Fortanix®

Integration Guide

FORTANIX SIGNING PROVIDER FOR TMKMS

VERSION 1.0

Fortanix, Inc. | 800 West El Camino Real | Suite 180 | Mountain View, CA 94040 | United States of America | +1 (650) 943-2484 |
info@fortanix.com | www.fortanix.com

TABLE OF CONTENTS

1.0 INTRODUCTION

This document describes how **Fortanix Data Security Manager (DSM)** integrates with **Tendermint Key Management System (TMKMS)** to provide remote signing capabilities that enable the signing of Ignite (previously, Tendermint) blockchain proposals and votes. With this integration, you can now create, manage, and use validator keys with Fortanix DSM SaaS. For any proposal or vote, TMKMS validates that double-signing is not being attempted and then signs the proposal or vote by invoking the sign operation offered by DSM SaaS which manages the validator keys.

2.0 DEFINITIONS

- **Fortanix Data Security Manager**

  Fortanix DSM is the cloud solution secured with Intel® SGX. With Fortanix DSM, you can securely generate, store, and use cryptographic keys and certificates, as well as secrets, such as passwords, API keys, tokens, or any blob of data.

- **Accounts**

  A Fortanix DSM account is the top-level container for security objects managed by the Fortanix DSM. An account is generally associated with an organization, rather than an individual. Security objects, groups, and applications belong to exactly one account. Different accounts are fully isolated from each other. *See support for more information.*

- **Fortanix Data Security Manager Security Objects**

  A security object is any datum stored in Fortanix DSM (for example a key, a certificate, a password, or other security objects). Each security object is assigned to exactly one group. Users and applications assigned to the group have permission to see the security object and to perform operations on it. *See support for more information.*

3.0 COMPILING TMKMS WITH FORTANIX DSM

Refer to the main README.md for compiling `tmkms` from the source code. You will need the prerequisites mentioned under the *Section: Supported Platforms* in the readme.

The following are the two ways to install `tmkms` with Fortanix DSM. In these methods, you must pass the `--features=fortanixdsm` parameter to cargo:

3.1 COMPILING FROM SOURCE CODE (USING THE `git` COMMAND)

`tmkms` can be compiled directly from the Git repository source code using the following command:

```
$ git clone https://github.com/iqlusioninc/tmkms.git && cd tmkms
[...]
$ cargo build --release --features=fortanixdsm
```

If successful, this will produce a `tmkms` executable located at `./target/release/tmkms`.

3.2 INSTALLING WITH `cargo install` COMMAND

With Rust (1.40+) installed, you can install `tmkms` using the following command:

```
cargo install tmkms --features=fortanixdsm
```

Or you can install a specific version (recommended), using the following command:

```
cargo install tmkms --features=fortanixdsm --version=0.4.0
```

This command installs `tmkms` directly from packages hosted on Rust's [crates.io] service. The package authenticity is verified using the [crates.io index] which is a Git repository and by SHA-256 digests of released artifacts.

However, if newer dependencies are available, it may use newer versions besides the ones which are "locked" in the source code repository. We cannot verify whether those dependencies do not contain malicious code. If you would like to ensure the dependencies in use are identical to the main repository, please build from the source code instead.

4.0 SET UP FORTANIX DSM

This section explains how to create secure keys in Fortanix DSM.

