AlphaRNG V1.0 – May 2021 – Rev 1.3

AlphaRNG is a Hardware Random Number Generator that can produce high quality true random numbers. Compared to other generators of its class, AlphaRNG implements data communications over a secure channel for confidentiality, integrity, and replay protection using a USB interface. It ensures the reliability of the built-in health tests using a technique known as hardware fault insertion (also known as fault injection). Hardware fault insertion is used to create temporary signal faults in the noise sources for the purpose of validating the expected behavior of the device under failure conditions. AlphaRNG is designed to comply with NIST SP 800-90B: "*Recommendation for the Entropy Sources Used for Random Bit Generation".*

Core capabilities

- Secure Data Communication using RSA, AES-GCM and HMAC
- Connectivity through a USB 2.0 High Speed interface.
- Max generation performance - 199 Mbit/s generation speed when used in non-secure mode
- Min generation performance – 3.4 Mbit/s when used with AES-256-GCM and HMAC-SHA256
- Compatible with Linux, 64bit Windows, macOS 10.15 Catalina and up (Intel and Apple Silicon)
- Device API enables independent security testing and validation of the entropy source in compliance with NIST SP 800-90B.
- On demand and start-up built-in diagnostics.

Description

The entropy byte-stream provided in the AlphaRNG output is based on electrical noise produced by two Zener diodes working in avalanche mode. The independent electrical noise created by each electrical circuit is amplified and sampled into independent raw byte-streams and have uniform distribution of the random values. The two resulting raw byte-streams are then combined and de-biased to produce the resulting entropy byte stream. The AlphaRNG software kit provides an API for retrieving data at each processing stage for the purpose of evaluation or, for example, when a different post-processing or conditioning algorithm is to be used.

The entropy byte-stream provided in the AlphaRNG output is based on electrical noise produced by two Zener diodes working in avalanche mode. The independent electrical noise created by each electrical circuit is amplified and sampled into independent raw byte-streams and have uniform distribution of the random values. The two resulting raw byte-streams are then combined and de-biased to produce the resulting entropy byte stream. The AlphaRNG software kit provides an API for retrieving data at each processing stage for the purpose of evaluation or, for example, when a different post-processing or conditioning algorithm is to be used.

AlphaRNG implements an API for establishing a secure connection with a computer over a USB interface. The secure connection is initiated on the computer by using the AlphaRNG software API (client). Once the client establishes a successful connection over the USB interface, the client creates a session request, encrypts it with the public RSA key, and sends it to the AlphaRNG (see Fig. 1). A session request consists of parameters such as symmetric cipher, randomly created cipher key, a new non reusable cipher IV, cipher AAD, MAC algorithm, randomly created MAC key, a new unique non-reusable request token, and a new generated MAC value of the plain request. The AlphaRNG receives the session request, decrypts it with its secret RSA key, generates its own MAC value of the plain request (using the MAC key provided in same request) and compares it with the MAC value provided in the requests. Upon successful validation of the session request, the AlphaRNG will store new session parameters and use those for the duration of the session. It then creates a response packet, encrypts it with the session cipher and key/IV/AAD provided and sends the response packet back to the client. A response packet consists of session cipher used, cipher tag, request token, MAC value of the plain response and RNG internal status code. The client receives the response, decrypts it with the session cipher using key/IV/AAD and cipher tag provided, generates its own MAC value of the plain response and matches it against the one provided, then it compares the provided token in the response with the one initially generated for the session request. Once the response is validated, the session is successfully established, and the client can securely communicate with AlphaRNG device over the USB interface.

Once a secure connection has been established, the client can send commands to AlphaRNG and receive responses. The client creates a command packet, generates the MAC value (using the session MAC algorithm and the MAC key) of the plain command data and includes it in the packet, it then encrypts the packet with the session cipher and the key/IV/AAD and sends it to the USB device. The command packet consists of parameters such as the cipher used, a new non-reusable cipher IV, cipher tag, MAC value of the plain command data, command ID and a freshly generated unique non-reusable command token. The AlphaRNG receives the command and decrypts it with session cipher and the key/AAD/IV, generates the MAC value using the session MAC algorithm and the MAC key and compares the MAC value with the one provided in command packet. Upon successful command validation, the AlphaRNG handles the command, creates a response packet, generates the MAC value, encrypts the response packet with the session cipher and the key/IV/AAD and sends the response packet back to the client. A response packet consists of session cipher used, command token, cipher tag, MAC value of the plain response, data payload and the RNG internal status code. The client receives the response, decrypts it with the session cipher using key/IV/AAD and cipher tag provided, generates its own MAC value of the plain response and compares it against the one provided in

the packet, then it compares the provided token in the response with the one initially generated for the command packet. Upon successful validation of the response packet, the client validates the RNG internal status and dispatches the response payload accordingly.

