SnT acceleration program proposal

Title: QRNG Entropy as a Service(EaaS) Platform for Quantum-Safe Entropy and key Delivery

Prepared by: Hungpu Chou (Hongfu Chou),
Date: December 19st, 2022

1. *Introduction to the motivation and technique on hand*

Random numbers [1] are widely used in numerical computing, statistical simulation, random sampling, etc. The security of random numbers is getting more attention. In addition to random numbers, information and network environment security [2] are also very important in real life, such as the generation of verification code for login, QR code for online payment, etc. All random numbers that involve important identity information must have extremely high security to protect personal privacy from being leaked or illegally stolen. At present, the mechanism for generating random numbers by computers is at risk of being attacked which is subject that the generated random numbers may be predicted in some cases.

Random number generation (RNG) [3] has always been one of the biggest problems. The problem with classic random number generators, i.e., pseudorandom number generators, consists in the possibility to know the deterministic process of pseudorandom generation by unwanted persons. This may result, in the case of cryptography, in compromising a myth of security. Another problem may be the incorrect handling of the generated sequence—mostly in cryptographic uses, the generated random sequence is applied once. Its multiple usage may lead to a security breach (e.g., in the case of the OTP cipher, a sufficiently long key should be truly random and used once in that protocol, otherwise it will be possible to break the code). The scale of threat can be illustrated by selected attacks and information about threats as listed below:

• 2006–2012—over the years there have been many reports of attacks on cryptographic keys generated by weak PRNGs (which allows for example to carry out a brute force attack on SSH secured with RSA keys)

• 2010—a spectacular attack was carried out on users of Sony's PlayStation 3 (PS3) game console (data was stolen as many as 77 million users). The attack was carried out using a flaw in the implementation of the ECDSA algorithm by Sony (disclosed materials reported that the same random number was mistakenly used multiple times as the so-called nonce for authentication)4

• 2012—two groups of researchers revealed numerous RSA encryption keys that were then actively used on the Internet as secure and were at risk of being broken due to insufficient random generator that was used to create them

• 2013—following Snowden's disclosure of these shortcomings to the U.S. National Security Agency (NSA), Reuters and New York Times conducted investigations revealing that the NSA was intentionally secretly lowered the security of the world's popular hardware and programming solutions for the purpose of crypto-attacks on encrypted content (including attacks on RNGs):

– Dual EC DRBG (Dual Elliptic Curve Deterministic Random Bit Generator) was used for this, a PRNG created and strongly pushed as a standard by the NSA. Only in 2013 it turned out that the NSA was the only one to have a backdoor for this generator and thanks to this the NSA was able to crack the cryptographic keys that had been generated using these generators. Upon disclosure, RSA Security and the US National Institute of Standards and Technology (NIST) instructed not to use the Dual EC DRBG generator.

– NSA carried out a secret project code-named Bullrun, focusing on exploiting vulnerabilities in a disseminated PRNG, to which it had access at random, in various devices (e.g., Juniper).

– Intel and Via on-chip HRNG motherboard random number generators probably also had backdoors. It has been indicated that the RdRand and Padlock instructions most likely have backdoors in Linux kernels up to v 3.13.

– Suspected scandal over NSA eavesdropping of 35-country leaders was just related to the use of attacks on RNG.

• 2013—Google confirmed that the IBM Java SecureRandom class in Java Cryptography Architecture (JCA) generated repetitive (and therefore predictable) sequences, which compromised application security made for Android to support the electronic currency Bitcoin – the equivalent of USD large amount in Bitcoins was stolen.

• 2014—It is suspected that the attack on the Tokyo cryptocurrency exchange MtGox, in which more than 800,000 Bitcoins were stolen (which resulted in the declaration of bankruptcy by MtGox) was related to an attack on RNG

• 2015—Hard-to-detect remote attack using an externally attached hardware Trojan horse on FPGA-based TRNGs presented13

• 2015—theft of 18,866 bitcoins from the Bitstamp exchange (12% of the currency traded on this exchange) –attack signature of the RNG attack

• 2017—ANSI x9.31 PRNG compliant to 2016 FIPS USA (Federal Information Processing Standards) – compromised if used with hard-coded seed (DUHK attack—Don't Use Hard-coded Keys)

The presented above examples clearly show that classic random number generators may be exposed to various attacks, or may have the so-called backdoors. This justifies the need to develop alternative technologies that could replace the classic generators on a large scale. The most promising, because they have a fundamental justification for the randomness in the formalism of quantum mechanics, are quantum random number generators.

