Genesys Interaction Recording Solution Guide

Deploying Recording Crypto Server

12/19/2025

Deploying Recording Crypto Server

Contents

Genesys Interaction Recording (GIR) needs the Recording Crypto Server (RCS) to manage the certificates and the encryption/decryption process when retrieving and playing back the stored recording files.

> **Important**
>
> - RCS does not support on-the-fly configuration changes. Restart RCS to apply changes to the Genesys Advanced Disconnect Detection Protocol (ADDP) configuration.
> - RCS will not start if Configuration Server is using UCS-2 encoding. In this scenario, use UTF-8 or set the Configuration Server option **[confserv] allow-mixed-encoding** to `true`.

Installing Recording Crypto Server

Preparing the Host

You must install the correct JRE version on the host machine where the Recording Crypto Server will be installed. For Recording Crypto Server 8.5.095.22 (or higher), download and install Java Runtime Environment (JRE) 17 from your preferred provider. For example, you can download this from use an OpenJDK version of the software. For Recording Crypto Server 8.5.095.17 (or lower), download and install Java Runtime Environment (JRE) 8 from your preferred provider. For example, you can download this from Oracle or use an OpenJDK version of the software.

> **Important**
>
> For more detailed information about the supported versions for each operating system, see the Genesys Supported Operating Environment Reference Guide.

To install JRE:

1. Perform one of the following:

   - For Recording Crypto Server 8.5.095.22 (or higher), download and install Java Runtime Environment (JRE) 17 from your preferred provider. For example, you can download this from use an OpenJDK version of the software.
   - For Recording Crypto Server 8.5.095.17 (or lower), download and install Java Runtime Environment (JRE) 8 from your preferred provider. For example, you can download this from Oracle or use an OpenJDK version of the software.

- Set the following environment variables for your host, as follows:
  - (Linux) Insert the following lines into the **/etc/profile** file:

```
export JAVA_HOME=/usr/lib/java/jre-<version of Java downloaded>/jre
```
Log out and log in again to activate the new environment variables in the current session.

- (Windows) Create a new System Variable named `JAVA_HOME` and use the path that was used during installation as the value. To do this, right-click your Computer icon. Select **Properties > Advanced System Settings > Environment Variables**, and then create the **JAVA_HOME** variable.

Installing Recording Crypto Server Using the Deployment Wizard

For instructions about installing Recording Crypto Server using the Genesys Administrator Extension, see the Solution Deployment section of the Genesys Administrator Extension User Guide.

When Recording Crypto Server (RCS) is started for the first time, and then terminated (either by using the Solution Control Interface or by killing the process) soon after, the RCS directory structure might be left in a partially initialized state. This can cause RCS to fail on subsequent attempts to start. To work around this, do not terminate RCS for at least 60 seconds starting it for the first time. If the directory structure is still invalid, delete all sub-directories in the RCS root directory, except for the conf and legal directories. When RCS is re-started, the required directories will be created.

Installing the Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files

Important

If you are using Java 17, this step is not required.

In older versions of Java 8, the default installation limits key sizes to 128 bits. Larger key sizes can be enabled by installing Java Cryptography Extension (JCE) Unlimited Strength Jurisdiction Policy Files.

Important

- If you are using an OpenJDK version of Java, no additional cryptography configuration is required.
- If the Java 8 version you are using is older than Java 8u151, then follow the installation steps for JCE described below.
- If you are using Java 8u151 or 8u152, you do not need to download and install JCE. However, you must make a change to the unlimited policy option in the **JRE_HOME/lib/security/java.security** file. Find the **#crypto.policy=unlimited** line and remove the hash (**#**) character to uncomment it.
- If you are using Java 8u161 or newer, no additional cryptography configuration is required.

To install:

1. If you are using the Oracle version of Java 8, download the Java 8 specific package from the **Oracle** website and follow the instructions provided with the package.
2. Copy the **Local_policy.jar** and **Us_export_policy.jar** files to the **JRE_HOME/lib/security** directory. If there are already copies of these files in that directory, make backup copies of these existing files in case you want to revert the installation.

> **Important**
>
> Make sure that the policy files are installed before starting the RCS for the first time. RCS will not start without these files.

