RSA SecurID® Software Token Deployment Planning Guide

Version 4

Contact Information

Go to the RSA corporate web site for regional Customer Support telephone and fax numbers**: www.emc.com/domains/rsa/index.htm.**

Trademarks

RSA, the RSA Logo, RSA Secured, SecurID, SecurCare, and EMC are either registered trademarks or trademarks of EMC Corporation ("EMC") in the United States and/or other countries. All other trademarks used herein are the property of their respective owners. For the most up-to-date listing of RSA trademarks, go to **www.emc.com/legal/emc-corporation-trademarks.htm#rsa**.

License agreement

This software and the associated documentation are proprietary and confidential to EMC, 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.

Note on encryption technologies

The referenced 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 the referenced product.

Distribution

Use, copying, and distribution of any EMC software described in this publication requires an applicable software license.

Disclaimer

EMC does not make any commitment with respect to the software outside of the applicable license agreement.

EMC believes the information in this publication is accurate as of its publication date. EMC disclaims any obligation to update after the date hereof. The information is subject to update without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." EMC CORPORATION 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.

All references to "EMC" shall mean EMC and its direct and indirect wholly-owned subsidiaries, including RSA Security LLC.

April 2011
Revised January 2013, December 2013, June 2014

Contents

Preface

About This Guide

This guide describes how to plan an RSA SecurID® software token deployment in an enterprise environment. This guide is intended for customers who are considering adding software tokens to their current RSA SecurID product deployment or who want to replace their hardware tokens with software tokens.

Product Information

For more information about RSA SecurID software token product offerings, and associated documentation, go to **http://www.emc.com/security/rsa-securid.htm**.

Related Documentation

Secured by RSA® Certified Partner Solutions directory. RSA has worked with a number of manufacturers to qualify products that work with RSA products. Qualified third-party products include virtual private network (VPN) and remote access servers (RAS), routers, web servers, and many more. To access the directory, go to **https://gallery.emc.com/community/marketplace/rsa?view=overview**.

RSA Authentication Manager 8.x Administrator's Guide. Provides information about how to administer users and security policies in RSA Authentication Manager 8.x.

RSA Authentication Manager 7.1 Administrator's Guide. Provides information about how to administer users and security policies in RSA Authentication Manager 7.1.

RSA Security Console Help. Describes day-to-day administration tasks performed in the RSA Security Console interface used with RSA Authentication Manager 8.x or RSA Authentication Manager 7.1. To view Help, click the **Help** tab in the Security Console.

RSA Authentication Manager 6.1 Administrator's Guide. Provides information about how to administer users and security policy in RSA Authentication Manager 6.1.

Database Administration Application Help. Describes day-to-day administration tasks performed in the Database Administration application used with RSA Authentication Manager 6.1.

RSA SecurID Authentication Engine 2.8.1 for Java Developer's Guide. Explains how to use the Java API to integrate RSA SecurID authentication features into an existing server application. The *Developer's Guide* is available from RSA SecurCare Online. Go to **https://knowledge.rsasecurity.com**.

RSA SecurID Software Token Best Practices Guide. Describes best practices designed to ensure the secure operation of RSA SecurID software token products. This guide is available from RSA SecurCare Online. Go to **https://knowledge.rsasecurity.com**.

RSA SecurID Software Token Converter. The Token Converter is a command line utility for converting individual RSA SecurID software token files into alternative delivery formats, including custom compressed token format (CTF) URLs and QR Codes. To download the Token Converter, go to **http://www.emc.com/security/rsa-securid/rsa-securid-software-authenticators/converter.htm**.

Support and Service

RSA SecurCare Online	**https://knowledge.rsasecurity.com**
Customer Support Information	**www.emc.com/support/rsa/index.htm**
RSA Solution Gallery	**https://gallery.emc.com/community/marketplace/rsa?view=overview**

RSA SecurCare® Online offers a knowledgebase that contains answers to common questions and solutions to known problems. It also offers information on new releases, important technical news and software downloads.

1 Overview of RSA SecurID Software Tokens

What Is a Software Token?

RSA SecurID® software tokens are designed to support the same RSA SecurID one-time password (OTP) algorithms as RSA SecurID hardware tokens. Software tokens consist of two separately installed components, an RSA SecurID software token app built for the intended device operating system and a unique token seed record. Instead of being stored in an RSA SecurID hardware token, the seed (cryptographic key) is stored on the user's smartphone, tablet, desktop, or laptop.

Once installed into an RSA SecurID app, a software token generates a 6-digit or 8-digit pseudorandom number, or tokencode, at regular intervals. When the tokencode is combined with a PIN, it is called a passcode. The tokencode or passcode serves as the OTP. Users enter OTP values, along with other security information, to verify their identity when accessing a protected resource. RSA SecurID software tokens are engineered to be compatible with existing RSA Secured® applications.