4.1 CREATE A FORTANIX DSM GROUP

1. To generate/import a Consensus key, first, create a Fortanix DSM group.

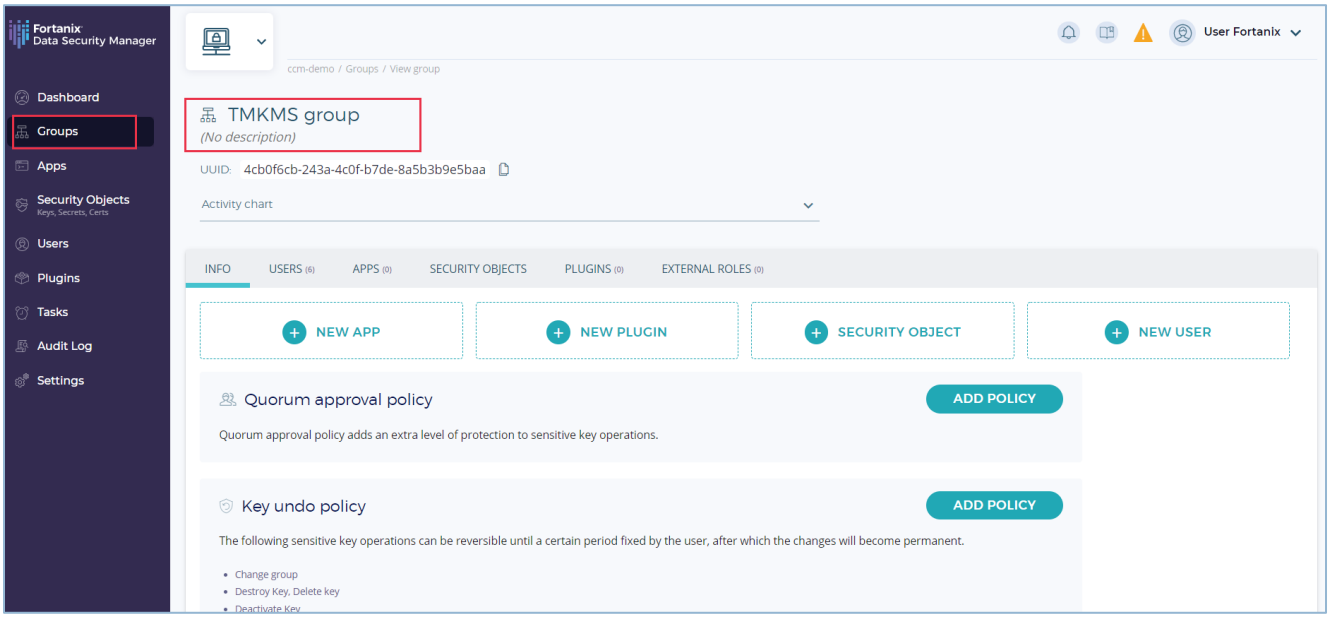

FIGURE 1: CREATE GROUP

4.2 CREATE AN APP IN FORTANIX DSM

Create an app in Fortanix DSM of type **REST API** and copy the app's **API KEY**. This API Key is added to the `tmkms.toml` configuration file later. *Refer to Section 4.4*.

