QEC: A Quantum Entropy Chip and Its Applications

Jungmin Park, *Member, IEEE*, Seongjoon Cho, Taejin Lim, and Mark Tehranipoor, *Fellow, IEEE*

***Abstract*— Quantum phenomena cannot be predicted by the uncertainty principle. As a quantum phenomenon, radioactive decay has been used as an entropy source to generate random numbers. In this article, we present the design and development of an innovative quantum entropy chip (QEC) that produces analog random pulses when emitted alpha particles resulted from radioactive isotope (americium-241) decay hit the sensor. The analog pulse generated by a QEC can be digitized into random numbers by an entropy extractor. The QEC provides security foundation for device authentication as well as a quantum random number generator (QRNG), especially suited for the Internet of Things (IoT) devices due to its small size. We have successfully designed and fabricated the QEC as a wafer for supporting a system-on-chip (SoC) Internet Protocol (IP) so that the QEC can be embedded into a microcontroller unit (MCU) or central processing unit (CPU). In addition, we built a stochastic model to estimate the entropy of the quantum source and evaluated statistical randomness and robustness against temperature, voltage variations, aging effects, and physical attacks. Finally, we demonstrate various applications using the QEC such as side-channel-resistant primitives and device authentication.**

***Index Terms*— Countermeasure, Internet of Things (IoT), radioactive decay, random number generator, side-channel attack (SCA).**

I. Introduction

IN MODERN cryptographic systems, the use of a random number generator is imperative to generate encryption keys, initial vectors, and nonces, as well as develop countermeasures against side-channel attacks (SCAs) such as random masking methods. Random numbers also play an essential role in complex scientific and financial simulations/transactions, modern lotteries, and gambling machines. Even in a quantum key distribution system based on BB84 protocol [1], the security depends highly on randomness that generates raw data and determines the encoding axis or phase.

The random numbers should be statistically unbiased, independent, and most importantly unpredictable [2]. It must be uniformly distributed in the same way as the ideal model of a fair coin toss that generates 1 (for the head) and 0 (for the tail) with 50% probability, respectively. In addition, it must be robust against aging effects and environmental variations such as temperature and electromagnetic (EM) emanations as well as malicious physical attacks, such as fault-injection attacks, using EM [3] or photons.

Two kinds of random number generators can be considered in modern systems, namely a pseudorandom number generator (PRNG) and a true random number generator (TRNG). When entering a seed value of hundreds of bits into the PRNG, such as linear feedback shift registers (LFSRs) or cryptographically secure PRNG (CSPRNG), we can generate a largely deterministic sequence of pseudorandom numbers. The pseudorandom numbers are uniformly distributed and pass statistical tests of uniformity just like true random numbers [4]. The PRNG is usually fast and can be implemented easily on field-programmable gate arrays (FPGAs) or in application-specific integrated circuits (ASICs). However, if both seed and deterministic algorithm to generate random numbers are known, all pseudorandom numbers are predictable or reproducible.

On the other hand, TRNGs produce an unpredictable sequence of random numbers based on the randomness from nonphysical or physical entropy sources. Operating systems can collect nonphysical entropy from disk access times, interrupt request lines (IRQs), or user interaction data, such as the keyboard stroke or mouse motion, called entropy pools. When a sufficient level of entropy is available in the pool, the operating system produces random numbers using a PRNG. Alternatively, physical entropy can be extracted from timing jitter [5], [6], thermal noise in electric circuits [7], biometric data[1] [8], or electrocardiogram (ECG) derived from the electrical activity of the heart [9].

In general, since these digitized sources of true randomness have a bias or insufficient entropy, reducing the bias or increasing entropy should be involved, which is called postprocessing. For forward and backward unpredictability as well as an increase of entropy, the National Institute of Standards and Technology (NIST)-approved cryptographic algorithm is used for deterministic random bit generators (DRBGs) such as HASH-DRBG, HMAC-DRBG, and CTR-DRBG [10]. In this case, the seed is acquired from true randomness sources so that the DRBG can be instantiated and unpredictable random numbers can be generated.

However, most of the existing TRNGs need to mitigate vulnerabilities under adversarial environments. First, sources of true randomness, e.g., thermal noise, timing jitter, or metastability, can be affected by temperature, voltage, and aging, and external attacks such as EM fault injection attacks [3]. Second, DRBGs included in the TRNG are susceptible to SCAs such as differential power analysis (DPA) [11] or correlation power analysis (CPA) [12], i.e., random numbers will be predictable

Manuscript received July 11, 2019; revised October 25, 2019 and January 13, 2020; accepted February 3, 2020. Date of publication March 12, 2020; date of current version June 1, 2020. This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) grant funded by the Korea Government (MSIT) (No. 12, technical development of security validation for firmware on IoT devices). *(Corresponding author: Jungmin Park.)*

Jungmin Park and Mark Tehranipoor are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: jungminpark@ufl.edu).

Seongjoon Cho and Taejin Lim are with the Research and Development Department, EYL Inc., Seoul 16954, South Korea.

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2020.2975091

[1] In [8], the randomness of two types has been checked, which are neuronal membrane voltages in the primary visual cortex of a cat during series of visual stimulation and the electrical conductance of the galvanic skin responses of humans.

if the secret key in DRBDs can be revealed by SCAs. Hence, we should consider the additional requirements for secure RNGs as well as sufficient entropy and unpredictability.

1) Random numbers generated by the RNG have sufficient entropy and unpredictability.
2) Random source is robust against environmental variations.
3) The RNG has robustness against SCAs.

Quantum phenomena can support true randomness and uncertainty which are an essential part of quantum mechanics: optics and radioactive isotope decay. Most well-known optical random entropy is constructed by sending a single photon to a balanced beam splitter [13]. Because the photon has characteristics of both wave and particle, an incoming photon is either transmitted or reflected equally likely, which is not influenced by the environmental variables. Another source of quantum randomness is based on the radioactive decay. In [14], alpha particles emitted during the radioactive isotope (americium-241[2]) decay are exploited. The time interval between consecutive radioactive decay or the number of radiated particles within a constant time is identically and independently distributed so that the entropy can be extracted or digitized.

In order to satisfy the third condition (i.e., a DRBG in a TRNG has robustness against SCAs), two kinds of countermeasures against SCAs can be exploited—hiding and masking. The masking method with random bits, such as threshold implementation [15], is widely used under the assumption that random bits for masking have full entropy. If masking random bits generated by another TRNG are biased or predictable by physical attacks such as fault injection attacks or SCAs, the masking countermeasure is nullified.

In this article, we propose an innovative approach to extract quantum randomness from a radioactive isotope of americium (Am), which is developed as a quantum entropy chip (QEC). The QEC provides random pulses to generate true random numbers as well as random frequency clocks so that it can offer a framework for a side-channel-resistant design using quantum random number generator (SCR-QRNG framework)[3] [16].

The QEC has the following features.

1) Offers small area overhead with a die size of 1.4 mm $\times$ 1.4 mm at 180-nm technology node.
2) Generates unbiased and nondeterministic random numbers without postprocessing.
3) Can be used as a stand-alone and embedded Internet Protocol (IP) in a system-on-chip (SoC) platform.
4) Provides robustness against malicious manipulation.

This paper is an extension of our previous work [16]. The list of differences from the previous work is as follows.

1) Our QRNG is evaluated in terms of three criteria, discussed in Section V, which are as follows.
   a) Characteristics related to the QRNG principle such as the source of randomness, and statistical and stochastic characteristics.
   b) The QRNG design.
   c) Robustness against temperature/voltage variations and aging effect as well as hardware security, such as optical and EM attacks.
2) Two kinds of quantum entropy sources and QRNG architecture are described comprehensively in Section II, and our QRNG is compared with existing commercial QRNGs in Section V-E.
3) A device authentication as another application for the QEC is presented in Section VII.

To the best of our knowledge, there are no pieces of literature to analyze and evaluate QRNGs in terms of comprehensive criteria at the time of this writing. In this extended version, we deal with a comprehensive and detailed study of QRNGs, our QEC, and applications such as a side-channel-resistant QRNG framework and device authentication.

This article is organized as follows. In Section II, we present preliminaries. Section III presents the hardware design of the QEC and QRNGs. Section IV provides the stochastic model of the quantum entropy caused by the radioactive decay. Experimental results are presented in Section V. The SCR-QRNG framework is presented in Section VI. Device authentication using the QEC is presented in Section VII. Finally, we conclude this article in Section VIII. The proof is given in the Appendix.

II. Preliminaries

A. Quantum Entropy Source

QRNGs are devices that produce random numbers from the quantum phenomenon, which cannot be predicted by any means according to the principles of quantum physics. QRNGs offer much better alternatives since their randomness comes from the unpredictable physical phenomenon that cannot be influenced by external variables. There are two quantum phenomena that randomness can be extracted from: optics and radioactive isotope decay. Many different optical methods have been proposed, tested, and commercialized in recent years, but they have common limitations.

1) Optical Quantum Entropy: The most well-known optical random number generator is constructed by sending a single photon to a balanced beam splitter [13]. Because the photons have characteristics of both wave and particle, incoming light is either transmitted or reflected equally likely. Whether a given photon is transmitted or reflected is a random phenomenon that cannot be influenced by the environmental variables. For example, if we can make a beam splitter that transmits precisely 50% of the light, and a hypersensitive sensor that can detect a single photon as 0 for a transmitted photon and 1 for a reflected photon, then one can produce ideal random numbers. However, optical quantum entropy has limitations as follows.

1) Because it is impossible to build a balanced beam splitter in such a way that precisely 50% of the light would be transmitted, it is inevitable to create a bias.
2) It requires a hypersensitive sensor that can detect a single photon as well as a device that can shoot a single photon to the half mirror at a precise angle.
3) The artificial act of postprocessing is needed that can correct the bias such as the XOR algorithm.
4) Due to the high cost of hypersensitive sensors, optical QRNGs too will be expensive.
5) All types of detectors need time delay to recover after a detection, known as the dead time.

2) Quantum Entropy Based on Radioactive Decay: Another source of randomness using quantum characteristic is based on the radioactive decay [14], which exploits alpha particles emitted during the decay of a radioactive isotope (americium-241). Each alpha particle consists of two protons and two neutrons, a nucleus of $^4_2$He. Alpha decay usually

[2]Americium-241 is commonly used for a household smoke detector in which the amount of elemental americium-241 is small enough to be exempt from the regulations (37 kBq or 1 μCi, corresponding to about 0.3 μg).

[3]In this article, SCR-QRNG framework means a framework for a side-channel-resistant design using QRNG. SCR-QRNG means side-channel-resistant QRNG.

occurs from the heaviest nucleus that has too many nucleons to be stable in a natural state. It can be expressed as: $^{241}_{95}\mathrm{Am} \rightarrow ^{237}_{93}\mathrm{Np} + ^{4}_{2}\mathrm{He}$, where the numbers in subscript and superscript represent the atomic number (also the number of protons in the nucleus) and a mass number of the nucleus (number of protons plus neutrons), respectively.

The particle radiation including beta particles[4] and gamma ray photons,[5] called ionizing radiation has enough energy to eject an electron from an atom. These particles ionize the matter and destroy molecular bonds, which can cause significant biological damage. The alpha particles are the least dangerous since they cannot penetrate very deeply into the skin and can be stopped by clothing as well [17].

A semiconductor sensor such as a p-i-n diode[6] is widely used for radiation monitoring or quantitative radiation measurement. When light or other forms of ionizing radiation are absorbed in the intrinsic area or in the depletion layer of a p-i-n diode, many electron–hole pairs are created. If the diode is reverse-biased, which results in the electric field between the p- and n-type regions, almost all of the accumulated charges are drawn away, and thus a small current pulse is generated. The small current can be amplified and processed in order to detect the very tiny current pulse.