Advantages of Software Tokens

RSA SecurID software is designed to protect an organization by helping to ensure that only authorized users are granted access to protected networked resources. RSA SecurID software tokens are engineered to provide the strength of RSA SecurID two-factor authentication combined with increased user convenience and efficient deployment and recovery.

Whether they have a personal smartphone or a corporate-issued laptop, users can use RSA SecurID software tokens on a device they already possess. Software tokens also provide the option of direct integration with enterprise applications through the software token automation API, further enhancing the user experience.

Software tokens can help reduce the overall cost of ownership of RSA SecurID technology for the organization in the following ways:

- Enable rapid electronic deployment to a distributed workforce
- Can be easily redeployed if users leave the organization
- Reduce Help Desk costs caused by lost or forgotten tokens

Software Token Apps

RSA SecurID software tokens are designed to be supported on smartphones, desktops and laptops, and web browsers. For current RSA software token offerings, go to **http://www.emc.com/security/rsa-securid/rsa-securid-software-authenticators.htm**.

In addition, the RSA Secured Partner Solutions directory provides software token solutions for other platforms and technologies, including USB flash drives and biometric devices. Go to **https://gallery.emc.com/tags?tags=rsa_securid_ready_authenticators**.

Software Token Seeds and Supported Token Types

All RSA SecurID software token apps are engineered to support RSA SecurID 820 software token seeds. RSA SecurID 820 seeds can be used with solutions developed and maintained by RSA, as well as with compatible solutions available from RSA SecurID Ready Authenticator Partners. Customers can acquire software token seeds in varying lifetimes (from 6 months to 10 years). Software tokens can be deployed to multiple devices over their lifetime, for example, iOS devices today and Android devices tomorrow.

RSA SecurID software token apps support 128-bit (AES algorithm) tokens; 64-bit (SID algorithm) tokens are not supported. The apps support time-based tokens only.

For more information or to purchase software token seeds, go to **http://www.emc.com/security/rsa-securid/rsa-securid-software-authenticators.htm** and click **Contact Sales**.

Token Assignment Limits

You can assign up to three RSA SecurID tokens to each authorized user, either all software tokens or a combination of hardware and software tokens. Most RSA SecurID software token apps support multiple software tokens to allow for users who need to log on to multiple domains under different identities. For token limits per app, see "Number of Tokens Supported Per App" on page 42.

Important: For security reasons, the token provisioning administrator should never deploy the same seed to multiple tokens. Each token should have a unique seed.

RSA SecurID PINs

RSA SecurID two-factor authentication requires using a personal identification number (PIN) as one of the factors. RSA strongly recommends using PINs with software tokens.

RSA Authentication Manager can be configured to require PINs of all the same length or to accept PINs that vary in length between 4 and 8 characters. For maximum security, RSA strongly recommends requiring complex, 8-character PINs. For details on PIN management in RSA Authentication Manager 8.x, see the *RSA Authentication Manager 8.x Security Configuration Guide*. For details on PIN management in earlier versions of RSA Authentication Manager, see the *RSA Authentication Manager 7.1 and 6.1 Security Best Practices Guide*.

When you add users to the RSA Authentication Manager database, you specify for each user individually whether the user can create the PIN or must accept a system-generated PIN. For PINPad-style software tokens (the default), the PIN must be numeric and cannot begin with a zero. For fob-style software tokens (supported with RSA Authentication Manager 8.x and RSA Authentication Manager 7.1), the PIN can be numeric (default) or alphanumeric. To require alphanumeric PINs, the administrator must configure the token policy in the RSA Security Console (**Authentication > Policies > Token Policies**).

If you use self-service provisioning, users with a valid Self-Service account can create their PINs when requesting a token.

Software Token Time Settings

The RSA SecurID algorithm uses time, expressed as Coordinated Universal Time (UTC or GMT), to calculate the current one-time password (OTP). Software tokens rely on the host device (for example, an iOS device) to provide the correct UTC time value. For this reason, the date, time, time zone, and Daylight Saving Time must all be set correctly to ensure that the software token can obtain the correct UTC time value. If possible, devices should be configured to use network time from a trusted time source, for example a network time protocol (NTP) server. The network time protocol is designed to synchronize the clocks of computers over a network.

Protecting Software Tokens During Provisioning

Software tokens should be protected during provisioning using device binding. Device binding associates a token with a device identifier. You can use device binding with all of the supported token provisioning methods described in Chapter 3, "Software Token Provisioning and Delivery."

