HowTo Guide: Protecting the Smartcrypt® Manager Master Key with Extended Key Protection

This custom enhancement extends the existing capability of the PKWARE Smartcrypt Enterprise Manager from protecting the SMDS master key using a single FIPS 140-2 Level 3 validated hardware security module (HSM) to using two HSMs.

This document supplements and changes some configuration steps in the Smartcrypt Manager Installation Guide and assumes familiarity with this document.

System Prerequisites

- Smartcrypt Manager 17.1 (See *Getting Started with Smartcrypt Manager* document to set up)
- FIPS Level 3-certified Hardware Security Modules of your choice. Recommended HSMs: Thales Vormetric DSM and Gemalto Luna SA
- HSMs must have a standard PKCS11 interface. You will need to know the path to the PKCS11 library (DLL) on each HSM.
- You will need to know the Slot Number and User PIN for each HSM.

Configuring the Hardware Security Modules

PKWARE provides the sample hsm.json configuration file to connect Smartcrypt to each of your HSMs. Choose your favorite text editor to configure your system.

IMPORTANT:
When editing settings in this file, ensure that all punctuation (quotation marks, braces, colons and the like) remain in place. In the example below, when you insert the User PIN, it should appear like this:
`"pin":"1111"`

Copy this file to the web server in a location readable by IIS but not accessible via HTTP.

```
{

    "wrap":

    {

        "device":{"name":"Gemalto Luna SA","module":"C:\\Program Files\\SafeNet\\LunaClient\\cryptoki.dll","slot":
1,"pin":"verify Slot # and insert PIN here"},

        "wrap":{"label":"SmartcryptWrapKey","create":true},

        "items":[{"name":"master","label":"SmartcryptMasterKey","create":true}]

    },


    "unwrap":

    {

        "device":{"name":"Thales","module":"C:\\Program
Files\\Vormetric\\DataSecurityExpert\\Agent\\pkcs11\\bin\\vorpkcs11.dll","slot":0,"pin":"verify Slot # and insert
PIN here"},

        "unwrap":{"label":"SmartcryptUnwrapKey","create":true}

    },

}
```

Configure Encryption Keys

Use the "wrap" section to identify and configure the Key Encryption and Master Encryption keys in your system.

Option	Description
device	The HSM containing the key encryption and master encryption keys
name	Identifier for a single HSM
module	Location of the PKCS11 library for the named HSM
slot	Specify the Slot number
pin	Insert User PIN

wrap	Identifies the key encryption key, an RSA public key on this device
label	Identifies the key encryption key on this device. You may point to any existing key, or use the default key name.
create	When set to true (recommended), if the script does not find a key in the store with the specified Label, a new key with that label will be added to the store. When set to false, and the Label doesn't exist, you will have to manually create the key. See Appendix for further configuration information.
items	Identifies the Master encryption key on this device.
name	Identifies this key for PKWARE software. **Must not change**.
label	Identifies the encryption key on this device. You may point to any existing key, or use the default key name.
create	When set to true (recommended), if the script does not find a key in the store with the specified Label, a new key with that label will be added to the store. When set to false, and the Label doesn't exist, you will have to manually create the key. See Appendix: for further configuration information.

Configure the Key Decryption Key

Use the "unwrap" section to identify and configure the key decryption key to be used.

Option	Description
device	The HSM containing the key decryption key
name	Identifier for a single HSM
module	Location of the PKCS11 library for the named HSM
slot	Specify the Slot number
pin	Insert User PIN
unwrap	Identifies the key decryption key (KDK), an RSA private key, on this device
label	File name of the decryption key. You may point to any existing key, or use the default key name.
create	When set to true (recommended), if the script does not find a key in the store with the specified Label, a new key with that label will be added to the store. When set to false, and the Label doesn't exist, you will have to manually create the key. See Appendix for further configuration information.

Edit Smartcrypt Web.config file

After completing the sample hsm.json, open Web.config in the Smartcrypt folder.

Edit the *<appSettings>* section:

- Delete this line:

```
<add key="SatellitePassword" value="" />
```

- Add this line, pointing to the location of hsm.json.

```
<add key="PKCS11MasterKeyConfiguration" value="C:\inetpub\wwwroot\hsm.json" />
```

Appendix: Creating Keys Manually

When *`create`* is set to true in the sample *hsm.json* configuration file, Smartcrypt will create any labeled key that it needs to work with, but allows (with `"create:false"`) for administrators to create their own valid keys. When Smartcrypt generates a labeled key, it generates an **RSA** key pair with **CKA_MODULUS_BITS: 2048** and **AES** key with **CKA_VALUE_LEN: 32**.

Smartcrypt requires manually-created keys to include the fields listed here. Items in **BOLD TYPE** must be set as defined.

Unwrap/private key:

CKA_CLASS: CKO_PRIVATE_KEY

CKA_KEY_TYPE: CKK_RSA

CKA_TOKEN: True

CKA_PRIVATE: True

CKA_SENSITIVE: True

CKA_SIGN: True

CKA_UNWRAP: True

CKA_DECRYPT: True

Wrap/public key – must match private key:

CKA_CLASS: CKO_PUBLIC_KEY

CKA_KEY_TYPE: CKK_RSA

CKA_TOKEN: True

CKA_ENCRYPT: True

CKA_VERIFY: True

CKA_WRAP: True

Symmetric key:

CKA_CLASS: CKO_SECRET_KEY

CKA_KEY_TYPE: CKK_AES

CKA_TOKEN: True

CKA_PRIVATE: True

CKA_SENSITIVE: True

CKA_EXTRACTABLE: True