Introduction

Random numbers are a key foundational component of many cryptographic algorithms - especially when generating keys for Public Key Infrastructure (PKI). The more entropy *(a measurement of randomness)* that is provided to the algorithm, the stronger the key and the difficulty to predict patterns of use with the key. With lower entropy, the resulting key is weaker, the key becomes more predictable, and the key is more susceptible to reasonable brute force.

Modern operating systems all employ a Random Number Generator (RNG) technique to generate entropy for the purposes of cryptographic uses which is often low on the entropy scale. Many application stacks utilize the operating system's entropy source or even simply use a basic pseudo-random number generator. Both of these sources generally have low entropy and are affected by user input, system load and processes. Further, many devices use the factory default keys, or use weak keys that are often repeated across multiple devices and therefore corruptible. Weak entropy enables signature forgery and private key theft.

Quantum sources of entropy can provide true randomness for cryptography as a whole. Common quantum sources come from measuring the quantum noise of light photons or random decays from radioactive particles. Injecting quantum entropy from these sources into new and existing software applications and PKI solutions significantly increases the strength of the keys and thereby the security of the device.

Both pre-quantum and post-quantum cryptography significantly benefit from quantum entropy.
QCloud is a technology and means by which to distribute and inject quantum entropy into existing cryptographic systems with little or no change to the software application stack. This applies to pre-quantum public key and signature algorithms as well as post-quantum solutions. Quantum entropy is one of the strongest sources of entropy. QCloud will deliver quantum entropy to client systems using algorithms from NIST PQC round 2 combined with AES. Despite the name, the technology supports on-premise deployment or air-gap scenarios.

Statement of Need

Low entropy used in current PKI and signature algorithms leaves algorithms susceptible to entropy side-channel attacks. This threat exists for current PKI and signature algorithms yet also exists for the NIST PQC round 2 algorithms. The following sections describe the importance of entropy, entropy attacks, and the details on the value-added for pre- and post-quantum cryptography.

Entropy Scoring & Implications

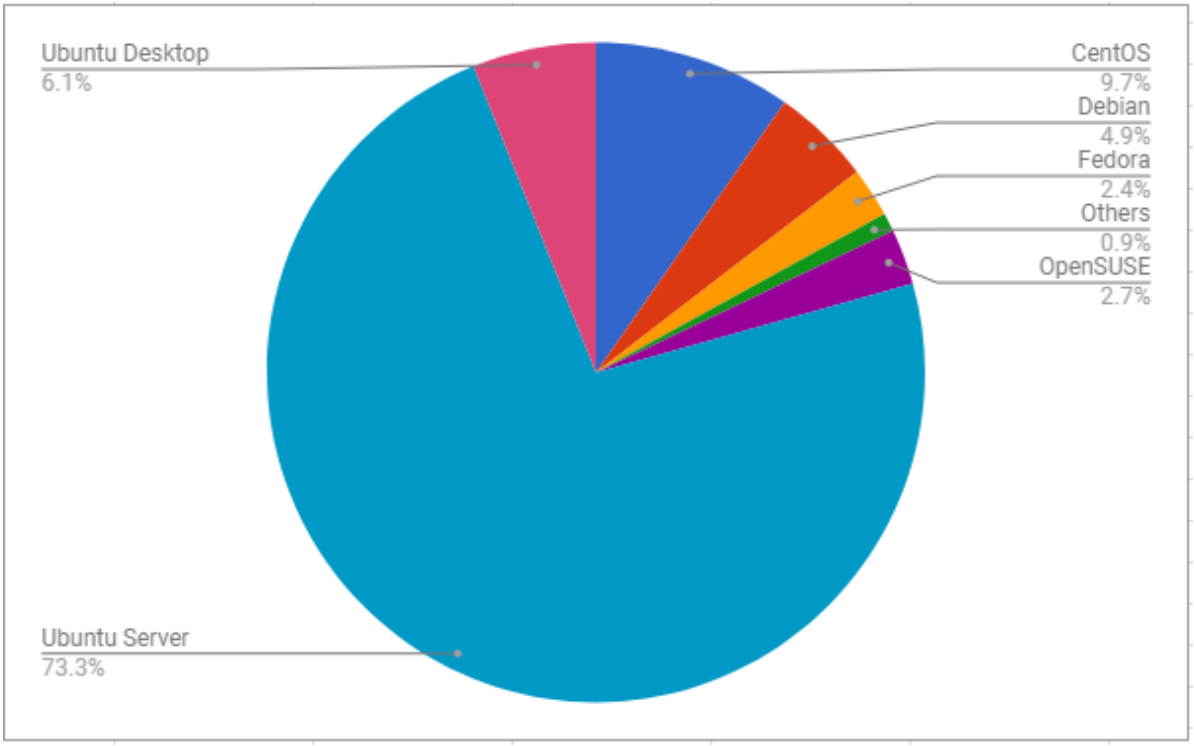
So why is entropy so important? The more random the numbers provided as a source to a cryptographic function, the more difficult it is to guess the key used to encrypt or sign sensitive data. By definition, non-random data also includes repetitions of sequences and patterns. Therefore, not only does a weak entropy source weaken the encryption key and thus make it susceptible to a brute force attack, but encrypted data itself may reveal patterns that can be used to uncover the underlying keys.