The time interval between two consecutive decay impulses can be modeled as an exponential distribution and provide the best entropy to generate random numbers [14] (the mathematical modeling will be discussed in Section IV), which is absolutely unpredictable. However, to the best of our knowledge, it has been difficult to miniaturize the device until now.

B. QRNG Architecture

The quantum entropy source, such as emitted photons or radioactive decay, generates randomness in the form of analog signals such as low-level electronic currents. Thus, the generated current should be amplified and digitized to distinguish the quantum signal from electrical noise. The digitized quantum signal is generated in various forms such as the time-to-digital conversion between detected decay events or a counting number of detected quantum particles. The combination of a quantum entropy source and a digitizer (or an entropy extractor) is called a digital entropy source. The stochastic model can be built based on the quantum entropy source and the extraction process.

If the digitized raw sequences, called raw random numbers have a negligible bias,[7] the postprocessing process is not required. However, if the raw random sequences have nonnegligible bias or the entropy of the raw random sequences is not sufficient, caused by an imperfection in the measurement or entropy sources with low entropy, the postprocessing reduces bias and compresses the raw random sequences to increase the entropy. The output of the postprocessing is as close as possible to a uniform distribution at the expense of a throughput [18]. The output of the postprocessing module is referred to as internal random numbers. The bias of raw random numbers and internal random numbers can be evaluated by statistical/online test suites such as NIST SP 800-22 [4]. There are various methods of postprocessing [19]; Von Neumann corrector, XORing, linear codes, and cryptographic algorithms such as AES and hashing.

If each bit of the biased raw random sequence is independently and identically distributed, a Von Neumann corrector can remove the bias at the cost of throwing away at least half of the bits, in which for every pair of raw random bits, 00 and 11 patterns are discarded, and 0 is assigned if a 01 pattern is given to the inputs, and 1 is assigned if a 10 pattern is given. The output bit of the Von Neumann corrector is theoretically uniformly distributed [19]. However, the throughput of the Von Neumann correct is at most $1/4$ of the rate of the inputs.[8]

If negligible bias is allowed, a linear code is a good candidate for the postprocessing. Let G be a linear corrector mapping n bits to k bits; $G\colon \mathbb{F}_2^n \rightarrow \mathbb{F}_2^k$. Then the bias of the output bits generated by a $[n, k, d]$ linear code is less than or equal to $2^{d-1}e^d$, where d is the minimum distance of the linear code constructed by the generator matrix G, and e is the bias of input bits [20]. For example, Dichtl's linear corrector [21] is given by $L\colon \mathbb{F}_2^8 \times \mathbb{F}_2^8 \rightarrow \mathbb{F}_2^8$

$$L(X, Y) = X \oplus (X \lll 1) \oplus (X \lll 2) \oplus (X \lll 4) \oplus Y.$$

If each input bit has $e = 0.05$ bias, the bias of each compressed output bit is reduced to $2^4 \times 0.05^5 = 0.000005$. In the case of XORing two inputs with the bias e; $L(X, Y) = X \oplus Y$, the output bias of each is equal to $2 \times e^2$. These linear codes have the same compression rate $1/2$, but Dichtl's linear code is more improved in terms of the bias.

Since the cryptographic postprocessing, such as hashing and AES, exploits both diffusion and confusion properties, it can also mask the imperfection of raw random numbers. Even if the source of randomness fails, it can provide a deterministic random number generator (DRNG) as well as unpredictability in forward,[9] backward,[10] or both directions. However, this method is more expensive than linear code-based methods in terms of time and area.

III. QEC Design and QRNG

Fig. 1 shows the block diagram of our QRNG which consists of a QEC, an entropy extractor and a postprocessing module. Since raw random numbers from the entropy extractor have enough entropy, postprocessing is not required.

A. QEC Design

The alpha particle used in the development of our QEC has an energy level of 4 MeV. Its radioactivity level is 0.11 μCi which is equal to 4.07 kBq. The applied americium-241 substrate is expected to emit 4070 decays/s. Due to the spherical diffusing characteristic, half of them vanish into the ground plate where the substrate is fixed. Because of their relatively large mass, higher electric charge ($+2e$) and

[4] β-Decay is a type of radioactive decay in which a β-ray (fast energetic electron or positron) and a neutrino are emitted from an atomic nucleus. Beta particles travel several feet when emitted from a radioactive source, but they are blocked by most solid objects [17].

[5] A γ-ray or γ-radiation is a penetrating EM radiation arising from the radioactive decay of atomic nuclei. It consists of photons in the highest observed range of photon energy. γ-Rays are the most dangerous form of ionizing radiation. These extremely high-energy photons can travel through most forms of matter because they have no mass. It takes several inches of lead or several feet of concrete to effectively block gamma rays [17].

[6] A p-i-n diode is a diode with a wide, undoped intrinsic semiconductor region between a p-type semiconductor and an n-type semiconductor region.

[7] Bias is defined as a deviation of probability that a random bit, r is equal to 1 from the ideal value, 0.5; $e = \Pr[r = 1] - 0.5$.

[8] Assuming that $\Pr[x_i = 1] = 0.5 + e$, $\Pr[x_i x_{i-1} = 01$ or $x_i x_{i-1} = 10] = 2(1/4 - e^2)$. Since the output is a bit, the rate of the output is given by $1/4 - e^2$.

[9] Subsequent (future) values cannot be determined (i.e., computed or guessed with nonnegligible probability) from current or previous output values.

[10] Previous values cannot be determined (i.e., computed or guessed with nonnegligible probability) from the current or future output values.

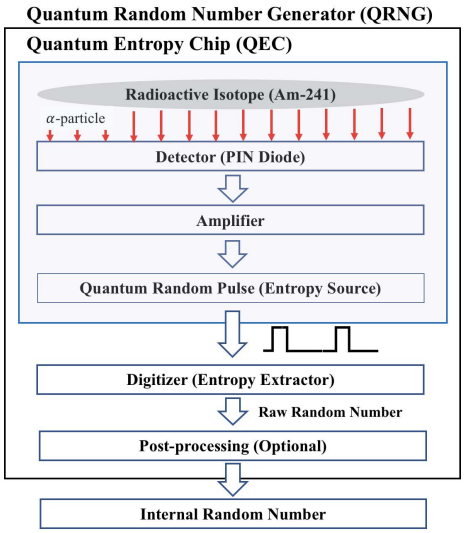


Fig. 1. Block diagram of the QRNG.

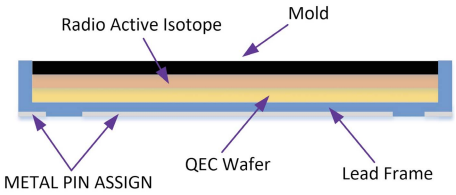


Fig. 2. Side view of the QEC.

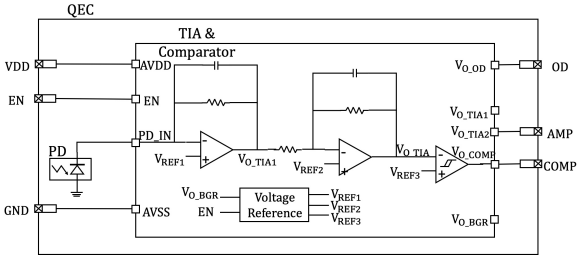


Fig. 3. Circuit design of the QEC.

relatively low velocity, the alpha particles are very likely to interact with other atoms and lose their energy. For example, alpha particles with 4-MeV energy can be stopped at a few centimeters of air. In order to prevent self-absorption, it needs to be as thin as possible. In practice, most of the alpha radioactive source is fixed on the substrate and covered by a very thin metal film, which prevents the radioactive source from being affected by external factors such as smoke (see Fig. 2). Fig. 3 shows the circuit design of the QEC which consists of a photodiode (PD) for detecting α-particles, two transimpedance amplifiers (TIAs), and a comparator.

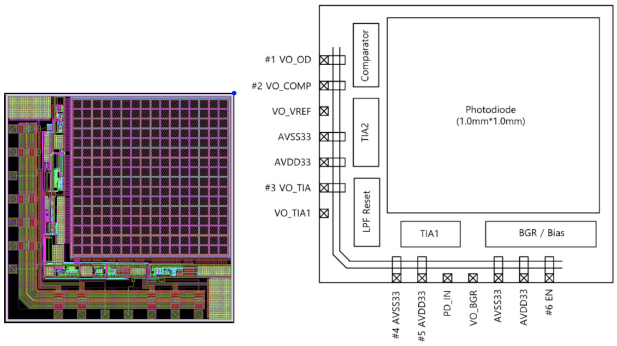
1) Detector: Alpha particles are detected by the following CMOS-type PD, and its structural characteristics are as follows.

1) *Type:* Surface-type p-n junction diode.
2) *Reverse Bias:* Typically 1.65 V.
3) *Particle Incidence:* Front side only.
4) *Charge Collection Range:* 3–8 μm (with surface insulating layer of 3 μm).
5) *Alpha Particle Energy Range:* 0.7–5.5 MeV (if SiO_2 has the same property to Si crystal).

The reverse-bias leakage current of the PD is less than 1 nA and when α-particles are absorbed in the depletion region of the PD, a photocurrent that has typically 1 μA is generated. Details of the technical specification are provided in Table I.

TABLE I
SPECIFICATION FOR THE DETECTOR

Item	Condition	Value
Unit PD area	Active area	50 μm × 50 μm
Total PD area	16 × 16 array	1 mm^2
Capacitance	-	106 pF
Effective resistance	-	57 Ω
Dark current	Reverse bias : 1.65V, 70°F	less then 1 nA
Light current	Reverse bias : 1.65V, 70°F	1 μA
Photo responsivity	550 nm light source	0.27A/W
Quantum efficiency	550 nm light source	60 %
Response speed	-	6.6 GHz



(a) (b)

Fig. 4. ASIC design of the QEC. (a) Layout of QEC: 1.4 mm × 1.4 mm. (b) Component blocks and pinouts.

TABLE II
PIN ASSIGNMENT AND DESCRIPTION

No.	Pin Assignment	Type	Description
1	OD	O	Open drain output (comparator)
2	COMP	O	Comparator output
3	AMP	O	Amplifier output
4	GND	Power	Ground
5	VDD	Power	Supply power (3.3V)
6	EN	I	Sensor enable
7 ∼ 12	NC	-	No connection
PAD	NC	-	No connection (center pad)

2) Amplifier: We use a TIA to amplify and detect a very low level of the light current by the absorption of an α particle. It is amplified in two stages. The transimpedance gain and the unit gain bandwidth of the first TIA are 10 MΩ and 10 MHz, respectively. The gain of the second amplifier is set to 10.

3) Comparator: The amplified voltage from TIAs is transformed to an analog pulse signal, V_{O_COMP}, called a quantum random pulse by a comparator with a reference voltage of 0.3 V such that

$$V_{O\_COMP} = \begin{cases} 3.3\ V, & \text{if } V_{O\_TIA2} \le V_{REF3} \\ 0\ V, & \text{if } V_{O\_TIA2} > V_{REF3}. \end{cases}$$

Therefore, the QEC finally produces a quantum random pulse when an α-particle caused by radioactive decay is detected.

Fig. 4(a) shows the layout design of the QEC based on 180 nm technology node and Fig. 4(b) shows components and pinouts in the QEC. Finally, the fabricated wafer is enclosed in a 12-pin quad-flat no-leads (QFN) package. The pinouts of the QEC is described in Table II. The QEC chip has a tiny size; 3 mm × 3 mm × 0.85 mm. The power consumption of the QEC is 3 mW.

