User Guide

FORTANIX - IGNITE ONE-TIME SIGNER SOLUTION

VERSION 1.0

TABLE OF CONTENTS

1.0 INTRODUCTION

This document describes how the **Fortanix "Ignite One-Time Signer"** solution improves the security of Proof-of-Stake blockchains that use the Ignite (previously known as Tendermint) software stack. The solution provides operators of validator nodes with a mechanism to prevent the double-signing of proposals and votes. This solution comprises a plugin that is securely deployed inside Fortanix Data Security Manager (DSM) Software-as-a-Service (SaaS) and a shim which facilitates communication between the validator and the plugin.

1.1 FORTANIX ONE-TIME SIGNER PLUGIN

The Fortanix "Ignite One-Time Signer" solution validates that the proposals and votes are well-formed, tracks the state of the consensus protocol, and ensures that double-signing is prevented since it can be harmful to the blockchain network. The plugin can be used to sign blockchain consensus messages of the following types:

- PrevoteType = 0x01
- PrecommitType = 0x02
- ProposalType = 0x20

The plugin is protected by a quorum policy that involves multiple admin users. Once deployed, the plugin code cannot be modified without explicit permissions from multiple administrators.
The solution offers the following benefits:

- High availability and disaster recovery.
- Secure key management.
- Secure execution of critical code.

NOTE:

- The `shim` utility is installed in the same environment as the validator node. It allows the validator node to interface with the "Tendermint One Time Signer" plugin. *Refer to Section 4.0 for more details*.

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.*

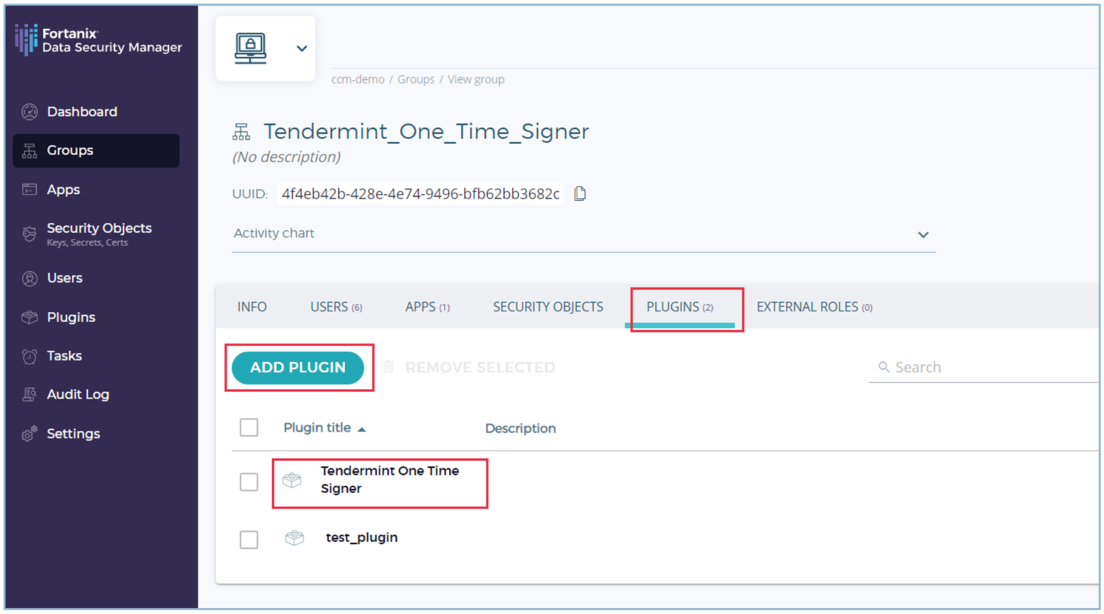
- **Fortanix Data Security Manager Security Objects**

  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 SET UP PLUGIN

3.1 CREATE A FORTANIX DSM GROUP

1. To use the Fortanix “Tendermint One Time Signer” plugin in Fortanix DSM, you must first create a Fortanix DSM group and add the plugin to this group.

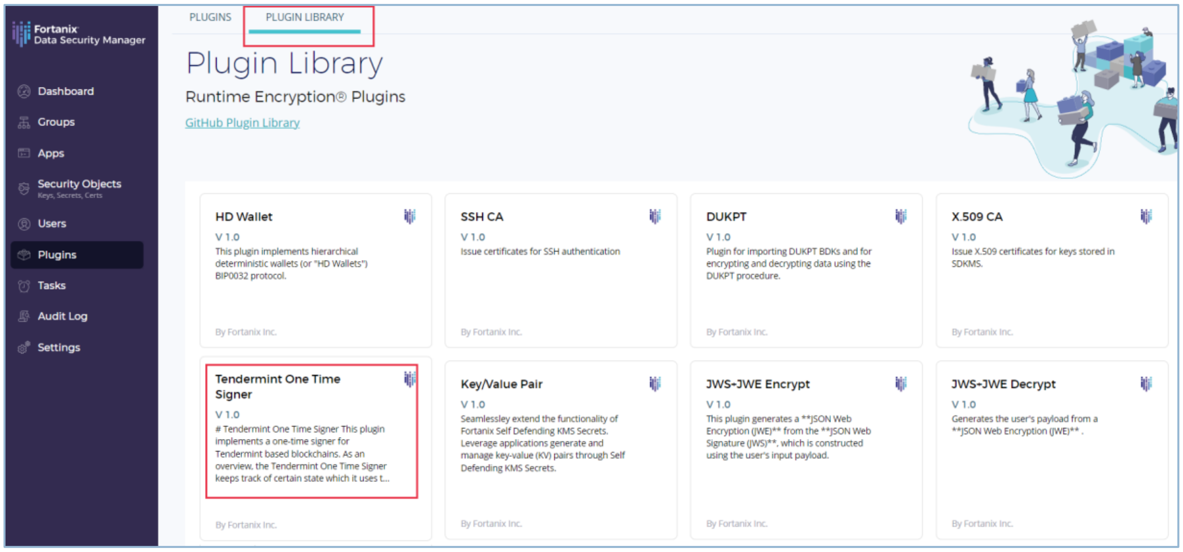


Tendermint_One_Time_Signer
(No description)
UUID: 4f4eb42b-428e-4e74-9496-bfb62bb3682c
Activity chart

	Plugin title	Description
□	Tendermint One Time Signer	
□	test_plugin	

FIGURE 1: IMPORT PLUGIN

Refer to the User’s Guide: Plugin Library for steps to [access](#) and [install](#) the plugin from the Fortanix DSM Plugin Library.