Entropy Scoring

NIST has a set of tools publicly available called SP800-90B for scoring the quality of entropy sources - or an entropy assessment. A true random score, according to the toolset, is about a 7.8. It turns out that the scoring of quantum sources ranges between 7.5 and 7.8. At Bright Apps we scored entropy sources available in general purpose Linux systems, and streamed random numbers scored on average about a 2.0. Once scores are less than 2.0, the brute force weak entropy attack becomes viable. In fact, Bright Apps is able to demonstrate brute force decrypt a public/private elliptical curve key set that was generated with a known low entropy source on a Raspberry Pi in minutes.

Entropy Sources

As of 2017, Ubuntu was arguably the largest install base for servers on the Internet. According to one source, 73.3% of servers on the internet run Ubuntu and its method for generating random numbers and keys. This means that for 2017 created keys, one might have a 73.3% chance that the Ubuntu RNG system of making a low entropy was utilized and therefore brute force attacks become viable. We know the bulk of entropy sources currently available *(side channel attack),* and they are not truly random entropy.



[1]=

[1] https://www.cloudbalkan.com/most-popular-linux-distributions-from-2017/

Entropy Attack

Any entropy source noticeably lower than a 7.8 begins to significantly decrease the computing power required for a brute force attack. For example, Bright Apps took an unviable random number generator from 1946 called the middle-square method. Although this method is completely impractical, its poor quality and repetitiveness quickly demonstrates an entropy attack and what happens when the weak link is the RNG on the existing system. By generating an elliptical key pair with this entropy source, we are able to recover public key A and its associated private key A. Our method in the attack was to simply use the same middle-square method to generate new public keys until our generated key X matched public key A. Once this happens, we then also have the private key A'. On a Raspberry PI, we generally obtained the key pair in less than a minute.

Effects on Pre- and Post-Quantum Cryptography

It is important to understand the differences between the effect quantum entropy has on pre- and post-quantum cryptography, as described below.

Pre-Quantum Computing

In the pre-quantum computing cryptography world, brute force entropy attacks are directly correlated to the entropy score. The lower the score, the less brute force that is required. This means with the average low scores for servers on the Internet, it does not require a quantum computer to brute force the private key, but rather just a bank of capable computers. Quantum entropy is important to ALL pre-quantum cryptography because it makes the computing power required to brute force a key impractical.

In a pre-quantum computing world, quantum entropy is an absolute necessity if the integrity of the cryptography is to be maintained.

Post-Quantum Computing

In the post-quantum computing (PQC) world, there are a number of algorithms that regardless of quantum entropy are instantly at risk. For example, a quantum computer using Shor's algorithm puts cryptography that base their mathematical problems on integer factorization, discrete logarithms, or elliptic-curve discrete logarithm at risk. This includes X.509, RSA, Elliptical Curve, and additional PKI systems. Quantum entropy does not assist these algorithms and hence the need for this RWP.

However, the proposed NIST PQC round 2 key and signature algorithms that are in theory quantum safe still require true random entropy to generate keys. Lower entropy leaves these algorithms vulnerable to non-quantum attacks; the same attacks that happen today. Each of these algorithms in a PQC world will benefit significantly from QCloud's quantum entropy.

QCloud plays a significant role in both pre-quantum and post-quantum worlds.

Quantum Entropy Distribution over a Quantum Resistant Network

Realizing that quantum entropy is critical in a pre- and post-quantum computing cryptographic world, there still remain challenges in securely delivering quantum entropy. In addition to the distribution of entropy, the ability to make use of the entropy without having to modify existing software stacks is the key to securing the pre- and post-quantum world.

Entropy Distribution

Bright Apps will utilize a key and signature algorithm from the NIST PQC round 2 list to encrypt (via AES) and sign entropy. Prospectively, BIKE and FALCON with AES.