B. QRNG

In order to generate raw random numbers as seen in Fig. 1, analog quantum random pulses should be digitized by

an entropy extractor. We can use a n-bit counter with a f-Hz clock frequency as an entropy extractor to measure each interval between consecutively detected decay. The raw n-bit random numbers from the counter originally has the exponential distribution with the parameter λ (not uniformly distributed). The entropy of the raw random number is close to 1/bit if three parameters, λ, n, and f, are satisfied with (2) based on a stochastic model (discussed in Section IV), which does not require any postprocessing.

IV. Stochastic Model

The event of the radioactive α-decay can be mathematically modeled by the Poisson distribution [14]. Let X be a random variable representing the number of α-decay per second. The probability mass function f_X is given by $f_X(x) = (e^{-\lambda_0}\lambda_0^x)/(x!), \quad x = 0, 1, 2, \ldots$, where λ_0 is the expected number of α-decay. The detecting component is located sufficiently close to the Am-241 disk. The rectangular detector composed of 16×16 grids captures 35% of emitted particles. The dead time of the PD detector is extremely low, which is 0.15 ns, so that we may ignore the possibility that more than one particle are detected simultaneously.

We introduce a random variable Y which represents the number of detected particles in a second. Then the following relationship is expected for some constant $k(<1)$: $Y = kX$. Y is also modeled by the Poisson distribution with the probability mass function f_Y as follows:

$$f_Y(y) = \frac{e^{-\lambda}\lambda^y}{y!} = \frac{e^{-\lambda_0 k}(\lambda_0 k)^y}{y!}, \quad \text{for } \; y = 0, 1, 2, \ldots$$

where $\lambda = \lambda_0 k$. An arbitrary time point to detect each α particle is denoted as d_i, for $i = 0, 1, \ldots$, and let T be a random variable representing the interarrival time between two consecutive detection; $t_i = d_i - d_{i-1}$ for $i = 1, 2, 3, \ldots$ By the Poisson process, the continuous random variable T follows an exponential distribution with the probability density function f_T given by $f_T(t) = \lambda e^{-\lambda t}$, for $t \geq 0$. The expected value of the interarrival time is $\mu = 1/\lambda$. Whenever the detector is activated, it requires the dead time T_d to recover during which it is impossible to be activated. The probability to detect the next decay after $t + T_d$ is not related to the dead time T_d because of the memoryless property or Markov property of the exponential distribution as follows [14]:

$$\Pr[D > t + T_d | D > T_d] = \Pr[D > t]$$

where D is a random variable representing the detection event. This also means that the next detection is not affected by the previous detection and dead time.

We can measure each interval between consecutively detected decay using a n-bit digital counter with an f-Hz clock signal. If the digitized value, $\mathbf{V} = [V_{n-1}V_{n-2}\ldots V_0]_2$, has the exponential distribution, the postprocessing is required for increasing entropy close to 1/bit.

Lemma 1: The ratio of $\Pr[V_i = 0]$ to $\Pr[V_i = 1]$ is as follows:

$$\frac{\Pr[V_i = 0]}{\Pr[V_i = 1]} = \frac{1}{e^{-\lambda 2^i/f}}, \quad \text{for } \; i = 0, 1, \ldots, n-1 \qquad (1)$$

where V_i is the ith bit of the counter, λ is the parameter of the exponential distribution, and f is the clock frequency of the counter. Both λ and f are design parameters. The proof of this lemma is given in the Appendix A.

Based on (1), as λ/f converges toward 0, the probability distribution of each bit, v_i approaches a uniform distribution; $\lim_{\lambda/f \to 0} (\Pr[V_i = 0]/\Pr[V_i = 1]) = 1$. Also, the probability of the least significant bit (LSB) is more similar to the uniform distribution.

Corollary 1: If two design parameters λ and f are satisfied with the condition that

$$\frac{2^{n-1}\lambda}{f} < \epsilon \qquad (2)$$

then the bias of digitized entropy sources is negligible as follows:

$$e = \Pr[V_i = 1] - 0.5 \approx 0, \quad \text{for } \; i = 0, 1, \ldots, n-1.$$

Corollary 2 (Entropy): The entropy per bit of the generated random numbers using a n-bit counter with an f-Hz clock signal is defined as follows:

$$H_1[\mathbf{V}] = \frac{1}{n}\sum_{i=0}^{n-1}\Big[\left(1 + e^{\lambda 2^i/f}\right)^{-1}\log_2\left(1 + e^{\lambda 2^i/f}\right) + \left(1 + e^{-\lambda 2^i/f}\right)^{-1}\log_2\left(1 + e^{-\lambda 2^i/f}\right)\Big]. \qquad (3)$$

The minimum entropy, $H_\infty[\mathbf{V}]$, is defined as

$$H_\infty[\mathbf{V}] = \log_2\left(1 + e^{-\lambda 2^{n-1}/f}\right). \qquad (4)$$

V. Experimental Results and Comparison

A. Evaluation Strategy

Ideal random numbers have the property that they are independent, uniformly distributed, and unpredictable on a finite range. However, practical random number generators may have potential weaknesses such as unsatisfied entropy, predictability caused by deterministic algorithm, loss of entropy by aging of the component, high power/resource usage or vulnerability against physical attacks, e.g., fault-injection attacks. In order to detect such defects, QRNGs should be evaluated in terms of three criteria; 1) characteristics related to the QRNG principle; 2) the QRNG design; and 3) robustness and hardware security [2].

1) QRNG Principle: This criterion includes the source of randomness, and statistical and stochastic characteristics.

a) Source of randomness: As discussed in Section II-A, since quantum entropy sources have an inherently unpredictable physical phenomenon, they are suitable for the root of randomness to generate random numbers. In order to validate whether the quantum entropy source (which corresponds to our QEC chip, in this article) is correctly working, total failure (tot) test[11] should be enabled to immediately perceive the failure of the source of randomness. For example, a QRNG including cryptographic postprocessing works as a DRNG if the source of randomness fails. The QRNG still can pass the statistical test even if its seed does not have true randomness. Therefore, the tot test of QRNGs with cryptographic postprocessing is required.

[11]Our tot test is performed with the output of an entropy source shown in Fig. 5, which is different from AIS methodology [22] based on the output of a digitizer.

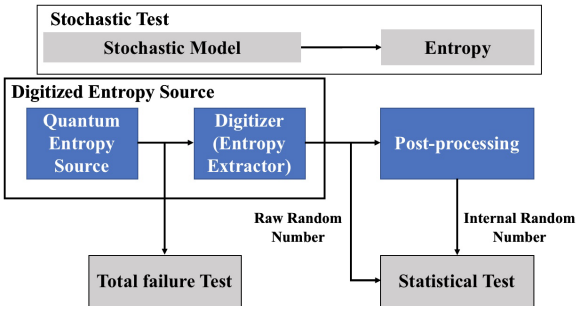


Fig. 5. Evaluation of the QRNG principle.

b) Statistical characteristic: Once raw random numbers or internal random numbers are generated, we need to validate the quality of the random numbers. The traditional approach to randomness testing is to perform a series of statistical tests such as NIST SP 800-22 [4], NIST SP 800-90b [23], BSI AIS 31 [22], TestU01 [24], Diehard [25], and Dieharder [26] suites. The χ^2 test is mostly used to compare the goodness of fit of the observed frequencies of a sample measure to the corresponding expected frequencies of the hypothesized distribution. The p-value of each test above the significance level indicates that the test is passed.

c) Stochastic characteristic: Only these statistical tests are not sufficient to evaluate truly randomness about unpredictability. PRNGs can pass the statistical tests but they are predictable. Similarly, false positives for statistical tests can occur. Stochastic testing focuses on estimating the entropy based on a stochastic model of the QRNG instead of statistical behavior, which corresponds to the P2-TRNG class in AIS 31 [22]. Fig. 5 shows the evaluation of the QRNG principle by three kinds of tests.

2) QRNG Design: In order to evaluate the practical usefulness of QRNGs, it is necessary to analyze the kind and number of resources such as the number of logic gates (in ASICs) or LUTs (on FPGAs), power consumption, technology possibility, and design automation possibility. In particular, QRNGs have many challenging problems in terms of technology and design automation possibilities. For example, it is very difficult or impossible to embed a QRNG IP to an SoC solution due to technology limitations. Most of the commercial QRNGs cannot be suitable for the Internet of Things (IoT) devices due to their large volume (seen in Table VII).

3) Robustness and Security: In addition to unpredictability, entropy, and design requirements, QRNGs must be robust against aging effects and environmental variations such as temperature, power supply or EM radiation. In order to analyze the robustness, randomness in various environment conditions should be validated by statistical tests.

Physical attacks on QRNGs should be taken into account since they are able to change the behavior of QRNGs, i.e., the entropy of generated random numbers is reduced significantly so that the whole cryptographic system can be broken easily. For example, by EM fault injection to the entropy extractor in the ring-oscillator-based TRNG, it is possible to control the bias of the TRNG [3]. If photons are able to be injected to optical sensors in an optical QRNG periodically, the output of the QRNG can be controlled by an adversary.

B. Evaluation: QRNG Principle

In order to analyze the source of randomness, statistical and stochastic characteristics, a tot test, NIST statistical tests, and stochastic tests are performed.

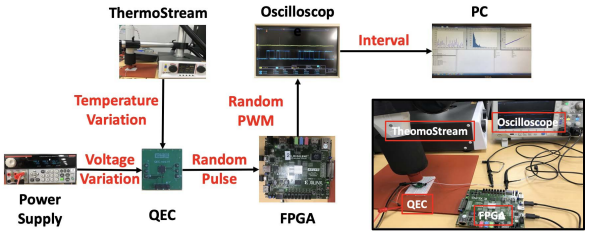


Fig. 6. Setup for failure and robustness tests.

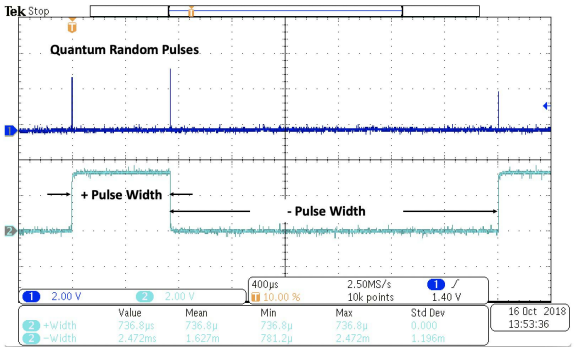


Fig. 7. Quantum random pulses and randomized PWM.

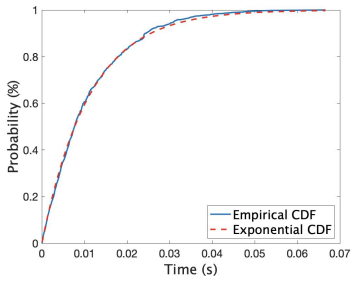


Fig. 8. Empirical CDF and exponential CDF based on measured quantum PWM under the normal condition (25 °C, 3.3 V).

1) Tot Test: In order to check if the QEC is generating enough entropy source, we measure each interval between consecutive random pulses that will be transformed into random numbers by an entropy extractor. Fig. 6 shows the setup of the tot test which consists of a QEC as the DUT, a power supply, a temperature test system, an FPGA development board, a Tektronix MDO3102 oscilloscope, and a PC. The random pulse signal generated by the QEC is inserted to the clock signal of a counter on an FPGA. At the rising edge of the random pulse, the counter increments. The pulsewidth of the LSB signal changes according to the interval between two consecutive pulses. Measured pulsewidths using an oscilloscope correspond to the time intervals, which are stored in a PC for analysis of the randomness. Fig. 7 shows measured quantum random pulses from the QEC and randomized pulsewidth modulation (PWM) from the counter. A quantum random pulse is generated at an average frequency of 100 Hz.