Plugin Library
Runtime Encryption® Plugins
GitHub Plugin Library

HD Wallet
V 1.0
This plugin implements hierarchical deterministic wallets (or "HD Wallets") BIP0032 protocol.
By Fortanix Inc.

SSH CA
V 1.0
Issue certificates for SSH authentication
By Fortanix Inc.

DUKPT
V 1.0
Plugin for importing DUKPT BDKs and for encrypting and decrypting data using the DUKPT procedure.
By Fortanix Inc.

X.509 CA
V 1.0
Issue X.509 certificates for keys stored in SDKMS.
By Fortanix Inc.

Tendermint One Time Signer
V 1.0
Tendermint One Time Signer This plugin implements a one-time signer for Tendermint based blockchains. As an overview, the Tendermint One Time Signer keeps track of certain state which it uses t...
By Fortanix Inc.

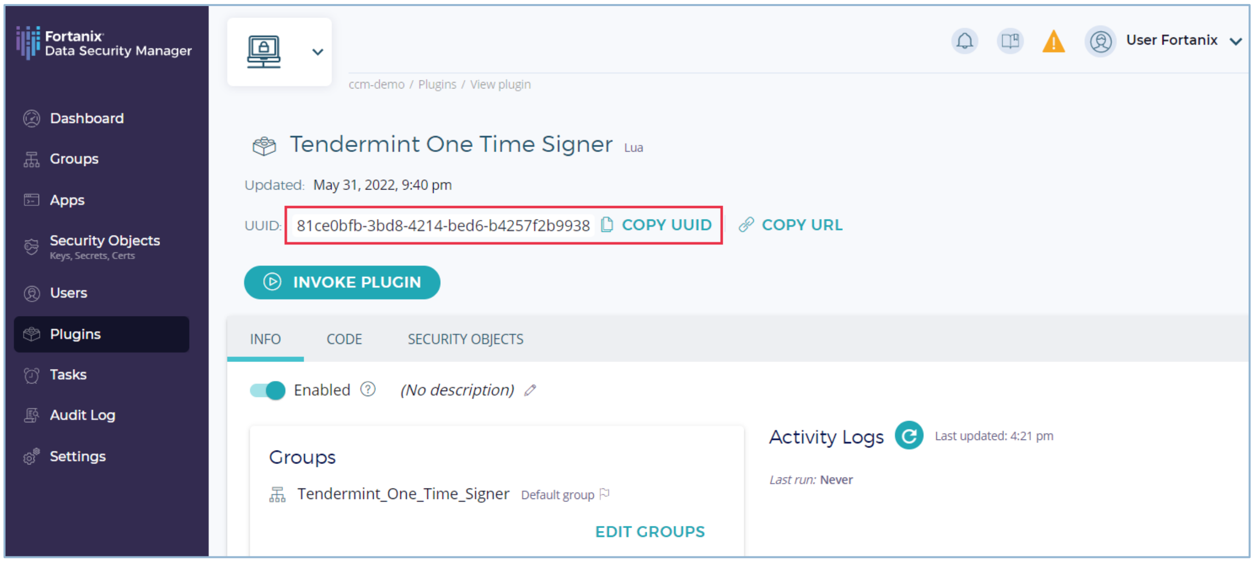
Key/Value Pair
V 1.0
Seamlessley extend the functionality of Fortanix Self Defending KMS Secrets. Leverage applications generate and manage key-value (KV) pairs through Self Defending KMS Secrets.
By Fortanix Inc.

JWS-JWE Encrypt
V 1.0
This plugin generates a **JSON Web Encryption (JWE)** from the **JSON Web Signature (JWS)**, which is constructed using the user's input payload.
By Fortanix Inc.

JWS-JWE Decrypt
V 1.0
Generates the user's payload from a **JSON Web Encryption (JWE)** .
By Fortanix Inc.

FIGURE 2: INSTALL PLUGIN FROM THE PLUGIN LIBRARY

2. Copy the UUID of the plugin to add it to the `shim` configuration file later. *Refer to Section 4.1.*



ccm-demo / Plugins / View plugin

Tendermint One Time Signer Lua

Updated: May 31, 2022, 9:40 pm

UUID: 81ce0bfb-3bd8-4214-bed6-b4257f2b9938

Enabled (No description)

