https://doi.org/10.7840/kics.2017.42.2.307

센서를 이용한 경량 난수발생기 설계 및 구현

강 하 나◆, 유 태 일*, 염 용 진°, 강 주 성**

A Design and Implementation of the Light-Weight Random Number Generator Using Sensors

Hana Kang◆, Taeil Yoo*, Yongjin Yeom°, Ju-Sung Kang**

요 약

암호시스템에서 난수발생기는 필수적인 요소이다. 최근에 IoT, Sensor Network, SmartHome와 같은 소형 디바이스를 사용하는 환경이 등장하면서, 이에 적합한 다양한 경량 암호들이 개발되고 있다. 하지만 리소스 제한, 엔트로피 수집의 어려움 등의 문제로 인하여, 기존의 데스크 탑에 초점을 두고 만들어진 난수발생기가 제대로 동작하는 것이 어려워지고 있다. 본 논문에서 경량 환경에서 안전한 난수를 생성하는 방법으로 경량 난수발생기 설계를 소개한다. 구조는 헨켈 매트릭스와 블록암호를 사용하고 잡음원으로 센서를 사용한다. 또한 소형 디바이스 중에서 가장 대표적인 Arduino보드에 설계한 경량 난수발생기를 구현하고, 구현 결과로 센서 데이터와 최종 출력 난수의 엔트로피 값을 측정하고 평가함으로써 효율성과 안전성을 확인한다.

Key Words : random number generator(rng), sensor, entropy, cryptography

ABSTRACT

Random number generator(RNG) is essential in cryptographic applications. As recently a system using small devices such as IoT, Sensor Network, SmartHome appears, the lightweight cryptography suitable for this system is being developed. However due to resource limitations and difficulties in collecting the entropy, RNG designed for the desktop computer are hardly applicable to lightweight environment. In this paper, we propose a lightweight RNG to produce cryptographically strong random number using sensors. Our design uses a Hankel matrix, block cipher as the structure and sensors values as noise source. Futhermore, we implement the lightweight RNG in Arduino that is one of the most popular lightweight devices and estimate the entropy values of sensors and random number to demonstrate the effectiveness and the security of our design.

Ⅰ. 서 론

암호시스템에서는 데이터 암호화 및 인증을 제공함으로써 안전한 통신을 보장한다. 암호시스템에서 난수발생기의 출력은 대칭키 알고리즘의 비밀키, 공개키 암호 알고리즘의 파라미터, 프로토콜의 논스, 양자키 분배 등에서 중요한 보안 매개변수로 사용된다. 따라서 암호시스템에서 난수발생기가 이상적인 난수를 출

※ 본 연구는 2016년 정부(미래창조과학부)의 재원으로 한국연구재단-차세대정보·컴퓨팅기술 개발사업의 지원을 받아 수행되었습니다.(NO.NRF-2014M3C4A7030648)
◆ First Author : Kookmin University Department of Financial Information Security, adrt1145@kookmin.ac.kr, 학생회원
° Corresponding Author : Kookmin University Department of Math / Financial Information Security, salt@kookmin.ac.kr, 종신회원
* Kookmin University Department of Financial Information Security, taeilyoo@kookmin.ac.kr, 학생회원
** Kookmin University Department of Math / Financial Information Security, jskang@kookmin.ac.kr, 정회원
논문번호 : KICS2016-08-231, Received August 31, 2016; Revised November 18, 2016; Accepted December 15, 2016

력한다는 가정 하에 암호 알고리즘, 프로토콜 등을 설계한다. 그러므로 난수발생기가 취약할 경우에는 전체 암호시스템의 안전성에 영향을 주게 된다.

1.1 연구 배경

최근에 IoT, Sensor Network, SmartHome 등과 같은 소형 디바이스를 많이 사용하는 경량 환경이 등장하면서, 이로 인해 새로운 보안취약점이 등장할 것으로 예상하고 있다.[1] 현재 많이 사용되는 암호 알고리즘은 데스크 탑에 초점을 두고 개발되었기 때문에, 리소스가 제한적인 경량 환경에서는 구현하기 어렵다. 그렇기 때문에 이러한 환경에 적합한 다양한 경량 암호가 개발되고 있다. 그러나 경량 블록암호, 경량 해시함수 개발에 비해 경량 난수발생기에 대한 연구가 부족한 상태이다.

난수를 생성하는 과정은 크게 엔트로피 수집 단계와 의사난수 생성 단계로 나눌 수 있는데, 특히 출력은 잡음원에 의존적인 출력을 제공하는 특징을 가지고 있다. 그렇기 때문에 난수발생기에서는 잡음원으로부터 엔트로피를 수집하는 단계가 중요한 역할을 한다. 그러나 데스크 탑 환경에서 경량 환경으로 바뀌면서 키보드, 마우스와 같은 사용자 입력이 사라지게 되었다. 이로 인해, 잡음원 수집에 대한 어려움이 존재한다. 난수발생기의 불충분한 엔트로피 수집으로 인하여 발생한 취약점으로는 안드로이드에서 사용되는 비트코인 지갑 탈취[2], 스마트카드의 RSA 인수분해로 인한 비밀키 노출[3], GitHub의 SSH Key 데비안 버그에 관한 취약성[4], 저사양 MCU의 난수발생기의 예측 가능한 난수 출력[5] 등이 존재한다.

