DEFINING A COSMIAN KMS TEST SUITE

Objective

The objective is to create a structured set of tests to ensure that the software functions as expected. The test suite will verify the functionalities and behavior of the system through well-defined test cases.

The test suite will contain the following components:

1. **Test Cases Definition:** Step-by-step instructions, expected results, and information about the Software Under Test (SUT), including version and execution date.
2. **Test Case Requirements:** Clear definitions of the requirements that each test case fulfills.
3. **Test Case Scenarios:** A detailed description of the scenarios to be tested.
4. **Test Case Descriptions:** Generic definition of each test case along with its purpose.
5. **Test Case Sequence:** The sequence in which test cases will be executed.
6. **Execution Template:** A template to document the execution of test cases.
7. **Automation Scripts:** Scripts that allow the automatic execution of the test cases and generation of the test report.

COSMIAN KMS Functionalities

The Cosmian KMS provides various functionalities, including but not limited to:

1. Access Rights Management

- **Manage Users' Access Rights:** Control and manage user permissions for cryptographic objects.

2. Covercrypt Management

- **Keys and Policies:** Manage Covercrypt keys and associated policies, including key rotation and encryption/decryption of data.

3. Certificates Management

- **Create, Import, Destroy, and Revoke Certificates:** Manage certificate lifecycle for encryption and decryption tasks.

4. Elliptic Curve Key Management

- **Elliptic Curve Keys:** Manage elliptic curve keys and perform encryption/decryption using ECIES (Elliptic Curve Integrated Encryption Scheme).

5. Attributes Retrieval

- **Get Attributes:** Retrieve attributes and tags associated with KMIP (Key Management Interoperability Protocol) objects.

6. Object Location

- **Locate Objects:** Search for cryptographic objects within the KMS.

7. Database Initialization

- **New Database:** Initialize a new user-encrypted database and return the secret (for SQLCipher).

8. RSA Key Management

- **Manage RSA Keys:** Handle RSA key generation, encryption, and decryption operations.

9. Server Information

- **Server Version:** Retrieve and display the version of the KMS server.

10. Symmetric Key Management

- **Symmetric Keys:** Manage symmetric keys for encryption and decryption tasks.

11. Authentication

- **Login/Logout:** Authenticate with the KMS server using the OAuth2 authorization code flow and log out from the Identity Provider.

12. Documentation Generation

- **Markdown Generation:** Automatically generate documentation in Markdown format.

13. Google API Management

- **Google Elements:** Manage Google keypairs and identities via the Gmail API.

Source: https://docs.cosmian.com/cosmian_key_management_system/

Selected Functionalities for the Test Suite

The following functionalities are selected for defining the test suite:

1. Certificates Management
2. Symmetric Key Management
3. Elliptic Curve Key Management (similar to Symmetric Key Management)
4. RSA Key Management (similar to Symmetric Key Management)

Requirements to Run the Test Suite

Python 3 must be installed on the system.

Test Suite Structure

The test suite is organized into three main Python scripts, each containing specific functions to perform the relevant operations for different categories:

1. CKMS_general.py

Contains the general functions required to support the tasks in the CKMS_keys.py and CKMS_certificates.py scripts.

2. CKMS_keys.py

Includes all the functions related to key management operations.

3. CKMS_certificates.py

Contains all the functions related to certificate management operations.

4. Test Case Scripts (CKMS_TC_XX_XX_XXXXXXXXXX.py)

In addition to the main scripts, there are separate test case scripts. These scripts evaluate the functions from the main scripts (CKMS_general.py, CKMS_keys.py, and CKMS_certificates.py) using different test scenarios.

- **Test Cases:** Each category (keys and certificates) has one corresponding test case.
- **Test Scenarios:** Each test case includes multiple test scenarios to verify different aspects of functionality.

The **unittest** library in Python has been utilized for writing and executing test cases due to its simplicity and ease of use in executing structured test cases.

5. RUN.py

Executing the RUN.py script will run all the test cases in the test suite sequentially.

