Cipher Management

Cipher Management

Cipher management is an optional feature that enables you to control the set of security ciphers that is allowed for every TLS and SSH connection. Cipher management allows you to disable weaker ciphers and thus enable a minimum level of security.

The **Cipher Management** page has no default values. Instead, the Cipher Management feature takes effect only when you configure the allowed ciphers. Certain weak ciphers are never allowed, even if they are configured on the **Cipher Management** page.

You can configure ciphers on the following TLS and SSH interfaces:

- **All TLS**—The ciphers that are assigned in this field are applicable to all server and client connections that support the TLS protocol on Unified Communications Manager and IM and Presence Service.
- **HTTPS TLS**—The ciphers that are assigned in this field are applicable to all Cisco Tomcat connections on ports 443 and 8443 that support the TLS protocol on Unified Communications Manager and IM and Presence Service.

Note If you assign ciphers on **HTTPS TLS** and **All TLS** fields, the ciphers that are configured on **HTTPS TLS** override **All TLS** ciphers.

- **SIP TLS**—The ciphers that are assigned in this field are applicable to all encrypted connections to or from the SIP TLS interfaces that support the TLS protocol on Unified Communications Manager. It is not applicable for SCCP or CTI devices.

  SIP interface in authenticated mode only supports NULL-SHA ciphers.

  If you configure ciphers in the SIP interface or All interface, authenticated mode is no longer supported.

  If you assign ciphers in **SIP TLS** and **All TLS** fields, then the ciphers you configured on SIP TLS override the All TLS ciphers.

- **SSH Ciphers**—The ciphers that are assigned in this field are applicable to SSH connections on Unified Communications Manager and IM and Presence Service.
- **SSH Key Exchange**—The Key Exchange algorithms that are assigned in this field are applicable to the SSH interface on Unified Communications Manager and IM and Presence Service.

Curve Negotiation

Following are the points for negotiating the curves:

- ECDSA ciphers are negotiated with different EC curves based on the key size of the ECDSA certificate.
- The RSA ciphers are negotiated with all the EC curves irrespective of key size of the certificate.
- The key size of a ECDSA certificate must be same as the curve size for the TLS negotiation to happen.

Example:

The 384 key certificate and ECDSA ciphers are negotiated, when the client offers P-384 EC curve.

Curve negotiation is based on the client preference for both RSA and ECDSA ciphers.

When the certificate size is 384 bits and client offerings are P-521, P-384, P-256 EC curves then TLS negotiation happen with the P-521 curve. Since curve offered by the client is P-521 at the first and P- 384 curve is also available on the list. When the certificate size is 384 bits and client offerings are P-521, P-256 EC curves then TLS negotiation will not happen because the P-384 curve is not offered by the client.

The following are the supported ciphers for EC curves:

```
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
```

Recommended Ciphers

By default, Unified Communications Manager and IM and Presence Service already uses a set of ciphers (see TLS and SSH Ciphers section below) that supports secure integration with most other products, including third-party products. Therefore, it is usually not required to make changes. If Cipher suite mismatches are causing TLS Handshake failures, Unified Communications Manager Cipher Management can be used to add additional ciphers to the list of supported Ciphers.

Cipher Management can also be used if customers want to be more restrictive and prevent certain Cipher suites from being negotiated during TLS handshake. After configuring the ciphers, restart the affected services or reboot the server for the changes to take effect.

Warning Configuring hmac-sha2-512 in SSH MAC interface affects the DRS and CDR functionality.

Configuring ciphers aes128-gcm@openssh.com, aes256-gcm@openssh.com in "SSH Cipher's" field or configuring only ecdh-sha2-nistp256 algorithm in "SSH KEX" will break the DRS and CDR functionalities.