Device Binding

To protect a software token during provisioning, the RSA Authentication Manager administrator should bind the token to a device class identifier or a device-specific identifier. You bind tokens during provisioning by setting the **DeviceSerialNumber** attribute in the RSA Security Console (RSA Authentication Manager 8.x or RSA Authentication Manager 7.1) or by creating token extension data (RSA Authentication Manager 6.1).

Device Class Identifier

A device class identifier is a globally unique identifier (GUID) associated with a specific class of devices, for example, iOS devices. RSA provides a specific GUID for each of its smartphone and desktop/laptop software token apps. This binding option allows the user to import the token to any device in a class of devices, for example, any device running a supported version of iOS. Binding to the device class prevents the token from being used on other types of devices running an RSA SecurID software token app. The GUIDs for the platforms supporting this attribute are listed in "Supported Device Binding Attributes" on page 41.

Device-Specific Identifier

A device-specific identifier (device ID or binding ID) is a unique value used to bind a token to one device in a class of devices. The identifier may come from the device hardware (BlackBerry 10, Windows Phone), or may be randomly generated for the device by RSA (iOS, Android). Device binding provides an increased level of assurance that a software token can be installed only on the authorized device it is intended for.

To use device binding, the administrator must have knowledge of the device-specific identifier. In most cases, the user can view the device ID in the installed RSA SecurID app and email this information to the administrator for device binding.

For a list of device-specific binding attributes supported by specific platforms, see "Supported Device Binding Attributes" on page 41.

Password Protection

RSA strongly recommends password protecting token files (SDTID files). All RSA Authentication Manager products can generate password-protected token files. Additionally, RSA Authentication Manager 8.x can generate password-protected custom CTF URLs containing token data.

Note: RSA Authentication Manager 7.1 and RSA Authentication Manager 6.1 do not natively generate custom CTF URLs. If you use one of these provisioning servers, you should generate an SDTID file and set a token file password. Then use the RSA SecurID Software Token Converter to convert the SDTID file to a password-protected custom CTF URL. For more information, see "RSA SecurID Software Token Converter" on page 21.

If you use the SAE API for token provisioning, you can specify a password string when exporting a software token to an SDTID file.

Assigning a unique token password can help protect against unauthorized users gaining access to a SDTID file (or to a custom CTF URL) and attempting to import the token to a different device. However, if the software token does not use device binding, the password does not prevent a user who has access to both the token file and the password from installing the token on multiple devices. For this reason, RSA strongly recommends using both device binding and password protection.

For more information about SDTID files, see "File-Based Provisioning" on page 19. For more information about custom CTF URLs, see "Compressed Token Format" on page 20.

Security Issues with Deploying Software Tokens to Multiple Devices

The software token seed license, RSA Authentication Manager, SAE, and software token apps do not prevent an administrator from deploying the same software token to multiple devices. (RSA Authentication Manager does prevent associating a single token with more than one user.)

RSA strongly discourages deploying the same token to multiple devices for the following reasons:

- **A software token provisioned to multiple devices cannot use device binding.**

  Because the token seed is used to generate one of the factors in two-factor authentication (something you have: the OTP), it is imperative that the token be used only on the intended device. For this reason, RSA strongly recommends using device binding, as described in "Device Binding" on page 10. However, if the administrator intends to provision the same software token to multiple devices owned by the same user (for example, an enterprise iPhone and a home PC), the administrator cannot use device binding, because different devices use different binding attributes. Without the protection of device binding, administrators must make sure they employ provisioning processes that prevent unauthorized users from accessing the software token.

- **Software tokens provisioned to multiple devices could experience time synchronization issues.**

  RSA Authentication Manager maintains information about clock skew for each software token seed to account for variations in device clock settings. When the same software token is used on multiple devices, each with its own clock settings and skew, users may experience problems. If the clock settings on the devices vary significantly, the user may be challenged to enter the next tokencode, or in some cases may even be blocked from authenticating.

Protecting Software Tokens Installed on the Device

After a software token has been imported to a device, it is stored in a token database and protected with a set of system attributes. When the software token app needs to open the token database, it queries the system for the set of attributes used and checks them for validity. If a user or malware attempts to copy the token database to another device, the user cannot obtain tokencodes or the software token app appears as though it does not have a token. If the user obtains a new device, the software token must be reissued.

Differences in the User Experience with Software Tokens

When adding software tokens to your RSA SecurID solution, you should be aware of differences in the user experience between hardware and software tokens.

Most RSA SecurID hardware tokens (for example, RSA SecurID 700, shown below) use a key fob form factor.

With a key fob, the user reads the current tokencode from the display, then enters an RSA SecurID PIN, followed by the tokencode, into the RSA SecurID protected resource. For example, the user would enter the PIN, followed by the tokencode, into a VPN client logon screen to protect the VPN session.