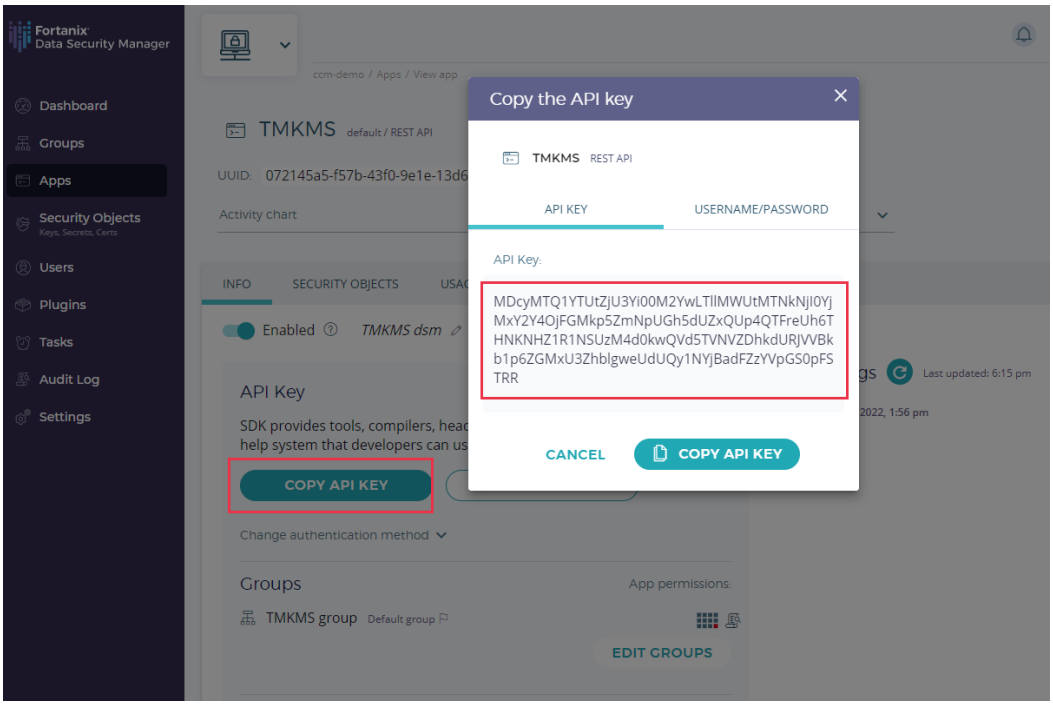

FIGURE 2: CREATE AN APP AND COPY THE API KEY

4.3 GENERATE A KEY

`tmkms` supports keys for accounts and consensus signing. The type of key must be **EC CurveEd25519** for the Consensus key and **Secp256k1** for the Account key. This guide explains the signing procedure using the Consensus key.

1. Generate a key called **`consensus-key`** in the same group created in *Section 4.1* so that the API key of the app created in *Section 4.2* can be used to access this key. The key type must be **EC** of Curve **Ed25519**. This key is used by the validator for consensus signing.

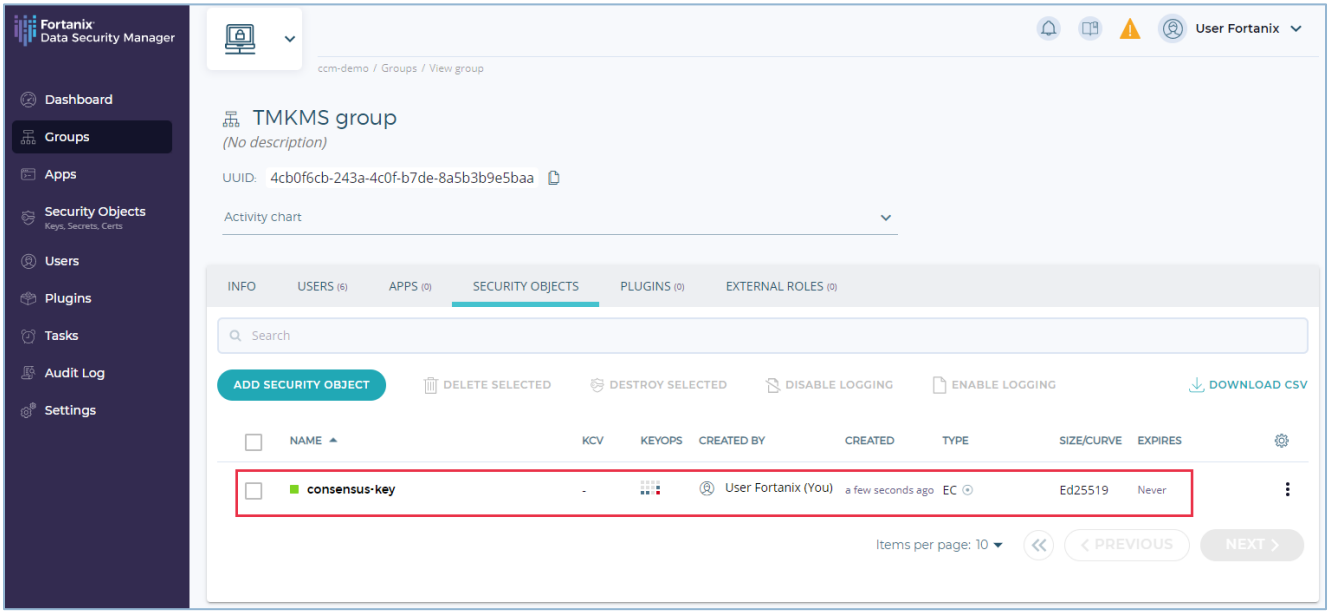

FIGURE 3: CONSENSUS KEY

2. Copy the Key ID or the key name of this key to add it to the `tmkms.toml` configuration file later. *Refer to Section 4.4*.

