TROPIC01

ODN_TR01_app_003

Device Identity and PKI Application Note

Version: 1.3

Abstract

This document describes TROPIC01 device identity provided by the X.509 chip certificate and the Tropic Square Public Key Infrastructure. This should enable developers to integrate the TROPIC01 securely into their products, namely through the factory provisioning systems.

February 5, 2026

tropic square

Contents

Glossary

- **API** – Application Programming Interface
- **ASN.1** – Abstract Syntax Notation One (as defined by [7])
- **BER** – Basic Encoding Rules (in the context of ASN.1)
- **CA** – Certification (Certificate) Authority (in the context of PKI)
- **CBOR** – Concise Binary Object Representation (as defined by [10])
- **COSE** – CBOR Object Signing and Encryption (as defined by [11])
- **CRL** – Certificate Revocation List
- **DER** – Distinguished Encoding Rules (in the context of ASN.1)
- **FW** – Firmware
- **HSM** – Hardware Security Module
- **HW** – Hardware
- **IETF** – Internet Engineering Task Force
- **ITU** – International Telecommunication Union (standardization body)
- **MCU** – MicroController Unit
- **Noise** – The Noise Protocol Framework as defined in [9]
- **OTP** – One-Time Programmable (memory)
- **PKI** – Public Key Infrastructure
- **P/N** – Part Number
- **RFC** – Request for Comments (in the context of IETF standardization)
- **RFU** – Reserved for Future Use
- **ROM** – Read-Only Memory
- **RoT** – Root of Trust
- **SoC** – System on Chip

- **S/N** – Serial Number
- **TLV** – Tag-Length-Value structure (in the context of ASN.1)
- **URI** – Uniform Resource Identifier (in the context of the Internet's World Wide Web)
- **UTC** – Coordinated Universal Time (as defined by ITU)
- **X.509** – ITU specification X.509 defining PKI [6]

1 Introduction

TROPIC01 is an openly auditable secure element that comes with its own unique cryptographic identity in the form of secure channel key pair and a certificate. The certificate is issued by Tropic Square PKI which provides framework for verifying the origin of each TROPIC01 chip ever produced.

The assurance of TROPIC01 genuine origin is an essential mitigation of supply chain attacks. The layered security model of the systems embedding secure elements assumes the security of the final product depends on all the underlying layers:

- Layer 0: Auditable HW design
- Layer 1: Auditable Bootloader (ROM) and Application Firmwares
- Layer 2: Cryptographically backed (verifiable) unique identity of system's RoT
- Layer 3: Customer-managed device identity backed by the system's RoT
- Layer 4: Customer-specific application security integrating with system's RoT (secure boot, user local authentication, remote admin authentication etc.)
- Layer 5: The actual application use cases and business logic of the product (access control of the doorlocks, data measurement and upload to mobile/cloud for smart sensors etc.)

This application note should help integrators with covering "Layer 2" of the security model outlined above. Specifically, it covers the following aspects of TROPIC01 identity:

- Chip Serial Number (S/N)
- Other chip identity metadata such as P/N
- Chip Secure Channel certificate
- Tropic Square PKI for issuing unique TROPIC01 certificates
- Recommended TROPIC01 certificate validation for integrators

Intended audience of this document:

- Embedded system developers developing firmware code which is interfacing with TROPIC01.
- Manufacturing engineering teams responsible for TROPIC01 provisioning.
- Supply chain engineers verifying TROPIC01 origin.

2 TROPIC01 unique identifiers

Each TROPIC01 is issued by Tropic Square with unique S/N. The S/N is present within two data structures provisioned to TROPIC01 in the factory:

- CHIP_ID
- X.509 Certificate Store (chip certificate)

Both data structures are stored in OTP memory, therefore they cannot be changed. To read both structures, the Host MCU issues ***Get_Info_Req*** L2 Request.

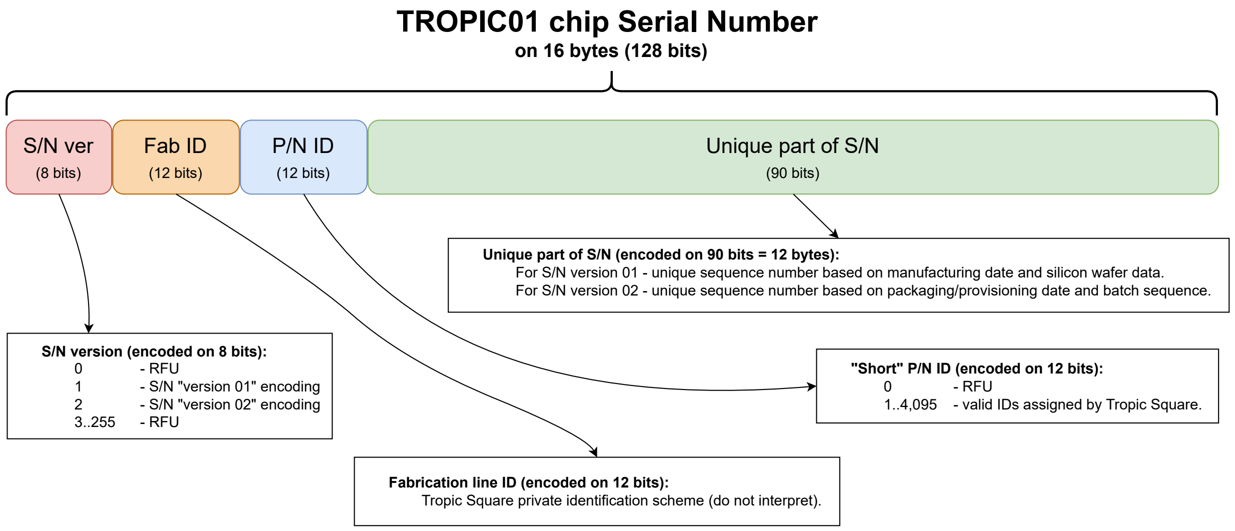
2.1 S/N structure

Tropic Square issues TROPIC01 with two versions of S/N:

- S/N version 01
- S/N version 02

All versions of S/N encoding have the length of 16 bytes (128 bits). This value (unsigned integer representation of the byte array) is mirrored in the chip certificate (refer to section 3.4). The first byte (8 bits) of the S/N denotes the version. The remaining 15 bytes are version specific.

