961, pp. 547-561.

[10] 소프트웨어 환경에서의 잡음원 엔트로피 검증 알고리즘, TTAS,KO-12.xxxx, 2017.6

부 록 Ⅰ-5

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

영문표준 해설서

해당 사항 없음

부 록 Ⅰ-6

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

표준의 이력

판수	채택일	표준번호	내용	담당 위원회
제1판	2013.12.xx	제정 TTAKS.KO-12.0235	-	응용보안/평가인증(PG504)
제2판	2017.6.xx	개정 TTAKS.KO-12.xxxx	운영체제별 잡음원 추가 및 응용지침 보완	응용보안/평가인증(PG504)

부 록 II

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

하드웨어 잡음원 수집방법 및 응용방법

II-1 하드웨어 잡음원 수집방법

하드웨어에 기반하여 잡음원을 수집하는 방법들은 다음과 같다.

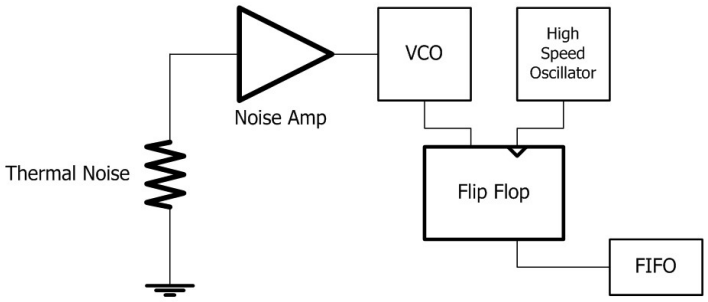
◦ 전자회로 상의 비결정론적 현상들을 이용하여 구현한 잡음원 생성기
  - 제너(Zener) 다이오드의 산탄 잡음(Shot noise)
  - 반도체 회로의 내재적인 열 잡음(thermal noise)
  - 자유 발진하는 링 오실레이터

◦ 물리적 현상들을 이용하여 구현한 잡음 생성기
  - 방사성 붕괴, 광전자 효과 등

다음에서는 난수발생기 개발자에 도움이 될 수 있는 하드웨어로 구현된 난수발생기의 사례를 설명한다.

II-1-1 인텔 난수발생기

(그림 II-1)은 인텔 난수발생기의 잡음원 생성을 나타내고 있으며 그림에서 보는 바와 같이 아날로그로 설계되어 있다[4].



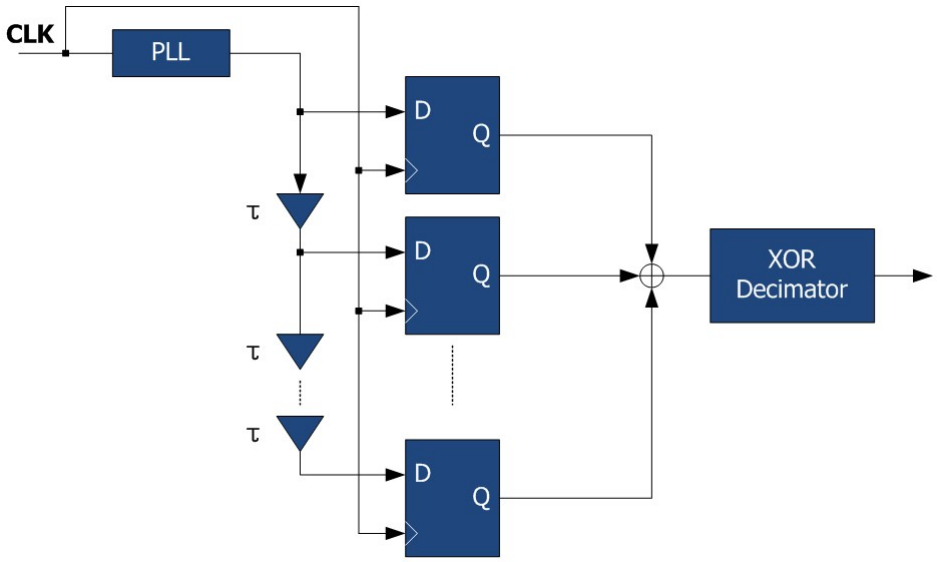
(그림 II-1) 인텔 난수발생기

인텔 난수발생기는 전자 회로 상에 존재하는 전기적 저항에 내재된 열잡음을 아날로그적으로 증폭시킨 후에 이 신호를 플립플롭(F/F)에 통과시킴으로써 이진 난수 값을 얻게 된다. 인텔 난수발생기는 기본적으로 두 개의 레지스터에서 발생한 열잡음을 OP-AMP를 사용해서 차분 증폭시키고 있다. 두 레지스터의 열잡음의 차이가 증폭된 후에는, 증폭된 값이 VCO(voltage controlled oscillator)를 구동시키고 VCO의 출력은 또 다른 오실레이