또한, 소형 디바이스의 경우 메모리가 2KB 수준으로 제공되기 때문에 표준 난수발생기를 구현하는 것이 어렵다.[6] 즉, 엔트로피 수집의 어려움과 메모리 제약으로 인해, 현재 난수발생기는 이러한 환경에서 제대로 동작하는 것이 어렵다.

1.2 논문의 주요 결과

본 논문은 경량 환경에 적합한 난수발생기를 설계하고 실험을 통해 유효성을 확인하고자 한다. 경량 난수발생기의 설계는 Barak, Halevi가 제안한 PRNG[7] 구조를 기반으로 하였으며, 실험으로는 Arduino에 설계한 경량 난수발생기를 구현하였다. 그리고 Arduino에 센서들을 연결하고, 센서들의 값을 잡음원으로 사용하여 난수를 출력하였다. 최종적으로 센서들의 엔트로피 값과 출력 난수의 엔트로피 값을 측정하고 평가함으로써 유효성을 확인하였다. 본 논문의 주요 결과를 요약하자면 아래와 같다.

- 설계한 PRNG의 출력이 통계적으로 우수하다는 것을 알 수 있다.
- 유휴(idle) 시간에 엔트로피 풀에 충분한 엔트로피를 모으면 Arduino에 최대 7,200 bps 속도로 안전한 난수를 출력할 수 있도록 설계하였다.
- 잡음원 수집이 어려운 환경에서 소형 디바이스를 안전한 난수발생기로 활용될 수 있다.

또한, 실제 사용되는 잡음원의 엔트로피 값을 측정함으로써, 잡음원의 엔트로피 값에 따라 엔트로피 수집 속도를 조절하여 출력 난수의 안전성을 향상시킬 수 있다.

본 논문의 구성은 2장은 난수발생기의 용도 및 중요성과 평가 기술, Arduino의 난수발생기에 대해서 소개하고, 3장은 본 논문의 설계에 기반을 둔 Barak, Halevi가 제안한 PRNG의 구조에 대해서 소개하고, 4장은 실제 설계한 경량 난수발생기의 구조 및 특징을 소개한다. 5장은 실험을 통해 구현한 경량 난수발생기의 구현 방법 및 결과에 대해 소개하고 마지막으로 6장은 결론으로 구성되어 있다.

Ⅱ. 난수발생기

2.1 난수발생기의 용도 및 중요성

현대 암호시스템에서는 안전한 통신을 하고자 데이터 암호화 및 인증을 제공하는 암호시스템을 이용한다. 암호시스템은 난수발생기를 이용하여 난수발생기의 출력 값인 난수를 공개키 암호 알고리즘의 파라미터, 대칭키 알고리즘의 비밀키, 프로토콜의 논스, 양자키 분배의 난수 등의 용도로 활용한다. 그리고 이를 통해 보안 문제를 해결하였다. 따라서 안전한 암호시스템은 난수를 사용하지 않고 정의될 수 없으며, 안전한 난수가 현대 암호의 안전성을 뒷받침하게 되었다. 암호시스템에서는 난수발생기가 이상적인 난수를 출력한다는 가정을 기반으로 암호 알고리즘, 프로토콜 등을 설계한다. 그러므로 아무리 안전한 암호 알고리즘, 프로토콜 등을 사용하였더라도 안전하지 않은 난수를 사용한다면 암호시스템이 공격될 수 있다. 따라서 안전한 난수를 출력하는 난수발생기를 사용해야 한다.

아래는 난수발생기의 용도에 대한 내용이다.

- 공개키 암호 알고리즘의 파라미터[8]
  - RSA의 경우, 공개키와 개인키를 생성하는데 사용되는 소수 p, q를 난수발생기를 이용하여 생성한다.
- 대칭키 암호 알고리즘의 비밀키와 IV (Initialization Vector) [9]
  - 난수발생기의 출력 값을 대칭키 암호 알고리즘의 비밀키로 사용한다.
  - 블록암호의 운영모드 중 초기 값을 필요로 하는 CBC, OFB 등에서는 난수발생기를 통해 초기 값 IV를 생성한다.
- 비밀번호 기반 알고리즘에 사용되는 솔트(Salt)
  - 솔트는 해시함수와 같은 일 방향 함수에서 추가적으로 입력되는 데이터로, 비밀번호를 사전공격으로부터 방어하기 위해 난수발생기를 사용하여 출력 값을 솔트로 사용한다.
- 프로토콜의 논스(Nonce) [10]
  - SSL/TLS 프로토콜에서 재전송 공격을 방어하기 위해 난수발생기로 생성한 논스를 포함하는 메시지를 통해 통신을 진행한다.
- 양자키 분배에 사용되는 난수 [11]
  - 양자키 분배는 양자 암호화 알고리즘의 핵심 부분으로, 처음에 제안한 BB84프로토콜에서 편광자 선택을 위해 난수가 사용된다.