Bright Apps LLC

Distribution and Usage

Once entropy is PQC-securely delivered to an endpoint for usage, it is critical that it does not require vendors integrate with an API when not necessary. If vendors wish to directly integrate with QCloud or the entropy source, that must be possible. Re-tooling the entire existing crypto infrastructure for a new entropy source would be too costly and time consuming.

QCloud integrates its securely delivered entropy by replacing operating system methods and device drivers that produce and deliver entropy to existing applications. This is removing the burden of forcing the industry to re-tool their software stacks.

Technical Benefits and Details

The following section walks through the technical benefits and details as it applies to PQC

NIST PQC Public-key Encryption and Key-gen Algorithms

Public/Private Key Algorithms	*Does Quantum Entropy Strengthen*
BIKE	yes
Classic-McEliece	yes
CRYSTALS-Kyber	yes
FrodoKEM	yes
HQC	yes
LAC	yes
LEDAcrypt	yes
NewHope	yes
NTRU-Prime	yes
NTRU	yes
NTS-KEM	yes
ROLLO	yes
Round5	yes
RQC	yes
SABER	yes
SIKE	yes
ThreeBears	yes

Bright Apps LLC

NIST PQC Digital Signature Algorithms

Digital Signature Algorithms	*Does Quantum Entropy Strengthen*
CRYSTALS-Dilithium	yes
Falcon	yes
GeMSS	yes
LUOV	yes
MQDSS	yes
Picnic	yes
qTESLA	yes
Rainbow	yes
SPHINCS	yes

As noted in the table above, QCloud affects the strength of many NIST PQC round 2 algorithms. Having said that, QCloud will utilize the BIKE algorithm for generating public and private keys along with Falcon for the digital signature algorithm. Quantum entropy shall be delivered with a **BIKE** key utilizing **AES** and a payload digital signature from **FALCON**.

Internet Protocols and Packet Quality/Loss

QCloud utilizes TCP and UDP to deliver packets of entropy. Each packet is encrypted with the same key size as the AES algorithm and digitally signed by Falcon. Since we have true random data in each packet, a loss of a packet is therefore irrelevant. What is important is that the packet is encrypted and digitally signed so that the client can trust the entropy to be used. *If needed, client-side augmentation of the entropy is possible as well without affecting the quantum score.*

2872 Ygnacio Valley Road #146 • Walnut Creek, CA 94598 • Info@brightappsllc.com

Migrating Existing Systems

In most cases, crypto systems read their entropy from a standard entropy source from the device operation system. Since QCloud replaces those device drivers with QCloud device drivers, the quantum entropy is automatically injected into the algorithms. In most cases, this is an update to the operating system, leaving no extra work for the existing application.

Standards and Performance

There are a few quantum number generator companies building hardware to generate quantum entropy. Existing QRNG vendors expect IT organizations to install custom hardware and then update their software stack to make use of that hardware entropy. Our proposed QCloud solution does not require the application providers to integrate new hardware, nor does it require the application provider to update their existing software. By installing the QCloud device drivers, as mentioned above, QCloud takes the quantum entropy from a remote hardware source and injects quantum entropy into the applications.

NIST currently has a toolset to measure the quality of entropy the SP 800-90B. Bright Apps is working with NIST to update and continue to improve entropy scoring. Defining a value for quantum RNG sources is one of our goals to develop.

Security Benefits and Details

Side Channel Attacks

By utilizing QCloud and quantum entropy, the entropy is truly random and therefore highly unpredictable. The entropy attack example for Ubuntu servers from the previous section is realistic when considering a side-channel attack. Simply by looking at server market penetration, we know the poor entropy sources to apply against PKI and signature schemes. The fact that most servers are a virtual machine (VM) makes the problem worse. Cloning a virtual machine literally clones the source of VM entropy. Using securely delivered quantum entropy sourced by QCloud not only eliminates the problem of re-used entropy, but also eliminates observation of key generation an attacker might use to exploit the cryptography.

Security, Strengths and Upgrades

Since the proposed solution for QCloud is installed device drivers on remote systems to receive quantum entropy from the cloud and inject that entropy into existing cryptographic functions, we automatically gain support for 192-256 classical security and 96-128 bit PQC security algorithms. When implemented in a complete solution, the application must provide random information from the operating system to the above algorithms to seed their use. This means that any system or application that utilizes the new PQC algorithms all receive this new quantum entropy. This is a significant enabler. This means that simply by installing the QCloud device driver that applications pre- and post-quantum are already enabled and upgraded.