Upgrading Recording Crypto Server

1. Make a backup copy of the **rcs.properties** file.
2. Make a backup copy of the **keystore** file.
3. Uninstall the Recording Crypto Server component.
4. Install the new Recording Crypto Server component.
5. Copy the settings from the backup copy of the **rcs.properties** file to the new **rcs.properties** file.
6. Copy the backup **keystore** file to the desired **keystore** file location and update the **rcs.properties** configuration file's **keystorepath** parameter to point to this file.

Configuring Recording Crypto Server

This section describes how to configure the Recording Crypto Server in your environment using Genesys Administrator Extension.

For more information about using Genesys Administrator Extension, see the Genesys Administrator Extension Help.

Configuring the KeyStore and Certificate Authority

For information on how Genesys supports TLS for secure data exchange, refer to Securing Connections Using TLS in the Genesys Security Deployment Guide.

The Recording Crypto Server stores certificate and key data files based keystores. Certificates uploaded to the server can be optionally validated against a Certificate Authority (CA).

> **Important**
>
> The CA configuration is used for recording certificates and not for TLS network

connections. This section describes the keystore and CA related configuration parameters.

To limit access, all recording encryption key related parameters are stored in a local **<Recording Crypto Server Install Directory>/conf/rcs.properties** configuration file.

The following table lists the parameters used in the **rcs.properties** configuration file.

Parameter Name	Default Value	Description
keystorepath	`keystore.bin`	Specifies the path to the keystore file. If HA is enabled, the keystore file should be accessed through a network share (see Configure HA).
keystorepassword	`genesys`	Specifies the password that accesses the keystore file. **Note:** The `keystorepassword` parameter can be overridden by the `RCS_KEYSTORE_PASSWORD` environment variable. In this case the same password is used for both `keystorepassword` and `keypassword`.
keypassword	`genesys`	Specifies the password used for each private key that is added to the keystore. **Note:** • The same password is used for each private key. • The `keypassword` parameter can be overridden by the `RCS_KEYSTORE_PASSWORD` environment variable. In this case, the same password is used for both `keystorepassword` and `keypassword`.
cacertstorepath	`Java-ROOT`	Specifies the CA certificate keystore. Possible values are: • `Java-ROOT`—The path to the default Java JRE CA certificate file. • `Windows-ROOT`—The path to the Windows system keystore. This is not valid for Linux systems.

Parameter Name	Default Value	Description
		• File Path—The path to use the CA keystore. This file must be a Java JKS keystore file. • None—Disables validation of certificates.
cacertstorepassword	changeit	Specifies the password for the CA certificate keystore.

The following shows an example **rcs.properties** configuration file:

```
keystorepath=keystore.bin
keystorepassword=keystorepassword
keypassword=keypassword
cacertstorepath=Java-ROOT
cacertstorepassword=capassword
```

Configuring the Connection to Interaction Recording Web Services (Web Services)

The Recording Crypto Server uses API calls to Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) for recording playback and archival operations. To configure the Interaction Recording Web Services (Web Services) connection, set the following parameters in the **[htcc]** section of the Recording Crypto Server application:

Parameter Name	Default Value	Description
baseurl	https://htcchost:8080	Specifies the base URL for the Interaction Recording Web Services (Web Services) connection. This parameter is dependent on the Interaction Recording Web Services (Web Services) server protocol (http or https), port, and URL suffix.
internalUrlPrefix	/api/v2	Controls the prefix added to requests sent to Interaction Recording Web Services to retrieve recording files. By default, or if a value other than **disable** is specified, RCS will concatenate the **baseurl**, **internalUrlPrefix**, and the **mediaPath** returned by RWS as the request URL. If the **internalUrlPrefix** value is set to **disable**, RCS will use the **mediaUri** from the metadata instead when fetching the recordings from RWS.
domain	Empty string	Specifies the domain of the Interaction Recording Web Services (Web Services) contact

Parameter Name	Default Value	Description
		center. This is the domain ID set for the contact center within Interaction Recording Web Services (Web Services).
user	`ops`	Specifies the name of the operations user for the Interaction Recording Web Services (Web Services) connection.
password	`opspassword`	Specifies the password of the operations user for the Interaction Recording Web Services (Web Services) connection.
max-sr-playback-connections	`50`	Specifies the maximum number of HTTP connections between Recording Crypto Server and Interaction Recording Web Services (Web Services) for screen recording playback.
contactcenterid	Empty string	Specifies the contact center ID value in the RCS requests sent to Interaction Recording Web Services (RWS). If this value is not specified, the contact center ID information is derived from the **/api/v2/ops/contact-centers** request sent to RWS. **Important:** If you are a Recording Crypto Server API user and you specify an empty Contact Center ID (CCID) when using the **/rcs/contact-centers/<ccid>/recordings/...** path, you will receive a misleading HTTP 403 Access is denied message.
trusted-ca	`false`	Configures TLS certificate validation when making a secure outbound connection to Interaction Recording Web Services (RWS). Valid values are `true`, `false`, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. RCS will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to Interaction Recording Web

