TROPIC01

ODN_TR01_app_005

First Pairing Key Application Note

Version: 1.0

Abstract

This application note explains usage of the first Pairing Key in TROPIC01.

September 30, 2025

tropicsquare

Contents

Glossary

- **CPU** – Central Processing Unit
- **SDK** – Software development kit

1 Introduction

This application note describes:

1. Meaning and Usage of Pairing Keys in TROPIC01.
2. Options Tropic Square provides when it comes to the key in the first Pairing Key slot (S_{H0PRIV}, S_{H0PUB}).
3. Best practices when it comes to handling of keys in Pairing Key slots.

1.1 Target Audience

This document is intended for Embedded Engineers, Product Managers and System Architects developing devices with TROPIC01.

1.2 Theoretical Background

TROPIC01 connects with Host MCU via asymetric key exchange mechanism. The key exchange establishes a Secure Channel. Secure Channel is a connection that is private between TROPIC01 and a remote party. For more details on Secure Channel, see [1].

This Secure Channel connection is diversified by a Paring Key pair (S_{Hi}). Different Pairing Keys allows TROPIC01 to connect securely with different remote parties. The Pairing Key pair consists of:

- Public Key (S_{HiPUB}) - Stored in TROPIC01
- Private Key (S_{HiPRIV}) - Known by a remote party

There can be up to 4 Public Pairing Keys in TROPIC01. TROPIC01 has a single Pairing Key slot for each Pairing Key.

Once a Secure Channel is established, the remote party can:

- Write new Pairing Keys to TROPIC01 (***Pairing_Key_Write***, see [2]).
- Invalidate Pairing Keys in TROPIC01 (***Pairing_Key_Invalidate***, see [2]).

The Host MCU can write a Pairing Key to each Pairing Key slot only once. Once written, the Host MCU can use the Pairing Key to establish a Secure Channel. Once a Pairing Key is invalidated, it is not possible to establish Secure Channel on such Pairing Key slot anymore.

1.3 First Pairing Key

To allow the very first Secure Channel to be established:

- Public part of the first Pairing Key (S_{H0PUB}) must be present in TROPIC01.
- Private part of the first Pairing Key (S_{H0PRIV}) must be known by the remote party.

Tropic Square stores the public part of the first Pairing Key to TROPIC01 during manufacturing.

2 Use Cases

Tropic Square provides following options of writing the first Pairing Key to TROPIC01:

- Default first Pairing Key.
- Customer specific first Pairing Key.

2.1 Default first Pairing Key

Tropic Square generates the first Pairing Key pair, and programs its public part to TROPIC01. The private Pairing Key is publicly available in TROPIC01 SDK - `libtropic`, see [3].

Pros:

- Simple setup - Device works out of the box with `libtropic`
- No special arrangement needed with Tropic Square

Cons:

- Common first Pairing Key for many customers
- One less Pairing Key slot for Keys that are private only to the customer.

! Warning !

When customer uses this option, they shall only use the Secure Channel on the first Pairing Key slot to store their own generated Pairing Key to the next slot. Then customers shall invalidate the first Pairing Key.

2.2 Customer specific first Pairing Key

With this option, the customer generates his own first Pairing Key, and delivers its public part to Tropic Square. Tropic Square writes this key to TROPIC01 during manufacturing. Customer does not need to reveal the first Pairing Key to anyone.

Tropic Square assigns custom Part Number to devices with this key. Such Part Number is available only to the customer who generated the Pairing Key pair. As the customer is the only party who knows the private key of the first Pairing Key Pair, they are the only party able to ever use such device.

Pros:

- Ultimate security.
- Devices with this Pairing Key will only ever establish Secure Channel with single customer.
- All Pairing Key slots can be used by the customer.
- No need to invalidate the key in the first Pairing Key slot.

Cons:

- Custom agreement needed with Tropic Square.
- `libtropic` needs to be configured and compiled for these devices - to contain your private key.

3 Usage Examples

3.1 How to set-up custom First Pairing Key.

The private part of the first Pairing Key is compiled into TROPIC01 SDK [3]. To set-up custom first Pairing Key:

1. Generate an X25519 Key Pair, e.g. via OpenSSL command in Linux terminal like so:

```
openssl genpkey -algorithm x25519 -out sh0-priv.pem
openssl pkey -in sh0-priv.pem -pubout -out sh0-pub.pem
```

   This will generate:

   - *sh0-priv.pem* - Private Key
   - *sh0-pub.pem* - Public Key

2. Reach out to Tropic Square at support@tropicsquare.com, and query TROPIC01 manufacturing with customer specific First Pairing key. During the bussines process you will be asked to deliver the *sh0-pub.pem* - Public Part of the first Pairing Key.

3. After you receive the TROPIC01 devices with your custom First Pairing Key, you need to configure `libtropic` to be compiled for your custom devices. To do so, set the value of **LT_SH0_PRIV_PATH** variable in *CMakeLists.txt* in the `libtropic` repository root. Set the value to the path of your generated *sh0-priv.pem*, e.g:

```
set(LT_SH0_PRIV_PATH
    "/home/user1/sh0-priv.pem"
    CACHE FILEPATH
    "Path to file with SH0 private key in PEM or DER format."
)
```

4. Proceed with compiling libtropic as explained in documentation in [3].

3.2 How to write a Pairing Key

To Write Pairing Key to a Pairing Key slot, call the following `libtropic` function:

```
/**
 * @brief Writes pairing public key into TROPIC01's pairing key slot 0-3
 *
 * @param h           Device's handle
 * @param pairing_pub 32B of pubkey
 * @param slot        Pairing key slot SH0PUB - SH3PUB
 *
 * @retval            LT_OK Function executed successfully
 * @retval            other Function did not execute successully, you might use
    lt_ret_verbose() to get verbose encoding
 * of returned value
 */
lt_ret_t lt_pairing_key_write(lt_handle_t *h, const uint8_t *pairing_pub, const uint8_t slot
    );
```

3.3 How to Invalidate a Pairing Key

To Invalidate Pairing Key in a Pairing Key slot, call the following `libtropic` function:

```
/**
 * @brief Invalidates pairing key in slot 0-3
 *
 * @param h           Device's handle
 * @param slot        Pairing key slot SH0PUB - SH3PUB
 *
 * @retval            LT_OK Function executed successfully
 * @retval            other Function did not execute successully, you might use
    lt_ret_verbose() to get verbose encoding
 * of returned value
 */
lt_ret_t lt_pairing_key_invalidate(lt_handle_t *h, const uint8_t slot);
```

Version history

Version	Date	Description
1.0	30.9.2025	Initial public release.

References

[1] ODD_TR01_datasheet, TROPIC01 Datasheet, Tropic Square

[2] ODU_TR01_user_api – TROPIC01 User API, Tropic Square

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

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.