Groups

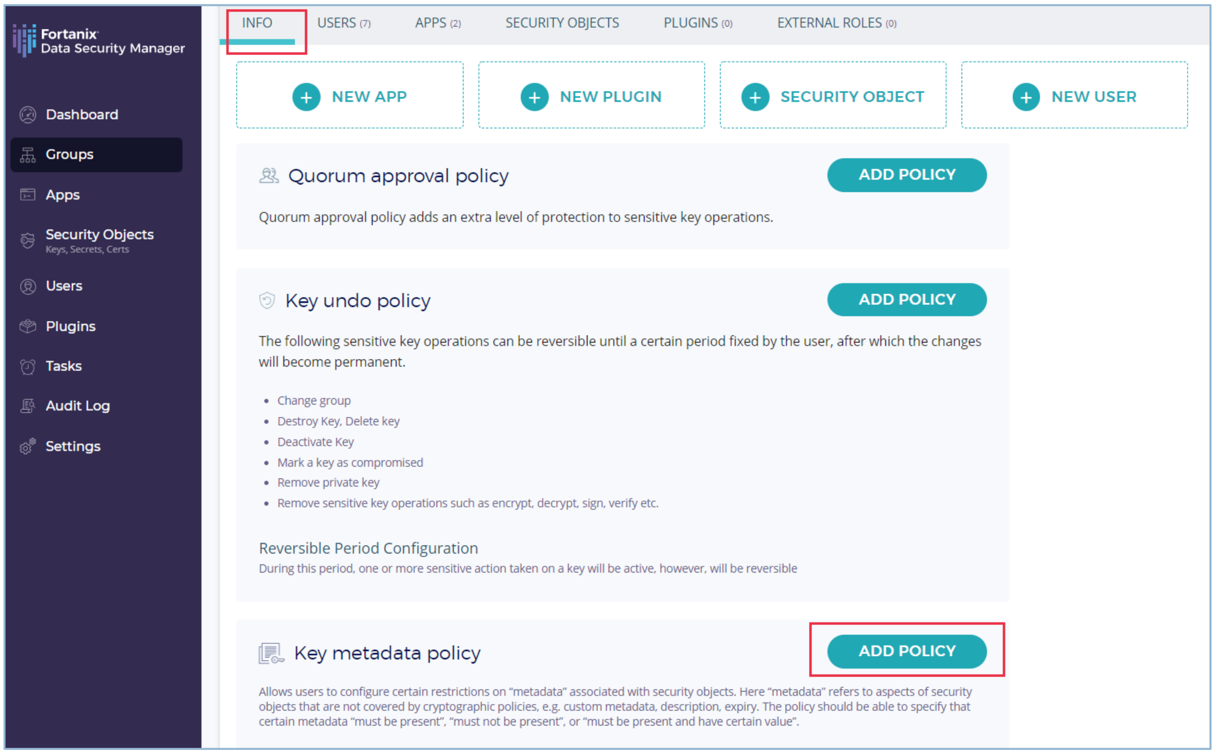
Tendermint_One_Time_Signer Default group

Activity Logs Last updated: 4:21 pm

Last run: Never

FIGURE 3: PLUGIN UUID

3. In the detailed view of the group, click the **INFO** tab and in the “Key metadata policy” section, click **ADD POLICY** to add a Key metadata policy for the group.



Quorum approval policy

Quorum approval policy adds an extra level of protection to sensitive key operations.

Key undo policy

The following sensitive key operations can be reversible until a certain period fixed by the user, after which the changes will become permanent.

- Change group
- Destroy Key, Delete key
- Deactivate Key
- Mark a key as compromised
- Remove private key
- Remove sensitive key operations such as encrypt, decrypt, sign, verify etc.

Reversible Period Configuration

During this period, one or more sensitive action taken on a key will be active, however, will be reversible

Key metadata policy

Allows users to configure certain restrictions on “metadata” associated with security objects. Here “metadata” refers to aspects of security objects that are not covered by cryptographic policies, e.g. custom metadata, description, expiry. The policy should be able to specify that certain metadata “must be present”, “must not be present”, or “must be present and have certain value”.

FIGURE 4: ADD KEY METADATA POLICY

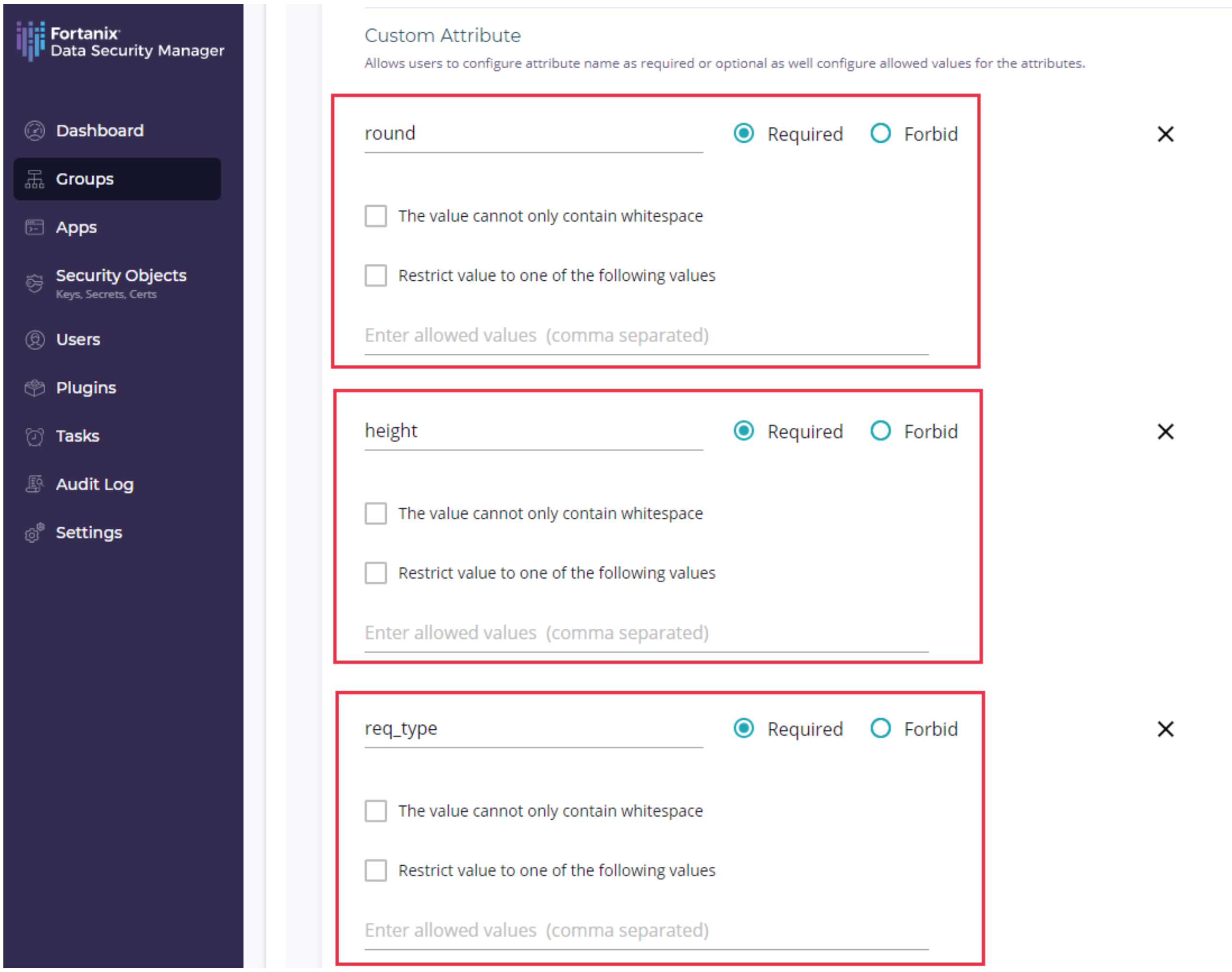
4. In the policy, add the following custom attributes as **Required**. These attributes maintain the state of the key for consensus signing.
    a. Name: **`round`**