Another inventive motivation is coming from RNG when working with smart contracts application [4]. A deterministic virtual machine is incapable of generating ‘true’ randomness. Due to this, RNG needs to be provided as an oracle service. The following demand can be satisfied by leveraging quantum physics. Quantum physics can be exploited to generate true random numbers, which have important roles in many applications, especially in cryptography. Quantum Random Number Generation (QRNG) [5] generates randomness via quantum phenomena. It uses a ‘true’ source of entropy using unique properties of quantum physics to generate true randomness. Therefore, QRNG is the gold standard for random number generation. As we known that LUQIA project support us to build the infrastructure of quantum key distribution. The IDQ QRNG solution in Figure 1 is purchased in our Quantum Lab of Luxembourg University.

Figure 1 IDQ QRNG solution

Once we have the true random entropy source QRNG, the next problem could be the transmission way of delivering to the user. This problem [4] is subject to suffering from the same issues as any other third-party oracle network. Setting up an oracle node that can provide PRNG exposes potential attack vectors like Sybil attacks, but also lacks source transparency and decentralization. For example, one needs to trust the governing entity to select the network participants, which means decentralized PRNG is only as secure and decentralized as the governing. To summarize above statement, it could open space for attack vectors by providing RNG through a third-party oracle network. But first-party oracles (Airnodes) [4] that are directly operated by the QRNG API Providers optimally counter the Sybil attack risk. API3 QRNG is a public utility offered through the Australian National University (ANU). It is powered by an Airnode hosted by ANU Quantum Random Numbers, meaning that it is a first-party service.

Australian National University's Quantum Optics Division is one of the worlds leading research institutions in the field. The division also operates a REST API, Quantum Random Numbers API, to serve QRNG in Web2. Thanks to this open-source IP to make the key and entropy source securely deliverable.

