SecurID Software Token Administrator's Guide

SecurID

Contact Information

RSA Link at https://community.rsa.com contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

RSA Conference Logo, RSA, and other trademarks, are trademarks of RSA Security LLC or its affiliates ("RSA"). For a list of RSA trademarks, go to https://www.rsa.com/en-us/company/rsa-trademarks. Other trademarks are trademarks of their respective owners.

License Agreement

This software and the associated documentation are proprietary and confidential to RSA Security LLC or its affiliates are furnished under license, and may be used and copied only in accordance with the terms of such license and with the inclusion of the copyright notice below. This software and the documentation, and any copies thereof, may not be provided or otherwise made available to any other person.

No title to or ownership of the software or documentation or any intellectual property rights thereto is hereby transferred. Any unauthorized use or reproduction of this software and the documentation may be subject to civil and/or criminal liability.

This software is subject to change without notice and should not be construed as a commitment by RSA.

Third-Party Licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed on the product documentation page on RSA Link. By using this product, a user of this product agrees to be fully bound by terms of the license agreements.

Note on Encryption Technologies

This product may contain encryption technology. Many countries prohibit or restrict the use, import, or export of encryption technologies, and current use, import, and export regulations should be followed when using, importing or exporting this product.

Distribution

Use, copying, and distribution of any RSA Security LLC or its affiliates ("RSA") software described in this publication requires an applicable software license.

RSA believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." RSA MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

June 2021

Contents

Preface

About This Guide

This guide is intended for RSA Authentication Manager administrators and IT personnel who will provision and deploy software tokens. Do not make this guide available to the general user population, with the exception of Authentication Procedures on page 18, which an administrator might choose to distribute.

This guide provides the following information:

- A description of the supported token types
- An overview of the methods for provisioning and deploying software tokens
- Information on security features provided for the software token app
- A troubleshooting section with workarounds for common issues, and a list of the error and informational messages provided by the app.
- Procedures for using the software token app that an administrator can distribute to users

For a complete list of SecurID documentation, see RSA Link.

SecurID Support and Service

You can access community and support information on RSA Link at https://community.rsa.com. RSA Link contains a knowledgebase that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Support for RSA Authentication Manager

Before you call Customer Support for help with the RSA Authentication Manager appliance, have the following information available:

- Access to the RSA Authentication Manager appliance.
- Your license serial number. To find this number, do one of the following:
  - Look at the order confirmation e-mail that you received when your ordered the product. This e-mail contains the license serial number.
  - Log on to the Security Console, and click **License Status**. Click **View Installed License**.
- The appliance software version. This information is located in the top, right corner of the Quick Setup, or you can log on to the Security Console and click **Software Version Information**.

Support for the Cloud Authentication Service and Identity Routers

If your company has deployed identity routers and uses the Cloud Authentication Service, RSA provides you with a unique identifier called the Customer Support ID. This is required when you register with RSA Customer Support. To see your Customer Support ID, sign in to the Cloud Administration Console and click **My Account > Company Settings**.

RSA Ready Partner Program

The RSA Ready Partner Program website at www.rsaready.com provides information about third-party hardware and software products that have been certified to work with RSA products. The website includes Implementation Guides with step-by-step instructions and other information on how RSA products work with third-party products.

Chapter 1: Overview

About the SecurID App

The SecurID app is authentication software that consists of a mobile app and separately installed software tokens. With a software token installed, the app generates 6-digit or 8-digit pseudorandom numbers, called tokencodes (one-time passwords), at regular intervals. Authorized users with supported Android or iOS devices can use a tokencode, in combination with a PIN, to access resources protected by SecurID, such as Virtual Private Networks (VPNs) and web applications.

Before provisioning and deploying software tokens, you must decide:

- How users will authenticate. See Supported Software Token Types below.
- Whether to generate SDTID files, CTF URL links, or CT-KIP URL links. See Provisioning and Distribution Methods on the facing page.
- Whether to bind each token to a specific Android or iOS device or leave the default binding (device class GUID.) See Device Binding on page 13.

Supported Software Token Types

SecurID supports the following software token types for user authentication:

- **PIN integrated with tokencode (PINPad-style).** The user enters a SecurID PIN in the Enter PIN screen on the Android or iOS device to produce a passcode (one-time password). The user authenticates by entering the passcode in the protected resource.
- **PIN followed by tokencode (fob-style).** The user authenticates by entering a SecurID PIN in the protected resource, followed by the current tokencode displayed on the device. The user experience is similar to authenticating with a hardware fob that displays tokencodes.
- **Tokencode only.** The user authenticates by entering the current tokencode displayed on the device (no PIN required).