We support the following cipher strings for the TLS and SSH interface configuration:

TLS

```
ECDHE-RSA-AES256-GCM-SHA384:ECDHE-RSA-AES256-SHA384:
ECDHE-RSA-AES256-SHA:AES256-GCM-SHA384:AES256-SHA256:AES256-SHA:
ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-SHA256:
ECDHE-RSA-AES128-SHA:AES128-GCM-SHA256:AES128-SHA256:AES128-SHA
```

SSH Ciphers

```
aes128-ctr,aes192-ctr,aes256-ctr,aes128-gcm@openssh.com,
aes256-gcm@openssh.com
```

SSH MAC

```
hmac-sha2-512,hmac-sha2-256,hmac-sha1
```

SSH KEX for FIPS

```
ecdh-sha2-nistp521, ecdh-sha2-nistp384, ecdh-sha2-nistp256,
diffie-hellman-group14-sha1, diffie-hellman-group16-sha512, diffie-hellman-group14-sha256
```

SSH KEX for Non-FIPS

```
ecdh-sha2-nistp521, ecdh-sha2-nistp384, ecdh-sha2-nistp256,
diffie-hellman-group14-sha1, diffie-hellman-group16-sha512, diffie-hellman-group14-sha256
```

Configure Cipher String

- Make sure you enter the cipher string in OpenSSL cipher string format in **All TLS**, **SIP TLS**, and **HTTPS TLS** fields.
- Make sure that you also enter the ciphers or algorithms in OpenSSH format in **SSH Ciphers**,algorithms in **SSH MAC**, and **SSH Key Exchange** fields.
- Review Recommended Ciphers, on page 2.

To configure the cipher string on different secure interfaces, see the Cipher Restrictions section.

Procedure

Step 1 From Cisco Unified OS Administration, choose **Security** > **Cipher Management**.
The Cipher Management page appears.

Step 2 To configure the cipher string in **All TLS**, **SIP TLS**, or **HTTPS TLS** field, enter the cipher string in OpenSSL cipher string format in the **Cipher String** field.

Step 3 If you don't configure the cipher string in the following fields:

- **All TLS or HTTPS TLS** field—the HTTPS TLS interface port (8443) takes configuration from the **Enterprise parameters** (HTTPS ciphers) page.
- **All TLS or SIP TLS** field—the SIP interface port (5061) takes configuration from the **Enterprise parameters** (TLS ciphers) page in encrypted mode and NULL-SHA ciphers in authenticated mode.

Note If you don't configure the cipher string in the **HTTPS TLS** or **SIP TLS** field, the system takes the configuration from the **All TLS** field by default.

For more information about OpenSSL cipher string format, see https://www.openssl.org/docs/man1.0.2/apps/ciphers.html.

Step 4 To configure the cipher string in the **SSH Ciphers** field, enter the cipher string in OpenSSH cipher string format in the **Cipher String** field.

For more information about OpenSSH cipher string format for SSH Ciphers, see https://www.ssh.com/manuals/server-admin/44/Ciphers_and_MACs.html.

If you don't configure any cipher string in the **SSH Ciphers** field, the following ciphers are applicable to all SSH connections by default:

In FIPS mode:

```
aes128-ctr, aes192-ctr, aes256-ctr,
aes128-gcm@openssh.com, aes256-gcm@openssh.com
```

In non-FIPS mode:

```
aes128-ctr, aes192-ctr, aes256-ctr,
aes128-gcm@openssh.com, aes256-gcm@openssh.com
```

Step 5 To configure the key exchange algorithm in the **SSH Key Exchange** field, enter the algorithm string in OpenSSH string format in the **Algorithm String** field.

For more information about OpenSSH algorithm string format for SSH Key Exchange, see the https://tools.ietf.org/id/draft-ietf-curdle-ssh-kex-sha2-09.html.

If you don't configure any key exchange algorithm in the **SSH Key Exchange** field, the following key exchange algorithms are applicable to all SSH connections by default:

In FIPS mode:

```
diffie-hellman-group1-sha1,diffie-hellman-group14-sha1,
diffie-hellman-group-exchange-sha1,diffie-hellman-group-exchange-sha256,
ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521
```

In non-FIPS mode:

```
diffie-hellman-group1-sha1, diffie-hellman-group14-sha1,
diffie-hellman-group-exchange-sha1,diffie-hellman-group-exchange-sha256,
ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521
```

Step 6 To configure MAC algorithm in the **SSH MAC** field, enter the algorithm string in OpenSSH string format in the **Algorithm String** field.

For more information about OpenSSH algorithm string format for SSH MAC, see https://www.ssh.com/manuals/server-admin/44/Ciphers_and_MACs.html.

If you don't configure any MAC algorithm in the **SSH MAC** field, the following MAC algorithms are applicable to all SSH connections by default:

In FIPS mode:

```
hmac-sha1
```

In non-FIPS mode:

```
hmac-sha1
```

Step 7 Click **Save**.

