# CSPN Security Target

## HP Endpoint Security Controller (EpSC) Platform Hardware Root of Trust NPCX499HA1 and NPCD324HA0

Revision Record

| Revision | Date | Comments |
|---|---|---|
| 1.0 | 2023-08-14 | First Version |
| 1.1 | 2024-04-04 | Updated Part Number, Chip Revision and ROM Version |
| 1.2 | 2024-06-20 | Minor revisions after ANSSI input |

# Table of Contents


1. Introduction ..... 5
   1.1 Document Context ..... 5
   1.2 Product identification ..... 5
2. TOE Summary Description ..... 6
   2.1 Identification ..... 6
   2.2 Features ..... 6
      2.2.1 Firmware Authentication ..... 6
      2.2.2 Verifiable HP Endpoint Security Controller Identity and Attestation Keys ..... 6
   2.3 Architecture ..... 6
   2.4 Assumptions ..... 8
3. Security problem definition ..... 9
   3.1 Assets ..... 9
   3.2 Threats ..... 11
      3.2.1 Replacement of Embedded Software Bootloader ..... 11
      3.2.2 Replacement of Embedded Software Bootloader Extension ..... 11
      3.2.3 Code Signing Subversion ..... 11
      3.2.4 Policy Settings Alteration ..... 11
      3.2.5 Crypto Attack ..... 11
      3.2.6 Confidential Assets Access ..... 11
   3.3 Security Functions ..... 12
      3.3.1 ROM based Integrity Protection ..... 12
      3.3.2 ROM Extension Integrity Protection ..... 12
      3.3.3 Code Signature Verification ..... 12
      3.3.4 OTP Based Integrity Protection ..... 12
      3.3.5 Confidentiality Protection ..... 12
      3.3.6 Availability of Code Authentication Mechanism ..... 12
   3.4 Cryptographic Properties ..... 12
      3.4.1 Crypto HW ..... 12
      3.4.2 Key Sizes ..... 13
      3.4.3 HP Public Key ..... 13

# List of Figures

Figure 1 Target of Evaluation (TOE) Overview ..... 7

# List of Tables

Table 1 Evaluated Product Identification ..... 5

# 1. Introduction

## 1.1 Document Context

This document is intended to define the target of evaluation used for the ANSSI[1] defined CSPN[2] security certification framework[3] for the "HP Endpoint Security Controller (EpSC) Platform Hardware Root of Trust – NPCX499HA1 and NPCD324HA0" implemented by HP Inc.

## 1.2 Product identification

| | |
|---|---|
| Editor | HP Inc.<br>1501 Page Mill Road, Palo Alto, CA 94304<br>P.C. 94304-1112<br>Palo Alto<br>United States |
| Developer | Nuvoton Technology Israel Ltd.<br>P.O. Box 12347<br>Herzliya B, 4673324<br>Israel |
| Link | http://www.hp.com |
| Products | HP Endpoint Security Controller (EpSC):<br>NPCX499HA1 die in a 144-Pin Very Thin Profile Fine-Pitch Ball Grid Array (VFBGA)<br>NPCD324HA0 die in a 128-Pin Low-Profile Plastic Quad Flat Package (LQFP-EP) |
| Part Number | NPCX499HA1BX<br>NPCD324HA0DX |
| Chip Revision | B2h |
| ROM Version | 21.22.0.3 |
| ROM Extension Version | 21.21.1.80 |
| Products Category | Hardware and embedded software |

*Table 1 Evaluated Product Identification*

[1] Agence nationale de la sécurité et des systèmes d'information
[2] Certification de Sécurité de Premier Niveau
[3] ANSSI-CSPN-CER-P-01_v5.0

# 2. TOE Summary Description

## 2.1 Identification

The Target of Evaluation (TOE) is the HP Endpoint Security Controller (EpSC) Platform Hardware Root of Trust hardware component which is core to the security and resilience properties of HP platform security capabilities like HP Sure Start, HP Sure Admin, HP Sure Recover, or HP Tamper Lock for example.

## 2.2 Features

### 2.2.1 Firmware Authentication

The TOE consists of hardware and software mechanisms designed to prevent unauthorized execution of low-level system firmware in a computing platform including cryptographic algorithms that are not weakened by quantum computing. This TOE is typically used in HP platform products to ensure that unauthorized modifications to system firmware (BIOS, and other critical system firmware) do not go undetected and that they can be recovered automatically. Hardware enforced firmware authentication is the foundation of the HP platform security features mentioned above.