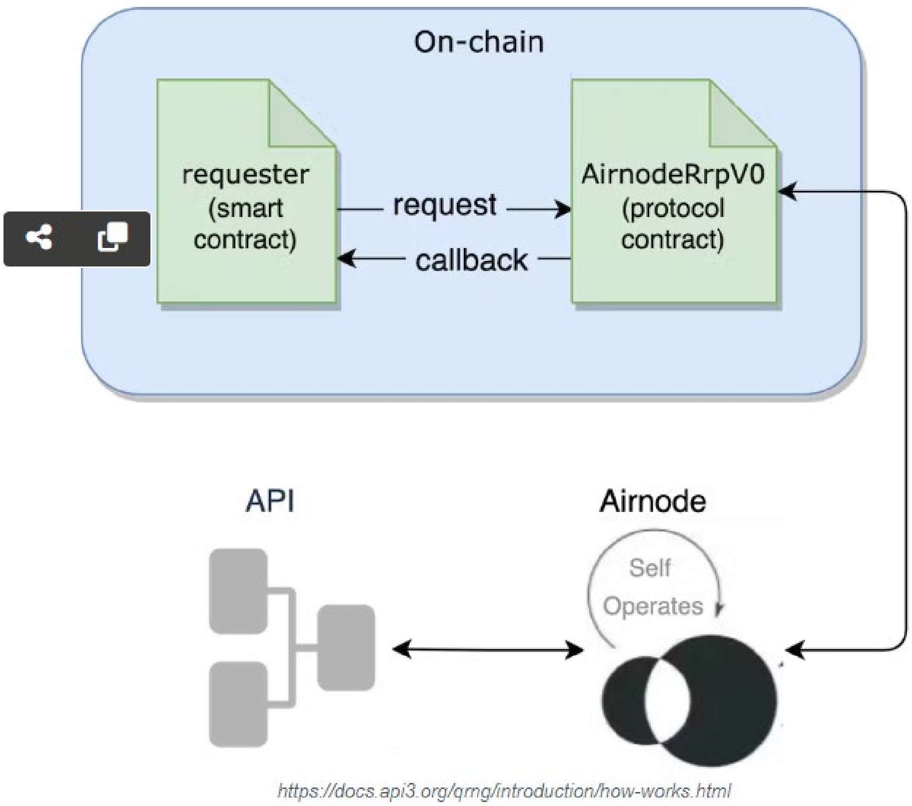

Figure 2 API3 QRNG request-response protocol

Briefly introduce the beginning stage of API3 QRNG request-response protocol in Figure 2 is to deploy and sponsor the QrngRequester with a matching sponsor wallet. The QrngRequester will be the primary contract that retrieves the random number. The QrngRequester submits a request for a random number to AirnodeRrpV0. Airnode gathers the request from the AirnodeRrpV0 protocol contract, retrieves the random number off-chain, and sends it back to AirnodeRrpV0. Once received, it performs a callback to the requester with the random number.

2. *The market survey of QRNG and EaaS application*

Now, it is essential for us to understand how broad the demand of QRNG and EaaS application is and the potential customers.

First, in [3], the turn of the 20th and 21st centuries can be considered the beginning of the currently observed rapid development and spreading of information technology in almost all areas of economy and science and in the sphere of utility. Information technology in many key aspects requires taking into account in algorithms the generating of random variables. Hence, the problem of random number generators plays a fundamental role in the field of information technology, in particular, of information security.

The current applications of random number generators (RNGs) extend to the area of information technology in terms of:

• Applications in the field of cryptography—for individual user applications;—for generation of random initialization sequences (so-called seeds) for encryption algorithms, authentication or digital signature;—for key generation (for asymmetric and symmetric cryptography, e.g., for the One Time-Pad cipher to ensure unconditional security), nonces/initiating vectors (IV), challenges for authentication, selection of exponents in the Dife-Hellman protocol

• other IT applications: e.g., tags/tokens for communication protocols, for indexing in databases, etc.

• statistical applications (e.g., selection of a representative sample for statistical analysis)

• numerical simulations of the Monte Carlo type

• nondeterministic behavior of artificial intelligence (AI)—AI in computer games, in self-controlled devices
(e.g., drones), etc.

• AI algorithms: neural networks (e.g., random weighting for networks) and genetic algorithms (e.g., randomly introducing mutations, randomly mixing representatives)

• structures and support services of currently popular cryptocurrencies (e.g., bitcoin wallets, bitcoin exchanges, etc.)

• games of chance (e.g., online casinos, also for cryptocurrencies)

• randomness in control processes (important problem of sample selection for control processes quality)

• randomness in administration (e.g., drawing the order on election lists)

Second, we summarise the demand of QRNG in the following points [6].

- Worldwide, the installed base of IoT devices will reach into the billions. These devices are particularly susceptible to hacking; IoT devices typically have minimal security measures. IoT chips are targeted by malware that is specifically aimed at such chips, as well as all the older security attacks that have long been familiar to cybersecurity experts. QRNGs seem like a way of addressing this problem and the new IQT Research report indicates QRNG-based security for IoT networks will exceed $600 million by 2030. An especially big opportunity for QRNGs is to be found in the Industrial Internet of Things (IIoT) which are to be found in automated factory, retail and office environments. QRNG are the only form of quantum cybersecurity that is low enough in cost to be used in an IoT environment.
- Another opportunity for QRNGs derives from using the Internet to distribute QRNG-derived entropy – entropy as a service (EaaS). This service delivers either (1) keys that

are created via a quantum entropy source, or (2) the raw entropy for use by the end user in a variety of innovative ways. This service could perhaps be used for smartphones, IoT devices, and maybe even data centers if the key distribution is secure and high rate enough. A number of high-profile firms are already in EaaS business and by 2032 this report predicts that this kind of service will generate around $310 million.

