

FIPS Mode Support

WB45NBT / Summit Linux

*Application Note**v3.0*

1 Scope

The WB45NBT, when paired with an Ezurio-approved FIPS-compliant software image, contains a FIPS 140-3 validated cryptographic module. As of the publication date of this document, the module is listed by the NIST Cryptographic Module Validation Program (CMVP) as certificate #5054. The authoritative source for validation status, certificate dates, sunset dates, security policy documents, and related validation information is the NIST CMVP certificate record. Current status should be verified directly through NIST. The platform and applications themselves are not FIPS certified. Rather, the platform contains a FIPS validated cryptographic module that has been validated on the platform. Some applications (for example, the supplicant and Wi-Fi driver) use the FIPS validated module when operating in FIPS modes.

1.1 Software configuration

Ezurio provides several software image default and customized configurations. Not all built software images are FIPS 140-3 compliant. Please coordinate with Ezurio support before deploying a WB45NBT platform to an environment that requires FIPS 140-3 compliance. The rest of the document assumes the use of a configuration based upon the EZ BSP reference `wb45n_legacy_fips_11_defconfig`.

1.2 NIST CMVP Reference

The NIST Cryptographic Module Validation Program (CMVP) is the authoritative source for FIPS validation status. Certificate status, validation dates, sunset dates, and associated security policy documents may change after publication of this document and should be verified directly through the NIST CMVP records. The following certificate numbers and links are provided for reference, and the security policy documents are available through the certificate records.

References

- NIST CMVP: <https://csrc.nist.gov/projects/cryptographic-module-validation-program>
- NIST CMVP Validated Modules Search: <https://csrc.nist.gov/projects/cryptographic-module-validation-program/validated-modules/search>
- NIST CMVP Certificate #5054: <https://csrc.nist.gov/projects/cryptographic-module-validation-program/certificate/5054>

2 FIPS Modes

The module supports two FIPS operating modes, selected with `fips-set` and applied after a reboot. The current mode can be read at any time with:

```
fips-set status.
```

Table 1: `fips-set` commands

Command	Result
<code>fips-set fips_wifi</code>	Enable FIPS Wi-Fi mode
<code>fips-set fips</code>	Enable FIPS mode
<code>fips-set unset</code>	Disable FIPS mode
<code>fips-set status</code>	Report current mode

`sdc_cli method` (enables `fips_wifi` mode):

```
# sdc_cli global set fips enable
Reboot required to activate fips mode

# sdc_cli global show fips
```

Four states are reported:

```
FIPS Mode: Disabled and Inactive
FIPS Mode: Inactive - Enabled on next start
FIPS Mode: Enabled and Active
FIPS Mode: Active - Disabled on next start
```

2.1.1 FIPS mode (`fips`)

The system's cryptographic services operate in a FIPS-approved state:

- System services and applications using OpenSSL operate with the standard default provider. Any application that requires FIPS-approved cryptography must explicitly activate the FIPS provider (via configuration or API); once activated, non-approved algorithms such as MD5, RC4, and DES fail within that process (see OpenSSL 3 FIPS Provider).
- The kernel's FIPS-validated cryptographic module is active.
- The module performs integrity self-tests of its software images at every boot; if an integrity check fails, the module halts and reboots rather than continuing in an unverified state.
- System random-number requests made through kernel RNG interfaces (`getrandom()`, `/dev/random`, and `/dev/urandom`) are serviced by the FIPS-validated Deterministic Random Bit Generator (CTR-DRBG using AES-256), which is seeded from a FIPS-validated entropy source. Applications that rely on these interfaces (e.g. SSH, TLS implementations, and the Wi-Fi supplicant) use this approved random-number generation path without application-specific changes. RNG-intensive operations may have lower performance than non-FIPS modes.

Wi-Fi network authentication is not affected in this mode; the supplicant continues to accept all supported authentication methods.

2.1.2 FIPS Wi-Fi mode (`fips_wifi`)

Everything in FIPS mode, plus the Wi-Fi supplicant and driver operate in a FIPS-approved state:

- Wireless enterprise authentication is restricted to FIPS-approved EAP methods: EAP-TLS, PEAP-GTC, PEAP-TLS, and EAP-TTLS (with GTC or PAP inner authentication). Methods relying on non-approved password protocols (MD5/CHAP/MSCHAP/MSCHAPv2) are rejected.
- TLS cipher suites used for wireless authentication are restricted to the NIST SP 800-52 Rev. 2 approved set.
- All Wi-Fi data encryption and decryption are moved out of the Wi-Fi firmware and performed by the FIPS-validated cryptographic module in the operating system (AES-128 CCMP).
 - Throughput may be reduced compared to non-FIPS operation, because every Wi-Fi data frame is processed by the host CPU instead of being encrypted on chip by the Wi-Fi firmware.

