C

Oracle Database FIPS 140-2 Settings

Oracle supports the Federal Information Processing Standard (FIPS) standard for 140-2.

- About the Oracle Database FIPS 140-2 Settings
  Federal Information Processing Standards (FIPS) are standards and guidelines for federal computer systems that are developed by the U.S. National Institute of Standards and Technology (NIST).
- Configuration of FIPS 140-2 Using the Consolidated FIPS_140 Parameter
  The consolidated `FIPS_140` parameter can be set for several different Oracle Database environments.
- Legacy FIPS 140-2 Configurations
  The legacy FIPS 140-2 configurations apply to Transparent Data Encryption (TDE), `DBMS_CRYPTO`, network native encryption, and Transport Layer Security (TLS).
- Postinstallation Checks for FIPS 140-2
  After you configure the FIPS 140-2 settings, you must verify permissions in the operating system.
- Verifying FIPS 140-2 Connections
  You can use trace files and other methods to verify the FIPS 140-2 connections.
- Managing Deprecated Weaker Algorithm Keys
  In Oracle Database release 23ai, several algorithms for both FIPS and non-FIPS have been deprecated.

C.1 About the Oracle Database FIPS 140-2 Settings

Federal Information Processing Standards (FIPS) are standards and guidelines for federal computer systems that are developed by the U.S. National Institute of Standards and Technology (NIST).

FIPS was developed in accordance with the Federal Information Security Management Act (FISMA). Although FIPS was developed for use by the federal government, many private sector entities voluntarily use these standards.

FIPS 140-2 specifies the security requirements that will be satisfied by a cryptographic module, providing four increasing, qualitative levels intended to cover a range of potential applications and environments. Security Level 1 conforms to the FIPS 140-2 algorithms, key sizes, integrity checks, and other requirements that are imposed by the regulations. FIPS 140-2 Security Level 1 requires no physical security mechanisms in the module beyond the requirement for production-grade equipment. As a result, this level allows software cryptographic functions to be performed in a general-purpose computer running on a specified operating environment.

When FIPS 140-2 settings are configured for Oracle Database, the database uses FIPS 140-2 Level 1 validated cryptographic libraries to protect data at rest and in transit over the network. Oracle Database uses these cryptographic libraries for native network encryption, Transparent Data Encryption (TDE) of columns and tablespaces (including Oracle SecureFiles), Transport Layer Security (TLS), and the `DBMS_CRYPTO` PL/SQL package.

Oracle Database has integrated the following FIPS 140-2 Software Level 1 validated cryptographic modules for authentication, network encryption, and data encryption:

- Oracle OpenSSL FIPS Provider Version 3.0:
  - NIST's Cryptographic Module Validation Program FIPS Certificate #4506. See the NIST Computer Information Technology Laboratory Security Resource Center page Cryptographic Module Validation Program Certificate #4506
  - Security Policy mapped to Certificate #4506. See Oracle FIPS 140-2 Non-Proprietary Security Policy
- RSA/Dell BSAFE Crypto-J 6.3 and RSA/Dell BSAFE Java Crypto Module 6.3:
  - NIST's Cryptographic Module Validation Program FIPS Certificate #4697. See the NIST Computer Information Technology Laboratory Security Resource Center page Cryptographic Module Validation Program Certificate #4697
  - Security Policy mapped to Certificate #4697. See BSAFE Java Crypto Module 6.3 Security Policy Level 1

See FIPS certifications for a complete list of Oracle product FIPS security certifications that are completed and are in progress.

To enable FIPS mode for Java components by configuring the `java.properties` file, see *Oracle Fusion Middleware Administering Security for Oracle WebLogic Server*.

Note that Oracle Database FIPS settings enforce the use of FIPS-approved algorithms for the Oracle database only. Third-party vendor software used with Oracle Database running in FIPS mode must use only these FIPS-approved algorithms, or else the vendor software will encounter failures.

C.2 Configuration of FIPS 140-2 Using the Consolidated FIPS_140 Parameter

The consolidated `FIPS_140` parameter can be set for several different Oracle Database environments.