Overview of test cases

1. Certificates Management

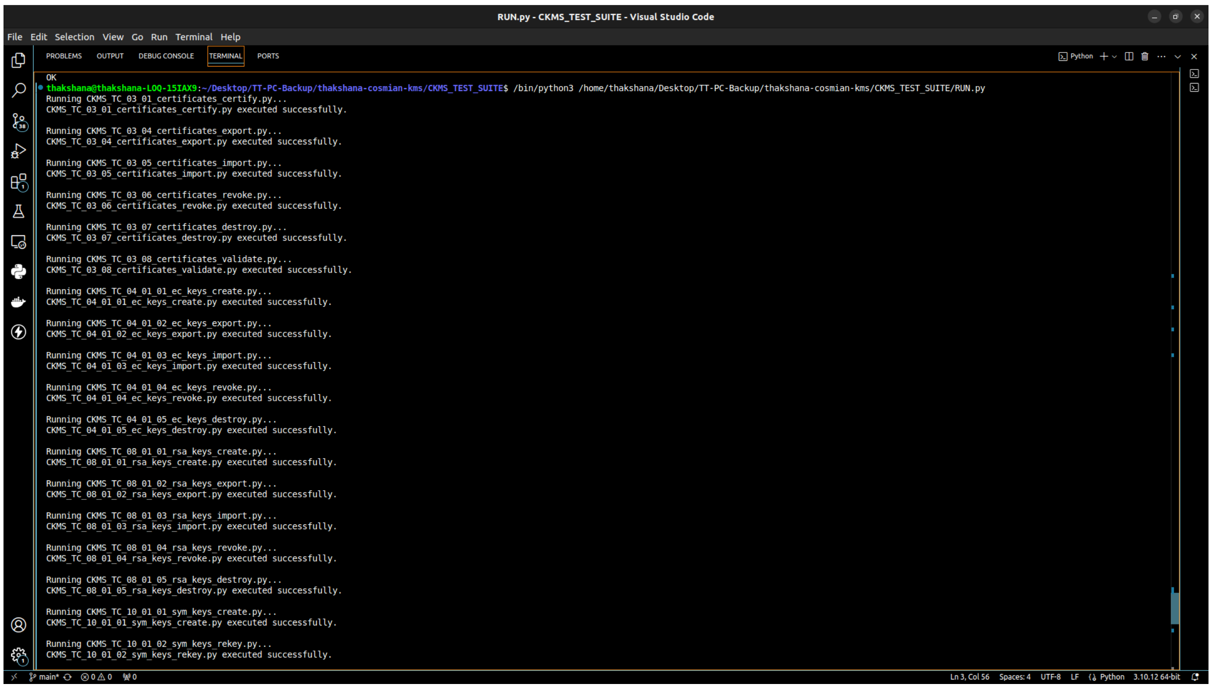
- Certify Certificates
- Export Certificates
- Import Certificates
- Revoke Certificates
- Destroy Certificates
- Validate Certificates