Note: Because tokencode-only authentication does not use two-factor authentication, RSA strongly recommends that you require the standard logon password in addition to the tokencode. For more information about the proper use of tokens that do not require a PIN, see the *RSA SecurID Software Token Security Best Practices Guide*.

Management Features for Software Token

The SecurID app supports the following features for managing software tokens:

- **Multiple Token Support.** Users can import up to 10 software tokens per device. An Authentication Manager server can provision three software tokens to an individual user. SecurID software tokens can be provisioned to the same device by different companies.
- **Token Nicknames.** Users can set token names to identify their tokens. Token names are called "nicknames" in the authentication servers. Nicknames can contain up to 32 alphanumeric characters. In

addition, nicknames must be unique, are case sensitive, and cannot consist entirely of spaces.

As the administrator, you can optionally set a nickname when configuring a token record. If you do not set a nickname, tokens are imported to the app with default names based on installation order: Token 1, Token 2, and so on. The user can rename tokens after importing them to the app.

If you use Self-Service provisioning with Authentication Manager 8.1 or later, you can allow users to set a nickname when they request a token. The token is imported into the app with the user-supplied nickname.

- **Delete Token option.** Users can delete any token. Users who delete all of their tokens must contact an administrator to request replacement tokens, or use Self-Service if available.
- **Token Expiration Warning.** Software tokens expire on the first second of the token expiration date (00:00:00 GMT). To ensure that the user always has a working software token installed, the app displays a warning indicating how many days remain before the token expires, starting 30 days before the expiration date. The user can contact the administrator or use a Self-Service (if available) to request a replacement token.

Provisioning Software Tokens

To provision software tokens and authenticate iOS device users, you need a supported version of Authentication Manager.

To provision software tokens and authenticate Android device users, you need a supported version of RSA Authentication Manager, as described in the *Release Notes*, or RSA SecurID Authentication Engine 2.8.1 for Java.

Authentication Manager supports two methods for deploying SecurID software tokens:

- **Security Console.** You initiate the process of assigning and distributing the user's token using the Security Console, a web-based administrative console.
- **Self-Service Console.** You configure Self-Service provisioning and allow the user to create an account. The user then enrolls to use Self-Service and requests a software token, using a web-based Self-Service Console.

  Self-Service provisioning is included with the Authentication Manager Enterprise Server license.

See RSA Authentication Manager documentation on RSA Link.

RSA SecurID Authentication Engine (SAE) is an Application Programming Interface (API) that provides the back-end authentication functions of RSA SecurID. After the API is successfully integrated into your environment, RSA SecurID users can be authenticated without needing an RSA Authentication Manager server. For more information, see RSA SecurID Authentication Engine Documentation on RSA Link.

Provisioning and Distribution Methods

This section provides an overview of the methods available for distributing software tokens to Android and iOS devices.

QR Codes

SecurID supports scanning a CTF URL or CT-KIP URL encoded in a QR Code. The user points the device camera at the QR Code to automatically scan the token into the SecurID app.

Use one of the following methods to create the QR Code:

- **Generate a QR Code in RSA Authentication Manager 8.x.** RSA Authentication Manager 8.x can generate QR Codes that each contain a CT-KIP URL. To use this feature, the Self-Service Console is required. An administrator must create a software token profile that uses the iOS 2.x device type, dynamic seed provisioning (CT-KIP), and QR Codes. For instructions, see the RSA Authentication Manager Administrator's Guide on the *RSA Authentication Manager Documentation* page on RSA Link.
- **Convert a CT-KIP URL to a QR Code with a Third-Party Conversion Tool.** RSA Authentication Manager 8.1 or later generates custom URLs containing CT-KIP data. The scheme portion of the custom CT-KIP URL is com.rsa.securid. This scheme is required when using custom CT-KIP URLs to provision software tokens to the SecurID app. After generating a custom CT-KIP URL, use a third-party QR Code conversion tool to embed the custom CT-KIP URL in a QR Code.
- **Convert a CTF URL or an SDTID file to a QR Code.** You can generate a legacy-format custom CTF URL containing token data using RSA Authentication Manager 8.1 or later, but you must use a third-party QR Code conversion tool to convert the custom CTF URL to a QR Code.