       Required
    b. Name: **`height`**

       Required
    c. Name: **`req_type`**

       Required

NOTE: The custom attributes are case-sensitive.



Custom Attribute

Allows users to configure attribute name as required or optional as well configure allowed values for the attributes.

round ● Required ○ Forbid ×

□ The value cannot only contain whitespace

□ Restrict value to one of the following values

Enter allowed values (comma separated)

height ● Required ○ Forbid ×

□ The value cannot only contain whitespace

□ Restrict value to one of the following values

Enter allowed values (comma separated)

req_type ● Required ○ Forbid ×

□ The value cannot only contain whitespace

□ Restrict value to one of the following values

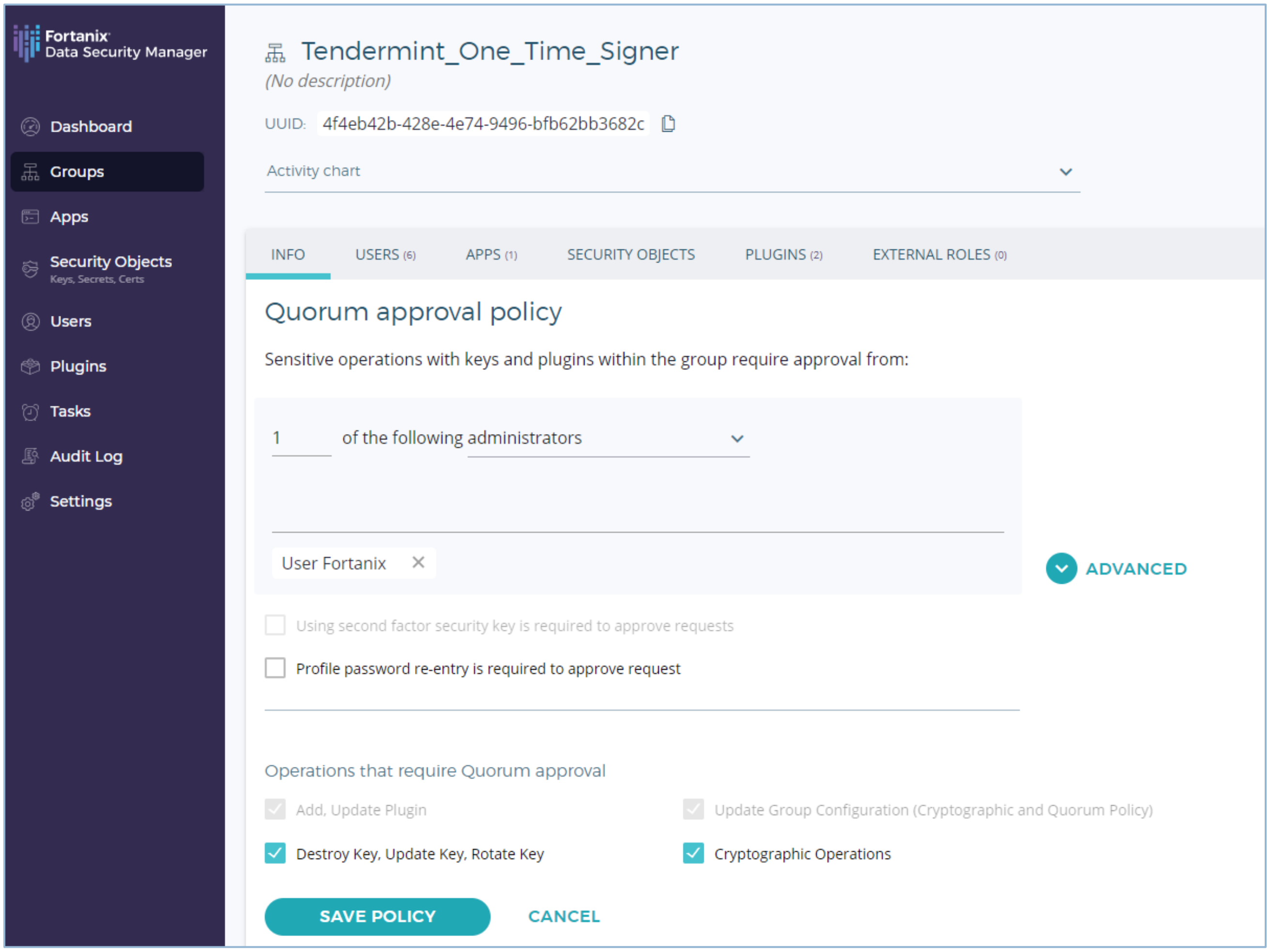
Enter allowed values (comma separated)

FIGURE 5: CREATE CUSTOM ATTRIBUTES

3.2 CONFIGURE A QUORUM POLICY FOR THE GROUP

After creating the Fortanix DSM group and adding the “Tendermint One Time Signer” plugin to this group, configure a Quorum Policy for the group to protect the plugin. This will ensure that the plugin code cannot be modified without the approval of the Group Administrator.