Following diagram presents the S/N version 01 and 02 structure:



Unique part of S/N (encoded on 90 bits = 12 bytes):
For S/N version 01 - unique sequence number based on manufacturing date and silicon wafer data.
For S/N version 02 - unique sequence number based on packaging/provisioning date and batch sequence.

S/N version (encoded on 8 bits):
0 - RFU
1 - S/N "version 01" encoding
2 - S/N "version 02" encoding
3..255 - RFU

"Short" P/N ID (encoded on 12 bits):
0 - RFU
1..4,095 - valid IDs assigned by Tropic Square.

Fabrication line ID (encoded on 12 bits):
Tropic Square private identification scheme (do not interpret).

Figure 1: TROPIC01 S/N version 01 and 02 structure

The 12-bit ("short") P/N notation is linked to "full" P/N in Tropic Square's catalogue. The "full" P/N can be obtained either from the **CHIP_ID** structure (described in section 2.2) or from [3] distributed through `https://github.com/tropicsquare/tropic01`.

2.2 CHIP_ID structure

The CHIP_ID field serves as "public manufacturer data" area. It contains the following information:

- CHIP_ID structure version
- S/N (refer to section 2.1 for more details).
- P/N ("short" and "full" length formats).
- Package type
- Silicon revision
- ROM/OTP data version (Tropic Square private formatting)
- Product factory test data (Tropic Square private formatting)
- Provisioning tracking data (Tropic Square private formatting)

Following table describes detailed encoding of the current CHIP_ID structure v1 for TROPIC01 ("big endian" encoded 128-byte array):

Item	Size (bits)	Description
CHIP_ID structure versioning (32 bits).		
CHIP_ID version	32	Chip ID structure version on 4 bytes, each interpreted as unsigned integer value 0..255. Example encoding: v1.2.3.4 = 0x01, 0x02, 0x03, 0x04.
Wafer level test info (128 bits).		
Factory Level Chip Info	128	Tropic Square private format (do not interpret).
Manufacturing level test info (128 bits).		
Function Test Info	64	Tropic Square private format (do not interpret).
Silicon Revision ID	32	ASCII encoded string value defined by Tropic Square. Example: ’ACAB’ = 0x41434142.
Chip Package Type ID	16	Chip Package Type ID defined by Tropic Square.
RFU	16	Padding for memory alignment, filled with 0xFFFF data (do not interpret).
Provisioning info (128 bits).		
Prov Info Version	8	Version of Provisioning Info structure (integer 0..255).
Fab ID	12	Tropic Square private format (do not interpret).
P/N ID	12	"Short" Part Number format as defined by Tropic Square.
Provisioning date	16	Tropic Square private format (do not interpret).
HSM version	32	Tropic Square private format (do not interpret).
Programmer version	32	Tropic Square private format (do not interpret).
RFU	16	Padding for memory alignment, filled with 0xFFFF data (do not interpret).
Serial Number (128 bits).		
S/N	128	Exact encoding as per description in section 2.1
"Full" Part Number string (128 bits).		
P/N length	8	Length of subsequent P/N data field, encoded as unsigned integer on single byte (values 0x00..0x0F = 0..15). Example: P/N length for ’TROPIC01-T001’ string = 0x0D (13).

Item	Size (bits)	Description
P/N data	120	P/N data encoded in ASCII, padded with 0xFF bytes. Example: 'TROPIC01-T001' = 0x54524F50494330312D54303031FFFF.
Provisioning Data version (160 bits).		
Provisioning Data version	160	Tropic Square private format (do not interpret).
CHIP_ID padding (192 bits), for alignment.		
Padding	192	Filled with 0xFF..FF data to align with 128-byte space dedicated for CHIP_ID in TROPIC01 memory map. (do not interpret).

Interpretation of selected sub-fields of CHIP_ID:

- P/N ID
  - "Short" form of P/N. It directly corresponds to the "Full" P/N string (1:1 equivalence).
  - Tropic Square is committed to keep the TROPIC01 P/N IDs within 12-bit range (up to 4,096 variants).
  - P/N ID is also present in the X.509 chip certificate through S/N field (refer to section 2.1 for more details).
  - The complete list of P/N IDs (and "full" P/Ns) can be found in [3].
- "Full" P/N string
  - ASCII (UTF-8) encoded string which is directly linked to order codes of TROPIC01.
  - "Full" P/N string of TROPIC01 has variable length 1 to 15 characters (bytes).
  - "Full" P/N string directly corresponds to the "short" for represented by P/N ID (1:1 equivalence).
  - The complete list "full" P/Ns (and "short" P/N IDs) can be found in [3].
- Chip Package Type ID
  - Chip Package Type ID is 2-byte value assigned by Tropic Square.