  If you use Authentication Manager to generate software token files (SDTID files), you can use the can use the RSA SecurID Software Token Converter 3.1 (Token Converter 3.1) utility to convert an individual token file to a QR Code that contains a custom CTF URL.

  RSA SecurID Authentication Engine (SAE) for Java does not natively generate QR Codes. You must use the Token Converter 3.1 utility to convert an SDTID file to a CTF URL embedded in a QR Code.

  When Token Converter 3.1 converts an SDTID file to a QR Code, the output is a JPEG file containing the QR Code image. The SecurID app can scan the QR Code to import the token. If you password-protect the SDTID input file, the app prompts for the password to complete the QR Code import.

  Download RSA SecurID Software Token Converter 3.1 from https://www.rsa.com/en-us/products-services/identity-access-management/securid/software-tokens/software-token-converter and follow the instructions in the *RSA SecurID Software Token Converter 3.1 Administrator's Guide*.

Dynamic Seed Provisioning

Dynamic seed provisioning uses the Cryptographic Token Key Initialization Protocol (CT-KIP) to eliminate the need for a token distribution file (SDTID file).

Note: RSA recommends using dynamic seed provisioning because the CT-KIP process helps prevent the potential interception of the token's seed. Only use SDTID or CTF if your company policy dictates that the SecurID apps cannot connect to the Internet or that a CT-KIP server cannot be set up.

You deliver a dynamically provisioned token to the SecurID app with a QR code or by sending an email message containing a custom CT-KIP URL hyperlink to the email client on the user's device. The user taps the URL link in the email or enters the link in the app to import the token.

To support dynamic seed provisioning (CT-KIP) on iOS devices, you must make sure that the RSA Authentication Manager server meets the App Transport Security (ATS) requirements. For more information, see App Transport Security Requirements for Dynamic Seed Provisioning on the facing page.

File-Based Provisioning (SDTID Files)

Authentication Manager and RSA SecurID Authentication Engine (SAE) for Java can generate software token (SDTID) files. RSA strongly recommends protecting SDTID files with a token file password as part of the provisioning process.

To deliver a token, you send an email with an SDTID file attachment to the email client on the user's device.

If you password-protect the file, RSA recommends sending the password separately, using a secure channel and best practices for communicating sensitive data.

Compressed Token Format (CTF Strings)

Compressed token format (CTF) is an alphanumeric or numeric format for delivering software tokens to mobile devices.

RSA Authentication Manager 8.1 and later generates CTF strings in a legacy numeric format, as described in the RSA Authentication Manager 8.2 Administrator's Guide. If you require alphanumeric CTF strings, use Authentication Manager to provision password-protected SDTID files and then convert them using the RSA SecurID Software Token Converter 3.1 (Token Converter) command line utility.

RSA SecurID Authentication Engine (SAE) for Java administrators obtain CTF strings by exporting the token to an SDTID file. Convert the password-protected SDTID file using the Token Converter 3.1.

Note: RSA strongly recommends protecting CTF strings with a password. Set the password on the SDTID file when provisioning the token in Authentication Manager. Use the -password option on the Token Converter command line.

By default, Token Converter 3.1 generates alphanumeric CTF strings appended to a URL. To deliver the CTF string, you send an email containing the URL to the user's device. The user taps the URL or enters the link in the app to import the token, and enters the password to complete the import.

To download the Token Converter and documentation, go to https://www.rsa.com/en-us/products-services/identity-access-management/securid/software-tokens/software-token-converter

App Transport Security Requirements for Dynamic Seed Provisioning

Apple introduced the App Transport Security (ATS) feature in iOS 9. This network encryption and security feature requires a server that supports Transport Layer Security (TLS) protocol version 1.2 or later with forward secrecy ciphers and certificates that are signed using a SHA-256 or later signature algorithm.

RSA Authentication Manager 8.1 Service Pack 1 (SP1) Patch 13 or later with the TLS 1.2 Mode update applied supports the required TLS encryption version, but you must ensure that the SSL console certificate used by RSA Authentication Manager meets the ATS requirements.

If the SSL certificate that you use to secure your CT-KIP connections does not use SHA-256 or later, then you must replace it. The default RSA Authentication Manager SSL console certificates do not meet the ATS requirement. For instructions on replacing the RSA Authentication Manager SSL console certificate, see the RSA Authentication Manager Administrator's Guide.