- QRNGs are an obvious way of protecting automobiles from hacking, which is a growing problem given that as much as 20 percent of the value of a car is now represented by electronics for automation or infotainment. Automobiles and trucks must be protected along with a number of dimensions including vehicle-to-vehicle networks, vehicle-to-infrastructure networks and back-end systems. This may be achieved using QRNGs and by 2030 we expect QRNGs in the automotive sector to approach $700 million.
- QRNGs are a genuinely disruptive technology, bringing quantum-level security to markets at a low cost for the first time. QRNG devices also present a relatively easy way for technology firms to enter the market for quantum technologies—substantially easier than with QKD or quantum computers. However, this ease of entry into the QRNG market might make it difficult for some companies to establish sustainable profits in the QRNG space. In part because of this situation, we expect a lot of M&A activity in the coming years, with a few leaders eventually emerging
- Although quantum-secure phones are the most fashionable QRNG products, the largest market for QRNGs will be in data centers – a $3.1 billion in 2026. While all of the quantum-enabled security modalities will be applied to data center applications going forward, QRNGs will be especially useful for supplementing other forms of encryption and in low-cost customer-controlled encryption. QRNGs intended to sell into the data center market should not create significant performance penalties.
- The financial service industry worldwide is expected to show considerable interest in QRNG technology as a way to combat hackers and improve Monte Carlo simulations in portfolio valuation. Financial institutions have been among the most enthusiastic when it comes to the adoption of quantum technology. By 2026 revenues from sales of QRNGs to financial institutions will reach $2.2 billion and will come mainly in the form of standalone systems.

To reveal a clearer scenario of the market of QRNG and EaaS, the following predictions are illustrated [7] and [8] in Figure 3, Figure 4 and Figure 5.