  - List of defined Chip Package Type IDs is published in [3].
  - Details about available TROPIC01 chip package types can be found in section 12 of [1].
- Silicon Revision ID
  - Silicon revision is encoded as 16-bit ID (assigned by Tropic Square).
  - Observing silicon revisions of the TROPIC01 product is important for tracking Errata and particular feature deviations.
  - One P/N always uses only one silicon revision. However one silicon revision may be used in multiple P/Ns.
  - For more details about TROPIC01 silicon revisions and their link to available P/Ns please refer to [3].

The CHIP_ID structure is readable by ***Get_Info_Req*** L2 Request. For more details refer to API specification [2].

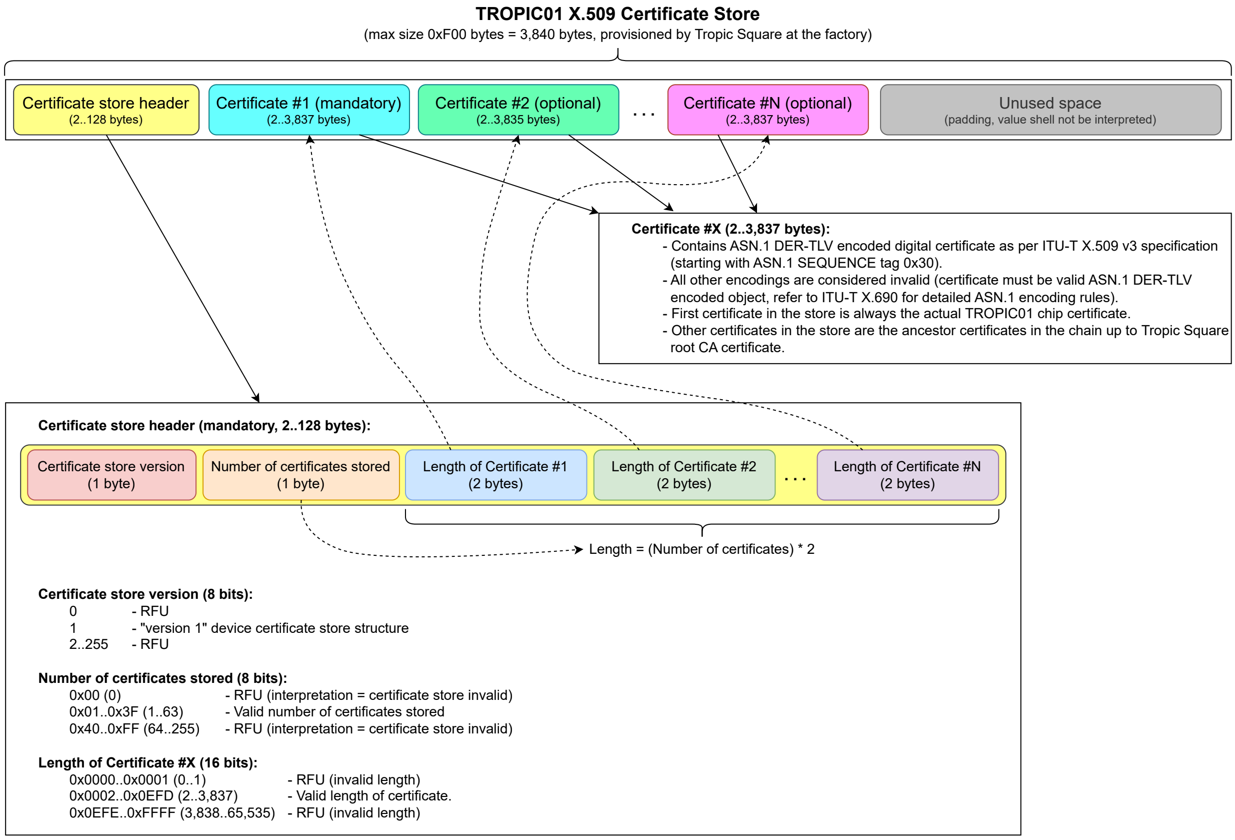
2.3 X.509 Certificate Store structure

Each TROPIC01 device contains X.509 Certificate Store data structure which consists of:

- Certificate Store header
- X.509 chip certificate (refer to 3.4)
- All ancestor certificates in the chip certiicate chain as issued by Tropic Square PKI (refer to section 3)
- Padding (optional)

The total size of all the memory allocated within TROPIC01 for X.509 Certificate Store (including proprietary header structure) is 0xF00 bytes (= 3,840 bytes). If the size of the actual valid certificates is less than the maximum allocated size, the trailing part of the Certificate Store memory space is padded by 0xFF.

Following diagram explains serialization (sequencing) of certificates inside the TROPIC01 X.509 Certificate Store. The header has a proprietary binary structure, the certificates are ASN.1 DER-TLV encoded as per [6] and [7]:


TROPIC01 X.509 Certificate Store
(max size 0xF00 bytes = 3,840 bytes, provisioned by Tropic Square at the factory)
Certificate store header
(2..128 bytes)
Certificate #1 (mandatory)
(2..3,837 bytes)
Certificate #2 (optional)
(2..3,835 bytes)
. . .
Certificate #N (optional)
(2..3,837 bytes)
Unused space
(padding, value shell not be interpreted)
Certificate store header (mandatory, 2..128 bytes):
Certificate store version
(1 byte)
Number of certificates stored
(1 byte)
Length of Certificate #1
(2 bytes)
Length of Certificate #2
(2 bytes)
. . .
Length of Certificate #N
(2 bytes)
Length = (Number of certificates) * 2

Certificate #X (2..3,837 bytes):

- Contains ASN.1 DER-TLV encoded digital certificate as per ITU-T X.509 v3 specification (starting with ASN.1 SEQUENCE tag 0x30).
- All other encodings are considered invalid (certificate must be valid ASN.1 DER-TLV encoded object, refer to ITU-T X.690 for detailed ASN.1 encoding rules).
- First certificate in the store is always the actual TROPIC01 chip certificate.
- Other certificates in the store are the ancestor certificates in the chain up to Tropic Square root CA certificate.