2.2 난수발생기의 구조 및 특징

암호시스템에서 필요로 하는 난수는 동전 던지기를 통해 얻은 수와 같이 예측 불가능하고 독립적이며, 비편향성 등의 조건을 만족하는 수이다. 하지만 이러한 수를 생성하는 것은 현실적으로 불가능하므로, 결정론적(Deterministic) 알고리즘으로 구성된 의사난수발생기(Pseudorandom Number Generator, 이하 PRNG)와 비결정론적(Non-deterministic) 알고리즘으로 구성된 진난수발생기(True Random Number Generator, 이하 TRNG)를 결합하여 난수발생기를 설계하고 난수를 생성한다.

난수발생기의 구조는 [그림 1]과 같이 크게 3단계로 구성된다.

- 엔트로피 수집 : 외부의 물리적 잡음원으로부터 엔트로피를 수집한다.
- 엔트로피 축적 및 시드 생성 : 엔트로피를 축적하여 시드 값을 생성한다.
- PRNG : 결정론적 알고리즘을 사용하여 난수를 생성한다.

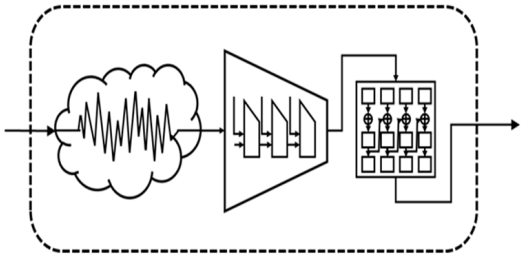

그림 1. 난수발생기의 구조
Fig. 1. Structure of RNG

난수발생기가 사용되는 환경에 따라 이용 가능한 엔트로피 소스가 다르기 때문에, TRNG와 PRNG를 결합한 구조를 사용한다.

난수를 출력하는 과정은 크게 엔트로피 수집 단계와 의사난수 생성 단계로 나눌 수 있는데, 잡음원에 의존적인 출력을 제공하는 특징을 지니고 있다. 즉, 난수발생기는 잡음원으로부터 엔트로피를 수집하는 단계가 중요한 역할을 한다. 그렇기 때문에 엔트로피 소스에 대한 정확한 측정과 통계적 특성의 평가가 중요하며, 난수발생기의 전체 안전성에 영향을 끼치기 때문에 다양한 환경에 적합한 엔트로피 소스의 발굴이 필수적이며 중요하다.

2.3 난수발생기의 평가기술

난수발생기의 평가에 대한 표준으로는 대표적으로 미국 국립표준기술연구소(National Institute of Standard and Technology, 이하 NIST)가 존재한다. 아래는 NIST에서 난수발생기에 대한 검증 기준 및 구현 적합성에 대해 제안한 문서들이다.

- FIPS 140 : Security Requirements for Cryptographic Modules [12]
  - CMVP(Cryptographic Module Validation Program)에 사용되는 표준으로써 하드웨어와 소프트웨어 암호모듈에 관한 안전성과 적합성을 다룬다.
- SP 800-90A : Recommendation for Random Number Generator Using Deterministic Random Bit Generators [13]
  - 결정론적 난수발생기(DRBG)의 매커니즘에 대한 권고사항을 기술한 문서이다.
  - 해시함수, HMAC, 블록암호, 타원곡선을 기반으로 한 다양한 난수발생기 매커니즘을 제시한 문

서이다.

- SP 800-90B : Recommendation for the Entropy Source Used for Random Bit Generation [14]
  - 엔트로피 소스 분포에 대한 특성을 결정하고, 이러한 특성에 따른 엔트로피 소스에 대한 테스트를 제시한 문서이다.
- SP 800-90C : Recommendation for Random Bit Generator(RBG) Construction [15]
  - SP 800-90A를 통한 DRBG의 알고리즘과 SP 800-90B의 엔트로피 검증 결과를 핵심으로 하여 난수발생기의 구조적 설계 방법을 제시한 문서이다.
  - 잡음원에서 변화된 디지털 입력의 엔트로피 타당성을 검증하고, 높은 엔트로피로 측정되었다면 입력으로 사용된다.
  - 여기서 추출된 입력은 DRBG를 통과하여 출력 난수로 생성된다.
- SP 800-22 : A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Application [16]
  - PRNG에 대한 통계적 검정을 제시한 문서이다.
  - 총 15가지의 검정방법이 있으며, p-value를 통한 귀무가설을 검증하는 단계를 거쳐서 통과유무를 확인한다.
  - 엔트로피 측정에는 활용되지 못하지만, 난수발생기 설계 시 널리 활용되는 통계적 평가이다.