1. Go to the detailed view of the group, and click the **INFO** tab.
2. In the “Quorum approval policy” section, click **ADD POLICY** to add a new quorum policy.
3. Configure the Quorum approval policy and click **SAVE POLICY**.



Tendermint_One_Time_Signer
(No description)
UUID: 4f4eb42b-428e-4e74-9496-bfb62bb3682c

Quorum approval policy

Sensitive operations with keys and plugins within the group require approval from:

1 of the following administrators

User Fortanix ✕

□ Using second factor security key is required to approve requests
□ Profile password re-entry is required to approve request

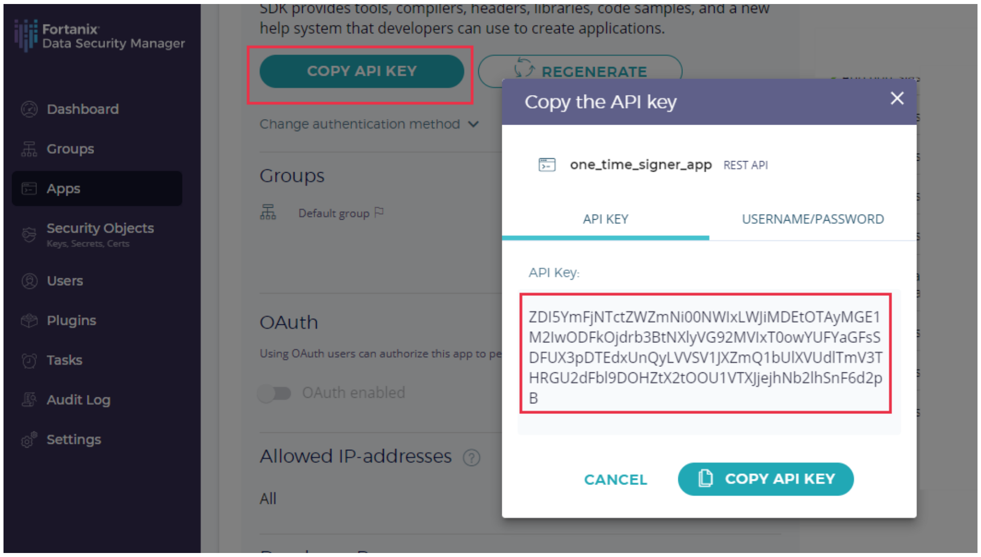
Operations that require Quorum approval

☑ Add, Update Plugin
☑ Update Group Configuration (Cryptographic and Quorum Policy)
☑ Destroy Key, Update Key, Rotate Key
☑ Cryptographic Operations

FIGURE 6: CONFIGURE QUORUM APPROVAL POLICY

3.3 CREATE AN APP IN FORTANIX DSM

Create an app in Fortanix DSM of type **REST API** for the “Tendermint One Time Signer” plugin and copy the app’s **API KEY**. This API Key is added to the `shim` configuration file later. *Refer to Section 4.1.*



SDK provides tools, compilers, headers, libraries, code samples, and a new help system that developers can use to create applications.

Copy the API key

one_time_signer_app REST API

API Key:

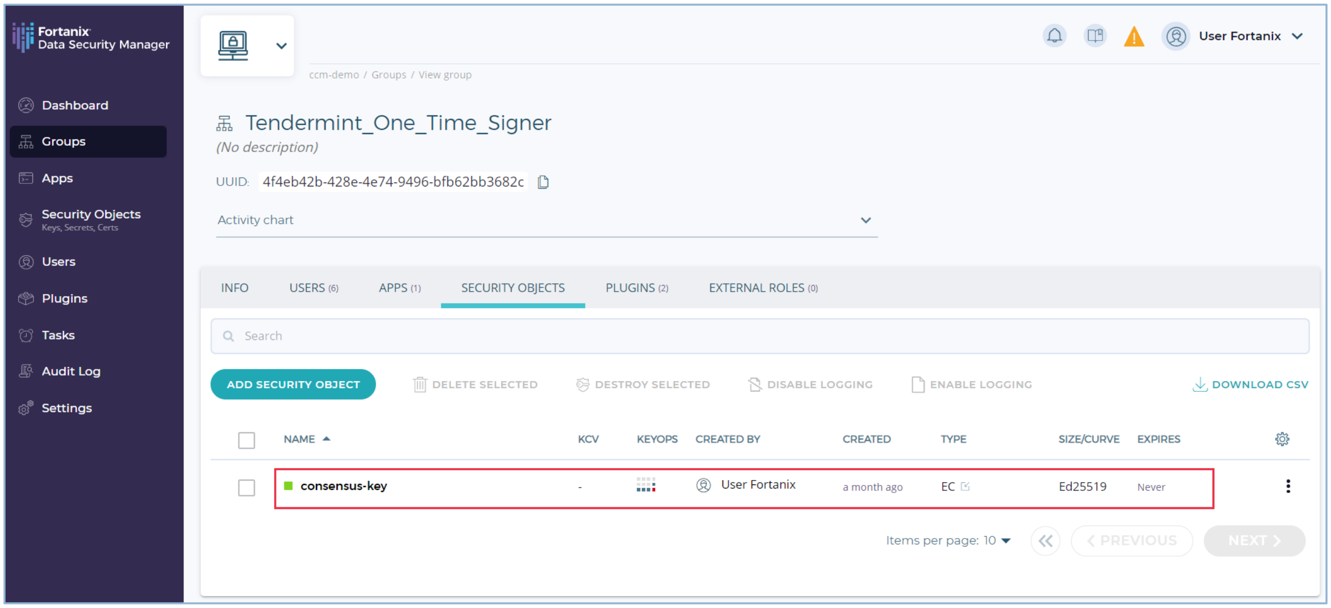
```
ZDI5YmFjNTctZWZmNi00NWIxLWJiMDEtOTAyMGE1
M2IwODFkOjdrb3BtNXlyVG92MVIxT0owYUFYaGFsS
DFUX3pDTEdxUnQyLVVSV1JXZmQ1bUlXVUdlTmV3T
HRGU2dFbl9DOHZtX2tOOU1VTXJjejhNb2lhSnF6d2p
B
```