Also ensure that your entire Authentication Manager CT-KIP provisioning infrastructure is ATS compliant. Non-compliant network appliances, such as proxy servers, firewalls, and load balancers, might prevent CT-KIP provisioning requests from reaching the RSA Authentication Manager CT-KIP server. These non-compliant appliances may require a simple SSL certificate replacement or more complicated firmware upgrades to achieve compliance. Please contact your appliance vendor for further assistance in ensuring that your appliances are ATS compliant.

If you meet these requirements, then iOS apps that are built with the RSA SecurID SDK 2.4 on Xcode 7.3.1 or later can perform CT-KIP provisioning with RSA Authentication Manager 8.1 Service Pack 1 (SP1) Patch 13 or later with the TLS 1.2 Mode update applied. Users who have SecurID Software Token installed are not required to download any additional updates to ensure iOS 9 or higher compatibility.

For more information on ATS, go to

https://developer.apple.com/library/ios/releasenotes/General/WhatsNewIniOS/A rticles/iOS9.html.

Provisioning Software Tokens Using the Security Console

RSA Authentication Manager includes the web-based Security Console that allows you to provision and distribute software tokens. An RSA Authentication Manager Super Admin must create a software token profile. The profile specifies software token configuration and distribution options.

If you plan to use several provisioning methods (for example, CT-KIP and CTF), create separate software token profiles for each method so that you do not have to edit the profile to change the distribution method.

When you add a software token profile, use the Android 2.x device definition file for Android apps and the iOS 2.x device definition file for iOS apps.

For more information, see the RSA Authentication Manager Administrator's Guide on the *RSA Authentication Manager Documentation page* on RSA Link.

Provisioning Software Tokens Using the Self-Service Console

RSA Authentication Manager 8.1 or later includes RSA Self-Service. The Self-Service Console provisioning component allows users to request SecurID tokens, including software tokens.

For more information, see the Help topic "RSA Self-Service Overview" on the *RSA Authentication Manager Documentation page* on RSA Link.

Security Features for Software Token

The SecurID app includes the security features described in this section.

Token Security on the Device

After a token is imported to an Android device, it is protected with a set of system attributes. After a token is imported to an iOS device, it is protected with unique application data that cannot be migrated to another device.

When the app needs to open the token database, it queries the system for the set of attributes and checks them for validity. If an unauthorized user or malware attempts to copy the token database to another machine or device, the user cannot obtain tokencodes or the app appears as not having a token. If the user obtains a new device, the software token must be reissued.

Next Tokencode Retrieval

RSA Authentication Manager and RSA SecurID Authentication Engine can detect when a user provides multiple incorrect one-time passwords (OTPs) in succession. (The default of invalid OTP entries is three.) This situation may be caused by user error, time drift on the device running the app, or it may indicate that an unauthorized user has gained access to the token and is attempting to use it. When this occurs, the authentication server places the token into Next Tokencode mode. The user must enter the next successive code (tokencode or passcode) to authenticate. Requiring the user to provide the next code helps ensure that the code is being generated by a token in the possession of the authorized owner.

When a user's token is in Next Tokencode mode, the user can tap an arrow on the token card in the SecurID app to immediately retrieve the next code without waiting for the next interval.

Show or Mask PIN for iOS Devices

By default, PIN characters are masked as the user enters them. The user can show or mask PIN characters in the native iOS Settings.

Detecting a Jailbroken or Rooted Device

Users should not download the SecurID app to a jailbroken or rooted device. The SecurID 3.0 app displays a warning message when a user attempts to do so. A future release will prevent the user from completing the download.

Software Token Configuration

RSA strongly recommends using the following for software tokens:

- Device binding
- Password protection for file-based tokens
- (iOS only) Policies for users that require complex passcodes for iOS data protection

Device Binding

When provisioning a software token record in Authentication Manager, bind the token by configuring a token extension attribute (DeviceSerialNumber). Binding allows installation only on a device or class of devices with a matching device ID.

RSA strongly recommends binding all tokens to a device class GUID.

Android Device Class GUID (globally unique identifier)

By default, software tokens provisioned for Android devices in RSA Authentication Manager 8.x are bound to the Android 2.x device class GUID (globally unique identifier). The Android device class GUID allows the user to import the token to any Android device that is supported by the SecurID app. It prevents the token from being imported to other types of devices running a SecurID app.

The Android device class GUID is:

a01c4380-fc01-4df0-b113-7fb98ec74694

iOS Device Class GUID (globally unique identifier)

Software tokens provisioned for iOS devices in Authentication Manager 8.1 or later are bound to a device class GUID (globally unique identifier), which is generated when the SecurID app is installed. This option allows the user to import the token to any iOS device that is supported by the SecurID app. It prevents the token from being imported to other device platforms or to desktops or laptops running a SecurID app.