Certificate store version (8 bits):

0	- RFU
1	- "version 1" device certificate store structure
2..255	- RFU

Number of certificates stored (8 bits):

0x00 (0)	- RFU (interpretation = certificate store invalid)
0x01..0x3F (1..63)	- Valid number of certificates stored
0x40..0xFF (64..255)	- RFU (interpretation = certificate store invalid)

Length of Certificate #X (16 bits):

0x0000..0x0001 (0..1)	- RFU (invalid length)
0x0002..0x0EFD (2..3,837)	- Valid length of certificate.
0x0EFE..0xFFFF (3,838..65,535)	- RFU (invalid length)

Figure 2: TROPIC01 X.509 Certificate Store structure

Following table describes TROPIC01 X.509 Certificate Store binary encoding aligned with the specific PKI defined in section 3. It is complementary to the diagram above. The entire structure is encoded as "big endian" byte array (3,840 bytes long, including padding):

Item	Size (bits)	Description
Certificate store header (up to 128 bytes)		
Certificate store version	8	X.509 certificate store structure version on single byte, interpreted as unsigned integer value 0..255. Shall have value: 0x01.

Item	Size (bits)	Description
Number of certificates	8	Number of X.509 certificates stored in the structure. Interpreted as unsigned integer value 0..255. At least one certificate is mandatory (the chip certificate). Shall have value 4 for the current release of TROPIC01 and Tropic Square PKI infrastructure.
Length of certificate number 1	16	Length of ASN.1 DER-TLV encoded structure of the first certificate in the store. Interpreted as unsigned integer value 0..65,535.
Length of certificate number 2	16	Length of ASN.1 DER-TLV encoded structure of the second certificate in the store. Interpreted as unsigned integer value 0..65,535.
Length of certificate number 3	16	Length of ASN.1 DER-TLV encoded structure of the third certificate in the store. Interpreted as unsigned integer value 0..65,535.
Length of certificate number 4	16	Length of ASN.1 DER-TLV encoded structure of the fourth certificate in the store. Interpreted as unsigned integer value 0..65,535.
Certificate number 1 (variable length), ASN.1 DER-TLV encoding.		
TROPIC01 chip (device) certificate	Variable	ASN.1 DER-TLV encoded device certificate as per subsection 3.4.
Certificate number 2 (variable length), ASN.1 DER-TLV encoding.		
TROPIC01 "product level" CA certificate	Variable	ASN.1 DER-TLV encoded device certificate as per subsection 3.3.
Certificate number 3 (variable length), ASN.1 DER-TLV encoding.		
TROPIC01 P/N level (group) CA certificate	Variable	ASN.1 DER-TLV encoded device certificate as per subsection 3.2.
Certificate number 4 (variable length), ASN.1 DER-TLV encoding.		
Tropic Square root CA certificate	Variable	ASN.1 DER-TLV encoded device certificate as per subsection 3.1.
Certificate store padding (variable length), up to the end of memory space allocated.		
Padding	Variable	Filled with 0xFF..FF data to align with 3,840-byte space dedicated for X.509 certificate store in TROPIC01 memory map (do not interpret).

The X.509 Certificate Store is readable by ***Get_Info_Req*** L2 Request. This command allows retrieving parts of the Certificate Store, so depending on the validation scope the host may choose to read only header and the first certificate (chip certificate) or the entire structure. For more details refer to API specification [2].

3 TROPIC01 certificate and issuing PKI

Following diagram summarizes the Tropic Square PKI hierarchy in terms of certificate chain and different subsystems (Certification Authority “perimeters”) for TROPIC01 chip certificate issuance:

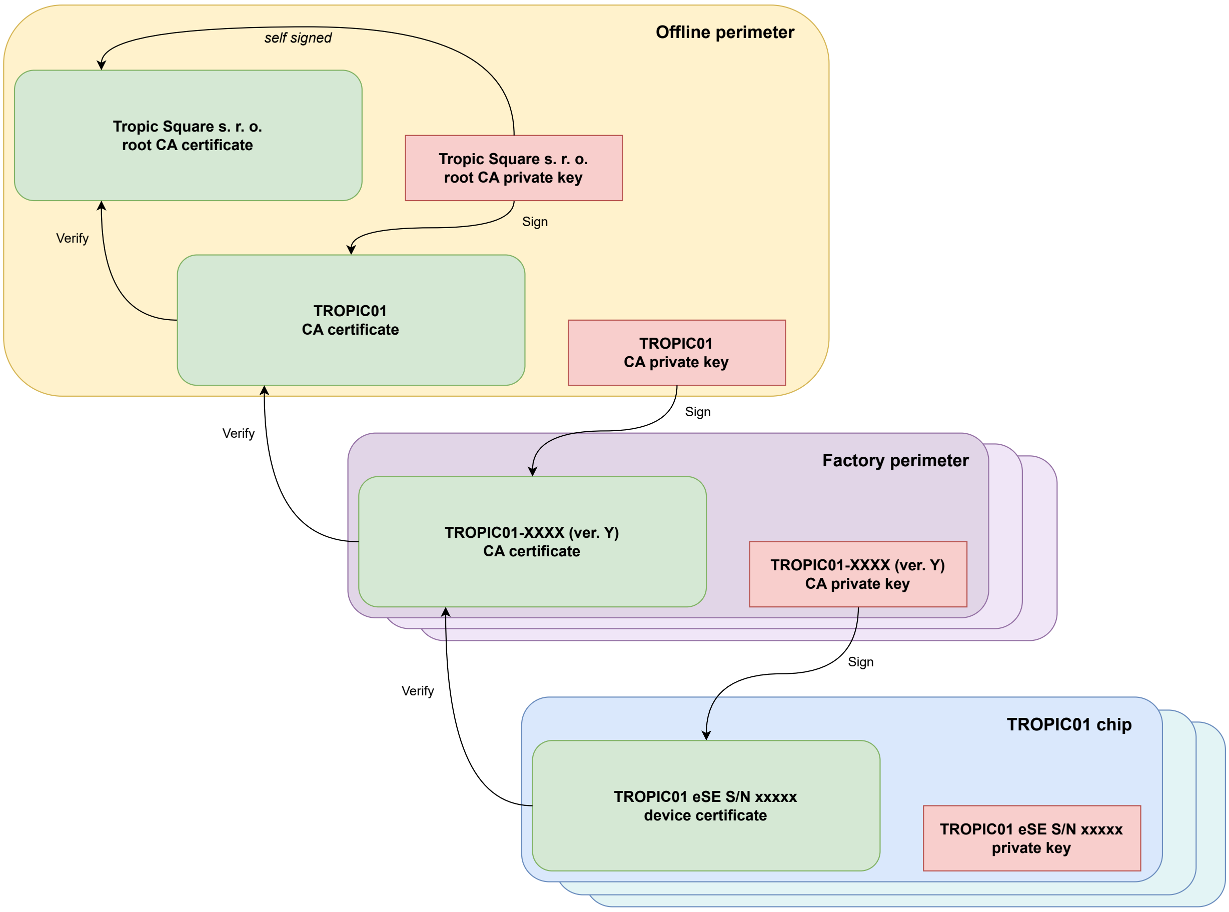

self signed
Offline perimeter
Tropic Square s. r. o. root CA certificate
Tropic Square s. r. o. root CA private key
Sign
Verify
TROPIC01 CA certificate
TROPIC01 CA private key
Sign
Verify
Factory perimeter
TROPIC01-XXXX (ver. Y) CA certificate
TROPIC01-XXXX (ver. Y) CA private key
Sign
Verify
TROPIC01 chip
TROPIC01 eSE S/N xxxxx device certificate
TROPIC01 eSE S/N xxxxx private key

Figure 3: PKI for TROPIC01

The arrows denote the certificate signing hierarchy: higher level key pair is used to issue (sign) the certificate on the lower level (in the direction of the arrow). The “Tropic Square root CA certificate” is self-signed. All the certificates (in green) are by definition public information, but all the private keys (in red) are secrets which never leave the given perimeter (HSM or chip).

To represent the structure of the X.509 certificates in this section, we use OpenSSL v3.1.1 notation (refer to [8]).

3.1 Tropic Square root CA certificate properties

The Tropic Square root CA certificate is the implicitly trusted root of TROPIC01 chip certificate chain. The root certificate needs to be observed for the revocation by manual observation methods (subscribing to Tropic Square updates, email newsletter, periodic screening of `https://pki.tropicsquare.com/` web page etc.)

The root certificate is used to issue "Product level" intermediate CA certificates and also for issuing of Certificate Revocation List (CRL) for "Product level" CAs.

Following diagram shows structure of the Tropic Square root CA certificate. For the interpretation of each sub-field consult ITU-T X.509 v3 structure in [6].

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: "Variable (16-byte array interpreted as integer)"
        Signature Algorithm: ecdsa-with-SHA512
        Issuer: C = CZ, O = Tropic Square s.r.o., CN = Tropic Square Root CA v1
        Validity
            Not Before: "Variable (Time of issuing)" GMT
            Not After : "Variable (Not before + 50 years)" GMT
        Subject: C = CZ, O = Tropic Square s.r.o., CN = Tropic Square Root CA v1
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (521 bit)
                pub: "Variable (From generated key pair)"
                ASN1 OID: secp521r1
                NIST CURVE: P-521
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                "Variable (Computed by issuing CA)"
            X509v3 Basic Constraints: critical
                CA:TRUE
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
    Signature Algorithm: ecdsa-with-SHA512
    Signature Value: "Variable (Computed by issuing CA)"
```

Listing 1: Tropic Square root CA certificate

3.2 TROPIC01 "product level" CA certificate properties

The first intermediate CA level within Tropic Square PKI is organized by products. TROPIC01 should have one dedicated intermediate CA at this level but in case of revocation it might be superseded by the CA of the same name but higher version. The revocation status of this intermediate CA should be observed though its declared CRL Distribution Points.

This intermediate CA certificate is used to issue "P/N level" intermediate CA certificates and also for issuing of Certificate Revocation List (CRL) for "P/N level" CAs.

Following diagram shows structure of the TROPIC01 "product level" CA certificate. For the interpretation of each sub-field consult ITU-T X.509 v3 structure in [6].

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: "Variable (assigned by issuing CA)"
        Signature Algorithm: ecdsa-with-SHA512
        Issuer: C = CZ, O = Tropic Square s.r.o., CN = Tropic Square Root CA v1
        Validity
            Not Before: "Variable (Time of issuing)" GMT
            Not After : "Variable (Not before + 40 years)" GMT
        Subject: C = CZ, O = Tropic Square s.r.o., CN = TROPIC01 CA v1
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (384 bit)
                pub: "Variable (From generated key pair)"
                ASN1 OID: secp384r1
                NIST CURVE: P-384
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                "Variable (Computed by issuing CA)"
            X509v3 Basic Constraints: critical
                CA:TRUE
                pathlen: 1
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Authority Key Identifier:
                "Variable (see Subject Key Identifier of the issuing CA)"
            X509v3 CRL Distribution Points:
                Full Name: "Variable URI pointing to http://pki.tropicsquare.com/ sub-page"
    Signature Algorithm: ecdsa-with-SHA512
    Signature Value: "Variable (Computed by issuing CA)"
```

Listing 2: TROPIC01 CA certificate

3.3 TROPIC01 Part Number (group) CA certificate properties

The second intermediate CA level within Tropic Square PKI is organized by P/Ns. TROPIC01 can have multiple intermediate CAs at this level, each can be used to issue chip certificates for one or more P/Ns ("groups"). The revocation status of this intermediate CA should be observed though its declared CRL Distribution Points.