2. Symmetric Key Management

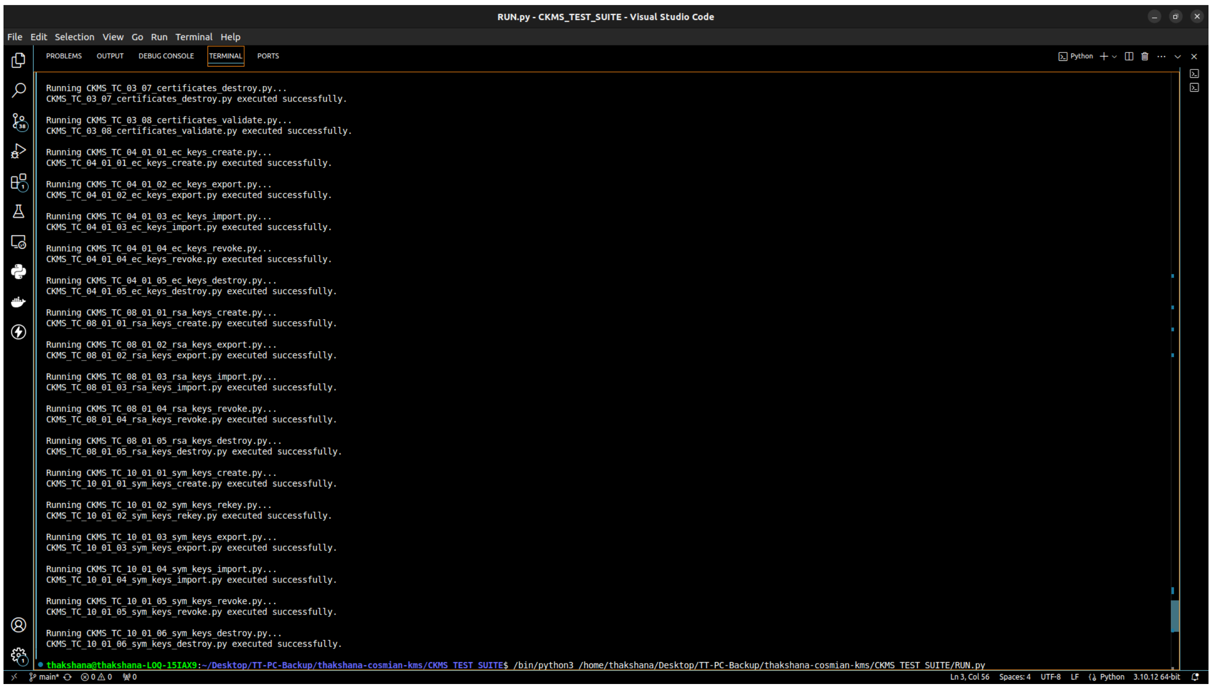
- Create Symmetric Keys
- Rekey Symmetric Keys
- Export Symmetric Keys
- Import Symmetric Keys
- Revoke Symmetric Keys
- Destroy Symmetric Keys