FIGURE 7: CREATE AN APP AND COPY THE API KEY

3.4 GENERATE A KEY

1. Generate a key called **`consensus-key`** in the same group created in *Section 3.1* so that the API key of the app created in *Section 3.3* can be used to access this key. The key type must be **EC** of Curve **Ed25519** since the `shim` accepts keys of this format only. This key is used by the validator for consensus signing.

NOTE: It is mandatory for the key name to be **`consensus-key`**.



Tendermint_One_Time_Signer

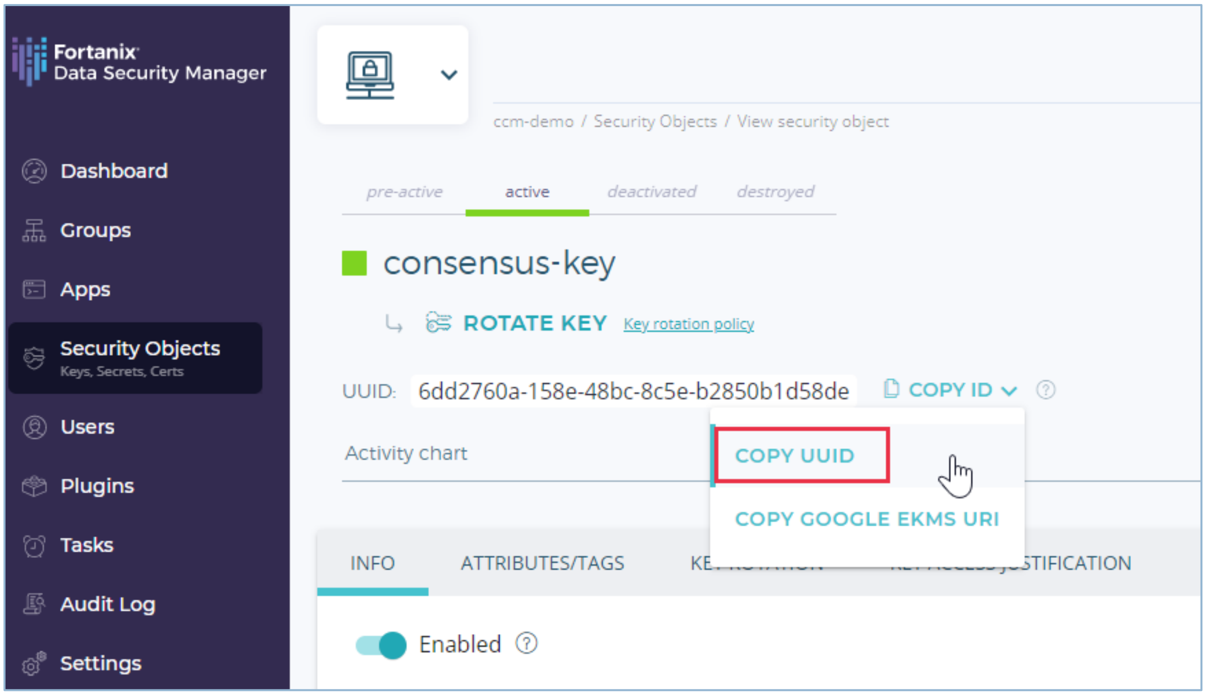
(No description)

UUID: 4f4eb42b-428e-4e74-9496-bfb62bb3682c

NAME	KCV	KEYOPS	CREATED BY	CREATED	TYPE	SIZE/CURVE	EXPIRES
consensus-key	-		User Fortanix	a month ago	EC	Ed25519	Never