터에 의해 동작하는 래치(latch)를 이용해 잡음을 취하게 된다.

Ⅱ-1-2 Fischer-Drutarovsky 난수발생기

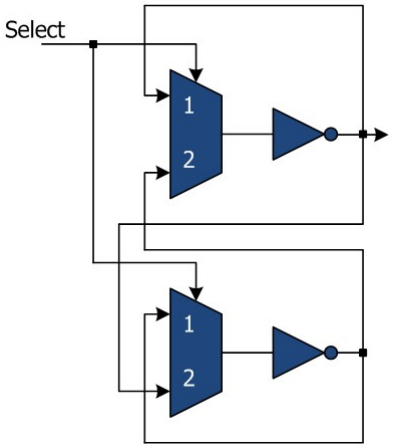
(그림 Ⅱ-2)는 Fischer와 Drutarovsky가 제안한 PLL(Phase Locked Loop)의 지터(jitter)에 기반한 난수발생기이다[5]. 이 난수발생기에서는 온칩(on-chip) PLL의 지터를 지연 버퍼(delay buffer)를 통해 서로 다른 시간으로 D 플립플롭에 입력이 들어가도록 하는 구조를 갖고 있다. D 플립플롭에서 취해진 잡음들은 이원가산기(XOR)를 통해서 연산되고 이 값은 다시 잡음의 품질을 개선시키기 위해 이원가산기 데시메이터(XOR Decimator)를 통과하게 된다.



(그림 Ⅱ-2) Fischer-Drutarovsky 난수발생기

Ⅱ-1-3 Epstein-Hars-Krasinski-Rosner-Zheng 난수발생기

(그림 Ⅱ-3)는 Epstein 등이 제안한 바이스테이블(bi-stable) 회로에 기반한 난수발생기이다[6].

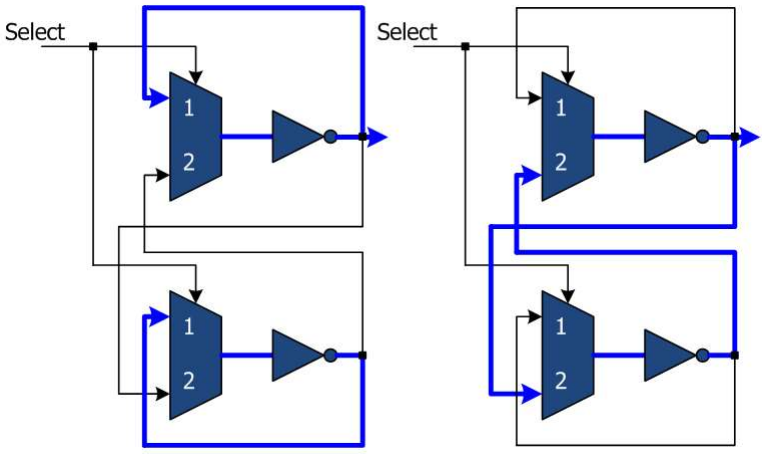


(그림 Ⅱ-3) 바이스테이블(bi-stable) 회로에 기반한 난수발생기

전체적으로는 이런 소자들이 여러 개 연결되어 난수발생기를 구성하게 된다. (그림 Ⅱ-4)에서 볼 수 있는 것처럼 Select 신호 값에 따라서 전체 회로는 다음 둘 중 하나로 동작하게 된다.

◦ 두 개의 분리된 단일 인버터 링
◦ 두 개의 안정된 캐스케이드 인버터

(그림 Ⅱ-4)에서는 두 가지 동작 모드를 보여준다.