Ten thousand pulsewidths are sampled under the normal condition (temperature: 25 °C and supply voltage: 3.3 V), and we evaluate how well they fit to an exponential distribution using the Kolmogorov–Smirnov (K–S) test [27] as a goodness-of-fit test. Fig. 8 shows that the empirical cumulative probability density (CDF) based on the measured samples is close to the exponential CDF. The p-value of the K–S test is 0.43, which fails to reject the null hypothesis. Under temperature and voltage variations, tot tests will be performed as discussed in Section V-D.

TABLE III

RESULTS OF NIST SP 800-22 STATISTICAL TESTS

Test	p-value	Pass/Fail
Frequency (monobit)	0.7743	Pass
Frequency within a block	0.4660	Pass
Runs	0.7516	Pass
Longest-run-of-ones in a block	0.4901	Pass
Binary matrix rank	0.4911	Pass
Discrete Fourier transform	0.4075	Pass
Non-overlapping template matching	0.0262	Pass
Overlapping template matching	0.9997	Pass
Maurer's *Universal Statistical*	0.3140	Pass
Linear complexity	0.3142	Pass
Serial	$> 0.1950^{\dagger}$	Pass
Approximate entropy	0.7951	Pass
Cumulative sums	$> 0.7190^{\dagger}$	Pass
Random excursions	$> 0.2725^{\dagger}$	Pass
Random excursions variant	$> 0.0334^{\dagger}$	Pass

†It is the minimum p-value of all p-values which are greater than 0.01.

TABLE IV

AIS 31 STATISTICAL TESTS OF QRNG

Test	value (y)	Condition	Pass/Fail
Disjointness	-	-	Pass
Monobit	9950.0	$9654 < y < 10346$	Pass
Poker	37.75	$1.03 < y < 57.4$	Pass
Run	-	-	Pass
Longrun	-	-	Pass
Autocorrelation	2559	$2326 < y < 2674$	Pass
Uniform distribution	0.00091, 0.00122	$y < 0.025$	Pass
Comparative multinomial	0.31752, 0.26912	$y < 15.13$	Pass
Entropy	7.99075	$y > 7.976$	Pass

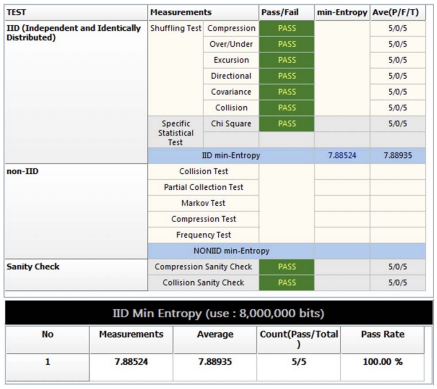


Fig. 9. NIST SP 800-90B test.

2) Statistical Test: A 16-bit counter with a 50-MHz clock as an entropy extractor and other modules, such as an asynchronous first-input–first-output (FIFO) and UART RX/TX for communication with a PC, are implemented on a Xilinx Spartan-6 FPGA. Raw random numbers generated by the counter are transmitted to a PC and then NIST SP 800-22, NIST SP 800-90B, and BSI AIS 31 statistical tests are performed multiple times with five sets of a million of collected bits using our test tool. Tables III and IV show that all NIST SP 800-22 [4] and BSI AIS 31 statistical tests [22] are passed, and measured entropy per byte is 7.99, which is very close to ideal entropy (=8). In addition, our QRNG is in compliance with NIST SP 800-90B [23] as shown in Fig. 9, in which measured entropy per byte is 7.889.

3) Stochastic Test: Based on the stochastic model described in Section IV, we can estimate the entropy of our QRNG. Using (3) and (4), both $H_1[\mathbf{V}]$ and $H_\infty[\mathbf{V}]$ are estimated to be all 1.

C. Evaluation: QRNG Design

The throughput of our QRNG using a 16-bit counter is 1.6 kb/s. The digital design for a counter, an FIFO, and UART RX and TX is implemented on a Xilinx Spartan 6 FPGA, which occupies 652 LUTs. The maximum frequency is allowed up to 179.96 MHz based on the timing analysis.

TABLE V

ROBUSTNESS TEST UNDER VARIOUS TEMPERATURES AND VOLTAGES

Temp./Volt.	$\mu(s)$	KS p-value	χ^2 p-value	Pass ($p > 0.01$)
0°C, 3.3V	0.0108	0.3418	0.2854	Pass
25°C, 3.3V	0.0111	0.4287	0.4372	Pass
50°C, 3.3V	0.0120	0.458	0.4068	Pass
75°C, 3.3V	0.0121	0.3783	0.2884	Pass
25°C, 2.97V	0.0144	0.3566	0.4625	Pass
25°C, 3.63V	0.0140	0.1024	0.0186	Pass

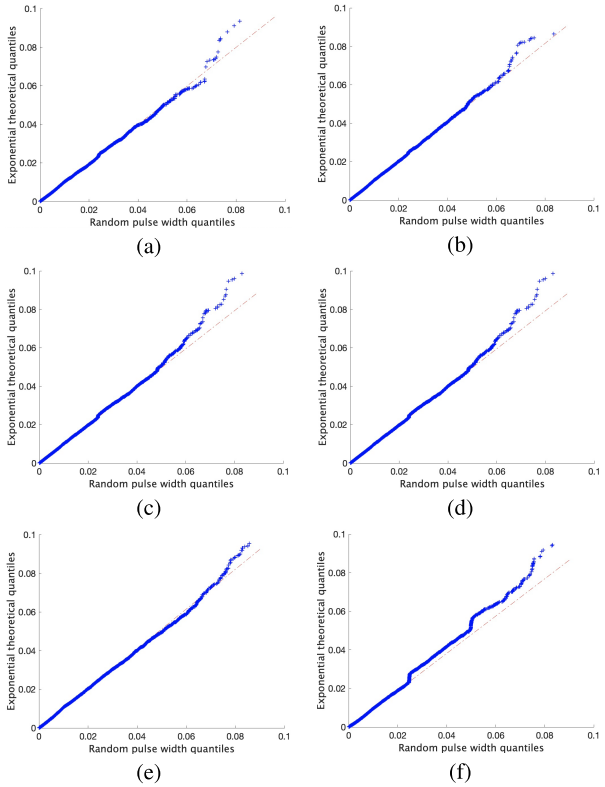


Fig. 10. Q–Q plots under various temperatures and voltages. (a) Temp.: 0 °C. (b) Temp.: 25 °C. (c) Temp.: 50 °C. (d) Temp.: 75 °C. (e) Temp.: 25 °C, 2.97 V. (f) Temp.: 25 °C, 3.63 V.

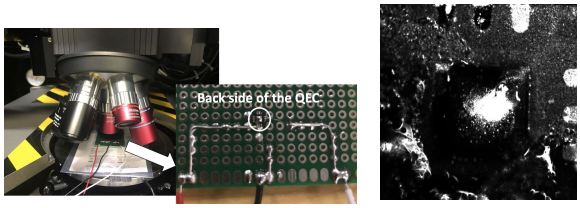
D. Evaluation: Robustness and Hardware Security

1) Temperature and Voltage Variations: In order to evaluate robustness against temperature and voltage variations, additional tot tests under different conditions are performed. In the case of temperature variations, the power supply voltage is fixed at 3.3 V and using a ThermoSteam in Fig. 6, various temperatures from 0 °C to 75 °C are given. For voltage variation tests, two different voltages such as 3.3 V ±10%, i.e., 2.97 and 3.63 V are given under the room temperature (25 °C). Ten thousand samples are collected from the QEC under different environments, and K–S and χ^2 tests are performed. Two p-values and Q–Q plots[12] under each setup in Table V and Fig. 10 show that the quantum random variable is exponentially distributed, and the randomness has robustness against given temperature (0 °C–75 °C) and voltage variations (2.97–3.63 V).

[12]In statistics, a Q–Q (quantile–quantile) plot is a probability plot, which is a graphical method for comparing two probability distributions by plotting their quantiles against each other [28]. If the two distributions being compared are similar, the points in the Q–Q plot will approximately lie on the line $y = x$.

TABLE VI
AGING EFFECT TEST

Eff. Aging	$\mu(s)$	KS p-value	χ^2 p-value	Pass ($p > 0.01$)
0 days	0.0111	0.4287	0.4372	Pass
11.3 days	0.0092	0.1311	0.2077	Pass
22.6 days	0.0089	0.1213	0.1338	Pass
33.9 days	0.0096	0.2093	0.3640	Pass
45.2 days	0.0091	0.0946	0.2130	Pass
56.5 days	0.0095	0.1827	0.0841	Pass



(a) (b)

Fig. 11. QEC optical attack. (a) Setup of PLS. (b) Laser scanning image of the back side of the QEC.

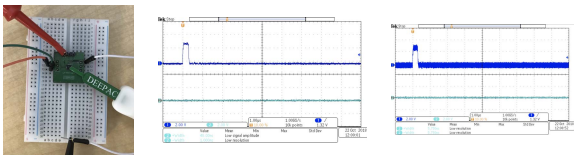
2) Aging Effect: The randomness of the QEC is capable of getting deteriorated by negative bias temperature instability (NBTI) aging. In order to evaluate the aging effect, the NBTI aging is considerably accelerated by increasing temperature and supply voltage with respect to its nominal conditions. In our experiments, we apply the stress with 80 °C and 3.8 V (3.3 V + 15%) to the QEC. The acceleration factor (AF) is given as follows [29]:

$$AF = \left(\frac{V_{\text{stress}}}{V_{\text{normal}}}\right)^{\frac{\alpha}{n}} \exp\left(\frac{E_{aa}}{K}\left(\frac{1}{T_{\text{stress}}} - \frac{1}{T_{\text{normal}}}\right)\frac{1}{n}\right)$$

where the gate voltage exponent, α, is 3.5, the time exponent, n, is 0.25, the apparent activation energy, E_{aa}, is -0.02 eV, Boltzmann's constant, k, is 8.62×10^{-5} eV/K, and the voltage and temperature under the normal condition are, respectively, 25 °C (=298.15 K) and 3.3 V. In the stress condition (80 °C = 353.15 K, 3.8 V), AF is 11.3, i.e., 1 h of the accelerated aging corresponds to 11.3 h of effective NBTI aging under normal conditions. Under constant accelerated aging, we perform tot tests after 24, 48, 72, 96, and 120 h which correspond to 11.3, 22.6, 33.9, 45.2, and 56.5 days under the normal condition. Both K–S and χ^2 statistical tests fail to reject the null hypothesis as seen in Table VI.

3) Hardware Security: If an adversary can control the QEC such as generating periodic pulses to remove entropy by fault-injection attacks based on photon or EM disturbance, countermeasures against these attacks should be considered. We do not try to execute sophisticated and powerful attacks but check if potential risks against fault-injection attacks exist. In our preliminary experiments, both optical attacks using a laser and EM attacks are considered.

a) Optical attacks: If photons with larger energy than the silicon band gap are injected inside a PD, photocarriers will be generated and then a pulse signal can be generated to an output of the QEC. Photoelectric laser stimulation (PLS) can be applied to the backside of a QEC using a Hamamatsu Phemos-1000 laser scan system as shown in Fig. 11(a). Since the bottom of a QEC wafer is covered with a lead frame as shown in Fig. 2, photons cannot be injected into the silicon and a high-contrast reverse-engineered image by scanning the backside of the QEC with a 1.3-μm laser cannot be obtained as shown in Fig. 11(b). Laser-induced fault-injection attacks are impossible without the decapsulation of the QEC.