### 2.2.2 Verifiable HP Endpoint Security Controller Identity and Attestation Keys

The TOE generates cryptographic key pairs in a manner that can be trusted by remote parties to be uniquely created and protected by a given EpSC component, running in the right state, and with the right firmware at the time of use. Digital signatures created by these keys can be provably linked to a specific device's EpSC running under such a trusted configuration. The security architecture is based on the Trusted Computing Group's Device Identity Composition Engine (DICE) standards[4] to enable a unique EpSC Device ID (DevID) key pair to be available to trusted, immutable EpSC firmware boot code. The DevID private key acts as an embedded certificate authority (ECA), which is used to certify further keys generated and protected by the EpSC immutable firmware that can be used for authentication, attestation, or both when a given version of the mutable firmware is loaded and running on the EpSC. Evidence of this protection is further demonstrated in the public key certificates signed by the EpSC DevID.

## 2.3 Architecture

The TOE consists of the following components:

- A microcontroller with built-in cryptographic hardware functionality
- Software stored in ROM (Read-Only-Memory), inside the microcontroller
- Embedded SRAM (Static Random Access Memory), inside the microcontroller
- Embedded Flash memory, inside the microcontroller
- One Time Programmable (OTP) memory (fused), inside the microcontroller
- The following Interfaces
    - A power line to receive power from the computer mainboard
    - A control line to connect to the chipset/CPU reset line on the computer mainboard
    - A memory access line to an External (or Internal) Flash Memory component which contains firmware to be loaded by the microcontroller