- About Configuration of FIPS 140-2 Using the FIPS_140 Parameter
  Configuring the `FIPS_140` parameter is the same for all supported environments.
- Configuring the FIPS_140 Parameter
  To configure FIPS 140-2, you must set the `FIPS_140` parameter in the `fips.ora` file.
- Running orapki in FIPS Mode
  Run `orapki` in FIPS mode by appending `-fips140_mode` at end of each `orapki` command for any wallet creation command.
- Configuring Standalone Java FIPS for Running Java Client Applications in FIPS Mode
  To configure standalone Java FIPS for running Java client applications in FIPS mode, you must check the `CLASSPATH` settings and set the appropriate FIPS-validated provider in the `java.security properties` file.
- Enabling FIPS by Running the enable_fips.py Python Script
  The `enable_fips.py` script enables FIPS mode for Java applications used with Oracle Database, such as Workload Manager, Oracle Database Configuration Assistant (DBCA), and Oracle Net Configuration Assistant (NetCA).
- FIPS-Supported Algorithms for Transparent Data Encryption
  FIPS-supported algorithms for Transparent Data Encryption (TDE) include AES algorithms.
- FIPS-Supported Cipher Suites for DBMS_CRYPTO
  The FIPS library supports the use of cipher suites for the `DBMS_CRYPTO` PL/SQL package.

- FIPS-Supported Cipher Suites for Transport Layer Security
  A cipher suite is a set of authentication, encryption, and data integrity algorithms that exchange messages between network nodes.
- FIPS-Supported Algorithms for Network Native Encryption
  The FIPS library supports both encryption and checksumming algorithms for native network encryption.

C.2.1 About Configuration of FIPS 140-2 Using the FIPS_140 Parameter

Configuring the `FIPS_140` parameter is the same for all supported environments.

The `FIPS_140` parameter has been consolidated for Oracle databases that use the following environments and features:

- Transparent Data Encryption (TDE)
- `DBMS_CRYPTO` PL/SQL package
- Transport Layer Security (TLS)
- Native network encryption

C.2.2 Configuring the FIPS_140 Parameter

To configure FIPS 140-2, you must set the `FIPS_140` parameter in the `fips.ora` file.

1. Locate the `fips.ora` file that is used by the database client or database server.

   Ensure that the `fips.ora` file is either located in the `$ORACLE_HOME/ldap/admin` directory, or is in a location pointed to by the `FIPS_HOME` environment variable.
2. Add the following line to the `fips.ora` file:

   ```
   FIPS_140=TRUE
   ```

   When you set `FIPS_140` to `TRUE`, cryptographic operations take place within a FIPS-validated cryptographic module.

   This parameter is `FALSE` by default. If you set `FIPS_140` to `FALSE`, then cryptographic operations take place in a cryptography module that is not validated for FIPS.

   For either setting, cryptographic operations are accelerated if possible.
3. Repeat this procedure in any Oracle Database home for any database server or client.

C.2.3 Running orapki in FIPS Mode

Run `orapki` in FIPS mode by appending `-fips140_mode` at end of each `orapki` command for any wallet creation command.

- Use the following syntax:

  ```
  orapki command -fips140_mode
  ```

C.2.4 Configuring Standalone Java FIPS for Running Java Client Applications in FIPS Mode

To configure standalone Java FIPS for running Java client applications in FIPS mode, you must check the `CLASSPATH` settings and set the appropriate FIPS-validated provider in the `java.security properties` file.

1. Navigate to the JDK home within the Oracle home.
2. Verify that the `CLASSPATH` includes the following jars: `cryptojce.jar`, `cryptojcommon.jar`, and `jcmFIPS.jar`.
3. In the `java.security` properties file, do the following:
   1. Set `com.rsa.jsafe.provider.JsafeJCE` as the first security provider. The default values of the `java.security` properties file are read from an implementation-specific location, which is typically the properties file `conf/security/java.security` in the Java installation directory.
   2. Move up the index of the existing security providers.

Related Topics

- orapki Utility Commands Summary
  The `orapki` commands perform a variety of wallet, certificate revocation lists (CRL), and certificate management tasks.

C.2.5 Enabling FIPS by Running the enable_fips.py Python Script

The `enable_fips.py` script enables FIPS mode for Java applications used with Oracle Database, such as Workload Manager, Oracle Database Configuration Assistant (DBCA), and Oracle Net Configuration Assistant (NetCA).

The `enable_fips.py` script updates the `fips.ora` file by setting the parameter `FIPS_140=TRUE` in the `fips.ora` file. It also sets `com.rsa.jsafe.provider.JsafeJCE` as the first security provider in the `java.security` file.