(a) (b) (c)

Fig. 12. QEC EM attack. (a) Setup of an EM attack. (b) Random pulse before an EM attack. (c) Random pulse after an EM attack.

b) EM attacks: EM injection can influence the randomness of a QEC. In order to evaluate EM effects, sinusoidal wave signals with 100 MHz and 10 peak-to-peak voltage by a function generator are injected through an EM probe, which is placed on the top of the QEC as shown in Fig. 12(a). Even if the EM injection makes noise included in a random pulse signal higher, it cannot generate or remove pulses. Fig. 12(b) and (c) shows a random pulse before and after an EM attack, respectively. However, we cannot guarantee that the QEC has robustness against EM attacks completely only by this preliminary experiment.

E. Comparison

Table VII shows a comparison of commercial QRNGs including our QRNG. Our QRNG can support an isolated quantum entropy source in the form of a tiny chip in contrast to other QRNGs in which all processing units and quantum entropy sources are included in the form of a PCB or a box case. The QEC is suitable to be combined with legacy or latest IoT devices because of its small size and simple interface.

An alpha particle consisting of two protons and two neutrons has a too heavyweight that its speed is much slower than a photon. Other commercial QRNGs using optical quantum entropy can generate high bit-rate random sequences inherently. However, they need sophisticated sensing devices, and their miniaturization is expensive and impractical. In order to increase the bit rate of our QRNG up to a few Gb/s, multiple QECs and high-frequency DRBGs can be used to generate high-bandwidth random sequences independently and parallelly.

VI. SIDE-CHANNEL-RESISTANT DESIGN USING QRNG

A framework for side-channel-resistant design using QRNG (SCR-QRNG framework [16]) includes a source of quantum randomness, security primitives, such as random frequency clocks and random numbers for side-channel-resistant design, a theoretical model of quantum randomness supported by the QEC, and a quantitative score to evaluate both side-channel resistance and throughput of side-channel-resistant implementations as shown in Fig. 13(a). Based on the theoretical model of quantum randomness, the entropy of generated random numbers and characteristics of random frequency clocks can be estimated; and using a quantitative score, called the SCR-T score, an optimized design is determined among various configured implementations.

Fig. 13(b) shows the block diagram of a side-channel-resistant QRNG (SCR-QRNG) used in the SCR-QRNG framework. It is composed of an analog part for the random frequency clock generator (RFCG) and a digital part for the side-channel-resistant CTR-DRBG to generate random numbers. We implement the CTR-DRBG on Xilinx Spartan-6 FPGA. Random frequency clocks and random numbers generated by the QRNG can be used as hiding and masking resources for side-channel-resistant design, respectively.

TABLE VII
COMPARISON OF COMMERCIAL QRNGS

Company	Name	Random Source	Bit Rate	Dimension (L x W x H)	Passed Test	Pros.	Cons.
ComScire [30]	PureQuantum Model PQ128MS	Shot noise of MOS transistors	128 Mbps	80 x 54 x 23 mm	NIST SP 800-22/90B Diehard	High Speed	High Cost & Volumn
Quintessence [31]	qStream True Random Number Generator	Quantum tunneling; Esaki diode	1 Gbps	169.4 x 64.3 x 23.5 mm	NIST SP 800-90A/B/C	High Speed	High cost & Volumn
IDQ [32]	Quantis-PCIe-16M	Photons; based on a half mirror	16 Mbps	167.7 x 106.7 mm	NIST SP 800-22 AIS 31	Medium speed	High cost & Volumn
MPD [33]	QRN	LED photons; # of detected photons	128 Mbps	-	Dieharder TestU01	High Speed	High Cost & Volumn
PicoQuant [34]	QPRNG 150	Photons; arrival times	150 Mbps	310 x 235 x 140 mm	TestU01	High Speed	High Cost & Volumn
Proposed	Quantum Entropy Chip	Radioactive α particles; arrival times between two detected particles	1.6 Kbps; 11 Mbps with post-processing	3 x 3 x 0.85 mm	NIST SP 800-22/90B AIS 31 Temperature ($0 \sim 75°$C) Voltage ($2.97 \sim 3.63$ V) Aging Optical and EM fault injection	Robustness SCA Resistance Low Cost & Volume	Low Speed

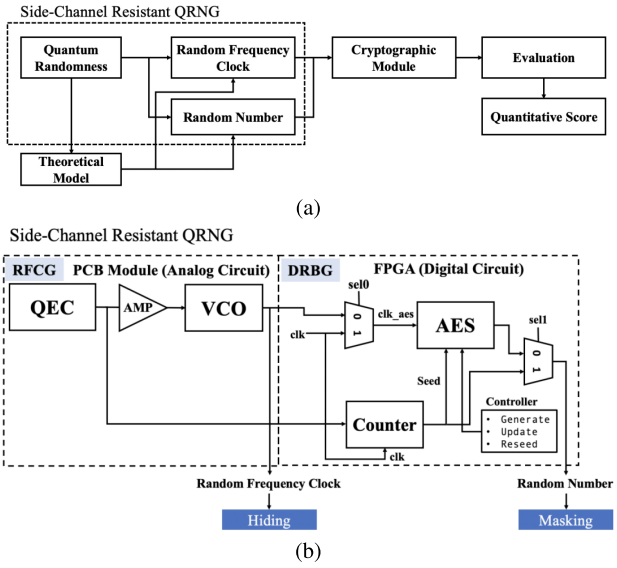


Fig. 13. SCR-QRNG framework. (a) Framework for side-channel-resistant design using QRNG. (b) Block diagram of side-channel-resistant QRNG.

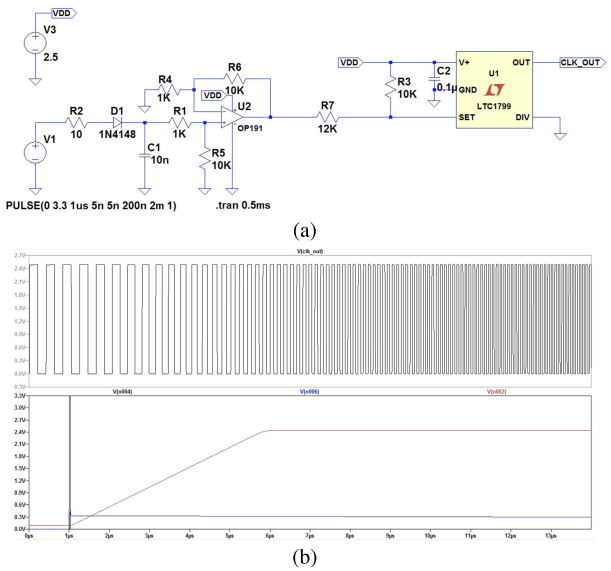


Fig. 14. Simulation of the RFCG. (a) Schematic capture of RFCG. (b) Waveform: V(clk_out): generated clock signal, V(n004): QEC pulse, V(n006): voltage of $C1$ and V(n002): amplified voltage.

A. SCR-QRNG

In order to generate raw random numbers, analog quantum random pulses should be digitized by an entropy extractor. We use a 16-bit counter with a 48-MHz clock cycle as an entropy extractor to measure each interval between consecutively detected decay. The raw 16-bit random numbers from the counter originally have the exponential distribution with the parameter $\lambda = 100$ (not uniformly distributed). The entropy of the raw random number is close to 1/bit if three parameters, λ, n, and f, are satisfied with (2) based on a stochastic model (discussed in Section IV).

Even though raw random numbers extracted by a 16-bit counter have sufficient entropy and unpredictability, the bit rate of the QRNG is low (1.6 Kb/s). In order to increase the bit rate (or throughput) without loss of entropy and unpredictability, a NIST-approved CTR-DRBG using the AES algorithm can be exploited. Extracted random numbers from the QEC are used as the seed of the DRBG in such a way that cryptographic random numbers can be generated with a high throughput until seeds are refreshed. In other words, within a time interval between consecutive detected radioactive decay, theoretically unpredictable random numbers with a high bit rate can be generated by a DRBG. However, if the secret key of the DRBG is revealed by SCAs, generated random numbers can be predicted.

We present a method to obfuscate power side-channel leakages to make SCAs more difficult and impractical in terms of computational complexity or measurements to disclosure (MTD). As a method of SCA obfuscation, the clock frequency of the DRBG is randomly modulated that both timing and power consumption of the target operation, e.g., the first round `SBOX` or `Addround_key` operation, can be changed randomly.

1) Random Frequency Clock Generator: Analog pulses generated by the QEC are used as sources of randomness to generate a random frequency clock. There exist two sources of randomness in the random pulses; the positive pulsewidth and the time interval between two consecutive random pulses which correspond to a Gaussian random variable, W, and an exponential random variable, T, respectively. A capacitor connected to the pulse signal of the QEC is charged during the positive pulsewidth and is discharged during the negative pulsewidth. Using a voltage-controlled oscillator (VCO), the charged voltage in the capacitor can determine the frequency of the clock signal. Since the voltage is charged and then discharged depending on the two random variables, W and T, the frequency is also a random variable, F.

Fig. 14(a) shows the schematic capture of the RFCG using LTspice SPICE simulation tool [35]. In Fig. 14(a), V1 represents the analog pulse of the QEC, which is connected to a low pass filter consisting of a register, $R2$, a capacitor, $C1$, and a switching diode, $D1$. During a positive pulse period, the capacitor, $C1$, is charged with a time constant, $\tau 1 = R2 \cdot C1$, and discharged with a time constant, $\tau 2 = (R1+R5) \cdot C1$, until the next positive pulse is generated by the QEC. A noninverting

Algorithm 1 CTR-DRBG Generate

procedure GENERATING RANDOM NUMBERS, R
Input: $S = (K, V, cnt), \beta$
Output: $S' = (K', V', cnt'), R$
$(K^0, V^0) \leftarrow (K, V)$
$temp \leftarrow \{\}; m \leftarrow \lceil \beta / l \rceil$
  for $j = 1, \ldots, m$ **do**
    $V^j \leftarrow (V^{j-1} + 1) \mod 2^{128}$
    $r^j \leftarrow \mathrm{AES}(K^0, V^j)$
    $temp \leftarrow temp \| r^j$
  end for
$R \leftarrow left(temp, \beta)$
$(K', V') \leftarrow$ Update$(\{0\}^{256}, K^0, V^m)$
$cnt' \leftarrow cnt + 1$
return $(K', V', cnt'), R$
end procedure

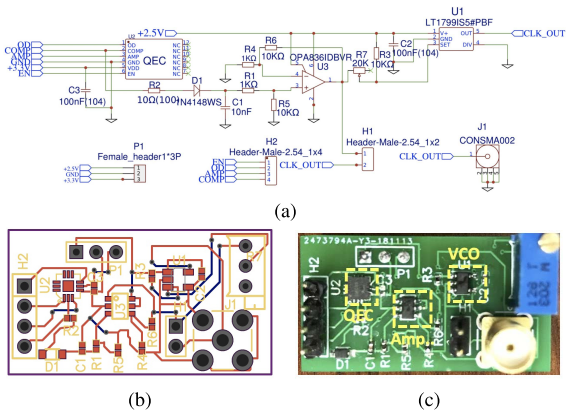


Fig. 15. PCB design of the RFCG. (a) Circuit design. (b) PCB design. (c) PCB module.

amplifier is utilized to amplify the charged voltage in the capacitor, $C1$. An LTC1799 [36] of Analog Devices, Inc., which has the frequency range from 1 kHz to 33 MHz, is used as a VCO. Fig. 14(b) shows the waveform of an analog pulse signal of the QEC, the voltage of $C1$, the amplified voltage, and the clock frequency changed by the amplified voltage. Both positive pulsewidth and negative pulsewidth decide the frequency range of the clock during two consecutive pulses, which will be mathematically analyzed in Section IV. Based on the simulation results, the PCB module of the RFCG is implemented, which can be mounted easily onto a SAKURA-G board. Fig. 15 shows the schematic, the PCB design, and the implemented PCB module of the RFCG.

