Mohit Agnihotri

Mohitkumar39@gmail.com

Test Specification

**General guidelines followed:**

1. Test name: <Component>\_<Feature>\_< Test case Objective Testcase>
2. Use this template to describe new test cases.

Test case description template:

|  |  |  |  |
| --- | --- | --- | --- |
| Test case |  | Test case Objective |  |
| Component | Rnpkeys | Test Case Name |  |
| Feature | Generate-key |  |  |
| Short Description: |  | | |

|  |
| --- |
| Precondition |
| 1. Initialize RNP 2. Set the default value for sshkeydir, res, format, hash via rnp\_setvar(). |

|  |  |
| --- | --- |
| Testing Step | Expected behavior |
| 1. Set the userId via rnp\_setvar() 2. Call the API to generate key (rnp\_generate\_key) | It is expected that the key is generated using the options set via rnp\_setvar() |

|  |  |
| --- | --- |
| Verification Step | Verification logic |
| 1. Load the newly generated RNP keys |  |

|  |
| --- |
| Comments (if any) |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | 1 | Test case Objective | VerifySupportedHashAlg |
| Component | Rnpkeys | Test Case Name | rnpkeys\_generatekey\_verifySupportedHashAlg |
| Feature | Generate-key |  |  |
| Short Description: | The test aims to test key generation with all possible hash algorithm.  Following hash algorithm are tested for the key generation.  "MD5", "SHA-1", "RIPEMD160", "SHA256", "SHA384", "SHA512", "SHA224" | | |

|  |
| --- |
| Precondition |
| 1. Initialize RNP 2. Set the default value for sshkeydir, res format via rnp\_setvar(). |

|  |  |
| --- | --- |
| Testing Step | Expected behavior |
| 1. Set the hash algorithm via rnp\_setvar() 2. Call the API to generate key (rnp\_generate\_key) | It is expected that the key is generated using the options set via rnp\_setvar() |

|  |  |
| --- | --- |
| Verification Step | Verification logic |
| 1. Load the newly generated RNP keys 2. Find the existence of the key via finding the key with the userId. Note: If userid variable is not set, default is always. | 1. This ensures the keys are loaded in the rnp control structure for verification. 2. Ensures the key exist by finding it. |

|  |
| --- |
| Comments (if any)   1. It is required to delete the old keys if the test case iterates over the hashing algorithm. |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | 2 | Test case Objective | VerifyUserIdOption |
| Component | Rnpkeys | Test Case Name | rnpkeys\_generatekey\_verifyUserIdOption |
| Feature | Generate-key |  |  |
| Short Description: | The test aims to test key generation with commandline options UserId.  Following different userid are tested.   * Rnpkeys\_Generatekey\_VerifyUserIdOption \_MD5", * " Rnpkeys\_Generatekey\_VerifyUserIdOption \_SHA-1", * " Rnpkeys\_Generatekey\_VerifyUserIdOption \_RIPEMD160", * " Rnpkeys\_Generatekey\_VerifyUserIdOption \_SHA256", * " Rnpkeys\_Generatekey\_VerifyUserIdOption \_SHA384", * " Rnpkeys\_Generatekey\_VerifyUserIdOption \_SHA512", * " Rnpkeys\_Generatekey\_VerifyUserIdOption \_SHA224" | | |

|  |
| --- |
| Precondition |
| 1. Initialize RNP 2. Set the default value for sshkeydir, res, format, hash via rnp\_setvar(). |

|  |  |
| --- | --- |
| Testing Step | Expected behavior |
| 1. Set the userId via rnp\_setvar() 2. Call the API to generate key (rnp\_generate\_key) | It is expected that the key is generated using the options set via rnp\_setvar() |

|  |  |
| --- | --- |
| Verification Step | Verification logic |
| 1. Load the newly generated RNP keys 2. Find the existence of the key via finding the key with the userId. | 1. This ensures the keys are loaded in the rnp control structure for verification. 2. Ensures the key exist by finding it. |

|  |
| --- |
| Comments (if any) |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | 3 | Test case Objective | VerifykeyRingOptions |
| Component | Rnpkeys | Test Case Name | rnpkeys\_generatekey\_verifykeyRingOptions |
| Feature | Generate-key |  |  |
| Short Description: | The test aims to test key generation with the user specified keyring. | | |

|  |
| --- |
| Precondition |
| 1. Initialize RNP 2. Set the default value for sshkeydir, res, format, hash via rnp\_setvar(). |

|  |  |
| --- | --- |
| Testing Step | Expected behavior |
| 1. Set the keyring via rnp\_setvar() 2. Call the API to generate key (rnp\_generate\_key) | It is expected that the key is generated using the options set via rnp\_setvar() |

|  |  |
| --- | --- |
| Verification Step | Verification logic |
| 1. Delete the default keyring i.e. pubring.gpg and secring.gpg found in the homedir 2. Load the newly generated RNP keys 3. Find the existence of the key. | 1. To ensure that default keyring is NOT available. 2. Ensure RNP loads the new keyring as specified by the options. 3. Ensure the keys were successfully written in the keyring. |

|  |
| --- |
| Comments (if any) |

|  |  |  |  |
| --- | --- | --- | --- |
| Test case | 4 | Test case Objective | VerifykeyHomeDirOption |
| Component | Rnpkeys | Test Case Name | rnpkeys\_generatekey\_verifykeyHomeDirOption |
| Feature | Generate-key |  |  |
| Short Description: |  | | |

|  |
| --- |
| Precondition |
| 1. Create new home dir with read/write permissions. 2. Delete the keys (if any) in the previous default directory. 3. Set the default value for sshkeydir, res, format, hash via rnp\_setvar(). 4. Initialize RNP |

|  |  |
| --- | --- |
| Testing Step | Expected behavior |
| 1. Call the API to generate key (rnp\_generate\_key) | It is expected that the key is generated using the options set via rnp\_setvar() |

|  |  |
| --- | --- |
| Verification Step | Verification logic |
| 1. Load the newly generated RNP keys 2. Find the newly generated key using default userid. | 1. This loads the new keys in the RNP 2. Successful execution of the find ensures the key was generated and added to the default keyring. |

|  |
| --- |
| Comments (if any) |