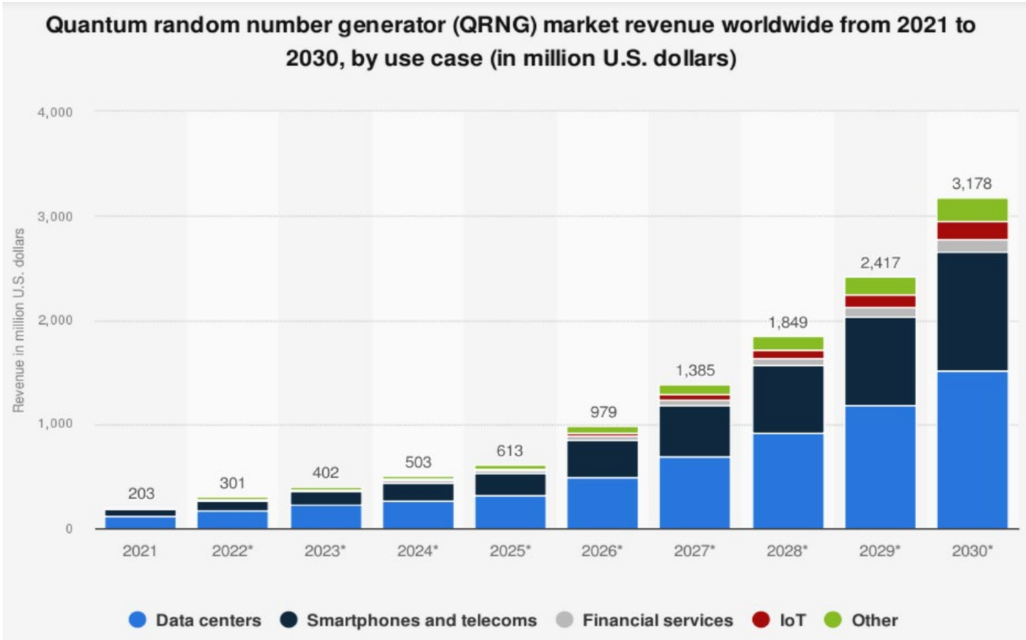

Figure 3 QRNG market revenue 1

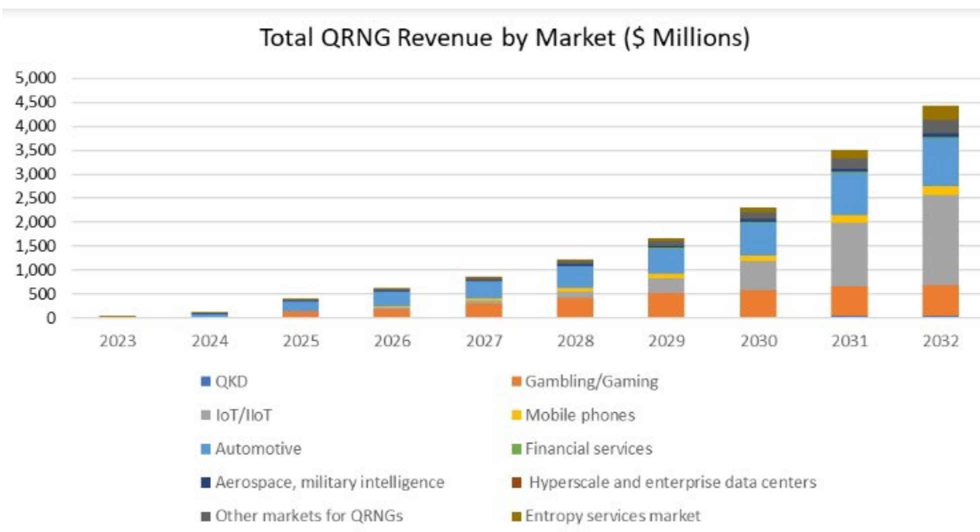

Figure 4 QRNG market revenue 2

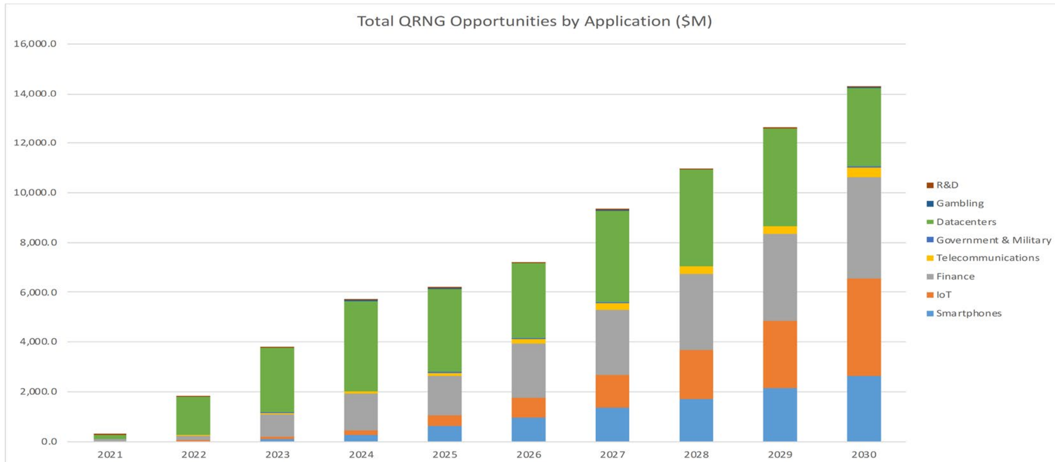

Figure 5 QRNG market revenue 3

Potential competitor

Alibaba, Cypto4A, Crypta Labs, Defense Research and Development Organization), EYL, InfiniQuant, KETS, QNu Labs, Qrypt, Quantaglian, Quantinuum, Quantropi, Quantum Dice, Quantum eMotion, Quantum Xchange, QuantumCTek, Quintessence Labs, Quside, Randaemon, Terra Quantum, Toshiba Europe

3. *Proposed Solution of the QRNG EaaS platform for entropy delivery*

In cryptography, entropy [9] is used to produce random numbers, which in turn are used to produce security keys to protect data while it's in storage or in transit. In today's world, many of our modern digital systems such as embedded devices, IoT devices, and virtual machines do not have direct human interaction to generate good enough entropy. The number of these devices now outnumbers the amount of people on the planet, and will continue to grow into the hundreds of billions over the coming years. The old assumption that you could always leverage a user as an easily-accessible and high-quality source of entropy in a digital system no longer holds. High-quality entropy is now a critical performance and security requirement.