2) DRBG: An NIST-approved CTR-DRBG using AES-128 encryption has access to the entropy source from the counter [see Fig. 13(b)] with full entropy during DRBG's instantiation in order to generate its initial state or during reseeding. Since the entropy extractor using the QEC provides the full entropy, a derivation function to condition entropy inputs and additional inputs are not required [37]. The initial state S^0 consists of a key, $K_0 \in \{0,1\}^{128}$, an initial vector, $V_0 \in \{0,1\}^{128}$, and a counter for reseeding, $cnt_0 = 1$; $S^0 = (K_0, V_0, cnt_0)$, where the initial seed supported by the QRNG corresponds to (K_0, V_0). The state, S^j, and random numbers, r^j, are updated and generated by a CTR-DRBG generate function which is given in Algorithm 1. The same key is used for the AES encryption until generating the required number of random bits, β. If the reseeding counter value, cnt, is greater than a reseeding period, p_{reseed}, reseeding is performed by the Reseed Algorithm 3.

Note that the key, K^0, at Algorithm 1 is not updated during m iterations, where the standard permits m to be as large as 2^{12} [37]. If an adversary is able to compromise the key, K^0, by SCAs, all future outputs can be recovered until the next reseed call. Generally, AES implementations are vulnerable to SCAs such as DPA and CPA. In order to prevent SCAs, the random frequency clock produced by the RFCG is used as a clock signal to a DRBG module to obfuscate timing and power side-channel leakages for cryptographic operations as shown in Fig. 13(b).

Algorithm 2 CTR-DRBG Update

procedure UPDATING K, V
Input: $provided\_data, K, V$
Output: K', V'
$temp \leftarrow \{\}$
  for $j = 1, 2$ **do**
    $V \leftarrow (V + 1) \mod 2^{128}$
    $C_j \leftarrow \mathrm{AES}(K, V)$
    $temp \leftarrow temp \| C_j$
  end for
$K' \| V' \leftarrow temp \oplus provided\_data$
return (K', V')
end procedure

Algorithm 3 CTR-DRBG Reseed

procedure UPDATING K, V
Input: $entropy\_input, K, V$
Output: K', V'
$K' \| V' \leftarrow$ Update$(entropy_input, K, V)$
return (K', V')
end procedure

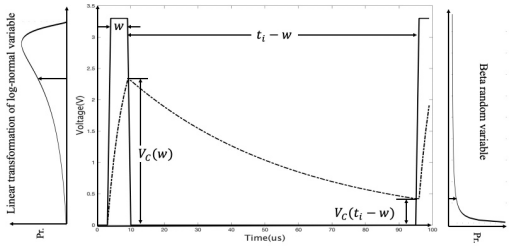


Fig. 16. Mathematical model of a random pulse.

B. Theoretical Model

Let W be the positive pulsewidth, which is a Gaussian random variable; $W \sim \mathcal{N}(\mu, \sigma^2)$. During the pulsewidth, the charged voltage in the capacitor, $C1$, in Fig. 14(a) is as follows: $V_C(w) = V(1 - e^{-w/\tau_1}) + V_C(0)$, where V is the positive peak voltage of the positive pulse, τ_1 is an RC time constant, and $V_C(0)$ is the initial voltage in C_1. Since W is a Gaussian random variable, $V_C(w)$ is the linear transformation, such as $V - V \cdot Y$ of a log-normally distributed random variable, $Y = e^{-W/\tau_1}$ with the parameter $-\mu/\tau_1$ and σ^2/τ_1^2 [38].

Lemma 2: Let $V_C(t_i - w)$ be the remaining voltage at the following decay time. Given $V_C(w)$, $V_C(t_i - w)/V_C(w)$ is a beta random variable with $\alpha = \lambda/\tau_2$ and $\beta = 1$, where τ_2 is an RC time constant.

Fig. 16 shows the random variables included in a random pulse. The charged voltage, $V_C(w)$ is reduced to $V_C(t_i - w)$ exponentially with a time constant, τ_2. The voltage level, $V_C(x)$, where $w \le x \le t_i - w$ controls the frequency of the clock signal generated by the RFCG is as follows [36]:

$$f_{\text{RFCG}} = 10\ \text{MHz} \cdot \frac{10\ \text{K}}{R_{\text{IN}} \| R_{\text{SET}}} \cdot \left[1 + \frac{V_C(x) - V^+}{V_{\text{RES}}} \frac{R_{\text{SET}}}{R_{\text{SET}} + R_{\text{IN}}} \right] \quad (5)$$

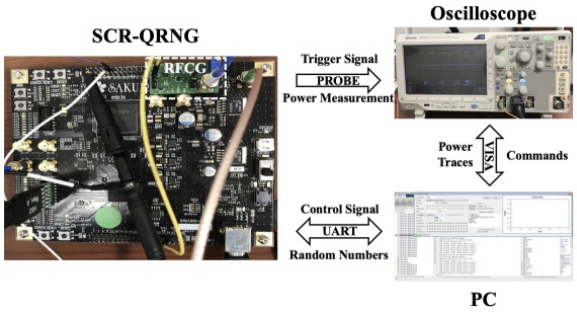


Fig. 17. Setup for power side-channel measurements: SCR-QRNG, oscilloscope, and PC.

TABLE VIII
FREQUENCY RANGE AND CHANGING RATE

	Mean(KHz)	Max.	Min.	σ	Rate(KHz/En.)
Config. 1	1005	25556	617	1489	515
Config. 2	1041	20147	677	1120	418
Config. 3	2245	29550	16222	1101	366

where V^+ is the power supply voltage of the VCO (=2.5 V), V_{RES} is 1.15 V, and R_{IN} and R_{SET} correspond to $R7$ and $R3$ in Fig. 14(a), respectively. In addition, the ratio of $R7$ to $R3$ is related to the ratio of the maximum frequency to the minimum frequency generated by the RFCG as follows. [36]:

$$\frac{R7}{R3} = c\frac{\chi}{\chi - 1} - 1 \tag{6}$$

where $c = (V^+ - V_{C(\mathrm{MIN})})/V_{\mathrm{RES}} = 2.16$, $\chi = (f_{\mathrm{RFCG(MAX)}}/f_{\mathrm{RFCG(MIN)}})$.

C. Setup of SCR-QRNG

Fig. 17 shows that the PCB of the RFCG is mounted on a SAKURA-G board in which a CTR-DRBG is implemented. Power side-channel leakages are measured by a Tektronix MDO3102 oscilloscope, and power traces are sent to a PC for SCAs through virtual instrument software architecture (VISA). Random numbers generated by the SCR-QRNG are transmitted to the PC for statistical and performance evaluation.

D. Side-Channel Leakage of SCR-QRNG

The clock frequency range and the changing rate of frequencies depend on the ratio of $R7$ and $R3$ as mentioned in Section IV. In addition, the ratio affects side-channel resistance. There are three different configurations in terms of the ratio of $R7$–$R3$; Config. 1—1.16 (R7:11.48 KΩ, R3: 9.9 KΩ), Config. 2—1.2 (R7:11.88 KΩ, R3: 9.9 KΩ), and Config. 3—1.5 (R7:14.85 KΩ, R3: 9.9 KΩ). The possible minimum ratio of $R7$–$R3$ is 1.16 by (6) which corresponds to Config. 1. As the ratio is smaller, the available frequency range and the changing rate of frequencies per encryption is larger. Table VIII shows the mean, maximum, and minimum of measured clock frequencies, the standard variation, and the changing rate of frequencies per encryption of three configurations.

In order to perform CPA attacks, 400 000 power traces per configuration are collected during AES encryption with the same key K^0. For comparison to noncountermeasure AES implementation, 100 000 power traces of a reference AES implementation with a 1.5-KHz clock signal are collected. Fig. 18(a)–(c) shows 100 collected power traces of three

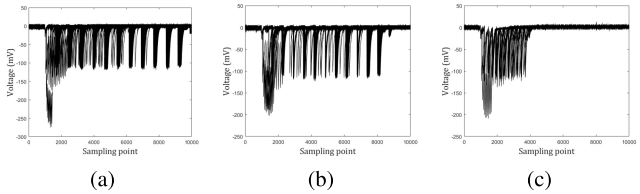


Fig. 18. Power traces of three different configured SCR-QRNGs. (a) Power traces of SCR-QRNG Config. 1. (b) Power traces of SCR-QRNG Config. 2. (c) Power traces of SCR-QRNG Config. 3.

different configurations during an AES encryption.[13] Assuming that an adversary knows the Hamming distance model of the target operations which is `Addround_key` (plaintext$_i$, K^0) $\oplus$ ciphertext$_{i-1}$ in this article, most powerful CPA attacks are performed. We define the oth success rate of the SCA as $\Pr[k^* \in \{g_1, g_2, \ldots, g_o\}]$, where k^* is the correct subkey and g_j is the jth highly probable candidate key for $j = 1, 2, \ldots, 256$. Since the secret key of AES-128 algorithm consists of 16 subkeys, an average of 16 success rates is estimated and we consider up to the second-order success rate.[14]

We define MTD as the minimum number of power traces with which the second-order success rate of the SCA is greater than 0.9. MTD decides the reseeding period of the DRBG, i.e., the reseeding period should be less than the MTD. We assume that an adversary can do two preprocessing of collected raw power traces. The first preprocessing is to filter highly correlated power traces with average power traces, called filtered power traces. The second preprocessing is to align filtered power traces to the average power traces, called aligned power traces. Fig. 19 shows success rates of CPA attacks using various numbers of raw, filtered, and aligned power traces when three different configured RFCGs are working. The first SCR-QRNG with a 1.16 ratio has the largest side-channel resistance based on the MTD even though its average frequency is the lowest. In the case of the third SCR-QRNG with a 1.5 ratio, both first and second success rates using aligned and filtered traces are close to those of the reference implementation as shown in Fig. 19(c).

E. Side-Channel Resistance and Throughput

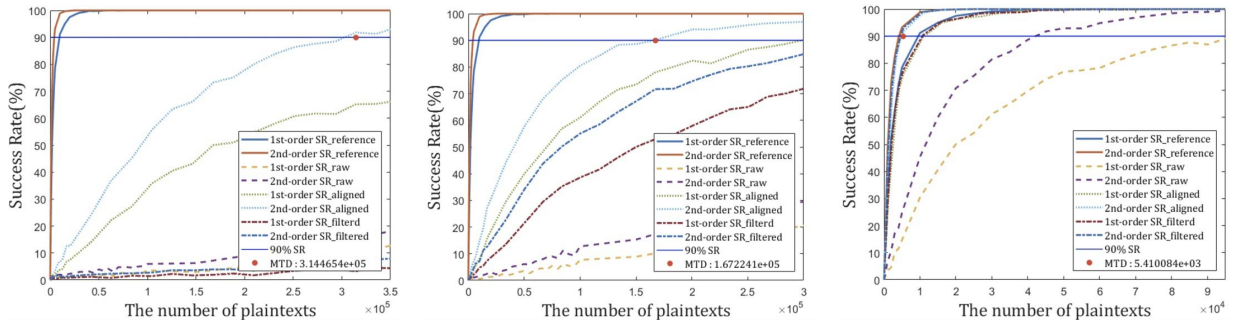
There exists a tradeoff between side-channel resistance and the average frequency of SCR-QRNGs. In other words, as side-channel resistance of an SCR-QRNG is larger, its average frequency is lower as shown in Table VIII. The low frequency caused by high side-channel resistance may result in low throughputs of the SCR-QRNGs. However, since the MTD can be increased by enhancing the side-channel resistance, the throughout can also be increased. Both side-channel resistance and throughput of the SCR-QRNG are increased by the RFCG while its average frequency is decreased, making total power consumption lower.