The iOS device class GUID is

556f1985-33dd-442c-9155-3a0e994f21b1

Android Device ID

A device ID is a unique sequence of 24 letters and numbers assigned to a specific Android device by the SecurID app. A token bound to a device ID cannot be used on any other device.

Note: Uninstalling and reinstalling the SecurID app generates a new device ID. If a user reinstalls the app, you must obtain the new device ID and update the user's software token record in your authentication server.

You bind tokens to a device ID when configuring the token in Authentication Manager. The user must first provide the device ID. After installing the SecurID app, the user can select **Device ID** on the Welcome screen and choose one of the following options:

- **Email Device ID.** This option opens an email that is prepopulated with the device ID. The user enters the administrator's e-mail address in the To: field. Make sure you provide an email address to users so they can send you the email containing their device ID.
- **Copy Device ID.** This option copies the device ID to the device clipboard. Users who have a Self-Service account on Authentication Manager 8.x can access the Self-Service URL through their device browser and paste the device ID into a device binding field when requesting a software token.

Instruct users to treat the device ID as sensitive information and to use a secure channel to deliver it to the administrator. After the user sends the administrator the device ID, the administrator should use a separate, secure channel to communicate the information needed by the SecurID app to complete the provisioning process, for example, the CT-KIP URL.

Determine Your Device Binding Option

Use the following information to decide which binding option best suits your requirements.

Binding Option	Comments
Android device ID	• The token allows installation only on the device with the specified device ID. • In an administrator-driven provisioning scenario, requires the administrator to obtain the device ID from the user before configuring the token record. • In a Self-Service provisioning scenario, the user can obtain the device ID from the SecurID app and enter the device ID when requesting a software token.
Android device class GUID	• The token can be installed on any Android device. • Prevents ability to import the token to a computer or mobile device other than Android. • Allows administrators to bind all tokens to the same device class. • For Authentication Manager 8.x, eliminates the need to configure a token extension attribute since the device class GUID is the default binding entry.

iOS Binding ID

A binding ID (called a device ID in previous versions of the Software Token app) is a unique, 24-character hexadecimal string generated by the SecurID app running on a specific iOS device. A token bound to a binding ID cannot be used by any other app running on the same device or a different device.

You bind a token when configuring it in Authentication Manager. The user must first provide you the binding ID, which is generated when the SecuID app is installed. To send the binding ID, the user must have an email account configured on the device.

You must provide users with an email address. Instruct users to treat the binding ID as sensitive information and to use a secure channel to deliver it to you.

To view and email the binding ID, from the Welcome screen or Home screen, the user taps **More > About**.

Determine Your Device Binding Option

Use the following information to decide which binding option best suits your requirements.

Binding Option	Comments
Binding ID	• The token can only be used by the app running on the device with the specified binding ID. • The administrator must obtain the binding ID from the user before configuring the token record. • If using Self-Service, the user can bind the token when requesting a software token.
iOS device class GUID	• The token can be installed on any iOS device. • Helps prevent importing the token to a computer or mobile device other than iOS. • Administrators can bind all tokens to the same device class. • For RSA Authentication Manager 8.1 or later, the iOS device class GUID eliminates the need to configure a token extension attribute since the device class GUID is the default binding entry.

Token Passwords

SDTID files and compressed token format (CTF) strings should be protected during transit by assigning a unique password in your provisioning server. The user must enter the password in the SecurID app to import the token. The CTF should not be communicated together with the device ID or binding ID.

Assigning a unique token password can help prevent unauthorized access. However, if the software token does not use device binding, the password does not prevent a user who has access to both the SDTID file or CTF URL and the password from installing the token on multiple devices. For this reason, RSA strongly recommends using both device binding and password protection of SDTID files and CTF strings.

iOS Data Protection

To enable iOS data protection, users must use passcodes on their devices. Passcodes comes in two types:

- Simple, four-digit, numerical passcodes
- More complex alphanumeric passcodes, which are normally much longer (preferred)

To ensure that a device's sensitive data is protected, RSA strongly recommends that your policies direct users to use passcodes, preferably complex, alphanumeric passcodes.

Chapter 2: Installing and Using the SecurID App

Install and Manage the SecurID for iOS App

The SecurID app can be installed directly onto a device from the App Store.

Install and Manage the SecurID for Android App

The SecurID app can be downloaded and installed for free from Google Play or the Amazon Appstore (China only). Software tokens must be purchased separately from RSA or an RSA Secured Partner.