2.4 Arduino의 난수발생기

본 논문에서는 소형 디바이스 중에 IoT 환경에 가장 많이 사용되는 Arduino를 기준으로 비교하고 활용하였다. Arduino에서 제공하는 난수발생기는 PRNG로 random, randomSeed 함수가 존재한다. 이때, PRNG는 아래의 수식 (1)와 같으며, 선형 합동 생성기(Linear Congruential Generator, 이하 LCG)이다.

$$X_{n+1} \equiv 7^5 \cdot X_n (\mathrm{mod} 2^{31}-1) \quad (1)$$

LCG는 특성상 최대 주기를 갖도록 인자를 선택해도 아주 좋은 난수를 생성하지 못한다는 단점이 존재한다.

일반적으로 random, randomSeed 함수를 사용할 때, 시드 값으로 analogRead 함수를 통해 얻은 아날로그 핀의 데이터를 사용한다. 하지만 아날로그 핀 데이터의 크기는 10비트이며, 사용되는 난수발생기가 LCG이기 때문에 쉽게 시드 값이 예측 가능하다.[17] 또한, 이전에 생성된 값을 알면 그 뒤에 만들어진 값을 모두 예측할 수 있다. 이와 같은 문제점으로 인하여, Arduino에는 암호학적인 목적으로 사용할 수 있는 난수발생기가 존재하지 않는다.[18]

Ⅲ. Barak, Halevi의 PRNG 구조[7]

Barak, Halevi가 제안한 구조는 아래와 같은 두 가지의 요소를 가지고 난수발생기를 구성한다.

- $extract(x)$: randomness extraction function
  - 입력 값으로 높은 엔트로피를 가지는 x를 넣어, 출력 값으로 입력 값보다 짧은 길이의 난수를 출력한다.
- $G(\bullet)$: PRG(Pseudorandom Generator)
  - 시드 값으로부터 긴 난수를 출력한다.
  - $extract(x)$의 출력 길이는 시드 값의 길이와 동일하다.

이때, 필요한 extraction function은 아래와 같은 정의를 따른다.

정의 1. Extraction Function

정수 m과 $\{0,1\}^{\geq m}$인 분포족(Family of distribution) H가 존재한다. 이때 모든 $D \in H$와 $y \in \{0,1\}^m$에 대해서

$$extract : \{0,1\}^{\geq m} \rightarrow \{0,1\}^m \quad (2)$$

가 아래의 수식을 만족하면, 이 $extract$을

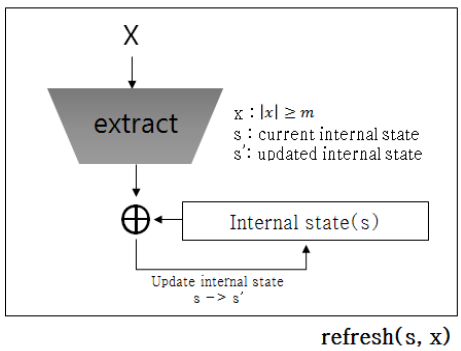

그림 2. refresh(s,x)의 구조
Fig. 2. refresh(s,x)

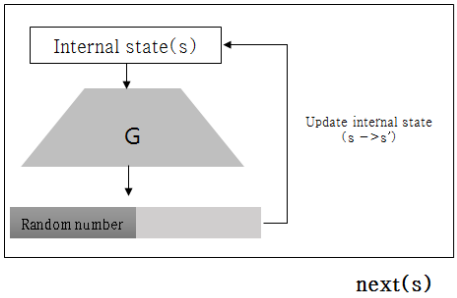

그림 3. next(s)의 구조
Fig. 3. next(s)

$H-extractor$라 한다.

$$2^{-m}(1-2^{-m}) \le \Pr_{x \xleftarrow{R} D}[extract(x)=y] \le 2^{-m}(1+2^{-m}) \quad (3)$$

위에서 설명한 $extract(x), G(\bullet)$으로부터 구성한 난수발생기는 아래 그림과 같이 $refresh(s,x), next(s)$ 두 가지 함수로 구성된다.

Ⅳ. 경량 난수발생기의 구조 및 특징

본 논문에서는 설계한 경량 난수발생기 모델은 Barak, Halevi가 제안한 PRNG 구조를 사용하였다.[7] 구조는 [그림 4]와 같이 잡음원으로부터 엔트로피 추출, 선형 되먹임 시프트 레지스터(Linear Feedback Shift Register, 이하 LFSR)를 통한 엔트로피 풀 갱신, 난수 출력 및 갱신의 3가지 단계로 구성된다. 설계한 난수발생기는 헨켈 매트릭스와 엔트로피 풀을 주기적으로 갱신하여 최종 출력 난수를 예측하기 어렵게 하는 구조를 지닌다. 또한, 본 논문에서 설계한 경량 난수발생기의 안전성은 Barak, Halevi가 제안한 PRNG 구조의 안전성[7]에 근거한다.