Note You can't edit **Cipher Expansion String** and **Algorithm Expansion String** fields.

The system validates the ciphers in the **All TLS**, **SIP TLS**, **HTTPS TLS** , and **SSH Ciphers** fields and auto populates ciphers in the **Cipher Expansion String** field.

If you enter invalid ciphers in the **Cipher String** field, the **Cipher Expansion String** field doesn't auto populate and the following error message appears:

```
You have entered an invalid Cipher String.
```

The system validates the algorithms in the **SSH Key Exchange** and **SSH MAC** fields, and auto populates the algorithms in the **Algorithm Expansion String** field.

If you enter invalid algorithms in the **Algorithm String** field, the **Algorithm Expansion String** field doesn't auto populate and the following error messge appears:

```
You have entered an invalid Algorithm String.
```

Note The ciphers or algorithms auto populated in **Cipher Expansion String** and **Algorithm Expansion String** fields are not the effective ciphers or algorithms. The system chooses the ciphers or algorithms from the **Cipher Expansion String** or **Algorithm Expansion String** field.

If you have configured ciphers in the corresponding fields, you have to either reboot or restart the respective services.

Table 1: Configured Ciphers and their corresponding Actions

Configured Cipher Fields	Action
All TLS	Reboot all nodes in the cluster for the cipher string to take effect.
HTTPS TLS	Restart the Cisco Tomcat service on all nodes for the cipher string to take effect.

Configured Cipher Fields	Action
SIP TLS	Restart Unified Communications Manager on all nodes for the cipher string to take effect.
SSH Ciphers	Reboot all nodes in the cluster for the cipher string to take effect.
SSH Key Exchange or **SSH MAC**	Reboot all nodes in the cluster for the algorithm string to take effect.

Note You can enable ciphers by entering them in the **Cipher String** fields of the **Cipher Management** page. If you don't enter them, all default ciphers supported by the application are enabled. However, you can also disable certain weak ciphers by not entering them in the **Cipher String** fields of the **Cipher Management** page.

Cipher Limitations

Although the **Cipher Management** configuration page allows you to configure any number of ciphers, each application has a list of ciphers it supports on its interfaces. For example, **All TLS** interfaces may show ECDHE or DHE or ECDSA-based ciphers, but an application such as Unified Communications Manager may not support these ciphers because EC curves or DHE algorithms are not enabled for this application's interfaces. For more information, see the "Application Ciphers Support" section for a list of ciphers supported by individual application interfaces.

Note If you are upgrading a cluster with ciphers configured in the Cipher Management page, ensure that you configure at least one common cipher between ALL and HTTPS fields.

Note Cisco Cloud Onboarding is not part of the Cipher Management suite and will use all the default ciphers that are supported in the server. However, this limitation has been fixed from 12.5(1) SU6 release onwards.

Validation in GUI

The ciphers on **Cipher Management** page are validated according to the OpenSSL guidelines. For example, if a cipher configured is ALL:BAD:!MD5, the cipher string will be considered as valid even though "BAD" is not a recognized cipher suite. OpenSSL considers this as a valid string. If AES128_SHA is configured instead of AES128-SHA (using an underscore instead of a hyphen) however, OpenSSL identifies this as an invalid cipher suite.

Authenticated Mode (NULL Ciphers)

If NULL ciphers are in use by an application interface, you can revoke the support for NULL ciphers by configuring any cipher list in **All TLS** or **SIP TLS** fields on **Cipher Management** page.

Examples of application interfaces that use NULL ciphers are:

- **All TLS Interface:** Unified Communications Manager SIP Proxy in IM and Presence through the **TLS Context Configuration** page.
- **SIP TLS Interface:** Unified Communications Manager through SIP or SCCP, when any **Device Security Profile** is set to **Authenticated** mode.

Don't configure ciphers for either of these two interfaces if NULL ciphers must be used.

Override Functionality

The settings on the **Cipher Management** page overrides the default settings for each application and any other location where ciphers have been configured. This means that if no ciphers are configured on the **Cipher Management** page, then the original functionality on all interfaces will be retained.