Given a reseeding period, $p_{\mathrm{reseed}}(\leq \mathrm{MTD})$, which is equal to the MTD, the throughput is calculated as follows:

$$\mathrm{Throughput} = \frac{128 \text{ bits} \times \mathrm{MTD}}{\text{Average seeding time} + \frac{11 \text{ clocks} \times \mathrm{MTD}}{f_{\mathrm{average}}}} \tag{7}$$

[13]An encryption consumes 11 clock cycles including a cycle for an `Addround_key` operation and ten cycles for ten-round iterations.

[14]If two highly probable candidates about each subkey are chosen correctly, one of 2^{16} possible combinations is the secret key.



(a) (b) (c)

Fig. 19. Success rate of CPA attacks. (a) Success rates of SCR-QRNG Config. 1. (b) Success rates of SCR-QRNG Config. 2. (c) Success rates of SCR-QRNG Config. 3.

TABLE IX
MTD, AVERAGE FREQUENCY, THROUGHPUT, AND SCR-T SCORES OF THREE SCR-QRNGS

	MTD	$f_{average}$ (KHz)	T (Mbps)	SCR-T
Config. 1	314465	1005	11.18	3514166
Config. 2	167224	1041	11.11	1857470
Config. 3	5410	2245	3.71	20086.65

Algorithm 4 `Random Encoding`

1: **procedure** GENERATING A RANDOM SEQUENCE
2: **Input:** An n-bit initial vector, IV, a sampling frequency, f_s, a sampling duration, D_s and a random pulse signal, $COMP$
3: **Output:** A m-bit sequence, and $RSA_{pub}[C]$
4: $SR \leftarrow IV$ % SR is a n-bit shift register.
5: $clk_{SR} \leftarrow COMP$ % clk_{SR} is a clock signal of the SR.
6: $cnt \leftarrow 0$
7: $idx \leftarrow 0$
8: **for** $i = 0, \ldots, m-1$ **do** % $m = f_s \cdot D_s$
9: $S[i] \leftarrow SR[n-1]$ % Sampling with a sampling rate, f_s
10: **if** $clk_{SR}! = 1$ **then**
11: $cnt \leftarrow cnt + 1$
12: **else**
13: $SR >> 1$ % Shifting right
14: $C[idx] \leftarrow cnt$
15: $idx \leftarrow idx + 1$
16: $cnt \leftarrow 0$
17: **end if**
18: **end for**
19: **return** $S, RSA_{pub}[C]$
20: **end procedure**

where the average seeding time is 160 ms and f_{average} is the average frequency of the RFCG. We define the SCR-T score considering both side-channel resistance and throughput as follows:

$$\text{SCR}-\text{T} = \text{MTD} \times \text{Throughput}. \tag{8}$$

Table IX shows the MTD, f_{average}, throughput, and SCR-T score of three different configurations. Based on the SCR-T score, the first SCR-QRNG with a 1.16-ratio configuration has the best side-channel resistance and throughput.

VII. DEVICE AUTHENTICATION

As another application using the QEC, a device authentication method is proposed in this section. Fig. 20 shows the block diagram of device authentication. First, the device generates a random pulse signal and sends it to the host (step 1: generating random pulse in Fig. 20). Since a random pulse signal generated by the QEC cannot be emulated by other devices, it is used for a host (or server) to identify trust devices in which the QEC is installed (step 2: randomness test in Fig. 20 such as the tot test in Section V-B). The host

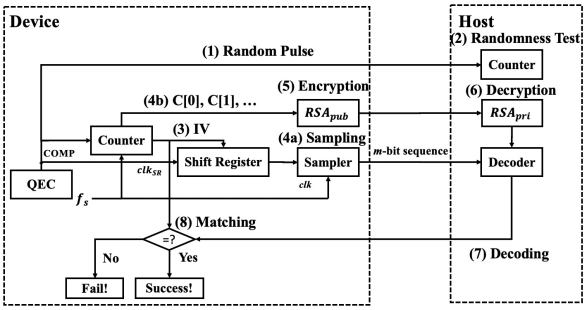


Fig. 20. Block diagram of device authentication.

determines the device as a genuine one if the received signal has an exponential random characteristic.

The devices also need to check if the connected host is authentic or not. For the authentication, the device sends a fixed-length random sequence with a set of encrypted numbers with a public key of the host for decoding the random sequence to the host, which is called random encoding, corresponding to steps 3, 4a, 4b, and 5 in Fig. 20. The host sends a decoded sequence using decrypted numbers with a private key of the host to the device (step 6 and 7 in Fig. 20). If the decoded sequence is matched to the original sequence, the host is authorized to communicate with the device (step 8 in Fig. 20). An unauthorized host does not have the correct private key so that it cannot decode the random code correctly.

Algorithm 4 shows the method of random encoding. A randomly generated n-bit initial vector, IV, is stored in an n-bit shift register, SR (line 4 in Algorithm 4 and step 3 in Fig. 20). Until a rising edge of the next random pulse signal, a counter, cnt, increases (line 10 and 11 in Algorithm 4, and step 4b in Fig. 20) and the MSB of the shift register is sampled with a frequency, f_s (line 9 in Algorithm 4 and step 4a in Fig. 20). When the random pulse increases, SR is left-shifted by a bit, cnt is stored in an array, $C[idx]$, and then reset (lines 13–16 in Algorithm 4). During D_s, by the above process, an m-bit random sequence, S, is generated (lines 8–18 in Algorithm 4) and the counting numbers, $C[i]$, are encrypted using RSA algorithm with a public key of the host (line 19 in Algorithm 4 and step 5 in Fig. 20). Note that the ownership of the public key is proven by a certificate authority (CA). For example, we assume that the 4-bit initial vector is $[1101]_2$, and the bit length of a random sequence is 16. The first time interval between two random pulses is three clocks, and the MSB of SR, "1" is encoded to "111." After shifting, the second time interval is two clocks and "0" is encoded to "00." "1" and "1" are encoded to "111" and "11111111" by the third and fourth

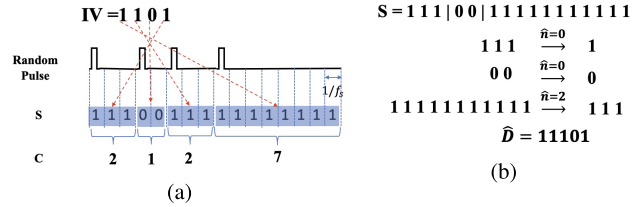


Fig. 21. Random encoding and MLE: $IV \neq \hat{D}$. (a) Example of random encoding. (b) Decoded sequence based on MLE.

time intervals, respectively. The 16-bit encoded sequence is $[1110011111111111]_2$ as shown in Fig. 21(a).

A. Attack Model

The trust host can obtain counting numbers, $C[i]$, by which the random sequence is decoded straightforwardly. It is infeasible for an untrusted host to decode the random sequence without counting numbers, $C[i]$. We assume that the adversary model is based on the maximum likelihood estimation (MLE). The n_e number of "1" (or "0") can be decoded in one of n_e ways, i.e., "1" (or "0"), "11" (or "00"), …, and $\underbrace{\text{'1}\ldots\text{1'}}_{n_e-1}$ (or $\underbrace{\text{'0}\ldots\text{0'}}_{n_e-1}$), which correspond to 0, 1, …, and $n_e - 1$ pulses generated by the QEC during n_e clocks (or n_e/f_s time interval), respectively. The likelihood function is defined as $\mathcal{L}(n; n_e) = f(n) = (e^{-\lambda n_e/f_s}(\lambda n_e/f_s)^n)/(n!)$, where $n = 0, 1, \ldots, n_e - 1$, the number of pulses in an n_e/f_s interval is a Poisson random variable with parameter $\lambda n_e/f_s$. Based on the MLE, a guessing code of an adversary is selected as follows: $\hat{D} = [\underbrace{1\ldots1}_{\hat{n}}]$(or $[\underbrace{0\ldots0}_{\hat{n}}]$), where $\hat{n} = \arg\max_{n\in\{0,1,\ldots,n_e-1\}} \mathcal{L}(n; n_e)$. Fig. 21(b) shows that an estimated sequence of $S = [1110011111111111]_2$ is $[11101]_2$ based on the MLE, which is not equal to $IV = [1101]_2$.

In order to estimate the success rate of the MLE attacks, we generate 1000 64-bit random sequences using Algorithm 4 and then perform the attacks. The success rate of the attack is 0.

VIII. Conclusion

Using the radioactive isotope americium-241, we developed a QEC which can be suitable for IoT devices, and be used as a stand-alone solution or embedded solution. The true quantum random entropy is used to generate the seed of the DRBG and cryptographic random numbers with unpredictability in both forward and backward directions are generated with high-frequency rates until receiving the next quantum seed from the QEC. Since it can also support the DRBG with side-channel robustness using the random frequency modulation, generated random numbers are side-channel-resistant as well as fully compliant with the statistical tests of SP 800-22, SP 800-90B, and AIS 31. Random frequency clocks and random numbers generated by the SCR-QRNG can be used as hiding and masking primitives for other side-channel-resistant implementations. In addition, our QEC provides a solution for device authentication.

In the future, the RFCG based on the QEC will be fabricated using the 180-nm technology node and packaged with the same size as our QEC chip. The RFCG IP will support random frequency clocks as well as random pulses for side-channel-resistant primitives as a stand-alone chip, which can be easily combined with legacy modules or chips without side-channel resistance.

An alternative method to increase the bit rate is to use another radioactive isotope, such as tritium, generating a beta-ray which is a fast energetic electron or positron. In this case, more sensitive sensing is required to detect beta particles with low-level energy. We will develop this method in the future.

Appendix
Proof of Lemma 1

We assume that a counter starts at time t_0. The probability that the LSB of the counter is 0 is equal to the probability that detection of an α particle occurs within $[t_0 + 2iT, t_0 + (2i + 1)T]$, for $i = 0, 1, \ldots$ as follows:

$$\begin{aligned}\Pr[V_0 = 0] &= \int_{t_0}^{t_0+T} \lambda e^{-\lambda t} + \int_{t_0+2T}^{t_0+3T} \lambda e^{-\lambda t} + \cdots \\ &= e^{-\lambda t_0}\left(1 - e^{-\lambda T}\right)\left(1 + e^{-2\lambda T} + e^{-4\lambda T} + \cdots\right) \\ &= e^{-\lambda t_0}\left(1 - e^{-\lambda T}\right)\frac{1}{1 + e^{-2\lambda T}}.\end{aligned}$$

The probability that the LSB of the counter is 1 is equal to the probability that detection of an α particle occurs within $[t_0 + (2i + 1)T, t_0 + (2i + 2)T]$, for $i = 0, 1, \ldots$ as follows:

$$\begin{aligned}\Pr[V_0 = 1] &= \int_{t_0+T}^{t_0+2T} \lambda e^{-\lambda t} + \int_{t_0+3T}^{t_0+4T} \lambda e^{-\lambda t} + \cdots \\ &= e^{-\lambda t_0}\left(1 - e^{-\lambda T}\right)\left(e^{-\lambda T} + e^{-3\lambda T} + \cdots\right) \\ &= e^{-\lambda t_0}\left(1 - e^{-\lambda T}\right)\frac{e^{-\lambda T}}{1 + e^{-2\lambda T}}.\end{aligned}$$

Thus, the ratio of $\Pr[V_0 = 0]$ to $\Pr[V_0 = 1]$ is

$$\frac{\Pr[V_0 = 0]}{\Pr[V_0 = 1]} = \frac{1}{e^{-\lambda/f}}$$

where $f = 1/T$.