This intermediate CA certificate is used to issue the final chip (device) certificates and also for issuing of Certificate Revocation List (CRL) for chips in the given P/N group.

Following diagram shows structure of the TROPIC01 "P/N level" CA certificate. For the interpretation of each sub-field consult ITU-T X.509 v3 structure in [6].

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: "Variable (assigned by issuing CA)"
        Signature Algorithm: ecdsa-with-SHA384
        Issuer: C = CZ, O = Tropic Square s.r.o., CN = TROPIC01 CA v1
        Validity
            Not Before: "Variable (Time of issuing)" GMT
            Not After : "Variable (Not before + 35 years)" GMT
        Subject: C = CZ, O = Tropic Square s.r.o., CN = TROPIC01-T CA v1
        Subject Public Key Info:
            Public Key Algorithm: id-ecPublicKey
                Public-Key: (384 bit)
                pub: "Variable (From generated key pair)"
                ASN1 OID: secp384r1
                NIST CURVE: P-384
        X509v3 extensions:
            X509v3 Subject Key Identifier:
                "Variable (Computed by issuing CA)"
            X509v3 Basic Constraints: critical
                CA:TRUE
                pathlen: 0
            X509v3 Key Usage: critical
                Certificate Sign, CRL Sign
            X509v3 Authority Key Identifier:
                "Variable (see Subject Key Identifier of the issuing CA)"
            X509v3 CRL Distribution Points:
                Full Name: "Variable URI pointing to http://pki.tropicsquare.com/ sub-page"
    Signature Algorithm: ecdsa-with-SHA384
    Signature Value: "Variable (Computed by issuing CA)"
```

Listing 3: TROPIC01-XXXX CA certificate

3.4 TROPIC01 chip (device) certificate properties

The chip certificate is the final level within Tropic Square PKI certificate chain. The certificate contains chip S/N which links to the P/N (refer to 2.1 for more). The certificate contains S_{TPub} key corresponding to device Secure Channel X25519 EC key pair (refer to [1]). The revocation status of chip certificate should be observed though its declared CRL Distribution Points.

Following diagram shows structure of the TROPIC01 "P/N level" CA certificate. For the interpretation of each sub-field consult ITU-T X.509 v3 structure in [6].

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: "Variable (chip S/N)"
        Signature Algorithm: ecdsa-with-SHA384
        Issuer: C = CZ, O = Tropic Square s.r.o., CN = TROPIC01-T CA v1
        Validity
            Not Before: "Variable (Time of issuing)" GMT
            Not After : "Variable (Not before + 20 years)" GMT
        Subject: CN = TROPIC01 eSE
        Subject Public Key Info:
            Public Key Algorithm: X25519
                X25519 Public-Key:
                pub: "Variable (Generated as STPUB)"
        X509v3 extensions:
            X509v3 Basic Constraints: critical
                CA:FALSE
            X509v3 Key Usage: critical
                Key Agreement
            X509v3 Authority Key Identifier:
                "Variable (see Subject Key Identifier of the issuing CA))"
            X509v3 CRL Distribution Points:
                Full Name: "Variable URI pointing to http://pki.tropicsquare.com/ sub-page"
    Signature Algorithm: ecdsa-with-SHA384
    Signature Value: "Variable (Computed by issuing CA)"
```

Listing 4: TROPIC01 eSE (chip) certificate

4 Binding TROPIC01 with host (factory process)

The initial "binding" of the host system (MCU/SoC) and TROPIC01 secure element typically happens during the manufacturing process of the device that integrates the TROPIC01. The TROPIC01 comes with the default Secure Channel Pairing Key (S_{H0PUB}). Such "transport" key pair is published by Tropic Square. For more details about Pairing Key slots and Secure Channel establishment refer to [1].

The recommended flow for the binding with TROPIC01 is:

- Validate TROPIC01 certificate chain.
- Verify the S_{H0PUB} key returned from TROPIC01 chip (device) certificate belongs to S_{H0PRIV} that is distributed by Tropic Square (refer to [1] for more details).
- Verify all the ECC key slots contain no ECC key (refer to [1] for more details).
- **Optional**: Issue customer key pair(s) and device certificate(s) usable for application level device identification (e.g. through schemes such as TLS 1.2/1.3, OSCORE, Noise protocol framework etc.)

4.1 Validate TROPIC01 certificate chain

To verify the authentic origin of the TROPIC01 chip, the Customer should perform the following verifications and actions during the initial binding with TROPIC01 chip:

1. Read TROPIC01 X.509 Certificate Store from chip memory by using ***Get_Info_Req*** L2 Request and validate it. To do this, use TROPIC01 SDK (Libtropic [5]) as shown in the example below.
2. Validate the chip (device) certificate:
   (a) ASN.1 DER-TLV structure should be valid and compliant to X.509 schema (refer to [6]).
   (b) Check the validity date of the certificate (requires access to real-time clock linked to UTC).
   (c) Check the revocation status of the certificate based on its CRL Distribution Points (requires on-line access to CRL URI).