For example, if the **Enterprise Parameter** "**TLS Ciphers**" is configured with "*ALL Supported Ciphers*" and the **Cipher Management** page is configured with ciphers "*AES256-GCM-SHA384:AES256-SHA256*" on **All TLS** interfaces, all application SIP interfaces will support only the "*AES256-GCM-SHA384:AES256-SHA256*" ciphers and ignores the **Enterprise Parameter** value.

Application Ciphers Support

The following table lists the application interfaces and the all corresponding ciphers and algorithms that are supported on TLS and SSH interfaces.

Table 2: Unified Communications Manager Cipher Support for TLS Ciphers

Application / Process	Protocol	Port	Supported Ciphers

Cisco CallManager	TCP / TLS	2443	`ECDHE-RSA-AES256-GCM-SHA384:`
			`ECDHE-RSA-AES256-SHA384:`
			`AES256-GCM-SHA384:`
			`AES256-SHA256:`
			`AES256-SHA:`
			`ECDHE-RSA-AES128-GCM-SHA256:`
			`ECDHE-RSA-AES128-SHA256:`
			`ECDHE-RSA-AES128-SHA:`
			`AES128-GCM-SHA256:`
			`AES128-SHA256:AES128-SHA:`
			`ECDHE-RSA-AES256-SHA:`
			Note The following ciphers are not supported from Release 14SU2 onwards:
			`CAMELLIA128-SHA`
			`CAMELLIA256-SHA:`

Application / Process	Protocol	Port	Supported Ciphers
DRS	TCP / TLS	4040	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA:CAMELLIA256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-RSA-AES256-SHA: DHE-RSA-CAMELLIA256-SHA: DHE-RSA-CAMELLIA128-SHA: CAMELLIA128-SHA
Cisco Tomcat	TCP / TLS	8443 / 443	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: DHE-RSA-AES256-GCM-SHA384: DHE-RSA-AES256-SHA256: DHE-RSA-AES256-SHA: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: DHE-RSA-AES128-GCM-SHA256: DHE-RSA-AES128-SHA256: DHE-RSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: DHE-RSA-CAMELLIA256-SHA: CAMELLIA256-SHA: DHE-RSA-CAMELLIA128-SHA: CAMELLIA128-SHA: ECDHE-RSA-DES-CBC3-SHA: EDH-RSA-DES-CBC3-SHA: DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA:

Application / Process	Protocol	Port	Supported Ciphers
Cisco CallManager	TCP / TLS	5061	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-GCM-SHA384 ECDHE-RSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256 ECDHE-ECDSA-AES128-SHA256: ECDHE-RSA-AES128-SHA ECDHE-ECDSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: ECDHE-ECDSA-AES256-SHA: CAMELLIA256-SHA: CAMELLIA128-SHA: ECDHE-ECDSA-DES-CBC3-SHA
Cisco CTL Provider **Note** Cisco CTL Provider is not available from Release 14SU3 onwards.	TCP / TLS	2444	AES256-GCM-SHA384:AES256-SHA256: AES256-SHA:CAMELLIA256-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA:CAMELLIA128-SHA:
Cisco Certificate Authority Proxy Function	TCP / TLS	3804	AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA:

Application / Process	Protocol	Port	Supported Ciphers
CTIManager	TCP / TLS	2749	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA
Cisco Trust Verification Service	TCP / TLS	2445	AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA
Cisco Intercluster Lookup Service	TCP / TLS	7501	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: AES256-GCM-SHA384: AES256-SHA256:AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA:

Application / Process	Protocol	Port	Supported Ciphers
Secure Configuration download (HAPROXY)	TCP / TLS	6971, 6972	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-SHA384: ECDHE-ECDSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: DHE-RSA-CAMELLIA256-SHA: CAMELLIA256-SHA: DHE-RSA-CAMELLIA128-SHA: ECDHE-ECDSA-AES256-SHA: ECDHE-ECDSA-DES-CBC3-SHA: CAMELLIA128-SHA:
Authenticated Contact Search	TCP / TLS	9443	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-SHA384: ECDHE-ECDSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: DHE-RSA-CAMELLIA256-SHA: CAMELLIA256-SHA: DHE-RSA-CAMELLIA128-SHA: CAMELLIA128-SHA: ECDHE-ECDSA-AES256-SHA: ECDHE-ECDSA-DES-CBC3-SHA:

Table 3: Unified Communications Manager IM & Presence Cipher Support for TLS Ciphers