In a similar manner, the ratio of $\Pr[V_i = 0]$ to $\Pr[V_i = 1]$ about the ith bit is

$$\frac{1}{e^{-\lambda 2^i/f}}, \quad \text{for } i = 0, 1, \ldots, n - 1.$$

References

[1] C. H. Bennett and G. Brassard, "Quantum cryptography: Public key distribution and coin tossing," *Theor. Comput. Sci.*, vol. 560, pp. 7–11, Dec. 2014.

[2] V. Fischer, "A closer look at security in random number generators design," in *Proc. Int. Workshop Constructive Side-Channel Anal. Secure Design*, vol. 2012, pp. 167–182.

[3] P. Bayon *et al.*, "Contactless electromagnetic active attack on ring oscillator based true random number generator," in *Proc. Int. Workshop Constructive Side-Channel Anal. Secure Design*, 2012, pp. 151–166.

[4] L. E. Bassham *et al.*, "A statistical test suite for random and pseudo-random number generators for cryptographic applications," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-20 Rev. 1a, 2010.

[5] B. Yang, V. Rožic, M. Grujic, N. Mentens, and I. Verbauwhede, "ES-TRNG: A high-throughput, low-area true random number generator based on edge sampling," *IACR Trans. Cryptogr. Hardw. Embedded Syst.*, vol. 2018, no. 3, pp. 267–292, 2018.

[6] P. Kohlbrenner and K. Gaj, "An embedded true random number generator for FPGAs," in *Proc. ACM/SIGDA 12th Int. Symp. Field Program. Gate Arrays (FPGA)*, New York, NY, USA, 2004, pp. 71–78.

[7] P. K. Benjamin. (Jun. 1999). *The Intel Random Number Generator*. Accessed: Apr. 10, 2018. [Online]. Available: https://www.rambus.com/intel-random-number-generator/

[8] J. Szczepanski, E. Wajnryb, J. M. Amigó, M. V. Sanchez-Vives, and M. Slater, "Biometric random number generators," *Comput. Secur.*, vol. 23, no. 1, pp. 77–84, Feb. 2004.

[9] C. Camara, P. Peris-Lopez, H. Martín, and M. Aldalaien, "ECG-RNG: A random number generator based on ECG signals and suitable for securing wireless sensor networks," *Sensors*, vol. 18, no. 9, p. 2747, Aug. 2018.

[10] E. B. Barker and J. M. Kelsey, "Recommendation for random number generation using deterministic random bit generators," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-90A, 2012.

[11] S. Belaid, L. Bettale, E. Dottax, L. Genelle, and F. Rondepierre, "Differential power analysis of HMAC SHA-2 in the Hamming weight model," in *Proc. 10th Int. Conf. Secur. Cryptogr. (SECRYPT)*. Reykjavik, Iceland: SciTePress, Jul. 2013, pp. 1–12.

[12] D. Oswald, B. Richter, and C. Paar, "Side-channel attacks on the Yubikey 2 one-time password generator," in *Proc. 16th Int. Symp. Res. Attacks, Intrusions, Defenses*, 2013, pp. 204–222.

[13] T. Jennewein, U. Achleitner, G. Weihs, H. Weinfurter, and A. Zeilinger, "A fast and compact quantum random number generator," *Rev. Sci. Instrum.*, vol. 71, no. 4, pp. 1675–1680, Apr. 2000.

[14] M. Rohe, "RANDy—A true-random generator based on radioactive decay," 2003.

[15] S. Nikova, C. Rechberger, and V. Rijmen, "Threshold implementations against side-channel attacks and glitches," in *Proc. 8th Int. Conf. Inf. Commun. Secur. (ICICS)*, 2006, pp. 529–545.

[16] J. Park, S. Cho, T. Lim, S. Bhunia, and M. Tehranipoor, "SCR-QRNG: Side-channel resistant design using quantum random number generator," in *Proc. IEEE/ACM Int. Conf. Comput.-Aided Design (ICCAD)*, Nov. 2019, pp. 1–8.

[17] Grabianowski. *How Radiation Sickness Works*. Accessed: Mar. 10, 2018. [Online]. Available: https://science.howstuffworks.com/radiation-sickness1.htm

[18] M. Herrero-Collantes and J. C. Garcia-Escartin, "Quantum random number generators," *Rev. Mod. Phys.*, vol. 89, no. 1, Jan. 2017, Art. no. 015004.

[19] S.-H. Kwok, Y.-L. Ee, G. Chew, K. Zheng, K. Khoo, and C.-H. Tan, "A comparison of post-processing techniques for biased random number generators," in *Proc. Inf. Secur. Theory Pract. Secur. Privacy Mobile Devices Wireless Commun.*, 2011, pp. 175–190.

[20] P. Lacharme, "Post-processing functions for a biased physical random number generator," in *Proc. Int. Workshop Fast Softw. Encryption*, 2008, pp. 334–342.

[21] M. Dichtl, "Bad and good ways of post-processing biased physical random numbers," in *Proc. Fast Softw. Encryption*, 2007, pp. 137–152.

[22] W. Killmann and W. Schindler, "AIS 31: Functionality classes and evaluation methodology for true (physical) random number generators, version 3.1," Bundesamt fur Sicherheit in der Informationstechnik (BSI), Bonn, Gemany, Tech. Rep., 2001.

[23] M. S. Turan, E. Barker, J. Kelsey, and K. McKay, "NIST special publication 800–90B, recommendation for the entropy sources used for random bit generation," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep., 2018.

[24] P. L'Ecuyer and R. Simard, "TestU01: AC library for empirical testing of random number generators," *ACM Trans. Math. Softw.*, vol. 33, no. 4, pp. 1–40, 2007.

[25] G. Marsaglia. (1995). *The Marsaglia Random Number CD-Rom Including the Diehard Battery of Tests of Randomness*. [Online]. Available: http://www.stat.fsu.edu/pub/diehard/

[26] R. G. Brown, D. Eddelbuettel, and D. Bauer, "Dieharder: A random number test suite," Duke Univ., Durham, NC, USA, Tech. Rep., 2004.

[27] F. J. Massey, "The Kolmogorov–Smirnov test for goodness of fit," *J. Amer. Stat. Assoc.*, vol. 46, no. 253, pp. 68–78, 1951.

[28] M. B. Wilk and R. Gnanadesikan, "Probability plotting methods for the analysis of data," *Biometrika*, vol. 55, no. 1, p. 1, Mar. 1968.

[29] R. Maes and V. van der Leest, "Countering the effects of silicon aging on SRAM PUFs," in *Proc. IEEE Int. Symp. Hardw.-Oriented Secur. Trust (HOST)*, May 2014, pp. 148–153.

[30] ComScire. *Purequantum Model pq128ms*. Accessed: Oct. 23, 2018. [Online]. Available: https://comscire.com/product/pq128ms/

[31] Q. Labs. *qStream True Random Number Generator*. Accessed: Oct. 23, 2018. [Online]. Available: https://www.quintessencelabs.com/products/qstream-quantum-true-random-number-generator/

[32] *Quantis Random Number Generator*. Accessed: Oct. 23, 2018. [Online]. Available: https://www.idquantique.com/random-number-generation/products/quantis-random-number-generator/

[33] Micro Photon Devices. *Quantum Random Number*. Accessed: Oct. 23, 2018. [Online]. Available: http://www.micro-photon-devices.com/Products/Instrumentation/Quantum-Random-Number

[34] PicoQuant. *Qprng 150*. Accessed: Nov. 19, 2018. [Online]. Available: https://www.picoquant.com/products/category/quantum-random-number-generator/pqrng-150-quantum-random-number-generator#documents

[35] Analog Devices. *LTspice Spice Simulation*. Accessed: Nov. 19, 2018. [Online]. Available: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html

[36] Linear Technology. *LTC1799 Datasheet*. Accessed: Nov. 19, 2018. [Online]. Available: https://www.analog.com/en/design-center/design-tools-and-calculators/ltspice-simulator.html

[37] J. Woodage and D. Shumow, "An analysis of the NIST SP 800–90A standard," in *Proc. IACR Cryptol. ePrint Arch.*, 2018, p. 349.

[38] E. Limpert, W. A. Stahel, and M. Abbt, "Log-normal distributions across the sciences: Keys and clues," *BioScience*, vol. 51, no. 5, pp. 341–352, May 2001.

Jungmin Park (Member, IEEE) received the Ph.D. degree in computer engineering from Iowa State University, Ames, IA, USA, in 2016.

He is currently a Research Assistant Scientist with the University of Florida, Gainesville, FL, USA. His research interests include side-channel disassembly, side-channel attacks (SCAs), SCA resistant hardware design, hardware Trojan, and QRNG.

Seongjoon Cho received the bachelor's and master's degrees in electronics engineering from Kookmin University, Seoul, South Korea, in 1996 and 1998, respectively.

He is currently the Chief Technology Officer and the Director of EYL Corporate Laboratory, Seoul. He is also the Chief Designer of the Quantum Entropy Chip and the main architect of its various applications. He has decades of experience in leading projects at the intersection of physical and cybersecurity. With a background in electrical engineering, he developed hardware and software solutions ranging from the Internet set-top box, space grade transmission control protocol (TCP)/Internet protocol (IP) board for satellite, various measurement devices, and anti-theft devices for mobile phones.

Taejin Lim received the bachelor's and master's degrees in electronics engineering from Kookmin University, Seoul, South Korea, in 1996 and 1998, respectively.

He worked at Samsung Electronics, Gyeonggi-Do, South Korea, where he developed a variety of hardware and software solutions, such as wireless application protocol (WAP) protocol, code-division multiple access (CDMA) modem chip, Linux porting and MPEG4 device driver, and audio and video capture board. He was a Professor of Computer Science with Sejong University, Seoul. He is currently the Chief Engineer of EYL Corporate Laboratory, Seoul. He is a Key Member of the Quantum Entropy Chip Development Team. He is also leading the project to develop quantum crypto chip, which is the foundation for this proposal.

Mark Tehranipoor (Fellow, IEEE) is currently the Intel Charles E. Young Preeminence Endowed Professor in Cybersecurity with the University of Florida, Gainesville, FL, USA. His current research projects include hardware security and trust, supply chain security, Internet of Things (IoT) security, VLSI design test, and reliability.

Dr. Tehranipoor is a Golden Core Member of the IEEE and a member of Association for Computing Machinery (ACM) and ACM Special Interest Group on Design Automation (SIGDA). He was a recipient of several best paper awards, including the 2008 IEEE Computer Society Meritorious Service Award, the 2012 IEEE CS Outstanding Contribution, the 2009 NSF CAREER Award, and the 2014 MURI Award. He serves on the program committee of over a dozen of leading conferences and workshops. He is also serving as an Associate Editor for the *Journal of Electronic Testing: Theory and Applications*, the *Journal of Low Power Electronics*, the IEEE TRANSACTIONS ON VERY LARGE SCALE INTEGRATION (VLSI) SYSTEMS, and the *ACM Transactions on Design Automation of Electronic Systems*.