   (d) Check the certificate signature based on Authority Key Identifier and issuing CA certificate. That can be obtained either from the TROPIC01 Certificate Store itself or from Tropic Square `https://pki.tropicsquare.com/` web page.

3. Validate "Part Number (group)" level issuing CA certificate. This step must be performed at least once, but it can be skipped for subsequent chip validations in case the issuing CA certificate is static for a given batch of TROPIC01 chips. The validation steps are analogous to the previous point:

   (a) ASN.1 DER-TLV structure should be valid and compliant to X.509 schema (refer to [6]).

   (b) Check the validity date of the certificate (requires access to real-time clock linked to UTC).

   (c) Check the revocation status of the certificate based on its CRL Distribution Points (requires on-line access to CRL URI).

   (d) Check the certificate signature based on Authority Key Identifier and issuing CA certificate. That can be obtained either from the TROPIC01 Certificate Store itself or from Tropic Square `https://pki.tropicsquare.com/` web page.

4. Validate "Product (TROPIC01)" level intermediate CA certificate. This step must be performed at least once, but it can be skipped for subsequent chip validations in case the intermediate CA certificate is static for a given batch of TROPIC01 chips. The validation steps are analogous to the previous point:

   (a) ASN.1 DER-TLV structure should be valid and compliant to X.509 schema (refer to [6]).

   (b) Checking the validity date of the certificate (requires access to real-time clock linked to UTC).

   (c) Checking the revocation status of the certificate based on its CRL Distribution Points (requires on-line access to CRL URI).

   (d) Checking the certificate signature based on Authority Key Identifier and issuing CA certificate. That can be obtained either from the TROPIC01 Certificate Store itself or from Tropic Square `https://pki.tropicsquare.com/` web page.

5. Validate Tropic Square Root CA certificate. This step must be performed at least once, but it can be skipped for subsequent chip validations in case the this inter-

mediate CA certificate is static for a given batch of TROPIC01 chips. The validation steps are following:

(a) ASN.1 DER-TLV structure should be valid and compliant to X.509 schema (refer to [6]).

(b) Check the validity date of the certificate (requires access to real-time clock linked to UTC).

(c) Check the revocation status of the certificate. This can be only done "out-of-band" by checking the status with Tropic Square (e.g. through `https://pki.tropicsquare.com/` web page, incident newsletter etc.)

(d) Check the certificate signature. The Root CA certificate is by definition self-signed, so the signature is validated by the public key presented within the certificate itself.

6. After successful validation of TROPIC01 X.509 certificate chain it is critical to validate additional metadata to prevent supply-chain attacks:

   (a) The S/N encoded in the chip certificate must encode the correct P/N as per 2.1 and [3].

   (b) Make sure that the selected P/N corresponds to the right P/N with the FW signing "vendor" key. Note that each TROPIC01 P/N hosts Application FW from exactly one vendor "source" which is determined by the FW signing key pair. For more details refer to [4].

   (c) **Optional**: Read **CHIP_ID** from chip memory (***Get_Info_Req*** L2 Request) and verify the metadata are matching the expectations set by P/N ID (refer to 2.2 for more details).

4.2 TROPIC01 certificate chain readout by TROPIC01 SDK

We provide an example program to read out the certificate chain for Linux and TROPIC01 USB DevKit as a part of the Libtropic [5] in the Full Chain Verification tutorial. Refer to Libtropic documentation for more information. An excerpt of the program is provided for reference in this section.

The following excerpt shows a function which dumps certificates to files. It is assumed that the `lt_handle` is already initialized and that the platform on which the program is compiled supports writing to files via file operations in C standard library. The excerpt does not show either initialization or proper deinitialization. Refer to the Libtropic [5] for more information and a complete example.

```c
lt_ret_t dump_cert_store(lt_handle_t *lt_handle)
{
    uint8_t cert1[TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE] = {0},
            cert2[TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE] = {0},
            cert3[TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE] = {0},
            cert4[TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE] = {0};

    struct lt_cert_store_t store
        = {.certs   = {cert1, cert2, cert3, cert4},
           .buf_len = {TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE,
                       TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE,
                       TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE,
                       TR01_L2_GET_INFO_REQ_CERT_SIZE_SINGLE}};

    // Reading X509 Certificate Store
    lt_ret_t ret = lt_get_info_cert_store(lt_handle, &store);
    if (ret != LT_OK) {
        return LT_FAIL;
    }

    // Dump the certificates to files
    const char *names[LT_NUM_CERTIFICATES]
        = {"t01_ese_cert.der", "t01_xxxx_ca_cert.der",
           "t01_ca_cert.der", "tropicsquare_root_ca_cert.der"};

    for (int i = 0; i < LT_NUM_CERTIFICATES; i++) {
        if (store.cert_len[i] == 0) {
            return LT_FAIL;
        }
        FILE *f = fopen(names[i], "wb");
        if (!f) {
            return LT_FAIL;
        }
        if (fwrite(store.certs[i], 1, store.cert_len[i], f) != store.cert_len[i]) {
            fclose(f);
            return LT_FAIL;
        }
        fclose(f);
    }

    return LT_OK;
}
```

4.3 TROPIC01 certificate chain validation example with OpenSSL

We provide an example validation script for Linux and TROPIC01 USB DevKit as a part of the Libtropic [5] in the Full Chain Verification tutorial. Refer to Libtropic documentation [5] for more information. A simplified version of the script is provided for reference in this section.