Application / Process	Protocol	Port	Supported Ciphers
Cisco SIP Proxy	TCP / TLS	5061	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: ECDHE-ECDSA-AES128-SHA: AES128-GCM-SHA256: AES128-SHA256: AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA: DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA: ECDHE-RSA-DES-CBC3-SHA: ECDHE-ECDSA-AES256-SHA:
Cisco SIP Proxy	TCP / TLS	5062	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA384: AES256-GCM-SHA384: AES256-SHA256:AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: ECDHE-ECDSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA: DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA: ECDHE-RSA-DES-CBC3-SHA: ECDHE-ECDSA-AES256-SHA:

Application / Process	Protocol	Port	Supported Ciphers
Cisco SIP Proxy	TCP / TLS	8083	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: ECDHE-ECDSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA: DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA: ECDHE-RSA-DES-CBC3-SHA: ECDHE-ECDSA-AES256-SHA:
Cisco Tomcat	TCP / TLS	8443, 443	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: DHE-RSA-AES256-GCM-SHA384: DHE-RSA-AES256-SHA256: DHE-RSA-AES256-SHA: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: DHE-RSA-AES128-GCM-SHA256: DHE-RSA-AES128-SHA256: DHE-RSA-AES128-SHA: AES128-GCM-SHA256: AES128-SHA256:AES128-SHA: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-SHA384: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA: ECDHE-RSA-AES256-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA128-SHA: CAMELLIA256-SHA: DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA: ECDHE-RSA-DES-CBC3-SHA: DHE-RSA-CAMELLIA128-SHA: DHE-RSA-CAMELLIA256-SHA: ECDHE-ECDSA-AES256-SHA: EDH-RSA-DES-CBC3-SHA:

Application / Process	Protocol	Port	Supported Ciphers
Cisco XCP XMPP Federation Connection Manager	TCP /TLS	5269	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: ECDHE-ECDSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA256-SHA: CAMELLIA128-SHA: DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA: ECDHE-RSA-DES-CBC3-SHA: ECDHE-ECDSA-AES256-SHA: ECDHE-RSA-AES256-SHA:
Cisco XCP Client Connection Manager	TCP / TLS	5222	ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA384: AES256-GCM-SHA384:AES256-SHA256: AES256-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: ECDHE-ECDSA-AES128-SHA: AES128-GCM-SHA256:AES128-SHA256: AES128-SHA: **Note** The following ciphers are not supported from Release 14SU2 onwards: CAMELLIA128-SHA: CAMELLIA256-SHA: DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA: ECDHE-RSA-DES-CBC3-SHA: ECDHE-ECDSA-AES256-SHA: ECDHE-RSA-AES256-SHA:

Table 4: Cipher Support for SSH Ciphers

Service	Ciphers/Algorithms

SSH Server	• Ciphers
	`aes128-ctr`
	`aes192-ctr`
	`aes256-ctr`
	`aes128-gcm@openssh.com`
	`aes256-gcm@openssh.com`
	• MAC algorithms:
	`hmac-sha2-256`
	`hmac-sha2-512`
	`hmac-sha1`

SSH Client	• Ciphers:
	`aes128-ctr`
	`aes192-ctr`
	`aes256-ctr`
	`aes128-gcm@openssh.com`
	`aes256-gcm@openssh.com`
	• MAC algorithms:
	`hmac-sha2-256`
	`hmac-sha2-512`
	`hmac-sha1`

Service	Ciphers/Algorithms

DRS Client	• Ciphers:
	`aes256-ctr`
	`aes256-cbc`
	`aes128-ctr`
	`aes128-cbc`
	`aes192-ctr`
	`aes192-cbc`
	• MAC algorithms:
	`hmac-md5`
	`hmac-sha2-256`
	`hmac-sha1`
	`hmac-sha1-96`
	`hmac-md5-96`
	• Kex algorithms:
	`ecdh-sha2-nistp256`
	`ecdh-sha2-nistp384`
	`ecdh-sha2-nistp521`
	`diffie-hellman-group14-sha1`
	`diffie-hellman-group-exchange-sha256`
	`diffie-hellman-group-exchange-sha1`
	`diffie-hellman-group1-sha1`
	Note The Kex algorithms diffie-hellman-group-exchange-sha256, diffie-hellman-group-exchange-sha1, and diffie-hellman-group1-sha1 are not supported from Release 12.5(1)SU4 if you have configured Cipher Management functionality in your Unified CM server. If the ciphers are not configured, DRS Client uses these algorithms.

