Intel Cloud Integrity Technology

**Master Password**

# Background

The server uses a variety of encryption keys and certificates and these need confidentiality and integrity protection while at rest.

# Architecture

Note: in this document, wherever a variable like KMS\_PASSWORD is mentioned, it can be substituted with KMSPROXY\_PASSWORD for the proxy, TRUSTAGENT\_PASSWORD for trustagent, MTWILSON\_PASSWORD for Mt Wilson, DIRECTOR\_PASSWORD for Trust Director. Similarly, wherever the file ~/.kms\_password is mentioned, it can be substituted with ~/.kmsproxy\_password, ~/.tagent\_password, ~/.mtwilson\_password, or ~/.director\_password for the corresponding applications.

## Purpose of master password

The Java keystore format provides adequate protection for the keys but each keystore itself must be protected with a password. These passwords, along with any other password such as a login credential to a remote web service, are stored in a password keystore (see password vault blueprint). The password to access the password vault is called the “master password”, and it is the single password that must be provided when starting the server in order to unlock everything else.

## Generation of master password

By default, a random master password is generated during installation. If administrator sets the environment variable KMS\_PASSWORD before running the installer, or writes it into the .env file before running the installer, the password generation is skipped and the provided password is used.

## Storage of master password

When stored on disk, the master password resides in a file ~/.kms\_password which must be owned by the user and be non-readable to others. For example, if installing the server as root, it would be in /root/.kms\_password.

If the installer randomly generated the master password, or if the user was prompted for the master password, the installer will automatically store it on disk.

If the user provided the master password via the KMS\_PASSWORD environment variable or .env file setting, the password will NOT be automatically stored on disk because we assume user will continue to provide KMS\_PASSWORD when running kms commands.

## Use of master password

The administrator must set the master password before running any server commands such as starting the server or changing the configuration. The administrator provides the master password by setting an environment variable KMS\_PASSWORD or KMSPROXY\_PASSWORD, for example:

export KMS\_PASSWORD=password

kms start

If the password is not provided in the environment, the service will check the file ~/.kms\_password and use the password contained in it if present.

# Design Choices

## Generation and storage of master password

If the KMS\_PASSWORD is set in the environment during installation, that password will be used and it will not be stored on disk. The user is expected to provide KMS\_PASSWORD in the environment prior to starting the service.

The following options are for what should be done when the user does not provide KMS\_PASSWORD during installation.

Table

|  |  |  |
| --- | --- | --- |
| **Alternative** | **+** | **-** |
| Generate random master password and store it into ~/.kms\_password | Automated installation | Storing master password on disk is less secure because an attacker that gains access to the host can then access all the keys even after service is shut down; also when changing it need to zeroize not just delete |
| Prompt user for master password and do not store it | Not storing master password on disk is more secure | Interactive installation; user can still automate installation by providing KMS\_PASSWORD when installing |
| Check KMS\_NOPROMPT variable, if set then use first alternative to generate and store password; if not set then use second alternative to prompt for password but in this case store it on disk |  |  |