By contrast, RSA SecurID software tokens (RSA SecurID 820) come configured as PINPad tokens by default. PINPad software tokens behave similarly to PINPad hardware tokens, for example, the RSA SecurID 520, shown below.

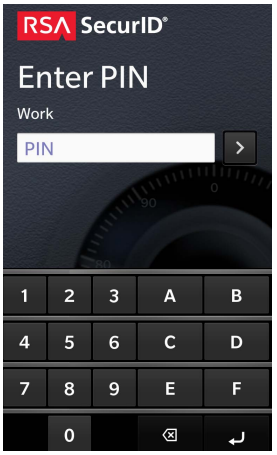
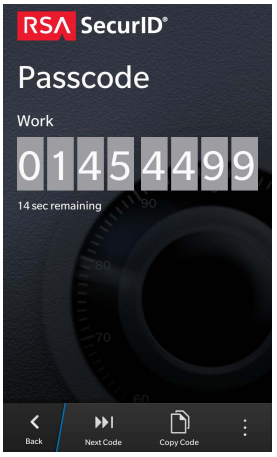
With a PINPad hardware token, the user enters the PIN on the PIN pad and then enters the resulting passcode into the RSA SecurID protected resource. With PINPad software tokens, the user enters the PIN into the software token app to generate a passcode. The user then enters the passcode into the RSA SecurID protected resource.

RSA SecurID software token apps support both PINPad-style and fob-style software tokens. If you want to issue fob-style software tokens, you must use RSA Authentication Manager 8.x, RSA Authentication Manager 7.1, or SAE.

Note: RSA Self-Service, RSA Credential Manager, and RSA Authentication Manager 6.1 (with RSA Authentication Deployment Manager) do not support fob-style software tokens.

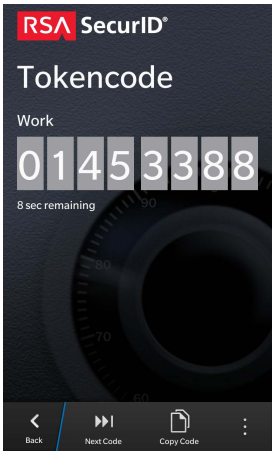
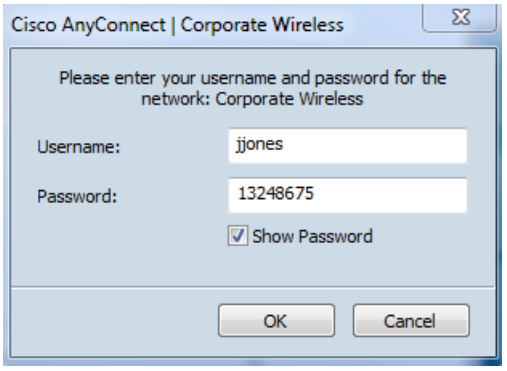
Passcode Authentication (PINPad-Style)

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

1	Enter the PIN in the RSA SecurID app on the device.	
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 table shows how a user authenticates to a VPN client with a fob-style software token (PIN entered in the protected resource followed by the tokencode).

1	View the tokencode in the RSA SecurID app on the device.	
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).	

Managing the Transition to Software Tokens

When making the transition to using software tokens, consider:

- How to manage the replacement of users' hardware tokens with software tokens.
- Which users and devices to support in your initial software token rollout.

Setting Up Software Tokens as Replacement Tokens

A convenient way to transition to software tokens is to replace users' hardware tokens with software tokens as their hardware tokens expire. You can also provision software tokens as needed to users who need both types of tokens and to users who have just joined your organization.

When you replace hardware tokens with software tokens, you can retain the existing PINs (default) or require users to set new PINs by setting the replacement tokens to New PIN mode. To retain a consistent user experience, you may want to reconfigure the replacement software tokens (RSA Authentication Manager 8.x and RSA Authentication Manager 7.1 only). For example, if you are replacing RSA SecurID 700 hardware fobs, you can configure the replacement software tokens as fob-style tokens. Users can continue to use their existing token until they first use their replacement token. At that time, the replaced token is unassigned.

For information on assigning replacement tokens in RSA Authentication Manager 8.x or RSA Authentication Manager 7.1, see the RSA Security Console Help. For information on assigning replacement tokens in RSA Authentication Manager 6.1, see the Database Administration Help.

2 Deploying Software Token Apps

This chapter provides a table showing the methods available for deploying RSA SecurID software token apps.

Application Deployment Options

RSA SecurID software token apps developed by RSA are packaged as standard applications for their respective platforms. You can use standard tools and techniques on each platform to deploy the apps, as shown in the following table. For details, see the *Administrator's Guide* for the app.