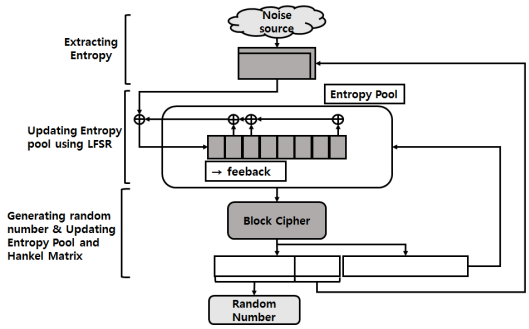

그림 4. 경량 난수발생기의 구조
Fig. 4. Structure of lightweight RNG

4.1 엔트로피 추출 단계

첫 번째 단계는 잡음원과 헨켈 매트릭스의 곱 연산을 통해 데이터를 압축하는 단계이다. 추출된 데이터는 엔트로피 풀에 제공된다. 헨켈 매트릭스의 구성은 아래 [그림 5]와 같이 기억자의 데이터를 가지고 행렬을 구성하며 구성 방식은 수식 (4)과 같다.

$$a_{ij} = a_{(i+1)(j-1)} \quad (a_{ij} = 0 \text{ or } 1) \quad (4)$$

헨켈 매트릭스는 입력되는 잡음원의 엔트로피에 따라 행렬의 크기를 조절할 수 있고, 크기는 m×n인 헨켈 매트릭스 구성하기 위해서 m+n+1 만큼의 데이터만 필요하다는 장점이 존재한다. 그러므로 제안하는 구조에서 무거운 해시함수를 구현하는 것보다 헨켈 매트릭스를 사용함으로써 소형 디바이스에서 메모리 문제를 해결할 수 있는 장점을 가지고 있다. 또한, 제안하는 구조의 마지막 부분에서 최종 출력 난수의 일부를 사용하여 헨켈 매트릭스를 주기적으로 갱신하기 때문에, 시드 공격에 대한 엔트로피 풀의 안전성을 향상시키는 장점을 가진다.

본 논문에서 설계한 헨켈 매트릭스의 크기는 8x16이며 각각 0 또는 1의 값을 가진다. 헨켈 매트릭스의 곱 연산 결과로는 8비트가 출력된다.

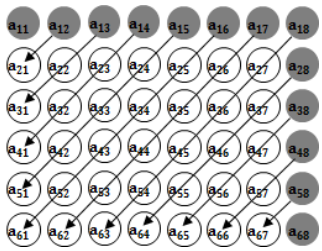

그림 5. 헨켈 매트릭스
Fig. 5. Hankel Matrix

4.2 LFSR을 통한 엔트로피 풀 갱신 단계

두 번째 단계는 첫 번째 단계에서 추출된 데이터를 가지고 선형 피드백 시프트 연산을 통한 엔트로피 풀을 갱신하는 단계이다. LFSR은 최대 순환 주기를 가질 때 최대 안전성을 가질 수 있다. 따라서 본 논문에서 제안한 구조에서 엔트로피 풀의 크기는 64비트이며, LFSR는 최대 순환 주기를 갖도록 아래의 수식 (5)과 같이 설계하였다.

$$x^6+x^5+1 \quad (5)$$

4.3 최종 난수 출력 및 갱신 단계

세 번째 단계는 최종 난수 출력을 위해 블록암호의 카운터 모드로 헨켈 매트릭스의 값과 엔트로피 풀의 값을 갱신하는 단계이다. 이때 블록암호는 최근에 경량 블록암호로 제안된 Speck[19]를 사용하였다. WSN430 sensor에 경량 블록암호 구현 결과에 의하면 현재 Speck은 블록암호 중에 가장 작은 메모리 사용량을 보여주고 있다.[20] 본 논문에서 사용된 Speck은 평문 64비트와 키 96비트를 입력으로 하여 암호문 64비트를 출력하는 알고리즘이다. Speck을 카운터 모드로 사용하여 최종적으로 128비트를 출력한다. 이때, 카운터 값은 초기화하지 않은 메모리의 값을 사용하여 암호화 할 때마다 1씩 증가하는 방식을 사용하고 있다. Speck의 키는 고정된 값을 사용한다. 최종출력 128비트는 [그림 6]과 같이 상위 40비트를 최종 출력 난수로 사용하고, 그 다음 24비트는 헨켈 매트릭스를 갱신하는데 사용되며 나머지 64비트는 엔트로피 풀을 갱신하는데 사용된다. 블록암호의 일부를 헨켈 매트릭스와 엔트로피 풀 갱신에 사용함으로써 안전성을 향상시키는 장점을 지닌다.