(그림 Ⅱ-4) 바이스테이블 회로에 기반한 난수발생기의 동작

(그림 Ⅱ-4)의 왼쪽 그림에서는 Select 신호가 1일 때의 회로의 구성을 보여준다. 이때의 회로는 두 개의 분리된 하나의 인버터가 연결된 링 오실레이터 구조를 보여준다. 인버터의 개수가 홀수이므로 이때 인버터의 출력은 일정한 주기를 갖고 0과 1이 반복되는 진동 현상을 보여준다. Select 신호가 2일 때는 (그림 Ⅱ-4)의 오른쪽과 같은 회로로 동작을 하게 된다. 이때에는 두 개의 인버터가 직렬로 연결된 구조를 갖게 된다. 인버터의 개수가 두 개이므로 회로는 하나의 상태로 고정이 된다. (그림 Ⅱ-4)에서 제안된 회로는 두

개의 안정된(stable) 조건 사이를 Select 신호에 따라 왕복하게 된다. 두 가지 동작 모드가 모두 안정적이지만, 스위칭 과정에서 링 오실레이터가 서로 반대 상태가 도달하는 경우가 발생할 수 있다. 이것이 이회로의 잡음원이 된다. 이 회로는 두 개의 안정된 상태 사이를 서로 전환하다가 전환 과정 중에 난수발생기의 잡음원 역할을 하는 불안정 상태가 발생한다.

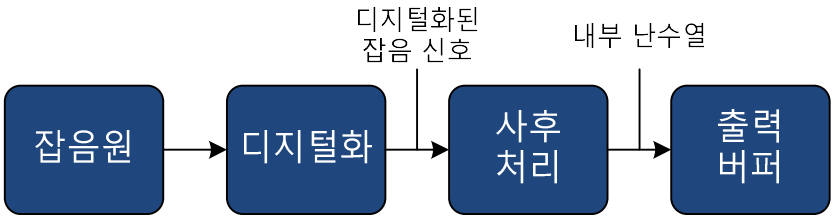
Ⅱ-2 하드웨어 잡음원 응용방법

윈도즈 운영체제에서 수집된 잡음원의 응용 분야는 다음과 같다.

- 결정론적 난수발생기의 씨드
- 비밀키 및 공개키 암호 알고리즘의 키 값 초기화
- 암호 프로토콜에서 난스(nonce) 값
- 랜덤한 패딩 값
- 블록 암호에서 랜덤한 입력값
- 인증을 위한 일회용 랜덤 값

하드웨어 잡음원의 응용 시 주의사항은 다음과 같다. (그림 Ⅱ-5)은 하드웨어 잡음원 생성 및 처리 과정을 나타내고 있다. 잡음원이 아날로그 신호로 생성되면 디지털화하고 디지털화된 잡음 신호는 바이어스가 존재하므로 통계적으로 난수 특성에 가깝도록 사후처리 과정을 거쳐 잡음원으로 사용한다.

- 하드웨어로 구현된 잡음원 생성기로부터 수집된 잡음원은 바이어스가 존재하므로 (그림 Ⅱ-5)과 같이 사후처리 과정을 수행한 후 응용되어야 한다.



(그림 Ⅱ-5) 잡음원 생성 및 처리 과정

- 잡음원은 결정론적 난수발생기의 씨드로 사용하기 위하여 저장될 경우 안전하게 보호하여야 한다.
- 하드웨어 잡음원 생성기에서 잡음원을 수집할 때마다 (그림 Ⅱ-5)의 디지털화된 잡음원에 대한 온라인 검사(Ⅲ-1 절 참조)를 수행하여야 하며 실패시 잡음원을 재수집해야 한다. 온라인 검사는 암호제품에 탑재되기 위해서는 빠르게 동작해야 하며, 적은 하드웨어 혹은 짧은 코드길이를 가져야 한다.

◦ 수집된 (그림 II-5)의 디지털화된 잡음원에 대한 오프라인 검사(III-2 절 참조)를 수행하여 잡음원의 엔트로피를 검사할 수 있다.

부 록 III

(본 부록은 표준을 보충하기 위한 내용으로 표준의 일부는 아님)

엔트로피 검사방법

잡음원 검사방법에는 온라인 검사방법과 오프라인 검사방법이 있다. 온라인 검사방법은 잡음원 수집과 동시에 엔트로피 검사를 수행할 수 있으며 검사기능이 잡음원 생성기와 함께 구현되어 있으며 잡음원 생성기의 자가시험으로 수행된다. 오프라인 검사방법은 수집된 잡음원을 데이터 파일 등을 사용하여 잡음원 수집 프로그램의 외부에서 엔트로피 검사를 수행하는 것이다.