Note: Software token seeds are provisioned separately from the software token apps. For more information, see Chapter 3, "Software Token Provisioning and Delivery,"and Chapter 4, "Sample Software Token Deployment Scenarios."

Software Token Platform	Deployment Options
Android	Google Play Store
iOS	Apple App Store
BlackBerry 10	BlackBerry World
Window Phone	Windows Phone Store
Windows and Mac OS X laptops and desktops	• Installation by the user • Software distribution platforms such as Microsoft Systems Management Server (SMS)
RSA Toolbar in web browser	Installation by the user
Active X control in a web application (Web SDK)	Installation of Active X component by the user

3 Software Token Provisioning and Delivery

Software Token Provisioning

RSA SecurID technology supports the following methods for provisioning software tokens:

- File-based provisioning using SDTID files is available in all RSA Authentication Manager products and the RSA SecurID Authentication Engine (SAE) API.
- Compressed token format (CTF) provisioning, which provides an alternative to token files, is available in RSA Authentication Manager 8.x. For other versions of RSA Authentication Manager, the Token Converter is required to generate custom CTF URLs.
- Dynamic seed provisioning, available in RSA Authentication Manager 8.x and RSA Authentication Manager 7.1.

File-Based Provisioning

STDID files are generated from token records imported into any RSA Authentication Manager product or SAE. The token file contains the seed (cryptographic key) used by the RSA SecurID algorithm and other data associated with the token (serial number, expiration date, number of digits allowed in the tokencode, and so on). The SDTID file must be imported into the software token app on the user's device to enable one-time password (OTP) generation.

For file-based provisioning scenarios, see Chapter 4, "Sample Software Token Deployment Scenarios." For detailed instructions on using SDTID files, see the *Administrator's Guide* for the RSA SecurID software token app you plan to use.

Compressed Token Format

Compressed token format (CTF) is an alternative to distributing software token data within SDTID files. RSA Authentication Manager 8.x can natively generate custom CTF URLs that can be imported into RSA SecurID apps that support this format. RSA Authentication Manager 7.1 and RSA Authentication Manager 6.1 do not generate Custom CTF URLs. If you use these provisioning servers, you should generate password-protected SDTID files and then convert them to custom CTF URLs using RSA SecurID Software Token Converter (Token Converter). For more information, see "RSA SecurID Software Token Converter" on page 21.

For CTF provisioning scenarios, see Chapter 4, "Sample Software Token Deployment Scenarios." For details on generating custom CTF URLs, see the Token Converter documentation. Software token and Token Converter documentation is available from the EMC web at **http://www.emc.com/security/rsa-securid/rsa-securid-software-authenticators.htm**.

Dynamic Seed Provisioning

The dynamic seed provisioning method uses the four-pass Cryptographic Token Key Initialization Protocol (CT-KIP) to exchange information between an RSA SecurID software token app (client app) running on a smartphone, tablet, desktop, or laptop, and the CT-KIP server, which is a component of the RSA Authentication Manager server or an RSA Authentication Manager web-tier server. The information exchanged between the client and server is used to generate a unique shared secret (token seed). Information critical to the seed generation is encrypted during transmission using a public-private key pair. The generated token seed value is never transmitted across the network.

RSA recommends using dynamic seed provisioning whenever possible, because the four-pass protocol prevents the potential interception of the token's seed during the provisioning process.

The four-pass CT-KIP protocol is initiated by a request from the RSA SecurID software token app to the CT-KIP server when the user selects an option to import a token. Dynamic seed provisioning uses a unique one-time provisioning activation code to ensure that the request is legitimate. The client app must be provided with the activation code, either through manual user entry or as part of a custom CT-KIP URL link sent to the user's device email. The CT-KIP server evaluates the submitted activation code, and if the server determines that the request is valid, the four-pass process continues, ultimately resulting in a successful token import.

For dynamic seed provisioning scenarios, see Chapter 4, "Sample Software Token Deployment Scenarios." For detailed instructions on using dynamic seed provisioning, see the *Administrator's Guide* for the software token app you plan to use.

RSA SecurID Software Token Converter

The RSA SecurID Software Token Converter (Token Converter) is a free command line utility for converting individual RSA SecurID software token files into alternative delivery formats, including custom compressed token format (CTF) URLs and QR Codes containing custom CTF URLs.

Custom CTF URLs can be imported into the RSA SecurID mobile apps for Android, iOS, and Windows Phone. The custom CTF URL is delivered as a hyperlink in the body of an email message.

QR Codes containing custom CTF URLs can be imported into the 2.0 releases of the RSA SecurID software token apps for Android and iOS. These apps contain a built-in QR Code reader that uses the device camera to scan the QR Code.