The following script assumes running on Linux machine with OpenSSL installed and certificates read from the TROPIC01 as shown in previous example. The script validates all certificates read from TROPIC01, and checks against their respective CRLs.

```
# Download certificate authorities from Tropic Square PKI web
curl --fail http://pki.tropicsquare.com/l0/tropic01_xxxx_ca_certificate_sn_30001.pem \
     -o "tropic01_xxxx_ca_certificate_sn_30001.pem"
curl --fail http://pki.tropicsquare.com/l0/tropic01_ca_certificate_sn_3001.pem \
     -o "tropic01_ca_certificate_sn_3001.pem"

# Parse CRLs from certificates read from device in previous example
L3=$(openssl x509 -in "t01_ese_cert.der" -inform DER -text | grep URI | cut -d ’:’ -f 2-)
L2=$(openssl x509 -in "t01_xxxx_ca_cert.der" -inform DER -text \
    | grep URI | cut -d ’:’ -f 2-)
L1=$(openssl x509 -in "t01_ca_cert.der" -inform DER -text | grep URI | cut -d ’:’ -f 2-)

# Download CRLs
curl --fail "$L3" -o "t01-Tv1.crl" # Downloads t01-Tv1.crl
curl --fail "$L2" -o "t01v1.crl" # Downloads t01v1.crl
curl --fail "$L1" -o "tsrv1.crl" # Downloads tsrv1.crl

# Verify (chip) device certificate
cat "tropic01_xxxx_ca_certificate_sn_30001.pem" \
    "t01-Tv1.crl" \
    "tropic01_ca_certificate_sn_3001.pem" \
    "t01v1.crl" \
    "./tropicsquare_root_ca_certificate_sn_301.pem" \
    "tsrv1.crl" > "chain.pem"
openssl verify -verbose -crl_check -CAfile "chain.pem" "t01_ese_cert.der"

# Verify the "Part Number (group)" certificate
cat "tropic01_ca_certificate_sn_3001.pem" \
    "t01v1.crl" \
    "./tropicsquare_root_ca_certificate_sn_301.pem" \
    "tsrv1.crl" > "chain.pem"
openssl verify -verbose -crl_check -CAfile "chain.pem" "t01_xxxx_ca_cert.der"

# Verify the "Product (Part Name)" certificate
cat "./tropicsquare_root_ca_certificate_sn_301.pem" "tsrv1.crl" > "chain.pem"
openssl verify -verbose -crl_check -CAfile "chain.pem" "t01_ca_cert.der"

# Verify Tropic Square Root Certificate
openssl verify -verbose -CAfile "./tropicsquare_root_ca_certificate_sn_301.pem" \
               "./tropicsquare_root_ca_certificate_sn_301.pem"
```

Version history

Version	Date	Description
1.0	30.9.2025	Initial public release.
1.1	20.10.2025	Fixes in OpenSSL validation script.
1.2	28.10.2025	Include needed Libtropic headers in the example.
1.3	02.02.2026	Update certificate chain readout and validation examples.

References

[1] TROPIC01 – Datasheet, Tropic Square

[2] TROPIC01 – User API, Tropic Square

[3] TROPIC01 – Catalog list, Tropic Square

[4] ODN_TR01_app_007 - TROPIC01 Firmware Update Application Note, Tropic Square

[5] TROPIC01 SDK – Libtropic,
Github: `https://github.com/tropicsquare/libtropic`
Documentation: `https://tropicsquare.github.io/libtropic/`

[6] ITU-T X.509 (10/2019), also published as ISO/IEC 9594-8
`https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.509`

[7] ITU-T X.690 (02/2021), also published as ISO/IEC 8825-1
`https://www.itu.int/ITU-T/recommendations/rec.aspx?rec=X.690`

[8] OpenSSL v3.1.1 x509 documentation
`https://docs.openssl.org/1.1.1/man1/x509/`

[9] The Noise Protocol Framework specification, Revision 34, 2018-07-11 `https://noiseprotocol.org/noise.html`

[10] Concise Binary Object Representation (CBOR), IETF RFC 8949, December 2020 https://datatracker.ietf.org/doc/html/rfc8949

[11] CBOR Object Signing and Encryption (COSE), IETF RFC 9053, August 2022 https://datatracker.ietf.org/doc/html/rfc9053

Legal Notice

Our mission is to provide you with high quality, safe, and transparent products, but to be able to do so, we also have to make the following disclaimers.

To verify the characteristics of our products, consult the repositories we make available on our GitHub. While we do our best to keep the content of these repositories updated, we cannot guarantee that the content of these repositories will always identically correspond to our products. For example, there may be delays in publication or differences in the nature of the software solutions as published and as included in the hardware products. Some parts of our products cannot be published due to third party rights.

We take pride in publishing under open-source license terms, but do not grant licenses to any of our patents. Please consult the license agreement in the repository. We reserve the right to make changes, corrections, enhancements, modifications, and improvements to our products, published solutions and terms at any time without notice.

Since we cannot predict what purposes you may use our products for, we make no warranty, representation, or guarantee, whether implied or explicit, regarding the suitability of our products for any particular purpose.

To the maximum extent permitted by applicable law, we disclaim any liability for any direct, indirect, special, incidental, consequential, punitive, or any other damages and costs including but not limited to loss of profit, revenue, savings, anticipated savings, business opportunity, data, or goodwill regardless of whether such losses are foreseeable or not, incurred by you when using our products. Further, we disclaim any liability arising out of use of our products contrary to their user manual or our terms, their use/implementation in unsuitable environments or ways, or for such use which may infringe third party rights. Notwithstanding the above, the maximum liability from the use of our products shall be limited to the amount paid by you as their purchase price.