This mode is required when the wireless network itself must remain inside the FIPS boundary (for example, WPA2-Enterprise on a regulated network).

3 Supported Wireless Capabilities

In FIPS Wi-Fi mode, WB45NBT support is limited to Tables 2, 3 and 4

Table 2: Wi-Fi capabilities

Category	Supported
Ciphers	AES-CCMP-128
Key management	WPA2-EAP, WPA2-PSK
802.1x methods	EAP-TLS, PEAP-TLS, PEAP-GTC, TTLS-PAP, TTLS-GTC

4 Certificates

Table 3: Accepted formats

Certificate	Formats	Contents
User / client	PKCS#12 (.pfx / .p12), PEM (.pem)	User certificate and user encryption key
CA	PEM (.pem), DER (.der)	CA certificate

4.1 Encoding

OpenSSL default encodings for PKCS#12 and for encrypted PEM private keys are not FIPS approved and must be overridden.

Table 4: Required PKCS#12 options

Parameter	Required option
MAC	-macalg sha256
Certificate encryption	-certpbe aes-128-cbc
Private key encryption	-keypbe aes-128-cbc

Encode a PEM certificate and key as PKCS#12:

```
openssl pkcs12 -export -in user-cert-key.pem -out user.pfx \
-macalg sha256 -certpbe aes-128-cbc -keypbe aes-128-cbc \
-passin pass:pem-password -passout pass:password-to-use
```

Convert PKCS#12 to PEM with an AES-encrypted private key (-aes128 or -aes256):

```
openssl pkcs12 -in user-cert-key.pfx -out user-cert-key.pem \
-passin pass:pfx-password -aes128 -passout pass:password-to-use
```

4.2 Passphrase-protected key material

Passphrase-protected private keys (encrypted PEM files and PKCS#12/PFX bundles) must use a passphrase of 14-63 characters to be usable in FIPS mode. NIST requires 112 bits (14 bytes) of strength for a passphrase. A passphrase drawn from a restricted character set (e.g., ASCII) will require more than 14 characters to meet this requirement.

These keys are externally loaded SSPs. The NIST CMVP record for [Certificate #5054](#) states: "When operated in approved mode. No assurance of minimum security of SSPs (e.g., keys, bit strings) that are externally loaded, or of SSPs established with externally loaded SSPs." The operator is responsible for the strength and protection of keys and passphrases before they are loaded.

For example, to generate a random ASCII passphrase with 112 bits of strength (14 random bytes, Base64-encoded to limit the output to usable characters):

```
openssl rand -provider fips -propquery fips=yes -base64 14
```

5 OpenSSL 3 FIPS Provider

External applications must load the OpenSSL 3 FIPS provider explicitly and request FIPS-approved algorithms. The `openssl.cnf` file does not load any providers by default.

For command line usage of the `openssl` utility in fips mode, pass `-provider fips -propquery fips=yes`:

```
openssl dgst -provider fips -propquery fips=yes -sha1 /etc/hosts
```

Non-approved algorithms fail. MD5, for example:

```
openssl dgst -provider fips -propquery fips=yes -md5 /etc/hosts
```

In application code, do both of the following (the Summit supplicant is the reference implementation):

1. Load the FIPS provider and default to FIPS algorithms. Reference: `openssl_load_fips_provider()` in `src/crypto/crypto_openssl.c`.
2. Create SSL contexts with the `fips=yes` property:

```
ssl = SSL_CTX_new_ex(NULL, "fips=yes", SSLv23_method());
```

6 Revision History

Version	Date	Notes	Contributor(s)	Approver
1.0	21 Apr 2016	Initial Release with WB50NBT added		Doug Smith
2.0	15 May 2025	Ezurio rebranding	Sue White	Dave Drogowski
3.0	24 Sep 2026	Updated for FIPS 140-3 (CMVP certificate #5054); WB45NBT only	John Nosky Doug Smith	Dave Drogowski