[4] https://trustedcomputinggroup.org/work-groups/dice-architectures/

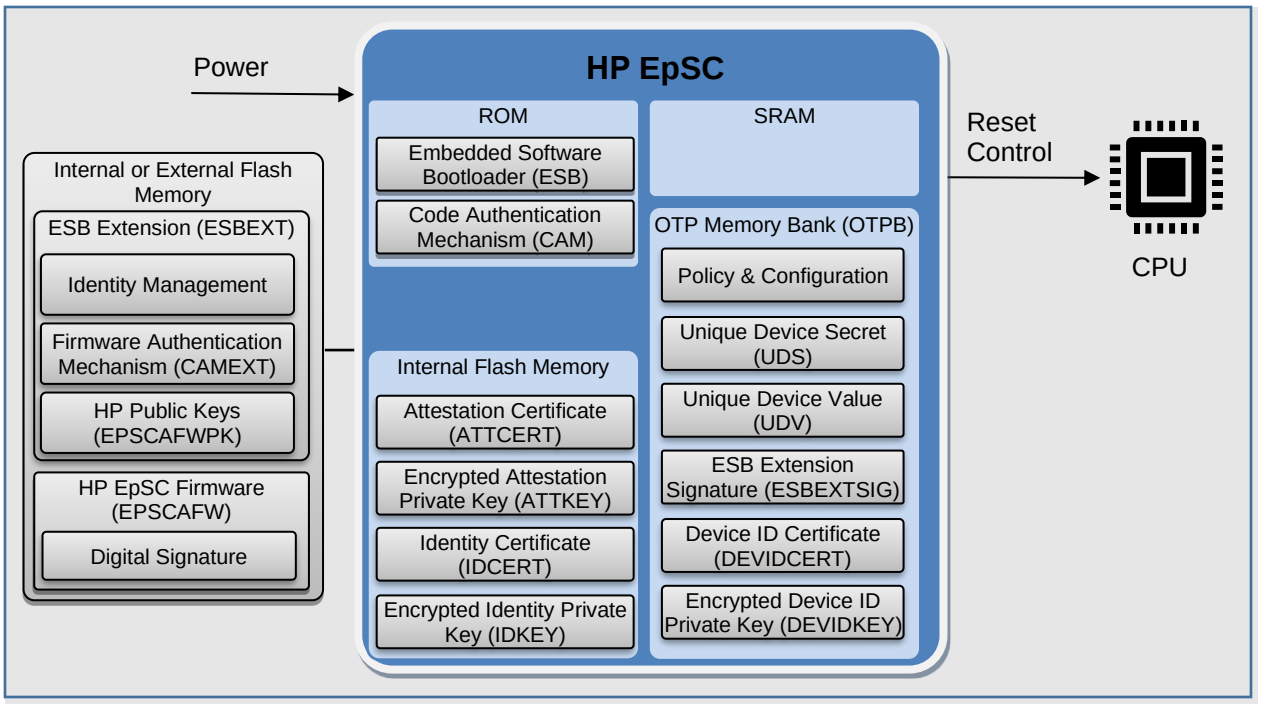



*Figure 1 Target of Evaluation (TOE) Overview*

The TOE acts as a platform hardware root of trust on the computer system where it is integrated. It is powered first when power is applied to the computer system, and it will ensure that only firmware appropriately authorized by the manufacturer will load into the HP EpSC before continuing with boot. This process is built of two steps:

1- First, ROM code (referred in the document as Embedded Software Bootloader – ESB), loads and verifies ROM extension firmware (referred in the document as Embedded Software Bootloader Extension – ESBEXT). ESB verifies the integrity and authenticity of the ESBXT by first computing an HMAC over the ROM extension firmware and configuration data in the One Time Programmable (OTP) internal memory. It uses an embedded Unique Device Value (UDV) as a key, and compares the result against an expected HMAC signature value for verification. Both the UDV and expected signature values are securely stored in OTP during manufacturing. This makes the ESBEXT functionally immutable as only one trusted firmware image will verify successfully.

2- Second, the ROM extension firmware loads and verifies HP EpSC firmware (referred to in the document as EpSC Authorized Firmware – EPSCAFW). Two Public Key Infrastructure (PKI) schemes are used (RSA, and a Hierarchical Signature System (HSS)) by the extension ROM to verify the digital signature of the HP EpSC firmware. HP Public Keys used for the verification are within the ROM extension firmware image.

   The ESBEXT authenticates the EPSCAFW in two steps. First, the firmware image header (holding header version data, firmware image authentication public keys and some configuration data) is authenticated. Then the ESBEXT, using the public key, authenticates the rest of the firmware image using the listed configuration. The header allows replacing the firmware image authentication keys with a new set and uses the header version data to revoke older keys if required.

The TOE then becomes the Platform Hardware Root of Trust for further functionality that will run on the HP EpSC, in support of platform security capability in HP PCs, such as HP Sure Start, HP Sure Admin, HP Sure Recover, HP Tamper Lock and other features.

An EpSC Device ID key pair is first generated from a Random Bit Generator (RBG) during manufacturing. An encryption key is then derived from Unique Device Secret (UDS), OTP configuration, and ROM extension firmware measurement (digest), and is used to encrypt the Device ID private key. The encrypted private key is then securely stored in OTP. The private key can only be decrypted on subsequent boots if the same firmware, OTP configuration, and UDS are present. Changes to these elements in the EpSC would render the EpSC ID private key inaccessible. Additionally, authentication and attestation key pairs are generated from RBG and their private keys are encrypted using keys derived from EpSC device state and HP EpSC firmware measurements.

This solution provides assurances to the computer user and remote parties that the system will only boot with validated firmware in the TOE and that specific identity and attestation keys are assured to represent that hardware and its firmware configuration. This is designed to occur without requiring any action on the part of the user.

## 2.4 Assumptions

The overall solution assumes that cryptographic materials used to digitally sign authorized firmware updates are generated in a trusted environment controlled by the manufacturer (HP), whereby private portions of those materials never leave that trusted environment.
It is also assumed that the complete chain from development to production is trusted.
The solution is typically deployed in mass market end-user devices, such as PCs or printers.

# 3. Security problem definition

## 3.1 Assets

- An Embedded Software Bootloader (A_ESB): This software is used to boot the TOE HP EpSC when power is applied, and the solution starts executing. Its integrity must be protected to ensure proper operation of the TOE.
    - The A_ESB loads the A_ESBEXT firmware from external or internal flash memory into internal RAM and uses the A_CAM to verify the A_ESBEXT integrity.

    - The A_ESB uses the A_UDS and measurements of A_ESBEXT firmware and configuration in A_OTPB as inputs to a one-way function (HMAC) to derive an intermediate value, described in the TCG DICE architecture as the Compound Device Identifier (CDI). The A_ESB passes the CDI to the A_ESBEXT.