Upgrades

To upgrade from the Software Token app, install the latest version of the SecurID app from Google Play. The software token that the user used with the earlier version of the app will continue to work with the new app.

Internet Connectivity for CT-KIP URL

If a user needs to import a token from a CT-KIP URL, the user's device must be connected to the Internet. For example, if a user's device is Wi-Fi only, the user must leave Wi-Fi on to import the token.

Font Size Setting

The SecurID app has been optimized for the Normal font size setting. Using a setting larger than the Normal setting affects the look of certain screens in the app.

Device Changes

The SecurID app is not included in a system backup. If a user needs to restore a device from a system backup, instruct the user to download the SecurID app from the app store (if necessary) and re-import the tokens.

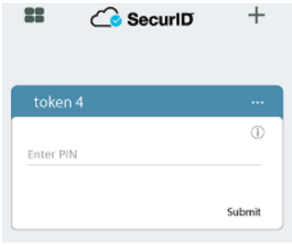
Authentication Procedures

This section describes three user authentication options. You can provide the appropriate procedures to your users. Instructions for users are also provided at https://help.access.securid.com/EN_US/Content/Production/ngx_c_what_is_product.html.

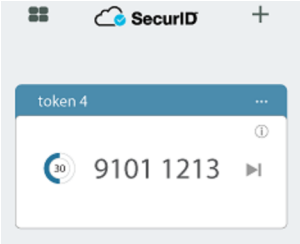
Passcode Authentication (PINPad-Style)

The following procedure shows how to authenticate to a VPN client with a PINPad-style software token (PIN integrated with tokencode).

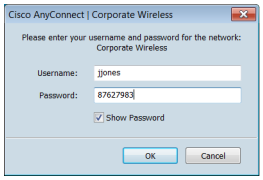
1. Enter the PIN in the SecurID app.



2. View the passcode (PIN integrated with the tokencode).



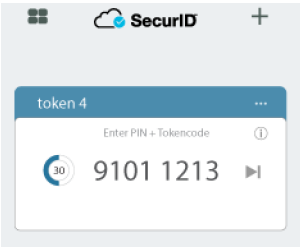
3. Enter the passcode in the protected resource (for example, a VPN).



Passcode Authentication (Fob-Style)

The following procedure shows how to authenticate to a VPN client with a fob-style software token (PIN entered in protected resource, followed by tokencode).

1. View the tokencode in the SecurID app.



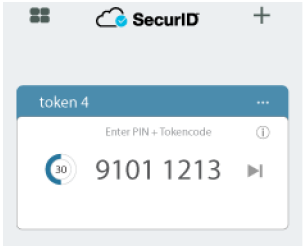
2. Enter the PIN in the protected resource (for example, a VPN). The PIN in this example is 13248675.

3. Enter the tokencode to the right of the PIN in the protected resource (for example, a VPN).

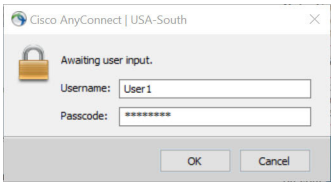
Tokencode-Only Authentication

The following procedure shows how to authenticate to a VPN client with a tokencode only. No PIN is required.

1. View the tokencode in the SecurID app.



2. Enter the tokencode in the protected resource (for example, a VPN).



Chapter 3: Troubleshooting

Problems Installing the SecurID App

This section describes problems that users might encounter when installing the app and provides workarounds.

Problem	Workaround
The SecurID for iOS app cannot be found in the Apple App Store.	The user has an unsupported version of iOS. The app supports devices running iOS 11 or later.
The SecurID for Android app cannot be found on Google Play.	The user has an unsupported device. In China, the app is only available in the Amazon Appstore.
The user cannot install the app.	The device does not have network connectivity, or a network failure occurred. Instruct the user to establish a network connection and retry. If this is unsuccessful, instruct the user to attempt to install another app.
The device does not have enough space to install the app.	Instruct the user to free up space on the device.
(Android only) A user cannot launch the SecurID app following installation.	Instruct the user to do the following: 1. Open the device Settings. 2. Disable Airplane mode (if enabled). 3. Turn on Wi-Fi.

Problem	Workaround
	4. Launch the SecurID app. **Note:** The device does not require a connection to a Wi-Fi network, and Wi-Fi can be turned off immediately after successfully launching the app.