1. Locate the `enable_fips.py` Python script in the `$ORACLE_HOME/bin` directory.
2. Run the `enable_fips.py` script.

   ```
   python enable_fips.py
   ```

3. In the scenario of running this script on the Oracle Database server, restart the server after the script completes running.

C.2.6 FIPS-Supported Algorithms for Transparent Data Encryption

FIPS-supported algorithms for Transparent Data Encryption (TDE) include AES algorithms.

- AES128
- AES192
- AES256

You can migrate the encryption algorithms in tables and tablespaces to the latest versions. Note that `3DES168` is no longer supported, starting with Oracle Database 23ai.

- For tables: *Oracle Database Advanced Security Guide*
- For tablespaces: *Oracle Database Advanced Security Guide*

C.2.7 FIPS-Supported Cipher Suites for DBMS_CRYPTO

The FIPS library supports the use of cipher suites for the `DBMS_CRYPTO` PL/SQL package.

For the `DBMS_CRYPTO` cryptographic hash:

- `HASH_SH256`
- `HASH_SH384`
- `HASH_SH512`
- `HASH_SHA3_256`
- `HASH_SHA3_384`
- `HASH_SHA3_512`
- `HASH_SHAKE128`
- `HASH_SHAKE256`

`DBMS_CRYPTO` MAC (Message Authentication Code):

- `HMAC_SH256`
- `HMAC_SH384`
- `HMAC_SH512`
- `HMAC_SHA3_256`
- `HMAC_SHA3_384`
- `HMAC_SHA3_512`

`DBMS_CRYPTO KMACXOF` (KECCAK Message Authentication Code):

- `KMACXOF_128`
- `KMACXOF_256`

`DBMS_CRYPTO` `ENCRYPT` and `DECRYPT`:

- `ENCRYPT_AES`
- `ENCRYPT_AES128`
- `ENCRYPT_AES192`
- `ENCRYPT_AES256`

`DBMS_CRYPTO` `PKENCRYPT` and `PKDECRYPT`:

- `PKENCRYPT_RSA_PKCS1_OAEP_SHA2`

`DBMS_CRYPTO` `SIGN` and `VERIFY`:

- `SIGN_SHA224_RSA`
- `SIGN_SHA256_RSA`
- `SIGN_SHA256_RSA_X931`

- `SIGN_SHA384_RSA`
- `SIGN_SHA384_RSA_X931`
- `SIGN_SHA512_RSA`
- `SIGN_SHA512_RSA_X931`
- `SIGN_SHA3_224_RSA`
- `SIGN_SHA3_256_RSA`
- `SIGN_SHA3_384_RSA`
- `SIGN_SHA3_512_RSA`
- `SIGN_SHA3_224_ECDSA`
- `SIGN_SHA3_256_ECDSA`
- `SIGN_SHA3_384_ECDSA`
- `SIGN_SHA3_512_ECDSA`

C.2.8 FIPS-Supported Cipher Suites for Transport Layer Security

A cipher suite is a set of authentication, encryption, and data integrity algorithms that exchange messages between network nodes.

During a TLS handshake, for example, the two nodes negotiate to see as to which cipher suite they will use when transmitting messages back and forth.

Configuring Specific Cipher Suites

Oracle Database TLS cipher suites are automatically set to FIPS approved cipher suites. If you want to configure specific cipher suites, then you can do so by setting the `SSL_CIPHER_SUITES` parameter in the `sqlnet.ora` or the `listener.ora` file.

```
SSL_CIPHER_SUITES=(SSL_cipher_suite1[,SSL_cipher_suite2[,..]])
```

You can also use Oracle Net Manager to set this parameter on the server and the client.

If a specific cipher suite is not specified, then Oracle Database will use the strongest cipher suite common to both the database server and client. The priority order of cipher suites to be selected are in order as they are listed in the preferred and less preferred cipher lists below. Oracle Database will not select 3DES cipher suites automatically due to their weakness; they must be configured explicitly.

Preferred Cipher Suites

The following cipher suites are approved for FIPS validation if you are using TLS version 1.3:

- `TLS_AES_128_CCM_SHA256`
- `TLS_AES_128_GCM_SHA256`
- `TLS_AES_256_GCM_SHA384`

The following cipher suites are approved for FIPS validation if you are using Transport Layer Security (TLS) version 1.2:

- `TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA`
- `TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256`

- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

3DES-Based Cipher Suites

Oracle does not recommend 3DES-based cipher suites because of a weakness in their design. Oracle Database release 21c and later contains support for the following 3DES-based cipher suites. However, they are not enabled by default and must be explicitly configured through the `SSL_CIPHER_SUITES` parameter in the `sqlnet.ora` or the `listener.ora` file.

- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_RSA_WITH_3DES_EDE_CBC_SHA

Related Topics

- Configuring TLS Cipher Suites
  A cipher suite is a set of authentication, encryption, and data integrity algorithms used for exchanging messages between network entities.

C.2.9 FIPS-Supported Algorithms for Network Native Encryption

The FIPS library supports both encryption and checksumming algorithms for native network encryption.

- Encryption algorithms: AES128, AES192, and AES256
- Checksumming algorithms: SHA1, SHA256, SHA384, and SHA512

C.3 Legacy FIPS 140-2 Configurations

The legacy FIPS 140-2 configurations apply to Transparent Data Encryption (TDE), `DBMS_CRYPTO`, network native encryption, and Transport Layer Security (TLS).

- About Legacy FIPS 140-2 Configurations
  The use of the legacy FIPS 140-2 configurations is still supported, but Oracle recommends that you use the consolidated `FIPS_140` parameter instead.
- Configuring FIPS 140-2 for Transparent Data Encryption and DBMS_CRYPTO
  The `DBFIPS_140` initialization parameter configures FIPS mode.
- Configuring FIPS 140-2 for Transport Layer Security
  To configure FIPS 140-2 for Transport Layer Security (TLS), you can set the `SSLFIPS_140` parameter.

- Configuring FIPS 140-2 for Native Network Encryption
  To configure FIPS 140-2 for native network encryption, you must set the `FIPS_140` parameter in the `sqlnet.ora` file.

C.3.1 About Legacy FIPS 140-2 Configurations

The use of the legacy FIPS 140-2 configurations is still supported, but Oracle recommends that you use the consolidated `FIPS_140` parameter instead.

The legacy FIPS 140-2 configurations apply to the following environments:

- Transparent Data Encryption (TDE)
- `DBMS_CRYPTO` PL/SQL packages
- Transport Layer Security (TLS)
- Network native encryption

Related Topics

- Configuration of FIPS 140-2 Using the Consolidated FIPS_140 Parameter
  The consolidated `FIPS_140` parameter can be set for several different Oracle Database environments.

C.3.2 Configuring FIPS 140-2 for Transparent Data Encryption and DBMS_CRYPTO

The `DBFIPS_140` initialization parameter configures FIPS mode.

This method of configuring FIPS 140-2 for TDE and `DBMS_CRYPTO` is considered a legacy configuration, but it is still supported. Oracle recommends that you use the consolidated `FIPS_140` parameter instead.

1. To configure Transparent Data Encryption and the `DBMS_CRYPTO` PL/SQL package program units to run in FIPS mode, set the `DBFIPS_140` initialization parameter to `TRUE`.

   The settings have the following effect for all platforms:

   - `TRUE`: TDE and `DBMS_CRYPTO` program units use a FIPS-validated cryptographic module.

     Be aware that setting `DBFIPS_140` to `TRUE` and thus using the underlying library in FIPS mode incurs a certain amount of overhead when the library is first loaded for each process. This is due to the verification of the signature and the execution of the self tests on the library. Once the library is loaded for each process, then there is no other impact on performance.

   - `FALSE`: TDE and `DBMS_CRYPTO` program units use a cryptographic module that is not validated for FIPS.

2. Restart the database.

Related Topics

- Configuration of FIPS 140-2 Using the Consolidated FIPS_140 Parameter
  The consolidated `FIPS_140` parameter can be set for several different Oracle Database environments.

C.3.3 Configuring FIPS 140-2 for Transport Layer Security

To configure FIPS 140-2 for Transport Layer Security (TLS), you can set the `SSLFIPS_140` parameter.

This method of configuring FIPS 140-2 for TLS is considered a legacy configuration, but it is still supported. Oracle recommends that you use the consolidated `FIPS_140` parameter instead.

1. Ensure that the `fips.ora` file is either located in the *`$ORACLE_HOME`*`/ldap/admin` directory, or is in a location pointed to by the `FIPS_HOME` environment variable.
2. In the `fips.ora` file, set `SSLFIPS_140` to `TRUE` so that the TLS adapter can run in FIPS mode.