To emphasize more on Entropy [10], in cryptography, entropy has a slightly different meaning. It refers to the randomness collected by a system for use in algorithms that require random seeds. A lack of good entropy can leave a crypto system vulnerable and unable to encrypt data securely. To sum up, random data is added to an "entropy pool" constantly. This randomness is based on "hard to predict" events within the machine. When a user desires randomness, a hash is taken of the entropy pool and the result is supplied to the user. When we call any secure randomness function on a Linux machine, we are likely using this driver or one very similar to it. If the pool of random data becomes fully empty, it can cause long wait times when an app or process requests random data for encryption purposes. More entropy must be generated and collected before the request can be fulfilled. Hence, the proposed service is required to have a good "entropy pool" to create quantum-safe entropy delivery.

To address more on EaaS, we first refer to NIST [11] to define and overview of EaaS. Cryptography is critical for securing data at rest or in transit over the IoT. But cryptography fails when a device uses easy-to-guess (weak) keys generated from low-entropy random data. Standard deterministic computers have trouble producing good randomness, especially resource-constrained IoT-class devices that have little opportunity to collect local entropy before they begin network communications. The best sources of true randomness are based on unpredictable physical phenomena, such as quantum effects, but they can be impractical to include in IoT devices. Standard deterministic computers have trouble producing good randomness, especially IoT-class devices that have little opportunity to build entropy locally before they begin network communications. The best sources of true randomness are based on unpredictable physical phenomena, such as quantum effects but they can be impractical to include in IoT devices. Finding ways to unlock the full potential of cryptography to secure data on the IoT can offer hope for better future.

Entropy as a Service (EaaS) is a novel Internet service architecture providing secure time and quantum entropy sources to IoT devices. The main components of the base EaaS architecture are shown in Figure 6. The critical components are the quantum entropy device, the EaaS server and a hardware root of trust device (TPM, Intel® IPT, ARM® TrustZone®) in the client system. EaaS does not generate keys; it only enables client systems to generate strong cryptographic keys without any possibility for the EaaS server to gain any insight into the client keys. Many today do not trust any centralized authority for a service of such fundamental importance. EaaS is designed to distribute and aggregate trust across a scalable collective of participants, yielding a collective authority. By combining known cryptographic techniques in novel ways, EaaS provides fresh timestamps and entropy to IoT devices on boot. The architecture distributes trust across thousands of servers scattered around the world: scalable enough that every country's government and every major technology company in the world could participate directly in the decentralized root of trust, each actively and independently ensuring that all others "stay honest." The architecture is open and reviewable by experts.