Parameter Name	Default Value	Description
		Services (Web Services) in the Configuring Transport Layer Security (TLS) Connections (Optional) section.

Configuring Cross Origin Resource Sharing (CORS)

If Interaction Recording Web Services (or Web Services if you're using version 8.5.210.02 or earlier) has Configuring Cross-Site Request Forgery (CSRF) protection enabled, CORS must be configured.

To configure CORS, set the following options in the **[cors]** section of the Recording Crypto Server application:

Parameter Name	Default Value	Description
allowed-origins	empty	Specifies the allowed origins list that is attached in the HTTP response Access-Control-Allow-Origins header, sent to a cross-origin request. In this case, it must be a list of all base URLs used by the users to connect to SpeechMiner Web.
allowed-headers	`X-Requested-With,Content-Type,Accept,Origin,Cookie,authorization,Access-Control-Allow-CSRF-TOKEN,Range`	Specifies the allowed headers list that is attached in the HTTP response Access-Control-Allow-Headers header, sent to a cross-origin request.
allowed-methods	`GET,POST,PUT,DELETE,OPTIONS`	Specifies the allowed methods list that is attached in the HTTP response Access-Control-Allow-Methods header, sent to a cross-origin request.
allow-credentials	`true`	Specifies the value sent in Access-Control-Allow-Credentials header of the HTTP response to cross-origin request.
enable-cors-filter	`true`	Enables handling cross-origin requests originating from other domains like SpeechMiner Web. The value should always be `true` for CORS security to be active.

Configuring SameSite Cookie for Screen Recording Playback (Optional)

The Recording Crypto Server provides the ability to enable the **SameSite=None** and **Secure** cookie attributes for the cookie used for screen recording playback in the SpeechMiner browser application. These attributes are not set by default.

To configure the **SameSite** and **Secure** cookie attributes, set the following option within the **[general]** section of the Recording Crypto Server application:

Important

Before enabling this option, ensure that the connection between the SpeechMiner browser application and Recording Crypto Server is configured to use HTTPS. If you set the value of this option to `true` and are using HTTP, the cookie will not be returned by the browser.

Parameter Name	Default Value	Description
samesite.enable	`false`	Specifies whether the **SameSite=None** and **Secure** cookie attributes are set during screen recording playback from the SpeechMiner browser application.

Configure Passwords

Important

- In a Linux or Windows environment, RCS supports reading the RCS keystore password from an environment variable instead of from the configuration file. When both are available, the environment variable takes precedence.
- **`RCS_KEYSTORE_PASSWORD`** - maps to the existing configuration parameters `keystorepassword` and `keypassword` in the RCS properties file. When specified the same password is used for both parameters.

In a Windows environment only, the Recording Crypto Server (RCS) can store the password in the Windows Vault instead of in the `rcs.properties` file.

For example, run the following commands for the Recording Crypto Server located at `<Recording Crypto Server Directory>\scripts\powershell`:

Command to store: `encryptPassword.bat [-store <path to credentials store>] -password <password>`

Command to start RCS: `startRCS.bat [-store <path to credentials store>] -rcs <command to start RCS>`

For example:

```
startRCS.bat -store C:\GCTI\RecordingCryptoServer\rcs.secret -rcs java %JAVA_OPTS%
-jar rcs.war -host host1.example.com -port 8888 -app RCS_Application
```

where:

- **`host1.example.com`** is the host for the Configuration Server.

- **8888** is the port for the Configuration Server.
- **RCS_Application** is the RCS application object.

> **Important**
>
> If the command <path to credentials store> contains a space, the path must be enclosed with quotation marks (").

Configuring Archiving

The Recording Crypto Server provides support for automatic archiving of recordings that are older than a predefined time.