Execution of the Test Suite

After creating all the test cases, the test suite can be executed sequentially using the **RUN.py** script.



```
OK
thakshana@thakshana-LOQ-15IAX9:~/Desktop/TT-PC-Backup/thakshana-cosmian-kms/CKMS_TEST_SUITE$ /bin/python3 /home/thakshana/Desktop/TT-PC-Backup/thakshana-cosmian-kms/CKMS_TEST_SUITE/RUN.py
Running CKMS_TC_03_01_certificates_certify.py...
CKMS_TC_03_01_certificates_certify.py executed successfully.

Running CKMS_TC_03_04_certificates_export.py...
CKMS_TC_03_04_certificates_export.py executed successfully.

Running CKMS_TC_03_05_certificates_import.py...
CKMS_TC_03_05_certificates_import.py executed successfully.

Running CKMS_TC_03_06_certificates_revoke.py...
CKMS_TC_03_06_certificates_revoke.py executed successfully.

Running CKMS_TC_03_07_certificates_destroy.py...
CKMS_TC_03_07_certificates_destroy.py executed successfully.

Running CKMS_TC_03_08_certificates_validate.py...
CKMS_TC_03_08_certificates_validate.py executed successfully.

Running CKMS_TC_04_01_01_ec_keys_create.py...
CKMS_TC_04_01_01_ec_keys_create.py executed successfully.

Running CKMS_TC_04_01_02_ec_keys_export.py...
CKMS_TC_04_01_02_ec_keys_export.py executed successfully.

Running CKMS_TC_04_01_03_ec_keys_import.py...
CKMS_TC_04_01_03_ec_keys_import.py executed successfully.

Running CKMS_TC_04_01_04_ec_keys_revoke.py...
CKMS_TC_04_01_04_ec_keys_revoke.py executed successfully.

Running CKMS_TC_04_01_05_ec_keys_destroy.py...
CKMS_TC_04_01_05_ec_keys_destroy.py executed successfully.

Running CKMS_TC_08_01_01_rsa_keys_create.py...
CKMS_TC_08_01_01_rsa_keys_create.py executed successfully.

Running CKMS_TC_08_01_02_rsa_keys_export.py...
CKMS_TC_08_01_02_rsa_keys_export.py executed successfully.

Running CKMS_TC_08_01_03_rsa_keys_import.py...
CKMS_TC_08_01_03_rsa_keys_import.py executed successfully.

Running CKMS_TC_08_01_04_rsa_keys_revoke.py...
CKMS_TC_08_01_04_rsa_keys_revoke.py executed successfully.

Running CKMS_TC_08_01_05_rsa_keys_destroy.py...
CKMS_TC_08_01_05_rsa_keys_destroy.py executed successfully.

Running CKMS_TC_10_01_01_sym_keys_create.py...
CKMS_TC_10_01_01_sym_keys_create.py executed successfully.

Running CKMS_TC_10_01_02_sym_keys_rekey.py...
CKMS_TC_10_01_02_sym_keys_rekey.py executed successfully.
```



```
Running CKMS_TC_03_07_certificates_destroy.py...
CKMS_TC_03_07_certificates_destroy.py executed successfully.

Running CKMS_TC_03_08_certificates_validate.py...
CKMS_TC_03_08_certificates_validate.py executed successfully.

Running CKMS_TC_04_01_01_ec_keys_create.py...
CKMS_TC_04_01_01_ec_keys_create.py executed successfully.

Running CKMS_TC_04_01_02_ec_keys_export.py...
CKMS_TC_04_01_02_ec_keys_export.py executed successfully.

Running CKMS_TC_04_01_03_ec_keys_import.py...
CKMS_TC_04_01_03_ec_keys_import.py executed successfully.

Running CKMS_TC_04_01_04_ec_keys_revoke.py...
CKMS_TC_04_01_04_ec_keys_revoke.py executed successfully.

Running CKMS_TC_04_01_05_ec_keys_destroy.py...
CKMS_TC_04_01_05_ec_keys_destroy.py executed successfully.

Running CKMS_TC_08_01_01_rsa_keys_create.py...
CKMS_TC_08_01_01_rsa_keys_create.py executed successfully.

Running CKMS_TC_08_01_02_rsa_keys_export.py...
CKMS_TC_08_01_02_rsa_keys_export.py executed successfully.

Running CKMS_TC_08_01_03_rsa_keys_import.py...
CKMS_TC_08_01_03_rsa_keys_import.py executed successfully.

Running CKMS_TC_08_01_04_rsa_keys_revoke.py...
CKMS_TC_08_01_04_rsa_keys_revoke.py executed successfully.

Running CKMS_TC_08_01_05_rsa_keys_destroy.py...
CKMS_TC_08_01_05_rsa_keys_destroy.py executed successfully.

Running CKMS_TC_10_01_01_sym_keys_create.py...
CKMS_TC_10_01_01_sym_keys_create.py executed successfully.

Running CKMS_TC_10_01_02_sym_keys_rekey.py...
CKMS_TC_10_01_02_sym_keys_rekey.py executed successfully.

Running CKMS_TC_10_01_03_sym_keys_export.py...
CKMS_TC_10_01_03_sym_keys_export.py executed successfully.

Running CKMS_TC_10_01_04_sym_keys_import.py...
CKMS_TC_10_01_04_sym_keys_import.py executed successfully.

Running CKMS_TC_10_01_05_sym_keys_revoke.py...
CKMS_TC_10_01_05_sym_keys_revoke.py executed successfully.

Running CKMS_TC_10_01_06_sym_keys_destroy.py...
CKMS_TC_10_01_06_sym_keys_destroy.py executed successfully.
thakshana@thakshana-LOQ-15IAX9:~/Desktop/TT-PC-Backup/thakshana-cosmian-kms/CKMS_TEST_SUITE$ /bin/python3 /home/thakshana/Desktop/TT-PC-Backup/thakshana-cosmian-kms/CKMS_TEST_SUITE/RUN.py
```

Note:

Elliptic Curve Key Management and RSA Key Management follow the same process as Symmetric Key Management. For simplicity, only Symmetric Key Management is detailed.

===

Prepared By:

Salitha Gunawardhana
[salitha.gunawardhana@thakshana.com]

===