- Code Authentication Mechanism (A_CAM): The A_CAM will validate the integrity of the A_ESBEXT firmware and execution environment by performing an HMAC over the firmware along with the policy and configuration data fused in the A_OTPB with A_UDV as key and comparing the result to A_ESBEXTSIG.
- One-Time-Programmable memory bank (A_OTPB): OTP bits within the TOE that are used by the A_CAM, A_CAMEXT, A_ESB and A_ESBEXT to control both code authentication policy parameters and secrets for managing device identity credentials, and that can retain their state in a reliable manner that is not reversible.
- An Embedded Software Bootloader Extension (A_ESBEXT): This firmware serves as an extension to the ROM code bootloader. It is used to generate and manage the EpSC Device ID, and the related attestation and Identity key pairs and certificates described in the TCG DICE standards. The A_ESBEXT uses the CDI provided by the A_ESB to derive secret keys for decrypting the identity and attestation private keys. The A_ESBEXT integrity is verified by A_CAM before subsequent execution by the TOE microcontroller. The A_ESBEXT code is functionally immutable because the A_ESBEXTSIG and A_UDV are fixed and locked in A_OTPB during the TOE manufacturing for the lifetime of the TOE.
- Code Authentication Mechanism Extension (A_CAMEXT): The A_CAMEXT will validate the integrity and authenticity of the firmware code (A_EPSCAFW) loaded by the A_ESBEXT boot loader.
- HP EpSC Authorized Firmware (A_EPSCAFW): Authorized HP EpSC firmware code, including a digital signature issued by HP, which is intended to be loaded into SRAM by A_ESBEXT and successfully validated using the A_CAMEXT, according to A_OTPB policy settings, before subsequent execution by the TOE microcontroller.
- HP EpSC Authorized Firmware Public Keys (A_EPSCAFWPK): The Public keys stored within A_ESBEXT Embedded Software Bootloader Extension used by A_CAMEXT to authorize HP EpSC firmware code.
- Unique Device Secret (A_UDS): The Unique Device Secret is a statistically unique, device-specific, secret value stored in the A_OTPB. The A_UDS with A_ESBEXT environment (firmware, configuration data, authorization, and execution policies plus lifecycle state) are used to derive the CDI.
- Unique Device Value (A_UDV): The Unique Device Value is a statistically unique, device-specific, value. The A_UDV is used as the key for the HMAC function to verify the integrity of the A_ESBEXT environment. The A_UDV is stored in the A_OTPB.

- An Embedded Software Bootloader Extension Signature (A_ESBEXTSIG): The A_CAM uses the A_ESBEXTSIG to verify the integrity of the A_ESBEXT firmware and the policy and configuration data fused in the A_OTPB. The A_ESBEXTSIG is created during manufacturing and stored in the A_OTPB.
- EpSC Device ID Key (A_DEVIDKEY) and Certificate (A_DEVIDCERT): The EpSC Device ID Key is an asymmetric key pair, long-lived identifier for a device, used as an ECA by the A_ESBEXT to issue (sign) EpSC Attestation (A_ATTCERT) and Identity (A_IDCERT) certificates of the keys generated for the A_EPSCAFW. The A_ESBEXT generates the EpSC Device ID Key, encrypts the private key with a key derived from the CDI and stores the encrypted private key in the A_OTPB. The A_ESBEXT generates a self-signed EpSC Device ID Certificate (A_DEVIDCERT) and stores it in the A_OTPB.
- EpSC Attestation Key (A_ATTKEY) and Certificate (A_ATTCERT): The EpSC Attestation Key is an asymmetric key used to sign attestation evidence (data structures that are known to the A_EPSCAFW). The A_ESBEXT generates the EpSC Attestation Key, and encrypts the private key with a key derived from the CDI, A_EPSCAFW code measurement and the OTP configuration. The A_ESBEXT signs the EpSC Attestation Certificate using the EpSC Device ID Key. The EpSC encrypted Attestation private key and its certificate are stored in EpSC internal non-volatile memory.
- EpSC Identity Key (A_IDKEY) and Certificate (A_IDCERT): The EpSC Identity Key is an asymmetric key used for signing authentication challenges (i.e., statements or challenge received from external source). The A_ESBEXT generates the EpSC Identity Key and encrypts the private key with a key derived from the CDI, A_EPSCAFW code measurement, and OTP configuration. The A_ESBEXT signs the EpSC Identity Certificate using the EpSC Device ID Key. The encrypted EpSC Identity private key and its certificate are stored in EpSC internal non-volatile memory.