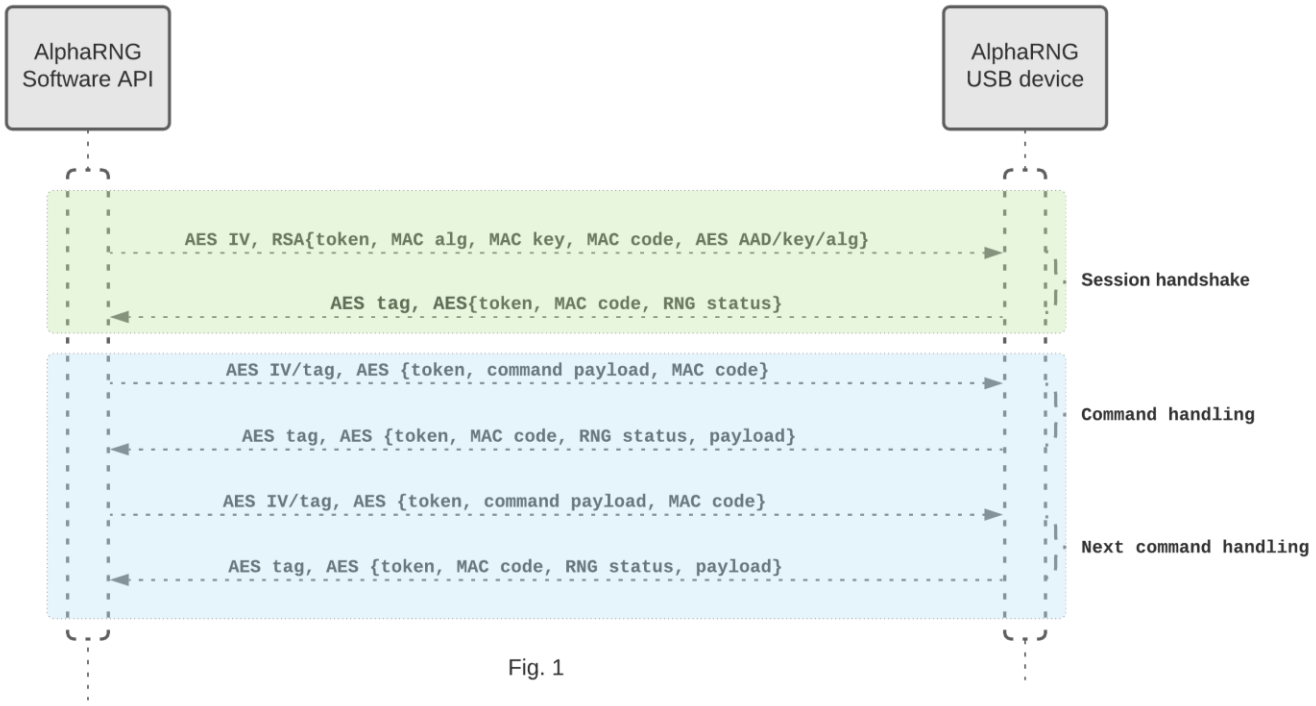

Fig. 1

The AlphaRNG software kit comes with two hard-coded public keys: one RSA 2048 bit key and one RSA 1024-bit key. Those two keys are used for establishing secure connections with any AlphaRNG device. In addition, each AlphaRNG is shipped with a custom public RSA 2048-bit key that can be exclusively used with one such device.

Supported systems

- Linux – data access provided through software API or utilities on Ubuntu, Red Hat, CentOS, and other Linux based x86-64 systems.
- macOS 10.15 Catalina and up (Intel and Apple Silicon) - data access provided through software API or utilities
- 64bit Windows 10 and 64bit Windows Server 2016/2019

Applications

- Live entropy source for seeding a Deterministic Random Bit Generator (DRBG)
- Cryptography
- Authentication
- Payment services
- Secure key generation
- Research (statistical sampling)
- Computer simulations
- Gaming and lotteries

Product Specifications

Product name	AlphaRNG
Interface	USB 2.0 high-speed interface with EMI filtering
Noise source	Two independent circuits based on avalanche breakdown effect in reversed-biased Zener diodes
Available asymmetric ciphers	RSA-2048, RSA-1024
Available symmetric ciphers	AES-256-GCM, AES-128-GCM
Available MAC algorithms	HMAC-SHA256, HMAC-SHA160, HMAC-MD5
Implemented statistical tests	Start-up and on-demand ‘Repetition Count’ and ‘Adaptive Proportion’
Continuous statistical tests	‘Repetition Count’ and ‘Adaptive Proportion’ implemented in the software kit
Health check test (HCT)	Start-up and on-demand Health diagnostics of both random noise sources
HCT validation	Hardware fault insertion
Data download speed	It varies depending on the cipher type and HMAC algorithm selected
Max data download speed	Up to 199 Mbps when used in non-secure mode
Min data download speed	3.4 Mbps when used with AES-256-GCM and HMAC-SHA256
Weight	Less than 22 grams (0.8 Oz)
Data connectivity and control interface	USB 2 high-speed interface with integrated EMI filtering and ESD protection
Device access	Locked, no debugging, AES encrypted firmware
Software Kit dependencies	OpenSSL 1.1+
Power supply	USB bus powered
Power consumption	220 mA in active mode, 90 mA when inactive
Dimensions	78mm * 23mm * 14mm
RoHS compliance	All parts and materials are RoHS compliant
Average EMF emission	Less than 1 μW/m² measured at the surface of the device

Operating Temperatures

- Maximum ambient temperature: 81°F (27° C). The connected device should be located at least 1 inch away from other USB devices in an area with a free or forced air flow circulation.

User Notes

- The AlphaRNG device can be plugged into one of the available USB 2.0 or 3.0 ports directly or by use of an USB 2.0 extension cable (extension cable not included).
- Do not immerse this product in any liquid or expose it to direct sunlight or high temperature environment.
- A quick start guide, including hardware and software requirements, installation, and verification steps for Linux, Windows and macOS can be found online at the following web address: https://tectrolabs.com/docs/alpharng/quick-start/
- The software installation and configuration instructions can be found online at the following web address: https://tectrolabs.com/docs/alpharng/

One-Year Limited Warranty: TectroLabs offers a 1-year limited and an optional 3-year extended warranty on AlphaRNG. We will repair or replace any device that fails due to defect in materials or manufacturing.