Problems Importing Tokens

The SecurID app stores information about successful and unsuccessful token imports in a log. The log messages are duplicates of the messages users receive when a token import succeeds or fails. The log stores up to 20 entries. After the maximum is reached, the oldest log messages are deleted. If a user cannot import a token, ask the user to send you the log file as follows.

1. From the app Home screen, tap **More**.
2. Tap **Email Logs**.

Problems and Workarounds

The following table lists problems users might encounter when attempting to import tokens and suggests workarounds.

Problem	Workaround
User Error	
The user cannot import a token because the SecurID app is not installed on the device.	The user must download and install the SecurID app before importing a token.
In a file-based import (SDTID or CTF), the user forgot the token file password or entered an incorrect token file password.	The user must retry with the correct password or contact the administrator for the password.
The user attempted to	The user must establish a network connection. Open the device’s browser and try connecting to the URL.

Problem	Workaround
import a dynamically provisioned token (CT-KIP), but the import failed because the device does not have network connectivity.	
Administrator Error One of the following errors occurred in configuring the token in RSA Authentication Manager:	
(Android only) The token is not intended for an Android device. (iOS only) The token is not intended for an iOS device.	(Android only) Verify that you selected the Android 2.x device type. (iOS only) Verify that you selected the iOS 2.x device type.
The token device binding is incorrect. For example, the administrator may have entered an incorrect binding ID.	Correct the token device binding and reissue the token.
The token type is not supported, for example, 64-bit SID.	Provision a 128-bit (AES) token.
The death date of the token lifetime configured in Authentication Manager has passed.	Provision a new token.
CT-KIP Errors	
The user cannot	Correct the CT-KIP URL link, and reissue the token.

Problem	Workaround
import a token because of an error in the CT-KIP URL link. The administrator probably used an older, unsupported link format.	• (Android only) The URL link must start with the following prefix text: http://127.0.0.1/securid/ctkip? scheme= • (iOS only) The URL link must start with the following prefix text: com.rsa.securid://ctkip?url= ctkipData=
The email message containing the URL link did not reach the user's device	In rare cases, this can occur due to a network communication failure. Instruct the user to refresh the mailbox. If necessary, re-send the email to the user's device.
The user cannot import a token because the wrong email message format was used.	Embed the custom CT-KIP URL within a hyperlink, and set the message format to HTML.
The device cannot interpret URL links. Nothing happens when the user taps the link.	(Android only) Instruct the user to copy the link from the email and click the plus sign (+) in the app to manually import the token. (iOS only) Instruct the user to copy the link from the email and use the app's **Enter Link** option to manually import the token.
(Android only) The user cannot launch the app or import a software token, because the app cannot retrieve device information.	• (Android only) This problem affects HTC devices on a CDMA network if the device was set to Airplane mode or turned on in an area that did not have a cell signal. Instruct the user to disable Airplane mode, (if enabled), verify that the device has a network connection, and restart the device. • (Android only) This problem can also occur if the user attempts to launch the app or import a token from a CT-KIP URL link on a Wi-Fi only device without the Wi-Fi connection turned on.
Compressed Token Format Errors	
The user cannot import a token because of an error in the CTF URL link. The administrator	Correct the CTF URL link format, and reissue the token. • (Android only) The URL link must start with the following prefix text: http://127.0.0.1/securid/ctf?ctfData= • (iOS) The URL link must start with the following prefix text:

Problem	Workaround
probably used an older, unsupported link format.	com.rsa.securid://ctf?ctfData=
The SDTID file was not converted properly with the Token Converter due to a command line error.	See the *RSA SecurID Software Token Converter 3.1 Administrator's Guide*.
The Token Converter could not convert the SDTID file because the file contained double-byte characters in the **UserFirstName**, **UserLastName**, or **UserLogin** fields.	Double-byte characters are not allowed in these fields.
The user successfully imported the first token from an SDTID file containing multiple tokens, but could not import the remaining tokens.	The SecurID app only imports the first token in a multi-token file and ignores the remaining tokens. Make sure each SDTID file contains only one token.
The user cannot import a token because the wrong email message format was used.	Embed the custom CTF URL within a hyperlink, and set the message format to HTML.
The device cannot interpret	(Android only) Instruct the user to copy the link from the email and use the app's Import Token option to manually import the token

Problem	Workaround
URL links. Nothing happens when the user taps the link.	(iOS only) Instruct the user to copy the link from the email and use the app's**Enter Link** option to manually import the token.
The user cannot import a software token file (.sdtid) from an email file attachment.	Some email clients do not list the SecurID app as option to open the file. In that case, try using a different email client.
App Transport Security (ATS) Errors	
(iOS only) The user cannot import a token because of a configuration error with CT-KIP and ATS.	(iOS only) Confirm that your app meets the requirements specified in "Requirements for Connecting Using ATS" in the "Cocoa Keys" section in the Information Property List Key Reference: https://developer.apple.com/library/content/documentation/General/Reference/InfoPlistKey Refere nce/Articles/CocoaKeys.html#//apple_ref/doc/uid /TP40009251-SW57

Problems Authenticating

This section describes problems that users might encounter when attempting to authenticate with workarounds.