Note: To import QR Codes into the iOS app, iOS devices must be running iOS 7 or later.

Custom CT-KIP URLs can be converted into QR Codes and scanned into the 2.0 releases of the RSA SecurID software token apps for Android and iOS. However, this requires the use of a third-party QR Code conversion tool, as the Token Converter does not convert custom CT-KIP URLs.

The Token Converter utility and documentation can be downloaded from **http://www.emc.com/security/rsa-securid/rsa-securid-software-authenticators/converter.htm**.

RSA Professional Services offers additional solutions for provisioning software token data within QR Codes. For more information, see "RSA Authentication Manager Prime Deployment Tools" on page 27.

Administrator-Driven and Self-Service Provisioning Models

RSA SecurID software token seeds are typically deployed in one of two provisioning models, described in the following sections.

- In an administrator-driven model, an enterprise administrator decides that a user should be enabled for an RSA SecurID software token and initiates the process to assign and deploy the user's token. Administrator-driven models may be used when a software token is issued as part of the standard process for bringing onboard a new employee, partner, or customer.
- In a user Self-Service model, the end-user requests a software token, potentially as part of requesting access to another enterprise service, for example VPN access.

Administrator-Driven Provisioning with RSA Authentication Manager

Note: This section describes software token provisioning tasks using RSA Authentication Manager. You can also provision software tokens using the RSA SecurID Authentication Engine (SAE) API. For more information, go to **http://www.emc.com/security/rsa-securid/rsa-securid-authentication-engine.htm**. Also see the *RSA SecurID Authentication Engine 2.8.1 for Java Developer's Guide*, available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

In an administrator-driven model, software tokens can be provisioned using:

- RSA Authentication Manager 8.x or RSA Authentication Manager 7.1 (with latest service pack). RSA Authentication Manager 8.x currently includes versions 8.0 and 8.1.
- RSA SecurID Appliance 3.0.
- RSA Authentication Manager 6.1.

Provisioning Tasks

To provision software tokens, the RSA Authentication Manager administrator must complete the following tasks.

1. Import token record files purchased from RSA.
2. Do one of the following, if needed:
   - For RSA Authentication Manager 8.x, add a software token profile for each platform for which the administrator plans to deliver software tokens. Each software token profile contains a "device definition type" that defines the software token attributes supported with a specific RSA SecurID software token app.
   - RSA Authentication Manager 7.1 does not use software token profiles. Instead the administrator may need to import a device definition file. This is an XML file that specifies the software token attributes supported with the specific RSA SecurID software token app. In most cases, RSA Authentication Manager 7.1 contains the required device definition files. For specific requirements, see the *Administrator's Guide* for the app.
   - RSA Authentication Manager 6.1 does not require software token profiles or device definition files.

3. Configure the tokens by entering the appropriate information. This includes:
   - Assigning a PIN or requiring the user to create a PIN or change the assigned PIN on first use of the token.
   - Binding the token. This requires obtaining device binding information from the user. For more information, see "Device Binding" on page 10. For supported device binding attributes, see "Supported Device Binding Attributes" on page 41.
   - Choosing a token distribution method.
4. Create users, if not already done.
5. Assign tokens to users.
6. Deliver the token using a platform-supported option. For more information, see "Delivery Options for File-Based Tokens, Custom CTF URLs, and QR Codes" on page 24 and "Delivery Options for Dynamically Provisioned Tokens" on page 26.

Self-Service Provisioning

RSA provides Self-Service provisioning options to automate and simplify the process of assigning and distributing RSA SecurID software tokens to users. RSA strongly recommends making Self-Service token provisioning accessible only when the user has already been authenticated by some other method, inside the network.

Self-Service provisioning reduces the amount of time needed to deploy software tokens. This significantly reduces administrative costs. Self-Service provisioning automates many of the complicated and time-consuming tasks administrators have traditionally performed when deploying RSA SecurID tokens.

RSA Self-Service

RSA Self-Service, available with RSA Authentication Manager 8.x, allows users to request tokens, set a PIN, create a nickname for a token, and perform other tasks. The administrator sets up Self-Service through the RSA Security Console. For example, you can use RSA Self-Service options to configure user logon options and define the token delivery settings for token files (SDTID files), custom CTF URLs, or custom CT-KIP URLs. Once you define the settings, users can log in to the Self-Service portal and request a token. When the request has been approved, the Service delivers the token using the appropriate method. For more information on configuring RSA Self-Service, including workflow options, see the *RSA Authentication Manager 8.x Administrator's Guide*.

Note: You must have the RSA Authentication Manager Enterprise Server license to use Self-Service provisioning. If you have Base Service license, you must upgrade to the Enterprise Service license to use RSA Self Service.