> **Important**
>
> Genesys recommends that the Media Lifecycle Management (MLM) functionality, which provides more flexible backup and purging rules, be used instead (see Media Lifecycle Management). New features, such as protecting recordings from deletion, are not supported with the Recording Crypto Server archiving mechanism.

To configure archiving, set the following options:

1. In the **[general]** section, set the **archive.block-size** option to the number of recordings RCS will fetch for archiving. The valid value ranges from 100 to 10000 and the default value is 5000. This option is used to verify that RCS does not run out of memory when it fetches all of the recordings at one time for archiving.

   > **Important**
   >
   > Genesys recommends setting the RCS maximum Java heap size to no less than 1024 MB when **archive.block-size** is 5000. This setting enables you to avoid RCS running out of memory. Increase the maximum Java heap size accordingly when you increase the **archive.block-size**. To set the maximum Java heap size for RCS, add the **JVM** option (`-xmx1024m`), to the RCS start script.

2. On the Annex tab of each Tenant (including the Environment Tenant), in the **[recording.archive]** section, set the following parameters:

Parameter Name	Default Value	Description
interval	1	Specifies how often, in days, the archiving process runs.
retentiontime	60	Specifies how long, in days, to keep the recordings before archiving them.

Parameter Name	Default Value	Description
speechminerurl	`https://host/interactionreceiver`	Specifies the SpeechMiner URL where the recording metadata is stored.
user	`archiveuser`	Specifies the SpeechMiner username used to authenticate the SpeechMiner database.
password	`changeit`	Specifies the SpeechMiner password that is used to authenticate the SpeechMiner database.
outputfolder	`archive`	Specifies the destination folder where the archived recordings are stored.
speechminer-trusted-ca	`false`	Configures TLS certificate validation when making a secure outbound connection to SpeechMiner Interaction Receiver. Valid values are `true`, `false`, and the path to a trusted certificate authority (CA) bundle. The CA file must be in PEM format. RCS will exit during initialization under the following conditions: CA path does not exist, CA file is not a valid PEM file, or CA file is corrupted. For more information, see Configuring TLS connection to SpeechMiner Interaction Receiver in the Configuring Transport Layer Security (TLS) Connections (Optional) section.

> **Important**
>
> Genesys does not recommend using a Network driver for Recording Crypto Server archive output. Therefore, set output to be a physical hard drive on the same machine.

Configuring High Availability

The Recording Crypto Server provides High Availability (HA) support to multiple Recording Crypto Server instances accessed through a load balancer. In this mode, all Recording Crypto Server instances use the same keystore file accessed through a network share, and are accessed through a single URL that utilizes the load balancer. To configure HA:

1. Set the `Redundancy Type` to `Hot Standby` on each Recording Crypto Server application instance. This setting enables logic for coordinated access to a shared keystore file.

2. Create a network share for the keystore file and set the **keystorepath** parameter in the Recording Crypto Server local configuration file to point to this file. Ensure that each Recording Crypto Server instance has read and write access to the keystore file.
3. Set the Recording Crypto Server URL parameter of the SpeechMiner application to the load balancer URL of Recording Crypto Server. If Genesys Administrator Extension is to be configured with a tenant specific URL for Recording Crypto Server, set this to the URL of the load balancer.
4. Create a Recording Crypto Server Cluster application using the `recording_crypto_850` application template, and set the following parameters:
   - On the `General` tab:
     - `Application Name`—The name of the cluster (for example, `RCS_Cluster`).
     - `Working Directory`—A period `"."`.
     - `Command Line`—A period `"."`.
     - `Command Line Arguments`—A period `"."`.
     - `Host`—The name of the host that the load balancer is installed on. This host must be in the configuration database.
   - On the `Ports` tab:
     - Configure the port by following the instructions provided in the Configure HTTP / HTTPS Port section.
5. Add a connection in the Genesys Administrator Extension application to the Recording Crypto Cluster application.

> **Important**
>
> For RCS HA configuration, each RCS instance operates in primary mode. The `Backup Server` setting on the `Server Info` tab of each RCS application should be set to `None`.