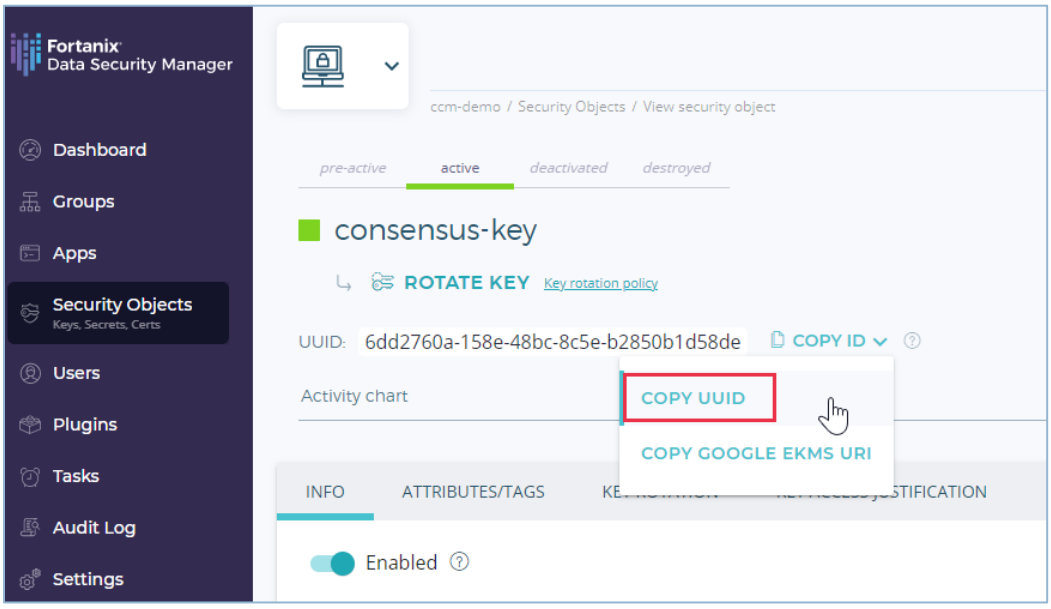

FIGURE 4: COPY KEY ID

Alternatively, you can also import an existing Tendermint key.

To import an existing Tendermint key, use the following script to convert a Tendermint key to Fortanix DSM accepted key format.

```
#!/bin/bash
# Usage: tendermint-ed25519.sh <input-tendermint> <output-private-p8der> <output-public-p8der>

gokey=$(jq -r .priv_key.value $1 | base64 -d| xxd -p -c 64)
echo 302e 0201 0030 0506 032b 6570 0422 0420 "${gokey:0:64}" | xxd -p -r > $2
echo 302a 3005 0603 2b65 7003 2100 "${gokey:64}" | xxd -p -r > $3
```

4.4 CONFIGURE `tmkms`

To perform the setup, `tmkms` needs a configuration file that contains the authentication details needed to authenticate to Fortanix DSM with an API key.

The file `tmkms.toml` contains this configuration. You can specify the path to the config with either `-c /path/to/tmkms.toml` or `tmkms` will look in the current working directory for the same file.

For example:

```
[[providers.fortanixdsm]]
api_endpoint = "https://<fortanix_dsm_url>"
api_key =
"Nzk5MDQ3ZGUtN2Q2NS00OTRjLTgzMDMtNjQwMTlhYzdmOGUzOlF1SU93ZXJsOFU4VUdEWEd
QMmx1dFJOVjlvMTRSd3lhNnVDNVNhVkpZOVhzYVgyc0pOVGRQVGJ0RjZJdmVLMy00X05iTEh
xMkowamF3UGVPaXJEWEd3"
signing_keys = [
    { chain_ids = ["$CHAIN_ID"], type = "account", key_id = "72e9ed9e-9eb4-46bd-a135-e78ed9bfd611" },
    { chain_ids = ["$CHAIN_ID"], type = "consensus", key_name = "My Key" },
]
```

Where,

- `api_key` is the Fortanix DSM app API Key. Refer to *Section 4.2* for steps to get the app API Key.