The security requirements for the critical assets are the following:

| **Assets** | Availability | Confidentiality | Integrity | Authenticity |
|---|---|---|---|---|
| An Embedded Software Bootloader (A_ESB) | X | | X | |
| Code Authentication Mechanism (A_CAM) | X | | X | |
| One-Time-Programmable memory bank (A_OTPB) | X | X | X | |
| An Embedded Software Bootloader Extension (A_ESBEXT) | X | | X | X |
| Code Authentication Mechanism Extension (A_CAMEXT) | X | | X | X |
| HP EpSC Authorized Firmware (A_EPSCAFW) | X | | X | X |
| HP EpSC Authorized Firmware Public Keys (A_EPSCAFWPK) | X | | X | X |
| Unique Device Secret (A_UDS) | X | X | X | |
| Unique Device Value (A_UDV) | X | | X | |
| An Embedded Software Bootloader Extension Signature (A_ESBEXTSIG) | X | | X | |
| EpSC Device ID Certificate (A_DEVIDCERT) | X | | X | X |
| EpSC Attestation Certificate (A_ATTCERT) | X | | X | X |
| EpSC Identity Certificate (A_IDCERT) | X | | X | X |
| EpSC Device ID Key (A_DEVIDKEY) | X | X | X | X |

| Assets | Availability | Confidentiality | Integrity | Authenticity |
|---|---|---|---|---|
| EpSC Attestation Key (A_ATTKEY) | X | X | X | X |
| EpSC Identity Key (A_IDKEY) | X | X | X | X |

## 3.2 Threats

Potential attackers may have local physical control of the device that incorporates the TOE, or they may be attempting to attack the TOE remotely. Attacks may involve software or hardware methods, or a combination thereof.

### 3.2.1 Replacement of Embedded Software Bootloader

An attacker manages to modify the microcontroller Embedded Software Bootloader (A_ESB) inside the microcontroller ROM and to maliciously load unauthorized firmware and execute it in the HP EpSC.

### 3.2.2 Replacement of Embedded Software Bootloader Extension

An attacker manages to modify the microcontroller Embedded Software Bootloader (A_ESBEXT) inside the microcontroller Flash Memory, for example to bypass the code authentication mechanism (A_CAMEXT), to ultimately load unauthorized firmware and execute it in the HP EpSC.

### 3.2.3 Code Signing Subversion

An attacker manages to modify the Target Firmware data in external Flash Memory in such a way that will bypass the code signing verification mechanism (A_CAMEXT), and result in the loading of unauthorized firmware to run in the HP EpSC.

### 3.2.4 Policy Settings Alteration

An attacker manages to manipulate A_OTPB policy control bits in such a way that results in rolling back state to authorize the loading of firmware (by A_ESB) for which authorization had been previously revoked, and run it in the HP EpSC.

### 3.2.5 Crypto Attack

An attacker manages to attack the cryptographic algorithm implementation used in the TOE in order to forge a digital signature that will appear legitimate and resulting in the loading unauthorized firmware in the HP EpSC or compromise any of the confidential assets or part of them (e.g., A_UDS, A_DEVIDCERT, A_ATTCERT or A_IDCERT) and abuse the EpSC cryptographic keys.

### 3.2.6 Confidential Assets Access

An attacker manages to access confidential assets in the A_OTPB (i.e., A_UDS), or in the internal SRAM (i.e., A_DEVIDCERT, A_ATTCERT and A_IDCERT private keys) when used.

## 3.3 Security Functions

### 3.3.1 ROM based Integrity Protection

The TOE contains a true Read-Only-Memory which protects the integrity of the A_ESB Embedded Software Bootloader (inclusive of the A_CAM) against any modification.