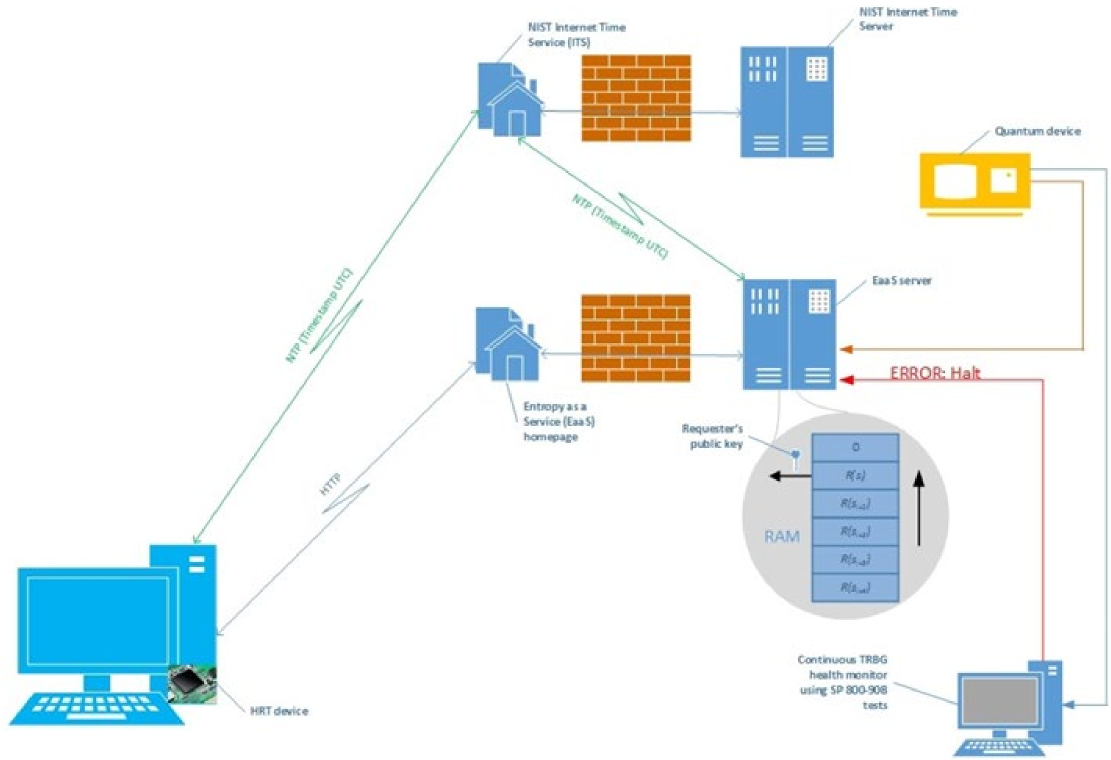

Figure 6 NIST EaaS example

Once our QRNG EaaS platform has been established, the RESTful API protocol [12] for quantum key distribution is delivered to all kinds of networked endpoint to access and illustrated as following Figure 7. It is worth to mention that our proposed QRNG EaaS platform in Figure 8 for entropy delivery is based on API3 first party oracle to alleviate third party oracle network attack.

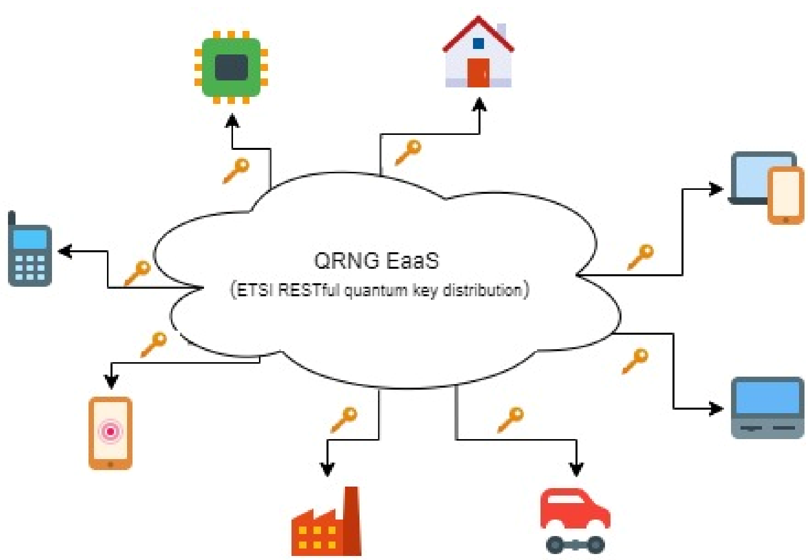

Figure 7 Proposed QRNG EaaS QKD scenario

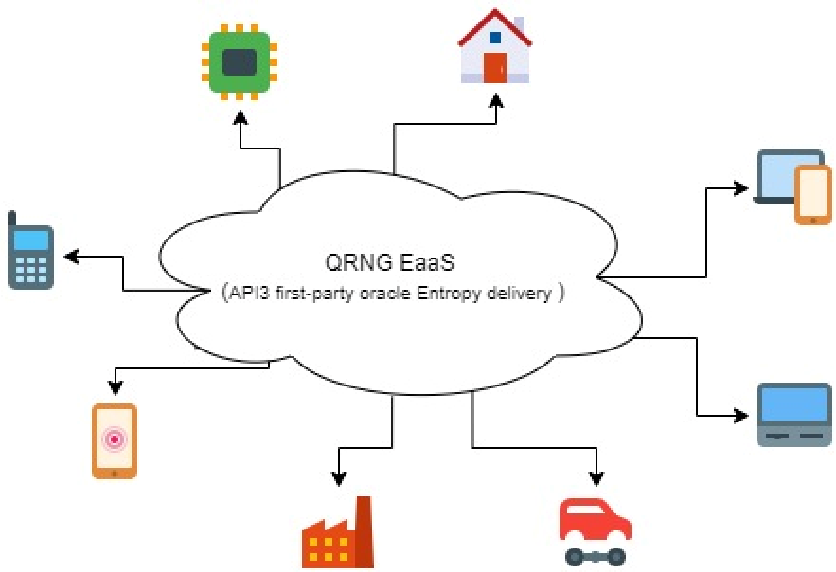

Figure 8 Proposed QRNG EaaS API3 scenario

In [13] , we specify the existing architecture as follows. The architecture of the Entropy-as-a-Service system consists of two main parts: the client-side and the server-side. The critical components of the system are the entropy source, the EaaS server and a secure device in the client systems capable of providing strong isolation and protection of cryptographic keys stored inside the device and offering a set of basic cryptographic services.