III-1 온라인 검사방법

온라인 검사는 잡음원의 생성기 자가시험의 한 항목으로서 잡음원 수집시 수행하며 수집된 잡음원의 엔트로피가 불만족이면 잡음원을 재수집한다.

온라인 검사를 위한 요구사항은 다음과 같다.

◦ 온라인 검사는 잡음원의 완전한 붕괴를 바로 검출할 수 있어야 한다.
◦ 온라인 검사는 (그림 II-5)의 디지털화된 잡음 신호에 대하여 수행하며 통계적인 취약성을 검출해야 한다.
◦ 디지털화된 잡음원의 통계적인 특성이 이상적인 수열의 특성과 허용 가능한 수준에서 차이가 난다면 잡음 경고가 일어날 확률도 작아야 한다.
◦ 온라인 검사는 빠르게 동작해야 하고 짧은 코드길이의 소프트웨어로 구현되어야 하거나 적은 하드웨어로 구현되어야 한다.

온라인 검사는 다음과 같은 방법들이 있다.

◦ FIPS 140-2 엔트로피 소스 건전성 시험[7]
  - 반복 카운터 검사(Repetition Count Test): 잡음원의 동일한 출력이 반복되는 현상을 조기에 검출하기 위함
  - 적합 비례 검사(Adaptive Proportion Test): 특정한 패턴의 빈도가 너무 많으면 잡음원의 오류로 판정함
◦ 비결정론적 난수발생기의 효율적 온라인 검사[8]
◦ Renyi 엔트로피 기반의 온라인 검사[9]
◦ 완전 붕괴 검사
  - 수집된 잡음원이 항상 엔트로피를 만족하지 못 할 경우 난수발생기 동작을 중단시키고 오류 메시지를 출력함(분포수: 4)