### 3.3.2 ROM Extension Integrity Protection

The TOE bootloader contains a code authentication (A_CAM) mechanism used to verify the integrity of the ROM Extension code it loads from external flash memory before it is executed by the TOE. The A_CAM uses HMAC with A_UDV as key to verify the signature A_ESBEXTSIG stored in A_OTPB. The A_UDV is set to random value, stored and protected from modification in the A_OTPB during TOE manufacturing. The A_CAM protects the integrity of the ROM Extension (A_ROMEXT), inclusive of the A_CAMEXT and the HP EpSC Authorized Firmware Public Key (A_EPSCAFWPK) against any modification.

### 3.3.3 Code Signature Verification

The TOE ROM Extension contains a code authentication (A_CAMEXT) mechanism used to verify both the integrity and the authenticity of a digital signature of the target firmware it loads from external flash memory before it is executed by the TOE.

### 3.3.4 OTP Based Integrity Protection

The TOE uses a One-Time-Programmable memory Bank (A_OTPB) to record the locking of specific TOE security parameters controlling the A_ESB for key authorization policy, A_UDV value and A_ESBEXTSIG value that are not reversible.

### 3.3.5 Confidentiality Protection

The TOE protects confidential assets by either:

- Erasing volatile secrets (maintained in RAM e.g., asymmetric private keys) generated by a given TOE software layer, after use, before passing control to the next software layer.
- Disable read/write access and latch setting till the next TOE Power On Reset (POR), for persistent secrets (e.g., A_UDS held in A_OTPB)

### 3.3.6 Availability of Code Authentication Mechanism

The TOE system design ensures that the EpSC is powered first and that the state of the component upon starting execution prevents the main CPU from starting execution, thus protecting against the possibility of an external agent attempting to execute code on the CPU in advance of A_CAMEXT execution.

## 3.4 Cryptographic Properties

### 3.4.1 Crypto HW

The A_CAM uses a hardware implementation of cryptographic HMAC verification, which will be used by the A_ESB Embedded Software Bootloader to reliably authenticate A_ESBEXT Firmware that is loaded from external or internal flash memory into A_SRAM before it is executed by the HP EpSC.
The A_CAMEXT uses a hardware implementation of cryptographic HSS and RSA signature verifications with a hybrid mechanism, which will be used by the A_ESBEXT Embedded Software Bootloader Extension to reliably authenticate A_EPSCAFW Firmware that is loaded from external or internal flash memory into A_SRAM before it is executed by the HP EpSC. The security tested by the CSPN certification has no dependency on the additional HSS signature verification process. Moreover, PQC security is not tested by the CSPN certification.

The A_ESB uses a hardware implementation of cryptographic HMAC based KDF, which will be used to generate the CDI. The A_ESBEXT uses the CDI to derive a sealing key to protect the A_DEVIDKEY private key, and combines the CDI with measurements of the A_EPSCAFW and execution environment (i.e., configuration data, authorization, and execution policies) to derive the sealing key to protect the A_ATTKEY and A_IDKEY private keys.
The A_ESBEXT uses a hardware implementation of cryptographic ECDSA key generation to generate the A_DEVIDKEY, A_ATTKEY and A_IDCERT keys and use ECDSA digital signature to generate their respective certificates (A_DEVIDCERT, A_IDCERT, A_ATTCERT) according to ANSI X9.62, using Curve P-384.
The A_ESBEXT uses a hardware implementation of cryptographic GCM block cipher encryption and decryption functions to protect access to A_DEVIDKEY, A_ATTKEY and A_IDKEY private keys.

### 3.4.2 Key Sizes

The A_CAM is designed to verify HMAC digital signature, using SHA-384 hashing, according to the RFC 2104.
The A_CAMEXT is designed to verify RSA and HSS digital signatures:

- RSA key size is 3072 bit and SHA-384 hashing according to the PKCS #1 v2.1 signature schemes RSASSA-PSS respectively
- HSS public key size 480 bit, using SHA-256, L (Level) 1, W (Winternitz parameter) 2 and H (Tree Height) 20 according to RFC8554

The A_CAMEXT is designed to generate ECDSA P-384 keys and signatures, using SHA-384 hashing according to FIPS PUB 186-4, using Curve P-384.

### 3.4.3 HP Public Key

The HP RSA 3072 and HSS 480 Public Keys (A_EPSCAFWPK) used by A_CAMEXT to verify the digital signature for the HP EpSC firmware are stored as part of the fixed bootloader extension (A_ESBEXT).