   For example:

   ```
   SSLFIPS_140=TRUE
   ```

   When you set `SSLFIPS_140` to `TRUE`, TLS cryptographic operations take place within a cryptographic module that is designed to comply with FIPS requirements.

   This parameter is `FALSE` by default. If you set `SSLFIPS_140` to `FALSE`, then TLS cryptographic operations take place in in a cryptography module that is not validated for FIPS, and as with the `TRUE` setting, the operations are accelerated if possible.
3. Repeat this procedure in any Oracle Database home for any database server or client.

> **Note:**
>
> The `SSLFIPS_140` parameter replaces the `SQLNET.SSLFIPS_140` parameter used in Oracle Database 10g release 2 (10.2). You must set the parameter in the `fips.ora` file, and not the `sqlnet.ora` file.

Related Topics

- Configuration of FIPS 140-2 Using the Consolidated FIPS_140 Parameter
  The consolidated `FIPS_140` parameter can be set for several different Oracle Database environments.

C.3.4 Configuring FIPS 140-2 for Native Network Encryption

To configure FIPS 140-2 for native network encryption, you must set the `FIPS_140` parameter in the `sqlnet.ora` file.

This method of configuring FIPS 140-2 for network native encryption is considered a legacy configuration, but it is still supported. Oracle recommends that you use the consolidated `FIPS_140` parameter instead.

1. Locate the `sqlnet.ora` file that is used by the database client or database server
2. Add the following line to the `sqlnet.ora` file:

   ```
   SQLNET.FIPS_140=TRUE
   ```

When you set FIPS_140 to TRUE, native network encryption cryptographic operations take place within a cryptographic module that is designed to comply with FIPS requirements.

This parameter is FALSE by default. If you set FIPS_140 to FALSE, then native network cryptographic operations take place in a cryptography module that is not validated for FIPS, and as with the TRUE setting, the operations are accelerated if possible.

3. Repeat this procedure in any Oracle Database home for any database server or client.

Related Topics

- Configuration of FIPS 140-2 Using the Consolidated FIPS_140 Parameter
  The consolidated FIPS_140 parameter can be set for several different Oracle Database environments.

C.4 Postinstallation Checks for FIPS 140-2

After you configure the FIPS 140-2 settings, you must verify permissions in the operating system.

The permissions are as follows:

- Set execute permissions on all Oracle executable files to prevent the execution of Oracle Cryptographic Libraries by users who are unauthorized to do so, in accordance with the system security policy.
- Set read and write permissions on all Oracle executable files to prevent accidental or deliberate reading or modification of Oracle Cryptographic Libraries by any user.

To comply with FIPS 140-2 Level 2 requirements, in the security policy, include procedures to prevent unauthorized users from reading, modifying or executing Oracle Cryptographic Libraries processes and the memory they are using in the operating system.

C.5 Verifying FIPS 140-2 Connections

You can use trace files and other methods to verify the FIPS 140-2 connections.

- Verifying FIPS 140-2 Connections When Using the FIPS_140 Parameter
  You can use trace files to check the FIPS 140-2 status when using the FIPS_140 parameter.
- Verifying FIPS 140-2 Connections for Transport Layer Security
  You can use trace files to check the FIPS 140-2 connections for Transport Layer Security (TLS).
- Verifying FIPS 140-2 Connections for Network Native Encryption
  You can use trace files to check the FIPS 140-2 connections for network native encryption.
- Verifying FIPS 140-2 Connections for Transparent Data Encryption and DBMS_CRYPTO
  You can check if FIPS mode is enabled by using SQL*Plus.

C.5.1 Verifying FIPS 140-2 Connections When Using the FIPS_140 Parameter

You can use trace files to check the FIPS 140-2 status when using the FIPS_140 parameter.

1. Set the environment variable ENABLE_TRACE to 1 to enable tracing.

- In C shell:

  ```
  setenv ENABLE_TRACE 1
  ```

- In bash:

  ```
  export ENABLE_TRACE=1
  ```

2. Check the trace files by searching for `FIPS`.

C.5.2 Verifying FIPS 140-2 Connections for Transport Layer Security

You can use trace files to check the FIPS 140-2 connections for Transport Layer Security (TLS).