FIGURE 8: CONSENSUS KEY

2. Copy the Key ID of this key to add it to the `shim` configuration file later. *Refer to Section 4.1*.

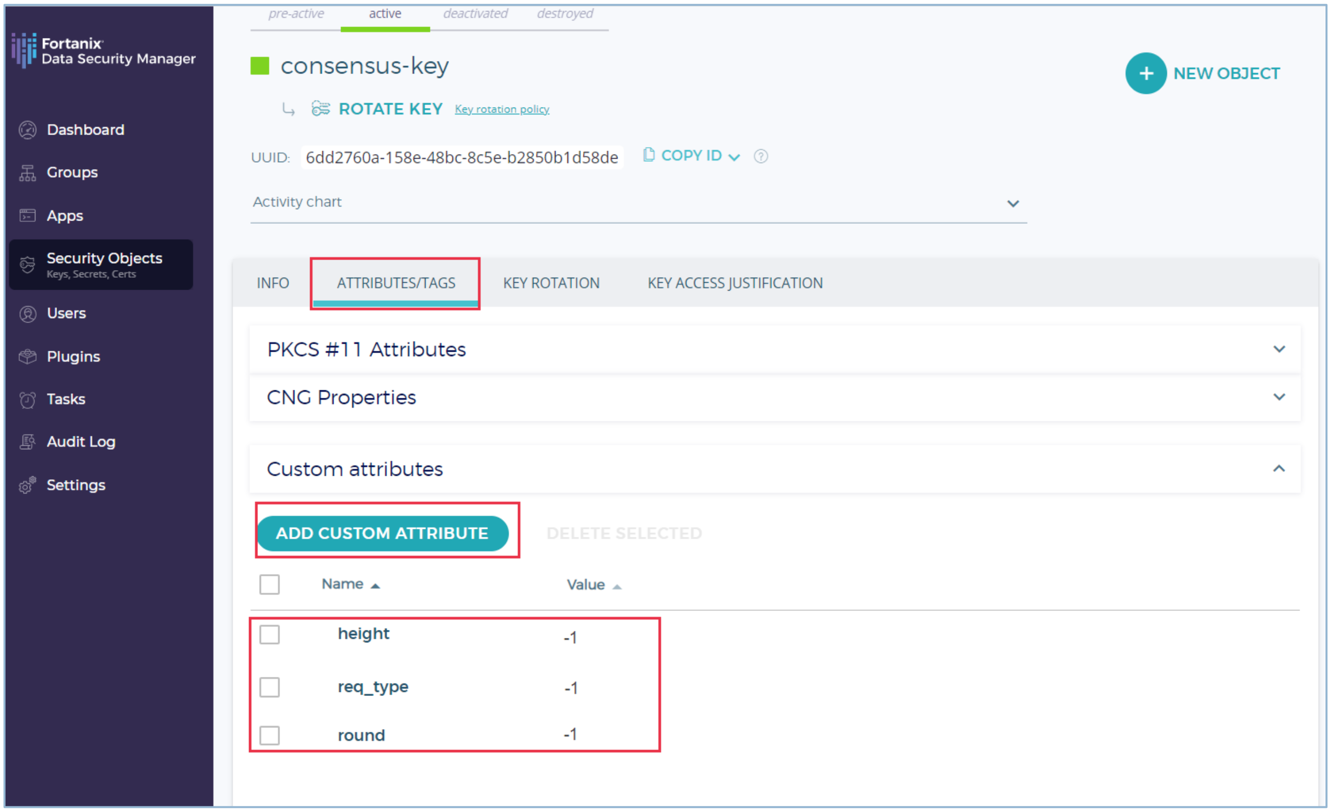


UUID: 6dd2760a-158e-48bc-8c5e-b2850b1d58de

FIGURE 9: COPY KEY ID

Alternatively, you can also import an existing Ignite key. *Refer to the section "Script to Convert Tendermint key to DER format" in the Plugin readme to convert an Ignite key to Fortanix DSM accepted key format*. The readme also has an example Ignite key.

3. In the detailed view of the key, click the **ATTRIBUTES/TAGS** tab and notice that the key has the custom attributes that were defined in the Key metadata policy while creating the group. For all the attributes, set the value as **`-1`**.



UUID: 6dd2760a-158e-48bc-8c5e-b2850b1d58de

	Name	Value
□	height	-1
□	req_type	-1
□	round	-1

FIGURE 10: KEY CUSTOM ATTRIBUTES

4.0 SET UP AND START THE SHIM

4.1 SET UP SHIM

To configure and use the `shim`, refer to the steps here. This contains the instructions to compile the `shim` utility and generate a `shim` executable.

`shim` requires a configuration file that contains the authentication details needed to authenticate to Fortanix DSM with an API key. The file `shim.toml` contains this configuration. The configuration file expects the following additional details:

- Fortanix DSM “Tendermint One Time Signer” plugin UUID – Add the Plugin UUID copied in *Section 3.1* to the `shim.toml` file.
- Fortanix DSM App API Key – Add the API Key copied in *Section 3.3* to the `shim.toml` configuration file.
- Fortanix DSM Key UUID – Add the Key UUID coped in *Section 3.4* to the `shim.toml` configuration file.

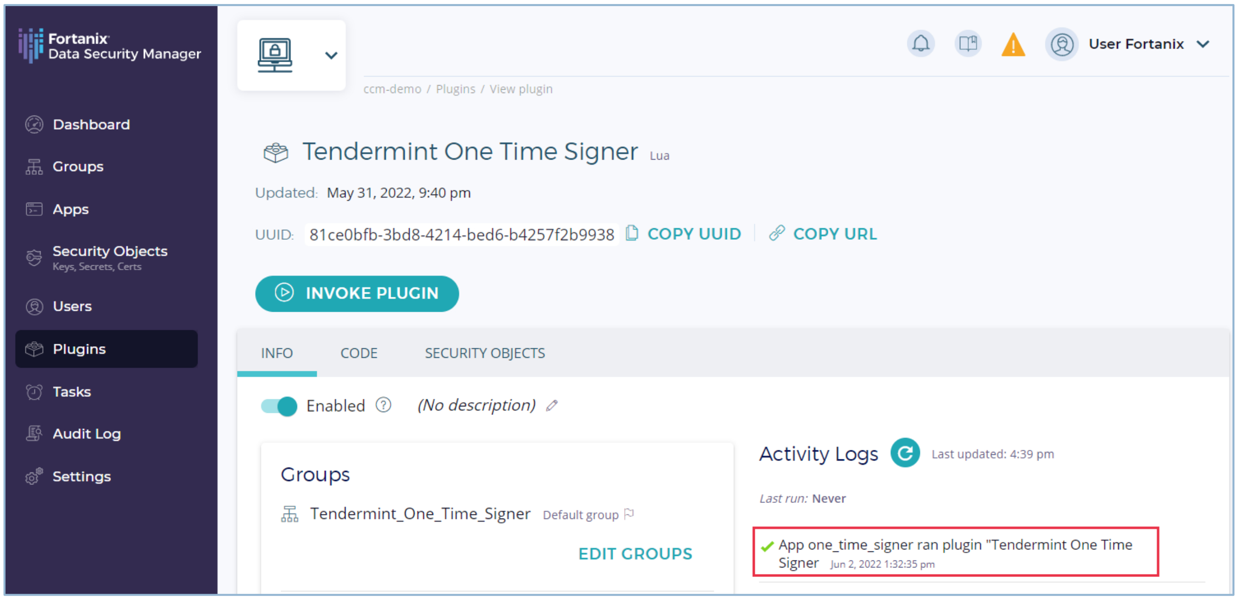
4.2 RUNNING SHIM

The `shim.toml` configuration file now has the details required for the Fortanix DSM Signer configuration:

- The Fortanix DSM “Tendermint One Time Signer” plugin ID for double-signing check.
- The Fortanix DSM App API Key to authenticate to Fortanix DSM.
- The Fortanix DSM Key ID is used by the Validator for consensus signing.