Example Load Balancer Configuration

The following is example configuration for the Apache load balancer. The details of setting up the required Apache modules are not shown. The load balancer setup must include "session sticky" so that a session that starts on a particular balancer member continues to be directed to the same member. This is achieved in the example below using the **route** and **stickysession** parameters. The **route** value must be set to the application name of the Recording Crypto Server instance, where " " characters in the name are replaced with the _ character. For example, if the application name is **RCS 1,** set the **route** value to RCS_1.

```
<Proxy balancer://rcscluster>
BalancerMember https://rcshost1:port/rcs disablereuse=On connectiontimeout=10000ms
route=RCS1_Application_Name
BalancerMember https://rcshost2:port/rcs disablereuse=On connectiontimeout=10000ms
route=RCS2_Application_Name
ProxySet stickysession=JSESSIONID
</Proxy>
ProxyPass /rcs balancer://rcscluster
```

If High Availability mode is not to be used, set the Recording Crypto Server's application Redundancy

Type to `Not Specified`. For this mode, the keystore file can be located on the local file system, a network share is optional.

Configuring an HTTP / HTTPS Port

To configure a port, follow these steps:

1. Log onto Genesys Administrator Extension (GAX).
2. In the GAX **Configuration** tab, choose **Environment**. Then, click **Applications** and select Recording Crypto Server application.
3. Go to the **Ports** tab in the Recording Crypto Server application.
4. Add a port or edit the existing one by entering values in the fields, **Port ID** and **Communication Port**. Note that there must be only one port.

You can configure either an unsecured port or a secured port based on your requirement.

Configuring an Unsecured (HTTP) Port

1. Enter the value `http` in the **Connection Protocol** field.
2. Enter the value `unsecured` in the **Listening Mode** field.
3. Leave the other fields empty and click **Save**.

Configuring a Secured (HTTPS) Port

1. Leave the **Connection Protocol** field empty.
2. Enter the value `secured` in the **Listening Mode** field. This sets the value `tls=1` in the **Transport Parameters** field automatically.
3. If you are setting up Mutual TLS, add `tls-mutual=1` to the **Transport Parameters** field.
4. Configure the secure port parameters at the appropriate level, as follows:

> **Important**
>
> If the protocol is set to `https` or left blank, a TLS server certificate and private key must be configured. This is done using the common method for Genesys applications as documented in the Genesys Security Deployment Guide. The certificate and private key can be configured in the host object, the application object, and the application port entry for HTTPS.

- Host Level
  a. In the GAX **Configuration** tab, choose **Environment** and click **Hosts**.
  b. Click on the host object on which the server is running and enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
  c. Restart the Recording Crypto Server.

- Application Level
  a. In the GAX **Configuration** tab, choose **Environment** and click **Applications**.
  b. In the **General** tab of the Recording Crypto Server application object, enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
  c. Restart the Recording Crypto Server.
- Port Level
  a. In the GAX **Configuration** tab, choose **Environment** and click **Applications**.
  b. In the **Ports** tab of the Recording Crypto Server application object, click on the port that you created earlier and enter the absolute paths to the certificate, certificate key, and Trusted CA in the corresponding fields.
  c. Restart the Recording Crypto Server.

Configuration of certificates at the port level has precedence over the application level, which has precedence over the host level.
The private key PEM file must be in PKCS8 format. This can be achieved using the following openSSL command:
```
openssl pkcs8 -topk8 -nocrypt -in private_keyfile.pem -inform PEM -out private_keyfile_pkcs8.pem
```

For more information on securing connections, refer to the Genesys Security Deployment Guide.

Configuring the Connection to the Primary Configuration Server

To work with Configuration Server High Availability, the Recording Crypto Server (RCS) requires a connection to the primary Configuration Server application. For information on how to set this connection, see the Framework 8.5.1 Management Layer User's Guide.

RCS supports an Advanced Disconnect Detection Protocol (ADDP) connection to the Configuration Server. To enable ADDP, perform the following:

- Add the Configuration Server to the RCS Connections tab.
- Specify the connection protocol as ADDP.
- Configure remote and local timeouts, valid values are 0-3600, where 0 means no timeout.
- Specify the required trace mode, either Local, Remote, or both.

For additional details, see the Advanced Disconnect Detection Protocol page in the Framework 8.5.1 Deployment Guide.