SFTP client	• Ciphers:
	`aes128-ctr`
	`aes192-ctr`
	`aes256-ctr`
	• MAC algorithms:
	`hmac-sha2-256`
	`hmac-sha1`
	• Kex algorithms:
	`ecdh-sha2-nistp521`
	`ecdh-sha2-nistp384`
	`diffie-hellman-group14-sha1`
	`diffie-hellman-group1-sha1`
	`diffie-hellman-group-exchange-sha256`
	`diffie-hellman-group-exchange-sha1`

End Users	`hmac-sha512` `SHA-256 – Hashing (salted)`

Service	Ciphers/Algorithms
DRS Backups / RTMT SFTPs	AES-128 - Encryption
Application Users	AES-256 - Encryption

Cipher Restrictions

The **Cipher Management** page allows configuration of ciphers supported by OpenSSL or OpenSSH. However, some of the ciphers are disabled internally based on Cisco's security standards to avoid accidental exposure of critical data.

When you configure ciphers on the **Cipher Management** page, the following ciphers are essentially disabled.

TLS Disabled Ciphers

```
EDH-RSA-DES-CBC-SHA:EDH-DSS-DES-CBC-SHA:ADH-DES-CBC-SHA:
DES-CBC-SHA:KRB5-DES-CBC-SHA:KRB5-DES-CBC-MD5:EXP-EDH-RSA-DES-CBC-SHA:
EXP-EDH-DSS-DES-CBC-SHA:EXP-ADH-DES-CBC-SHA:EXP-DES-CBC-SHA:EXP-RC2-CBC-MD5:
EXP-KRB5-RC2-CBC-SHA:EXP-KRB5-DES-CBC-SHA:EXP-KRB5-RC2-CBC-MD5:EXP-KRB5-DES-CBC-MD5:
EXP-ADH-RC4-MD5:EXP-RC4-MD5:EXP-KRB5-RC4-SHA:EXP-KRB5-RC4-MD5:ADH-AES256-GCM-SHA384:
ADH-AES256-SHA256:ADH-AES256-SHA:ADH-CAMELLIA256-SHA:ADH-AES128-GCM-SHA256:ADH-AES128-SHA256:
ADH-AES128-SHA:ADH-SEED-SHA:ADH-CAMELLIA128-SHA:ADH-DES-CBC3-SHA:ADH-RC4-MD5:
AECDH-AES256-SHA:AECDH-AES128-SHA:AECDH-DES-CBC3-SHA:AECDH-RC4-SHA:AECDH-NULL-SHA:
DES-CBC3-MD5:IDEA-CBC-MD5:RC2-CBC-MD5:RC4-MD5:ECDHE-RSA-RC4-SHA:ECDHE-ECDSA-RC4-SHA:
ECDH-RSA-RC4-SHA:ECDH-ECDSA-RC4-SHA:RC4-SHA:RC4-MD5:PSK-RC4-SHA:KRB5-RC4-SHA:
KRB5-RC4-MD5:IDEA-CBC-SHA:KRB5-IDEA-CBC-SHA:KRB5-IDEA-CBC-MD5:DHE-RSA-SEED-SHA:
DHE-DSS-SEED-SHA:SEED-SHA:KRB5-DES-CBC3-MD5:NULL-MD5:PSK-AES256-CBC-SHA:
PSK-AES128-CBC-SHA:PSK-3DES-EDE-CBC-SHA:ECDHE-RSA-NULL-SHA:ECDHE-ECDSA-NULL-SHA:
ECDH-RSA-NULL-SHA:ECDH-ECDSA-NULL-SHA:NULL-SHA256:NULL-SHA
```

SSH Disabled Ciphers

```
3des-cbc,aes128-cbc,aes192-cbc,aes256-cbc,rijndael-cbc@lysator.liu.se
```

SSH Disabled KEX Algorithms

```
curve25519-sha256@libssh.org,gss-gex-sha1-,gss-group1-sha1-,gss-group14-sha1-
```

SSH Disabled MAC Algorithms

```
hmac-sha1-etm@openssh.com,hmac-sha2-256-etm@openssh.com
```