1. Add the following lines to `sqlnet.ora` to enable tracing:

   ```
   trace_directory_server=trace_directory
   trace_file_server=trace_file
   trace_level_server=trace_level
   ```

   For example:

   ```
   trace_directory=/private/oracle/owm
   trace_file_server=fips_trace.trc
   trace_level_server=16
   ```

   Trace level 16 is the minimum trace level required to check the results of the FIPS self-tests.

2. Check the trace files by searching for `Provider Type: FIPS140`.

C.5.3 Verifying FIPS 140-2 Connections for Network Native Encryption

You can use trace files to check the FIPS 140-2 connections for network native encryption.

1. Add the following lines to `sqlnet.ora` to enable tracing:

   ```
   trace_directory_server=trace_directory
   trace_file_server=trace_file
   trace_level_server=trace_level
   ```

   For example:

   ```
   trace_directory=/private/oracle/owm
   trace_file_server=fips_trace.trc
   trace_level_server=16
   ```

   Trace level 16 is the minimum trace level required to check the results of the FIPS self-tests.

2. Check the trace files by searching for `FIPS mode activated successfully`.

C.5.4 Verifying FIPS 140-2 Connections for Transparent Data Encryption and DBMS_CRYPTO

You can check if FIPS mode is enabled by using SQL*Plus.

1. Connect to the database instance by using SQL*Plus.
2. Run the following `SHOW PARAMETER` command:

   ```
   SHOW PARAMETER DBFIPS_140
   ```

   Output similar to the following should appear:

   ```
   NAME                                 TYPE        VALUE
   ------------------------------------ ----------- ------------------------------
   DBFIPS_140                           boolean     TRUE
   ```

C.6 Managing Deprecated Weaker Algorithm Keys

In Oracle Database release 23ai, several algorithms for both FIPS and non-FIPS have been deprecated.

The security strength of the cipher algorithms has been strengthened in Oracle Database 23ai. The following cipher algorithms are deprecated or removed:

- For FIPS mode:
  - The FIPS security strength of 80 is no longer supported. The new default security strength for FIPS mode is 112. Currently, this is the only supported FIPS security strength.
  - RSA, Diffie Hellman, and Digital Signature Algorithm (RSA/DH/DSA) with 1024 key size are no longer supported. The new minimum supported key size is 2048.
- For non-FIPS mode:
  - Security Strength 0 (RSA/DH/DSA key length 512) is deprecated. By default, Security Strength support is now 80. Security strength 0 (RSA key 512 and equivalent) is still available, but not recommended for use. Available security strengths for non-FIPS use are 0 (deprecated), 80, and 112.

Oracle recommends that you find existing use of RSA/DH/DSA 512 /1024 key sizes (along with ECC equivalents) and replace these with RSA/DH/DSA 2048 key size and equivalents.

The following tables describe the security strength of various encryption keys.

You can use the `orapki` command line utility to create signed certificates, manage Oracle wallets, and manage certificate revocation lists. It has the same default key sizes as listed in the following tables.

FIPS Default Setting (Starting with Oracle Database 23ai)

Table C-1 FIPS Default Setting (Starting with Oracle Database 23ai)

Algorithm Key Type	Security Strength
-	Default Security strength: 112 (was 80) Security strength: 0, 80 are not supported and not available for FIPS use
Default RSA/DH/DSA (Diffie Hellman, Digital Signature Algorithm)	2048 key size (Key size support for less than 2048 bits key size is not supported)
Default ECC (Elliptic Curve Cryptography)	ECC curves with minimum ECC curve key length 224, ECC names curves P192, K163, and B163 and lower are not supported

Non-FIPS Default Setting (Starting with Oracle Database 23ai)

Table C-2   Non-FIPS Default Setting (Starting with Oracle Database 23ai)

Algorithm Key Type	Security Strength
-	Default Security strength: 80 Security strength: 0, 112 (available)
Default RSA/DH/DSA (Diffie Hellman, Digital Signature Algorithm)	1024 key size (512 and 2048 are also available by setting `ORACLE_MIN_KEY_STRENGTH_SUPPORT`). To change Non-FIPS security strength to 0 or 112, set the `ORACLE_MIN_KEY_STRENGTH_SUPPORT` parameter in the `fips.ora` file to 0 or 112. This file is either in `$ORACLE_HOME/crypto/admin` or in a location pointed to by the environment variable `FIPS_HOME`.
Default ECC (Elliptic Curve Cryptography)	ECC curves with minimum ECC curve key length 163. ECC names curves lower than K163, B163 are not supported.