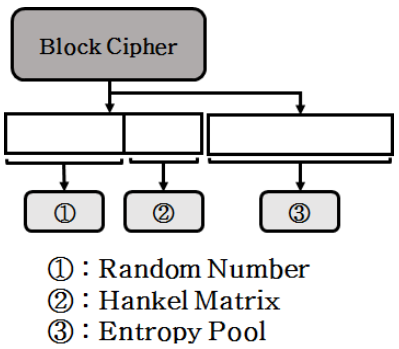

그림 6. 블록암호의 출력의 사용
Fig. 6. Usage of block cipher output

V. 경량 난수발생기의 구현 방법 및 결과

본 장에서는 위에서 설계한 경량 난수발생기의 효율성을 보이기 위하여 구현 방법에 대해 소개한다. 구현 대상은 소형 디바이스 중에 가장 대표적인 Arduino Uno를 선택하였다. 잡음원으로는 부착된 센서들의 데이터를 가지고 설계한 경량 난수발생기를 구현하였다. 설계한 경량 난수발생기의 구현은 크게 초기화 단계와 최종 난수 출력 단계로 나눈다. 첫 번째 초기화 단계에서는 센서들의 데이터를 가지고 헨켈 매트릭스와 엔트로피 풀을 초기화한다. 두 번째 최종 난수 출력 단계에서는 부착된 센서들의 데이터를 가지고 위에서 언급한 1, 2, 3 단계를 수행한다.

5.1 잡음원 수집

본 논문에서는 난수발생기의 입력인 잡음원을 부착된 센서들의 데이터로 사용하였다. 사용한 센서들의 종류는 아래 [표 1]과 같다. 실제 수집되는 센서들의 데이터는 10비트이지만, 잡음원의 엔트로피 측정에 적합하도록 SP 800-90B 기준에 맞추어 변화가 큰 상위 8비트만 추출하여 잡음원으로 사용하였다.[14]

5가지 센서 이외에 아무것도 연결되지 않은 아날로그 핀으로부터 들어오는 데이터를 추가적으로 사용하였다.

표 1. 센서들에 대한 설명
Table 1. Description of the sensors.

Type of sensors	Description
Accelerometer	Sensor for measuring gravity acceleration
Temperature	Sensor for measuring the ambient temperature
Ambient Light	Sensor for measuring the ambient light
Distance	Infrared sensor for distance measurement
Grayscale	Sensor for detecting the ambient brightness

5.2 초기화 단계

먼저 헨켈 매트릭스와 엔트로피 풀을 초기화하기 위해서 센서 중 가속도 센서 데이터를 가지고 헨켈 매트릭스를 구성한다. 나머지 센서들의 데이터와 구성된 헨켈 매트릭스의 곱 연산을 통해 나온 결과를 가지고 엔트로피 풀을 초기화해준다. 그 다음으로 블록암호 Speck을 카운터 모드로 사용하여 엔트로피 풀을 갱신한다.

안전한 난수를 출력하기 위해서는 충분한 엔트로피가 수집되어야 한다. 따라서 본 논문의 구현에서는 충분한 엔트로피를 수집하기 위해 보드에 전원이 공급되는 순간부터 헨켈 매트릭스와 센서 데이터의 곱 연산, LFSR을 통한 엔트로피 풀 갱신, Speck의 카운터 모드를 통한 헨켈 매트릭스와 엔트로피 풀 갱신 총 3단계를 75,000번 수행하였다. 이때, 초기화 단계를 수행하는 횟수는 사용하는 환경에 따라 다양하게 변경 가능하다.

5.3 최종 난수 출력 단계

초기화 단계로부터 엔트로피 풀에 충분한 엔트로피가 수집된다. 엔트로피 수집 단계가 완료되면 위에서 언급한 1, 2, 3단계를 거쳐 7,200 bps 속도로 최종 40비트 난수를 출력한다. 최종 출력 128비트에서 40비트를 제외한 나머지 데이터를 헨켈 매트릭스와 엔트로피 풀을 갱신하는데 사용한다.

센서들의 데이터를 수집하는 속도는 10초에 1번 8가지 종류의 데이터를 수집하도록 설정하였다.[7]

이 또한, 사용하는 환경에 따라 다양하게 변경이 가능하다.

5.4 구현 결과

[표 2]는 대표적인 소형 디바이스 Arduino의 스펙을 보여주고 있다. [표 3]은 제안 된 난수발생기의 구현 결과를 나타내고 있다.