RSA Credential Manager

RSA Credential Manager (available with RSA Authentication Manager 7.1) provides most of the same features as RSA Authentication Manager 8.x Self-Service. The Self-Service console can be configured to use either SDTID files or dynamic seed provisioning (custom CT-KIP URLs) to distribute a software token once a user request has been approved. For a scenario on using RSA Credential Manager software to deploy software tokens, see "Android, BlackBerry 10, iOS, and Windows Phone Devices with Self-Service" on page 39. For platform-specific information on provisioning software tokens using RSA Credential Manager software, see the *Administrator's Guide* for the software token app you plan to use. For more information on configuring RSA Credential Manager software, including workflow options, see the *RSA Authentication Manager 7.1 Administrator's Guide*.

Note: RSA Credential Manager software is included with RSA Authentication Manager Enterprise Edition. You must have the Enterprise Edition to use RSA Credential Manager.

RSA Authentication Deployment Manager

RSA Authentication Deployment Manager 1.3 (formerly RSA Web Express and available with RSA Authentication Manager 6.1) supports delivering SDTID files once the request has been approved. Options for distributing the token include sending the token as an email attachment. For more information on configuring RSA Authentication Deployment Manager, including workflow options, see the RSA Authentication Deployment Manager documentation available from RSA SecurCare Online at **https://knowledge.rsasecurity.com**.

Delivering Software Tokens

The following sections list the methods available for delivering file-based and dynamically provisioned tokens.

Delivery Options for File-Based Tokens, Custom CTF URLs, and QR Codes

The following options are available for delivering file-based software tokens and custom CTF URLs:

- SDTID files are usually delivered to the device as an email attachment.
- SDTID files that have been converted into custom CTF URLs are delivered to the device as a hyperlink in an email.
- QR Codes containing custom CTF URLs can be scanned into the 2.0 releases of the RSA SecurID apps for Android and iOS. iOS devices must be running iOS 7 or later. RSA places no restrictions on how you deliver QR Codes to Android and iOS devices. Users point the device camera at the QR Code to automatically scan and import the token.

The following table lists the delivery options by platform. For platform-specific details on delivering token files, see the *Administrator's Guide* for the software token app.

Platform	Delivery Options for File-Based Tokens, Custom CTF URLs, and QR Codes
RSA SecurID Software Token 2.0 for Android	• Send an SDTID file as an email attachment. • Convert an SDTID file to a custom CTF URL using the Token Converter and send the CTF URL in an email. • Convert an SDTID file to a custom CTF URL embedded in a QR Code (requires Token Converter 3.0 or 3.1.). The app can automatically scan the QR Code to import the token.
RSA SecurID Software Token 2.0 for iOS	• Send an SDTID file to the device as an email attachment. • Convert an SDTID file to a custom CTF URL using the Token Converter and send the CTF URL in an email. • Convert an SDTID file to custom CTF URL embedded in a QR Code (requires Token Converter 3.1). The app can automatically scan the QR Code to import the token. The iOS device must be running iOS 7 or later.
RSA SecurID Software Token 1.0 BlackBerry 10	Send an SDTID file as an email attachment.
RSA SecurID Software Token 1.0 Windows Phone	Convert an SDTID file to CTF using the Token Converter and send the CTF string in an email. The user must copy the CTF string and paste it into the app.
RSA SecurID Software Token 4.1.x for Windows and Mac OS X laptops and desktops	• Send an SDTID file as an email attachment. • Import an SDTID file with the RSA SecurID Software Token Import utility. • Copy an SDTID file to a specified folder, for example, **My Documents** on Windows, where the software token app will automatically import it. The file can be delivered by a software management system (such as Microsoft Systems Management Server).
RSA SecurID Toolbar 1.4.2	Copy an SDTID file to a specified folder, for example, **My Documents**, where the Toolbar app will automatically import it.
Active X control in a web application (Web SDK)	File-based provisioning is not supported.

Delivery Options for Dynamically Provisioned Tokens

Dynamically provisioned software tokens are typically delivered as a custom CT-KIP URL hyperlink in an email.

The following table lists the CT-KIP delivery options by platform. For platform-specific details, see the *Administrator's Guide* for the specific software token app.