The EaaS server is continuously fed random data from the attached quantum source. The data enters a FIFO-like buffer in the server's RAM, and when a client request comes, the server reads the top value off of the buffer, signs and encrypts it, and then sends it to the requester. The FIFO buffer shifts after every request and when new data comes from the random source. The EaaS server ensures that the FIFO buffer is erased prior to server shutdown and never paged to disk. Open implementations can help provide trust that this does, in fact, occur.

The client side of the system consists of a classic computing device enabled with a dedicated hardware component capable of storing secret cryptographic keys and seeds. The client system has a dedicated software application bridging the communication between the EaaS and the hardware component. Examples of secure hardware components are the "Trusted Platform Module", or "TPM", TrustZone in ARM processors, and the IPT technology in Intel processors. Additionally, if a client system or device does not have a secure hardware component, it can still use the EaaS system. The presence of a hardware component simply provides further guarantees to the system or device user, when present.

4. *Working progress and planning*

Entropy as a Service (EaaS) scaling experiment [14] and especially in 5G and cloud environments.

- Tentative WP structure
- API3 EaaS Scaling Experiments:
    - Single Server
        - Number of IoT devices that entropy source can support
        - Performance impact of CPU pinning
    - 2 Servers, Network Connection
        - Number of IoT devices that can be comfortably supported over a network
        - Client and VM clone boot time entropy: Time to accumulate 256 bits of entropy after boot
        - Testing scheme with real IoT devices

Bibliography

[1] X. Y. Q. Z. ,. M. L. CHAO-HSIEN HSIEH, "BCsRNG: A Secure Random Number Generator Based on Blockchain," *IEEE Access,* pp. 98117-98126, 2022.

[2] H. W. G. G. S. Shin, "A first step toward network secu635 rity virtualization: From concept to prototype," *IEEE Trans. Inf. Forensics Security,* p. 2236–2249, 2015.

[3] P. J. J. N. J. E. J. Marcin M. Jacak, "Quantum generators of random numbers," Scientifc Reports nature profolios, 2021.

[4] API3, "How you can use Quantum random numbers as a public good, on 13 smart contract protocols," Hackernoon, 2022.

[5] Q. R. N. Generation, "Quantum Random Number Generation," *Natural partner journal quantum information ,* p. doi:10.1038/npjqi.2016.21;, 2016.

[6] I. Researchers, "Quantum Random Number Generators: Market and Technology Assessment 2023-2032," INSIDE QUANTUM TECHONOLGY, 2022.

[7] [Online]. Available: https://thequantuminsider.com/.

[8] "Statista," [Online]. Available: https://www.statista.com/.

[9] CRYPTO4A, "CRYPTO4A," 2021. [Online]. Available: https://crypto4a.com/blog/why-we-need-entropy-in-cybersecurity/.

[10] L. Wagner, "What Is Entropy In Cryptography?," 2020. [Online]. Available: https://blog.boot.dev/cryptography/what-is-entropy-in-cryptography/.

[11] NIST, "An official website of the United States government," [Online]. Available: https://csrc.nist.gov/projects/entropy-as-a-service.

[12] ETSI, "Quantum Key Distribution (QKD); Protocol and data format of REST-based key delivery API," ETSI Group Specification, 2019.

[13] A. Vassilev and R. Staples, "Entropy as a Service: Unlocking Cryptography's Full Potential," *Computer,* vol. 49, no. 9, pp. 98 - 102, 2016.

[14] R. Jagannathan, "Entropy as a Service: A Framework for Delivering High-quality Entropy," RSAConference2020, 2020.