SP 800-90B[14]는 잡음원이 Non-IID(Non-Independent and Identically Distributed)일 때, Collision Test, Compression Test, Partial Collection Test, Markov Test, Frequency Test 5가지 검정방법을 적용하여 가장 작은 값을 잡음원의 Min-Entropy 값으로 측정한다. 각 센서에서 수집된 데이터를 SP 800-90B 엔트로피 측정에 맞도록 8비트씩 1,000,000개의 샘플을 수집하여 엔트로피를 측정하였다. 이때, 데이터 수집에 문제가 없도록 나머지 환경은 통제 하에 진행되었다. 아래의 [표 4]는 실제 구현을 통해 측정된 결과와 Hennebert, Hossayni의 논문에서 측정한 결과[21]을 비교한 표이다. 표를 통해 비교해본 결과,

표 2. 아두이노 우노의 사양
Table 2. Specification of Arduino Uno.

Types of memory	Specification
Flash Memory	32KB
SRAM	2KB
EEPROM	1KB

표 3. 구현 결과
Table 3. Result of implementation.

Types	Memory
프로그램 저장 공간	4,750 bytes
전역 변수	270 bytes
SRAM	1,727 bytes
출력 속도	7,200 bps

표 4. 센서들의 엔트로피 값
Table 4. Entropy of the sensors.

Type of sensors	Entropy	Hennebert, Hossayni's entropy[21]
Temperature	0.021 ~ 0.759	0 ~ 0.05
Grayscale	0.033 ~ 0.822	-
Ambient Light	0.175 ~ 1.010	-
Distance	0.139 ~ 0.996	-
Analog Pin	0.045 ~ 1.003	-
Accelerometer X-axis	0.021 ~ 0.759	0.22
Accelerometer Y-axis	0.020 ~ 0.996	0.42
Accelerometer Z-axis	0.021 ~ 0.996	0.36

측정 결과가 유사하다는 것을 알 수 있다. [표 4]는 본 논문의 실험 결과는 Non-IID 5가지 검정 방법의 결과 최솟값과 최댓값을 나타낸 값이다. 이때, 엔트로피 측정은 SP 800-90B를 기반으로 구현된 평가도구 이용하여 측정하였다.[22]

최종 출력된 난수의 엔트로피를 측정하기 위해 40비트씩 1,000,000개의 샘플을 수집하였다. [표 5]는 출력 난수의 비트 당 엔트로피와 샘플 당 엔트로피 측정한 결과이다. 실제 실험에서 출력한 샘플의 크기는 40비트이지만, 출력 데이터의 엔트로피를 측정하기 위해 SP 800-90B의 근거하여 변화가 가장 큰 값 8비트를 샘플로 하여 엔트로피를 측정하였다.[14]

또한, Dieharder[23]을 통해 출력 난수의 통계적 검정을 실행한 결과 모두 통과하였음을 확인할 수 있었다. 이와 같은 통계적 검정을 통해서, 본 논문에서 설계한 경량 난수발생기의 PRNG의 출력이 통계적으로 우수하다는 것을 알 수 있다.

표 5. 출력 난수의 엔트로피 값
Table 5. Entropy of random number output.

Tests	1 bit	8 bits
Collision Test	0.880	7.032
Compression Test	0.880	6.928
Partial Collection Test	0.932	6.990
Markov Test	0.995	6.990
Frequency Test	0.984	7.566

Ⅵ. 결 론

본 논문에서는 소형 디바이스를 사용하는 경량 환경에 적합한 난수발생기를 설계하고 구현 결과를 제시하였다. 제안하는 경량 난수발생기는 이론적으로도 충분한 엔트로피를 제공하는 구조이며, PRNG의 출력을 평가한 결과 통계적으로 우수하다는 것을 알 수 있었다. 실험 결과 고속 난수발생기로 사용할 수 없으나 유휴(idle)시간에 엔트로피 풀에 엔트로피를 계속 모으면 난수가 필요할 때 Arduino에서 최대 7,200 bps 속도까지 출력할 수 있다는 것을 확인하였다. 이 경량 난수발생기는 메모라 사용량이 극히 적은 특징이 있기 때문에, 잡음원 수집이 어려운 디바이스 환경에서 활용될 수 있을 것으로 기대된다. 특히, 구현된 소형 디바이스 자체를 하드웨어 난수발생기로도 활용될 수 있음을 보였다.

References

[1] D. H. Kim, S. U. Yun, and Y. P. Yi, "The security of IoT service," in *Proc. KICS Int. Conf. Commun.*, pp. 53-59, Jeju Island, Korea, Jul. 2013.

[2] K. Michaelis, C. Meyer, and J. Schwenk, *Randomness fail! The State of randomness in current java implementations*, Springer Berlin Hedelberg, pp. 129-144, Feb. 2013.

[3] D. J. Bernstein, Y. A. Chang, C. M. Cheng, L. P. Chou, N. Heninger, T. Lange, and N. V. Someren, "Factoring RSA keys form certified smart cards : Coppersmith in the wild," in *Int. Conf. Theory and Appl. Cryptololgy and Inf. Secur.*, Springer Berlin Heidelberg, pp. 341-360, Dec. 2013.