Important

- You will see log messages about ADDP activity in the RCS logs despite switching ADDP Trace Mode to **Trace Is Turned Off** or **Trace On Server Side**. This is due to the underlying libraries handling ADDP protocol functionality.
- ADDP debug logging can be suppressed by modifying the **suppress-debug-loggers** value in the **[log]** section of the RCS configuration to contain:
  ```
  com.genesyslab.platform.commons.connection.interceptor.AddpInterceptor
  , com.genesyslab.platform.commons.timer.impl.SchedulerImpl
  ```

- Genesys Advanced Disconnect Detection Protocol (ADDP) will appear in the **[log]** section of the Configuration Server log files when **verbose**=`all`.

Configuring Log Output

The Recording Crypto Server supports the Genesys Management Framework log configuration. For information on how to set up log output appropriate for your Recording Crypto Server application, see the Common Log Options section of the Framework 8.5.1 Configuration Options Reference Manual.

Configuring the Connection to Message Server

The Recording Crypto Server must have a connection to the Message Server application to enable central auditing and alarming. For information on how to set this connection, see the Framework 8.5.1 Management Layer User's Guide.

Configuring Transport Layer Security (TLS) Connections (Optional)

Configuring TLS connection to Interaction Recording Web Services (Web Services)

1. Set up TLS on Interaction Recording Web Services. For more information, see Configuring TLS on the Server-Side for Interaction Recording Web Services section. For information on acquiring TLS certificates and private keys, see Genesys Security Deployment Guide.
2. In the [**htcc**] section of the Recording Crypto Server configuration file, set the `baseurl` parameter to use `https`.
3. In the [**htcc**] section of the Recording Crypto Server configuration file, configure the **trusted-ca** parameter as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted-ca** to `true`.
- If the TLS certificate being used by RWS is a self-signed certificate, set **trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

> **Important**
>
> If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and use TLS only for encrypted transmission, set **trusted-ca** to `false`. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime

Environment 8 versions disallow MD5 signatures for certificates.

> **Important**
>
> The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to SpeechMiner Interaction Receiver

1. Set up TLS on SpeechMiner Interaction Receiver. For more information, see SpeechMiner Server-Side Configuration.
2. On the **Annex** tab of each Tenant (including the Environment Tenant), in the [**recording.archive**] section, set the `speechminerurl` parameter to use `https`.
3. In the [**recording.archive**] section, configure the `speechminer-trusted-ca` parameter as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **speechminer-trusted-ca** to `true`.
- If the TLS certificate is a self-signed certificate, set **speechminer-trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

> **Important**
>
> If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and TLS is used only for encrypted transmission, set **speechminer-trusted-ca** to `false`. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 8 versions disallow MD5 signatures for certificates.

> **Important**
>
> The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to Message Server

1. Set up TLS on Message Server. For more information, see Securing Core Framework Connections section in the *Genesys Security Deployment Guide*. For information on acquiring TLS certificates and private keys, see Genesys Security Deployment Guide.
2. To connect to the secure TLS port, see Configuring a Secure Client Connection to Other Genesys Servers

section in the *Genesys Security Deployment Guide*.

3. In the properties of the **Connection** table, configure the **trusted-ca** parameter as follows:

- If the TLS certificate was issued by a well-known certificate authority such as Verisign, set **trusted-ca** to `true`.
- If the TLS certificate is a self-signed certificate, set **trusted-ca** to the path to a file containing the CA that generated the self-signed certificate. The file containing the certificate must be in PEM format.

> **Important**
>
> If there are intermediate certificate authorities forming a chain of trust, then the certificate of the root certificate authority must be the certificate being set.

- If you do not wish to verify the TLS certificate and TLS is used only for encrypted transmission, remove the **trusted-ca** parameter from the configuration. If verification is not configured, certificates will not be checked if they have expired or the server hostname is matching the certificate's common name or subject alternative name. However, certificates will be checked if they are signed with a strong signature algorithm. Newer Java Runtime Environment 8 versions disallow MD5 signatures for certificates.

> **Important**
>
> The statement about JRE 8 disallowing MD5 signatures for certificates was relevant for Java 7/8 transitions. It is no longer applicable for latest Java versions.

Configuring TLS connection to Configuration Server

1. Set up TLS on the Configuration Server. For more information, see Configuring TLS on Configuration Server in the *Genesys Security Deployment Guide*. Refer to Genesys Security Deployment Guide to acquire TLS certificates and private keys.
2. In the command line arguments of start information in the RCS application properties, change the port to use the Configuration Server Auto-Detect port.

For more information about the Recording Crypto Server options, see the Genesys Interaction Recording Options Reference.