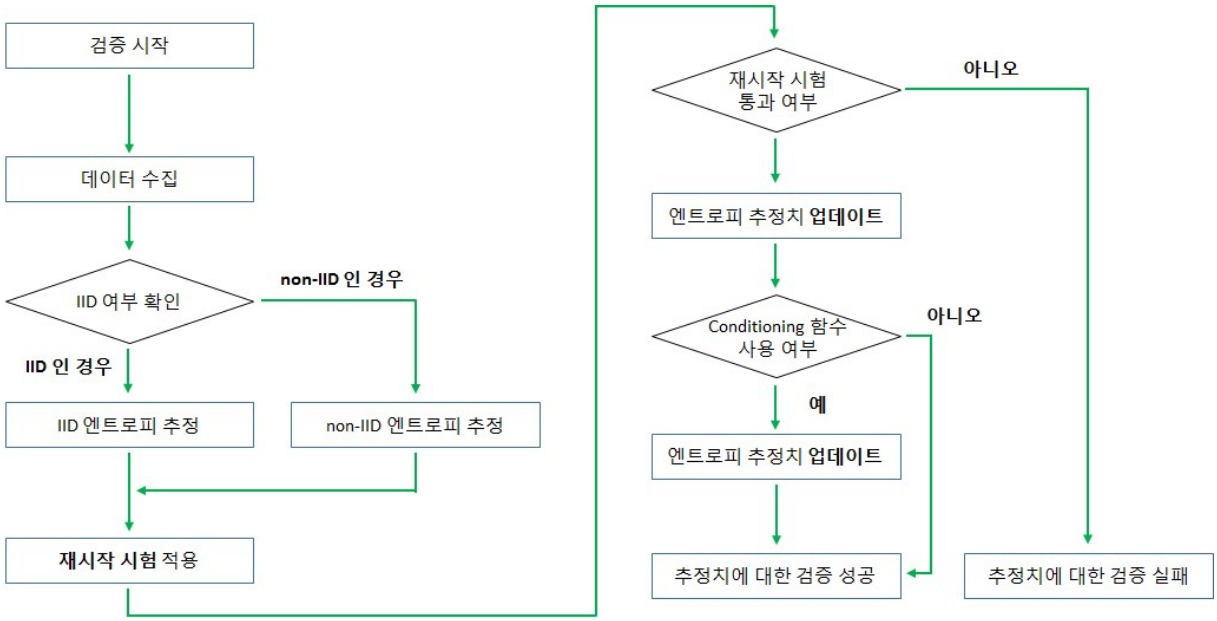
III-2 오프라인 검사방법

본 절에서 NIST SP 800-90B[2]에 따른 잡음원 엔트로피 검사방법을 기술한다. 잡음원에 대한 오프라인 검사는 주어진 데이터에 대하여 최소 엔트로피를 측정하는 것을 중심으로 구성되어 있다. 출력 비트열이 최대 엔트로피(full entropy)를 갖는 경우는 독립적이며 이상적인 분포(IID)를 갖는 것으로 간주될 수 있다. 검사 절차는 평가대상인 잡음원의 출력 비트열이 IID로 주장되는 경우와 그렇지 못한 경우로 구분되어 진행된다. IID가 아닌 경우는 별도의 최소 엔트로피를 측정하는 과정을 추가하며, IID가 주장된 경우에는 이를 검증하기 위한 통계적인 테스트를 수행한다.

<표 III-1> 엔트로피 측정을 위한 검사 종류

<table>
<tr><td rowspan="7">IID
(Independent and Identically Distributed)
분포인지 검정하는
테스트</td><td rowspan="6">Shuffling Test</td><td>Compression Score</td></tr>
<tr><td>Over/Under Runs Scores</td></tr>
<tr><td>Excursion Score</td></tr>
<tr><td>Directional Runs Scores</td></tr>
<tr><td>Covariance Score</td></tr>
<tr><td>Collision Scores</td></tr>
<tr><td rowspan="2">Specific Statistical Test</td><td>카이제곱(Chi Square) Test</td></tr>
<tr><td>기타 테스트(추가선정 예정)</td></tr>
<tr><td>IID의 min-entropy를
측정하는 방법</td><td colspan="2">IID 테스트를 모두 통과하는 경우,
각 테스트 중 최소 엔트로피</td></tr>
<tr><td rowspan="5">non-IID의
min-entropy를
측정하는 방법</td><td colspan="2">Collision Test</td></tr>
<tr><td colspan="2">Partial Collection Test</td></tr>
<tr><td colspan="2">Markov Test</td></tr>
<tr><td colspan="2">Compression Test</td></tr>
<tr><td colspan="2">Frequency Test</td></tr>
<tr><td rowspan="2">Sanity Check</td><td colspan="2">Compression Sanity Check</td></tr>
<tr><td colspan="2">Collision Sanity Check</td></tr>
</table>

엔트로피 검사의 진행에 대한 흐름은 (그림 III-1)을 따른다. (그림 III-1)의 각 단계에 필요한 검사의 종류는 <표 III-1>와 같다. 상세한 검사내용은 NIST SP 800-90B[2]에 따른 잡음원 엔트로피 검사방법을 참고한다.



(그림 III-1) 엔트로피 검사 수행 절차

NIST SP 800-90B에 따른 오프라인 검사방법은 미국 CMVP와 한국 KCMVP에서 검증도구로 구현되어 있으며 잡음원 엔트로피 검사에 활용하고 있다.

NIS SP 800-90B에 따른 검증도구를 사용하기 위한 데이터 수집 및 검사방법은 다음과 같다.

◦ 데이터 수집

1. 데이터 검사를 위한 인터페이스가 준비되어야 하며 사용된 데이터는 다른 용도로 재사용될 수 없다.
2. 수집되는 데이터는 (그림 II-5)의 디지털화된 잡음 신호이어야 한다.
3. 검사에 필요한 큰 데이터 집합은 잡음원으로 부터 직접, 최소한 1백만번 이상 연속적인 샘플링으로 얻어져야 한다.

◦ 검사 방법

1. 연속동작 검사로 반복 카운터 검사와 적합 비례 검사를 수행한다.
2. 수집된 데이터에 대하여 (그림 III-1)의 엔트로피 검사 수행 절차에 따라 검사를 수행한다.

◦ 검사 도구

1. 엔트로피 검사 프로그램: (Second DRAFT) NIST SP 800-90B NIST SP 800-90B Entropy Estimation Suite
2. 다운로드 정보: https://github.com/usnistgov/SP800-90B_EntropyAssessment