[4] *Audit of github SSH Keys finds many still vu lnerable to old debian bug* (2015), Retrieved November, 18, 2016, from https://threatpost.com/audit-of-gitbub-ssh-keys-finds-many-still-vulnerable-to-old-debian-bug/113117/

[5] *True random number generator for a true hacker* (2015), Retrieved November, 18, 2016, from http://hackday.com/2015/06/29/true-random-generator-for-a-true-hacker

[6] J. Y. Park, S. M. Shin, and N. H. Kang, "Mutual authentication and key agreement scheme between lightweight devices in internet of things," in *Proc. KICS Int. Conf. Commun.*, pp. 707-714, Jeju Island, Korea, Jul. 2013.

[7] B. Barak and S. Halevi, "A model and architecture for pseudo-random generator and application to d/dev/random," CCS'05, Nov. 2015.

[8] Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone, *Handbook of applied cryptography*, Massachusetts Inst. Technol., 1996.

[9] NIST SP 800-38A, *Recommendation for block cipher modes of operation methods and techniques*, 2011.

[10] S. Thomas, *SSL and TLS essentials*, New York, John Wiley & Sons, Inc., 2000.

[11] C. H. Bennett, "Quantum cryptography: Public key distribution and coin tossing," in *Int. Conf. Comput. Syst. and Sign. Process. IEEE*, pp. 175-179, 1984.

[12] FIPS PUB 140-3, *Security Requirements for Cryptographic Modules*, 2009.

[13] NIST, SP 800-90A, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, 2015.

[14] NIST, SP 800-90B, (Second Draft) *Recommendation for the Entropy Sources Used for Random Bit Generation*, 2016.

[15] NIST, SP 800-90C, *Recommendation for Random Bit Generator(RGB) Construction*, 2012.

[16] NIST, SP 800-22, *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Application*, 2010.

[17] B. Kristinsson, "Ardrand : The arduino as a hardware random-number generator," *Cryptography and Secur., Inf. Theory*, Dec. 2012.

[18] Wikipedia, *Linear congruential generator*, https://en.wikipedia.org/wiki/Linear_congruential_generator

[19] National Security Agency, *The simon and speck families of lightweight block ciphers*,

Jun. 2013.

[20] *Implementations of lightweight block ciphers on a WSN430 sensor*, http://bloc,project.citi-lab.fr/library_option_02.html

[21] C. Hennebert, H. Hossayni, and C. Lauraoux, “Entropy harvesting from physical sensors,” *Wisec’13*, pp. 149-154, Budapest, Hungary, Apr. 2013.

[22] H. Kang, Y. Yeom, and J. S. Kang, “An implementation of integrated tool for statistical randomness tests and entropy estimations,” in *Proc. KICS Winter Conf.*, pp. 229-230, Jeongseon, Korea, Jan. 2013.

[23] Diharder, Retrieved November, 18, 2016, from http://www.phy.duke.edu/~rgb/General/deiharder.php

강 하 나 (Hana Kang)

2016년 2월 : 국민대학교 수학과 학사

2016년 3월~현재 : 국민대학교 금융정보보안학과 석사

<관심분야> 암호이론, 난수성 분석, 암호시스템 분석

유 태 일 (Taeil Yoo)

2010년 2월 : 국민대학교 수학과 학사

2014년 8월 : 국민대학교 수학과 석사

2016년 12월~현재 : 국민대학교 금융정보보안학과 박사과정

<관심분야> 암호시스템 분석, 안전성 분석, 병렬 구현

염 용 진 (Yongjin Yeom)

1992년 2월 : 서울대학교 수학과 학사

1994년 2월 : 서울대학교 수학과 석사

1999년 2월 : 서울대학교 수학과 박사

2000년 4월~2012년 2월 : ETRI 부설 연구소 책임연구원/팀장

2006년 12월~2007년 12월 : Columbia 대학교 방문연구원

2012년~현재 : 국민대학교 수학과 부교수

2013년~현재 : 국민대학교 BK21+ 미래 금융정보보안 인력양성사업단 교수

<관심분야> 암호구현 및 분석, 보안시스템 평가

강 주 성 (Ju-Sung Kang)

1989년 2월 : 고려대학교 수학과 학사

1991년 2월 : 고려대학교 수학과 석사

1996년 2월 : 고려대학교 수학과 박사

1977년~2004년 : 한국전자통신연구원 선임연구장/팀장

2001년~2002년, 2010년 : 벨기에 루벤대학 COSIC 방문 연구원

2004년~현재 : 국민대학교 수학과 교수

2013년~현재 : 국민대학교 BK21+ 미래 금융정보보안 인력양성사업단 교수

<관심분야> 암호이론, 정보보안 프로토콜, 안전성 분석 및 평가