Problem	Workaround
User Error	
The token was disabled due to too many failed logon attempts.	Check the Authentication Manager logs. If the token is not disabled (or expired), ask the user to read you the current tokencode and the next tokencode. After you obtain the pair of tokencodes, resynchronize the token in Authentication Manager. **Note:** Instruct users with PIN-enabled tokens to tap Submit to display the tokencode. No PIN is required.
The user attempted to authenticate before setting a PIN.	(Android only) Instruct the user to follow the instructions in the SecurID app Help to set a PIN. (iOS only) Make sure the protected application prompts the user to set a PIN.
The user entered an incorrect PIN or entered the PIN in the wrong location. For example, when authenticating with a fob-style token, the user may have entered the tokencode, followed by the PIN, instead of entering the PIN, followed by the tokencode.	Instruct the user on how to authenticate with the specific token type. Instructions are provided in the app Help.

Problem	Workaround
Other	
The time on the Android device or the iOS device may be out of sync with the clock settings in Authentication Manager.	Instruct the user to access the Information screen in the app and read you the time shown in the **GMT** field. SecurID uses Coordinated Universal Time (UTC or GMT) settings to calculate the current one-time password. Software tokens rely on the client device to determine the correct UTC time value. For this reason, the local time, the time zone, and Daylight Saving Time must all be set correctly so that users can perform RSA SecurID authentication from their devices. Users who cross time zones with their devices only need to change the time zone to reflect the correct local time.
One or more tokens have expired.	The user can delete the tokens and contact the administrator to request replacement tokens or use Self-Service, if allowed.

Error Messages

The following tables list error conditions that users might encounter and associated error messages displayed by the SecurID app.

Message	Condition
Registration data expired. Contact your IT Help Desk.	The user attempted to register using invalid data.
Your device has reached the maximum token limit.	The user already has 10 tokens and attempted to import another token.
Enable camera permission to scan QR code.	The user must allow the app to access the camera to scan a QR code.
Enter a unique name for your token. or Token name already exists on this device.	Instruct the user to enter a unique name for each token on a device.
Your device is rooted. SecurID recommends that you do not proceed. Contact your IT Help Desk.	The user is attempting to download the app to a device that is rooted or jailbroken. The app will

Message	Condition
	download, but the user should be discouraged from using the app on this device.
Your token expired on [DATE]. Contact your IT Help Desk.	The token has already expired and the user must download a new token.
Your token [TOKEN NAME] expires on [DATE]. Contact your IT Help Desk.	The named token will expire on the given date.
Multiple Tokens About to Expire	Multiple tokens will expire soon.
You need an email account to share the Binding ID and email logs.	The user must be able to access an email account from the device where the app is installed.
Invalid PIN characters. PIN must contain 4 to 8 digits.	The user entered a PIN containing fewer than 4 digits or more than 8 digits.
Token import unsuccessful. Try again or contact your IT Help Desk.	The token import was unsuccessful but they user should try again before calling the IT Help Desk.
Token import unsuccessful. Invalid token data. Contact your IT Help Desk.	The token data is invalid and the token cannot be imported.

Information Messages

The following messages provide feedback and instructions to the user.

Message	Condition
Your token has been added.	The token has been imported to the user's device.
This server certificate was signed by an unknown certifying authority. If you trust this server, tap Accept to continue.	This message may be displayed during a CT-KIP

Message	Condition
	import for a variety of reasons, for example, if your RSA Authentication Manager CT-KIP implementation uses a self-signed certificate.
[TOKEN NAME] Will Expire on[DATE]	The token will soon expire. The user must contact the IT Help Desk.
A screenshot was taken of this screen on [DATE] at [TIME]. If you did not take this screenshot, contact your administrator.	This message warns the user that another person may have take a screenshot of the user's app screen. If this is the case, the app may be compromised.