- `key_id` and `key_name` is the Fortanix DSM key UUID and key name respectively. You can either pass the key UUID or key name for signing. Refer to *Section 4.3* for steps to create a Consensus key or import an existing key.

4.5 RUNNING THE TMKMS APPLICATION

The `tmkms.toml` configuration file now has the required details:

- The Fortanix DSM App API Key to authenticate to Fortanix DSM.
- The Fortanix DSM Key ID/Key name is used by the Validator for consensus signing.

1. Start `tmkms` using the following command:

```
$ tmkms start
```

This will read the configuration from the `tmkms.toml` file in the current working directory.

or

To explicitly specify the path to the configuration, use the `-c` flag:

```
$ tmkms start -c /path/to/tmkms.toml
```

2. Run the tmkms application. Go to the detailed view of the **consensus-key** and in the **Activity Logs** section notice that this key is used by the Validator for consensus signing.

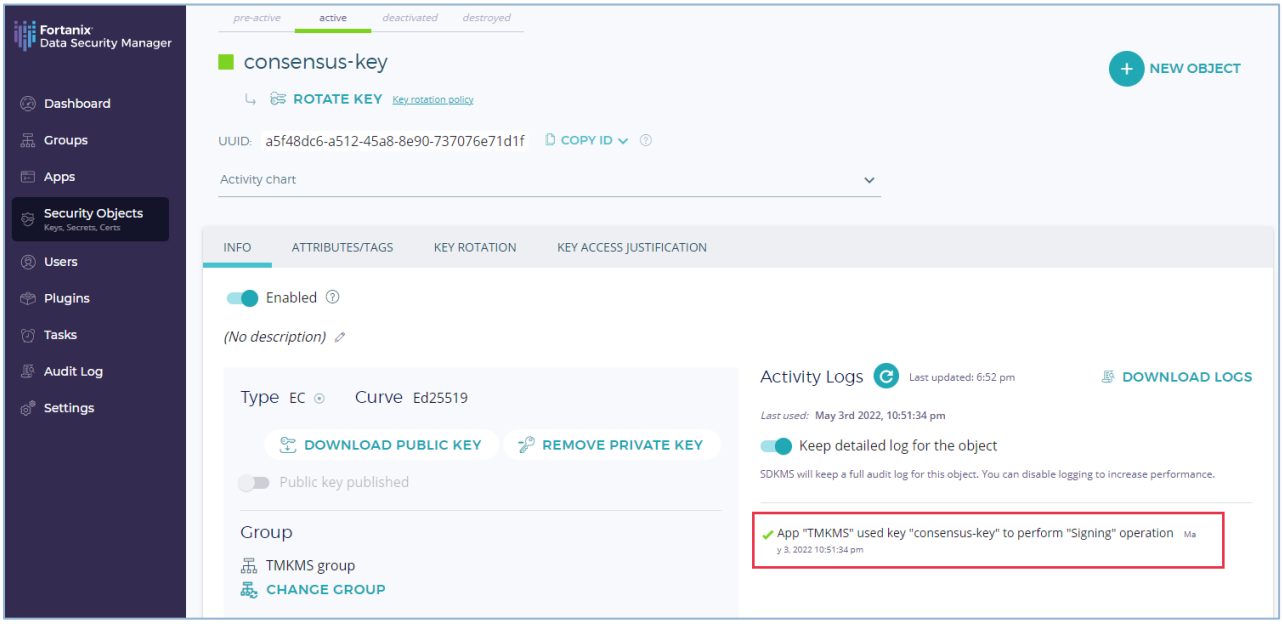

FIGURE 5: EXECUTION LOG FOR KEY

5.0 DOCUMENT INFORMATION

5.1 DOCUMENT LOCATION

The latest published version of this document is located at the URL:

https://support.fortanix.com/hc/en-us/articles/6999736664468-Using-Fortanix-Signing-Provider-for-Tendermint-KMS

5.2 DOCUMENT UPDATES

This document will typically be updated on a periodic review and update cycle.

For any urgent document updates, please send an email to: support@fortanix.com