# PCIe QRNG ACCEL GEN 3 - Usage Manual

## Table of Contents

1. Scope ..... 2
1.1. Overview ..... 2
2. Referenced Documents ..... 2
3. Document History ..... 2
4. PCIe QRNG Acceleration Board Setup Steps ..... 3
4.1. Overview ..... 3
4.2. Diagnostics Test ..... 4
5. Extras ..... 4
5.1. OpenSSL 3.6.5 Integrated with EigenQ QMA ..... 4


© 2026 EigenQ, Inc

# 1. Scope

## 1.1. Overview

This document provides guidance on installing the EigenQ PCIe QRNG Acceleration Board and validating random number generation.

# 2. Referenced Documents

None

# 3. Document History

| DATE | VERSION | AUTHOR | DESCRIPTION |
|---|---|---|---|
| 29-Sept-26 | 1.0 | Pedro | Document Creation |
| 02-Oct-26 | 1.1 | Pedro | Update to RPM installation steps. |
| 14-Oct-26 | 1.2 | Pedro | Adding MOK Key Steps. |
| 15-Oct-26 | 1.2 | Pedro | Updating GPG Public KEY. |


© 2026 EigenQ, Inc

# 4. PCIe QRNG Acceleration Board Setup Steps

Guidance for a technician to set up, install, and use the PCIe QRNG Acceleration Board. The following are required:

- Red Hat Enterprise Linux 9
- PCIe QRNG Acceleration Board with Gen 3 firmware
- eigenq-firm-gen03-packages-rhel9.zip downloaded and available on the OS

## 4.1. Overview

This section guides you through setting up and validating the EigenQ PCIe QRNG Acceleration Board on Red Hat 9: confirm prerequisites, install the hardware, deploy EigenQ services, load the driver, and verify service health and entropy generation.

1. Verify that the target machine is running Red Hat 9.
2. Install the PCIe QRNG Acceleration Board in the machine. Shut down the machine, install the board, and then power the machine on.
3. Confirm that the PCIe QRNG Acceleration Board is recognized by the OS. Run the following command in a terminal:

   ```
   lspci | grep -i accelerators
   ```

   Example expected result (PCI bus address may vary):

   ```
   86:00.0 Processing accelerators: EigenQ, Inc. QMA Board PCIe Gen2
   ```

4. If the command above does not return the EigenQ device description, run the following command to update the PCI ID database:

   ```
   sudo update-pciids
   ```

   Run the `lspci` command again and confirm that EigenQ is listed.

5. Import the EigenQ RPM Signing Public Key:

   ```
   curl -fsSLo /tmp/RPM-GPG-KEY-EigenQ \
                https://support.eigenq.com/keys/GPG-KEY-EigenQ-101520261419
   gpg --show-keys --with-fingerprint /tmp/RPM-GPG-KEY-EigenQ
   sudo rpm --import /tmp/RPM-GPG-KEY-EigenQ
   ```


© 2026 EigenQ, Inc

6. Install the EigenQ EQDMA drivers and dependencies.

   ```
   unzip eigenq-firm-gen03-packages-rhel9.zip
   sudo dnf install -y https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
   sudo dnf install -y "kernel-devel-$(uname -r)"
   sudo dnf install -y ./*.rpm
   ```

If Secure Boot is enabled, the DKMS signing certificate must be enrolled before the EigenQ kernel driver can load.

After installing the EigenQ RPM, run:

```
sudo /usr/bin/eigenq-enroll-mok
```

You may be asked to set a password for MOK enrollment. Memorize this password for Step 8, where it will be required after the machine reboots and you need to enroll the MOK Key.

7. Check the driver installation status.

   ```
   sudo dkms status
   ```

   Example expected result (kernel version may vary):

   ```
   eigenq-eq_qma/3.0, 5.14.0-687.53.1.el9_8.x86_64, x86_64: installed
   ```

8. Reboot.

   ```
   sudo reboot
   ```

   During startup, the blue **Shim UEFI Key Management** screen will appear. Select the option to enroll the MOK key, then enter the password you created in Step 6.

9. Confirm that the driver loaded successfully.

   ```
   ls /dev/eq*
   ```

   Expected result:

   ```
   /dev/eq_qma0
   ```

   If no devices are returned, reboot the machine and run the command again.

10. If the driver is loaded, start the EigenQ service.

    Check whether the service is running:

    ```
    sudo eq-qes status
    ```

    If the service is not running, initialize it:


**© 2026 EigenQ, Inc**

```
sudo eq-qes init
```

11. Verify that the service is operating correctly.

```
sudo eq-qes monitor
```

Press Ctrl+C to stop monitoring.

## 4.2. Diagnostics Test

Use the EigenQ diagnostics script to run diagnostics on the PCIe QRNG Acceleration Board. First create a directory for the eigenq logs to be stored. For now let's go with /home/eigenq/logs.

```
mkdir -p /home/eigenq/logs
```

Then we can start the diagnostics script, run:

```
bash /usr/lib/eq-diagnostics.sh
```

# 5. Extras

## 5.1. OpenSSL 3.6.5 Direct Integration with EigenQ QMA

The following steps install an OpenSSL 3.6.5 build integrated with the EigenQ Board. This build uses /dev/hwrng as its primary entropy source, where the EigenQ service provides entropy directly. Please download openssl_bin_rh9.tar and transfer it to the tmp folder. You can use wget if you would like.

```
tar -xf openssl_bin_rh9.tar
sudo mkdir /app
sudo mv openssl_bin /app/openssl_bin
sudo chmod 644 /dev/hwrng
```

To use this OpenSSL version in the current shell session, run:

Become the root user:

```
sudo su
```

Set the openssl on the linux path:

```
export PATH=/app/openssl_bin/bin:$PATH
export LD_LIBRARY_PATH=/app/openssl_bin/lib64:$LD_LIBRARY_PATH
```

Confirm that the OpenSSL version is 3.6.5:

```
openssl version
```

Verify that OpenSSL is integrated with the EigenQ Board.

Stop the EigenQ service:


© 2026 EigenQ, Inc

```
sudo eq-qes stop
```

Clear the hardware RNG buffer:

```
timeout 5s cat /dev/hwrng | pv > /dev/null
```

Run the following test:

```
timeout 5s openssl rand -hex 32
echo $?
```

Expected result: exit code 124. This indicates that the command did not complete within the 5-second timeout while the EigenQ service was stopped.

Initialize the EigenQ service:

```
sudo eq-qes init
```

Run the following test:

```
timeout 5s openssl rand -hex 32
echo $?
```

Expected result: the command returns the requested random data and exit code 0.
Re-run the diagnostics script to include the OpenSSL benchmark test:

```
bash /usr/lib/eq-diagnostics.sh
```


**© 2026 EigenQ, Inc**