The configuration file also has the configuration for the Validator, which tells the Validator to use the Fortanix DSM Signer for consensus signing. To do this, you must pass the address of the `shim` for Ignite to listen for connections from the external validator process.

1. Run the shim configuration using the command `shim start -c </path/to/shim.toml>`. The detailed steps to run the `shim` configuration are described here.
2. Add the remote signer address for the Validator, for example, `tcp://127.0.0.1:26690`. This address must then be added to the `shim.toml` configuration file.
3. Start the Ignite Validator.
4. After the Validator starts, the proposal is now signed with a hash at a particular height, round, and step. These values are updated in the **consensus-key**’s custom attributes section.
5. The Validator receives the signature from the plugin.
6. Go to the detailed view of the plugin in Fortanix DSM and notice the **Activity Logs** section on the right. You will see the log showing that the Fortanix DSM app ran the plugin successfully.



Tendermint One Time Signer Lua

Updated: May 31, 2022, 9:40 pm

UUID: 81ce0bfb-3bd8-4214-bed6-b4257f2b9938

Enabled (No description)

Groups

Tendermint_One_Time_Signer Default group

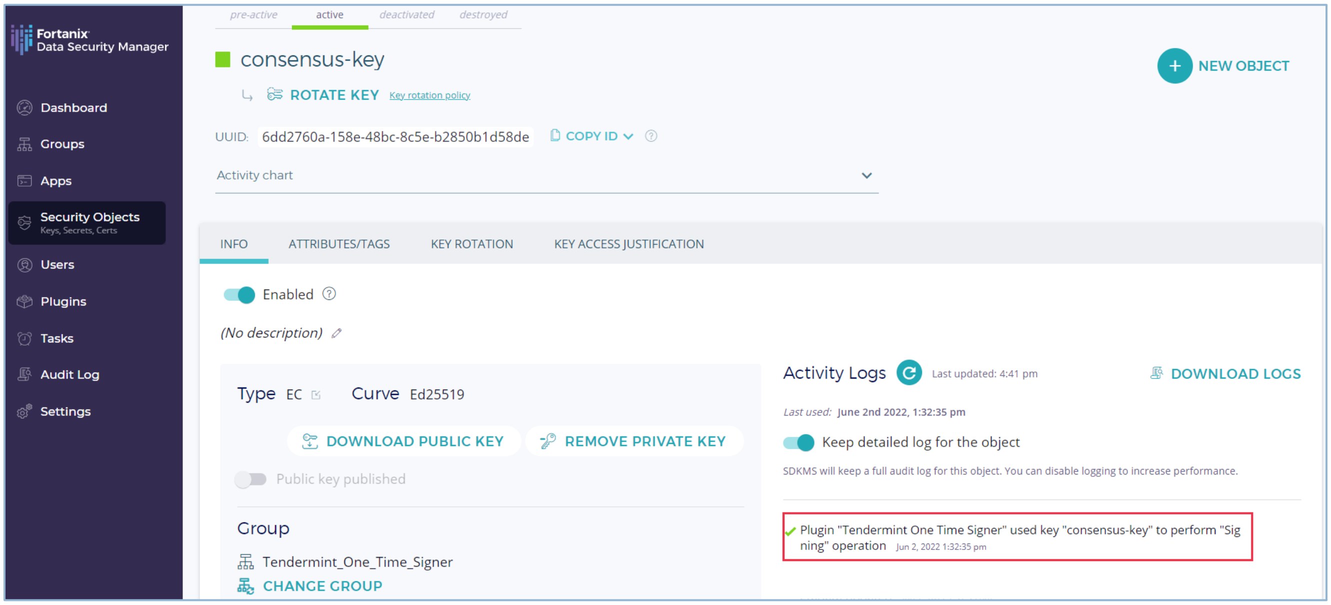
Activity Logs Last updated: 4:39 pm

Last run: Never

✓ App one_time_signer ran plugin "Tendermint One Time Signer Jun 2, 2022 1:32:35 pm

FIGURE 11: PLUGIN EXECUTION LOG

7. 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.



consensus-key

UUID: 6dd2760a-158e-48bc-8c5e-b2850b1d58de

Enabled

(No description)

Type EC Curve Ed25519

Public key published

Group

Tendermint_One_Time_Signer

Activity Logs Last updated: 4:41 pm

Last used: **June 2nd 2022, 1:32:35 pm**

Keep detailed log for the object

SDKMS will keep a full audit log for this object. You can disable logging to increase performance.

✓ Plugin "Tendermint One Time Signer" used key "consensus-key" to perform "Signing" operation Jun 2, 2022 1:32:35 pm

FIGURE 12: 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/6678703159700-Using-Fortanix-Ignite-One-Time-Signer-Plugin

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

NOTICE: This document was produced by Fortanix, Inc. (Fortanix) and contains information which is proprietary and confidential to Fortanix. The document contains information that may be protected by patents, copyrights, and/or other IP laws. If you are not the intended recipient of this material, please destroy this document and inform info@fortanix.com immediately.