Platform	Delivery Options for Dynamically Provisioned Tokens
RSA SecurID Software Token 2.0 for Android	• If using RSA Authentication Manager 8.x, send the natively generated custom CT-KIP URL link to the device in email. RSA Authentication Manager 8.x generates the correct URL format for Android. • If using RSA Authentication Manager 7.1, construct a custom CT-KIP URL link and send to the device in email. • Convert the custom CT-KIP URL into a QR Code using a third-party QR Code conversion tool. The app can automatically scan the QR Code to import the token.
RSA SecurID Software Token 2.0 for iOS	• If using RSA Authentication Manager 8.x, send the natively generated custom CT-KIP URL link to the device in email. RSA Authentication Manager 8.x generates the correct URL format for iOS. • If using RSA Authentication Manager 7.1, construct a custom CT-KIP URL link and send to the device in email. • Convert the custom CT-KIP URL link into a QR Code using a third-party QR Code conversion tool. The app can automatically scan the QR Code to import the token. The device must be running iOS 7 or later.
RSA SecurID Software Token 1.0 for BlackBerry 10	Construct a custom CT-KIP URL link and send to the device in email.
RSA SecurID Software Token 1.0 Windows Phone	Construct a custom CT-KIP URL link and send to the device in email.
RSA SecurID Software Token 4.1.x for Windows and Mac OS X laptops and desktops	• Supply a custom CT-KIP URL and activation code for entry in software token app. • Supply only the activation code for entry in software token app (requires custom policy). Applies to both Windows and Mac OS X laptops/desktops. • Configure automatic import (requires custom policy). Applies to Windows desktops/laptops only.
RSA SecurID Toolbar 1.4.2	A custom CT-KIP URL and activation code can be imported directly by clicking the URL link or can be copied and pasted from a custom CT-KIP URL link in an email.
Active X control in a web application (Web SDK)	A custom CT-KIP URL and activation code can be embedded as a hidden field in a web page to provision the token through CT-KIP automatically.

RSA Authentication Manager Prime Deployment Tools

RSA Professional Services offers RSA Authentication Manager Prime, a suite of highly flexible and extensible software components that enable streamlining of RSA SecurID life cycle management. The components described in this section facilitate RSA SecurID software token deployment and management.

To learn more about the RSA Professional Services suite, go to http://www.emc.com/security/rsa-securid/rsa-authentication-manager-prime.htm. To contact RSA Professional Services, click **Contact Sales**.

AM Prime Self-Service Portal

The RSA Authentication Manager Prime Self-Service Portal (AM Prime SSP) allows users to request, activate, replace, and manage tokens.

The 2.0 releases of the RSA SecurID software token apps for Android and iOS contain a built-in QR Code reader for importing a token by scanning a QR Code. The AM Prime SSP supports token distribution using QR Codes directly in the portal UI. When an authorized user enters the SSP to retrieve a token, a QR Code that is valid for a short period of time is presented to the user to initiate the token import process. The QR Code can contain either a custom CTF URL or a custom CT-KIP URL. The QR Code is scanned automatically, using the device camera, and the token is quickly imported and ready to use.

Note: Customers who want to add or build similar QR Code token distribution functionality into their own in-house portal can do so by using the AM Prime AM Integration Services (AMIS) component.

RSA Authentication Manager Bulk Administration

RSA Authentication Manager Bulk Administration (AMBA, available for RSA Authentication Manager 8.x or RSA Authentication Manager 7.1) is a standalone utility for preparing bulk software token deployments. AMBA supports scripting operations against RSA Authentication Manager software, such as token assignment and issuance.

AMBA can be configured to use a configuration file as input to drive bulk operations. AMBA can generate token files (SDTID files) or set up tokens for dynamic seed provisioning (CT-KIP). AMBA does not provide its own methods for delivering the SDTID files or the CT-KIP activation codes to the user. However, AMBA can be incorporated in a larger script or automation process that can handle the delivery of the SDTID files or activation codes.

Redistributing Software Tokens

When planning software token provisioning, consider how you will handle redistributing software tokens in the following cases:

- The user obtains a new device.
- The user wipes the device.
- The installed token nears its expiration date or expires.
- You suspect your environment has been compromised.

Note: On certain platforms, such as iOS, a new software token is required if the user removes and then reinstalls the RSA SecurID app.

Redistributing a Software Token to a New Device

With most RSA SecurID software token apps, the user must reinstall the software token app and import a new token if the user obtains a replacement device, due to the token being bound to the device (through the device ID) or copy protection of the token.

Redistributing a Software Token to a Wiped Device

Most RSA SecurID software token apps require the user to reinstall the app and re-import the token or tokens if the user wipes the device. However, BlackBerry 10 and iOS devices can make use of data backup options.

- For BlackBerry 10 devices, RSA strongly recommends having users back up their data (personal or work) through BlackBerry Link. Backing up their data (including their token data), allows them to restore the data to the same device. (They cannot restore token data to a different device due to copy protection.) Users see an option to download BlackBerry Link when they first connect their devices to their computers. They can also download the software by going to **http://us.blackberry.com/software/desktop/blackberry-link.html**. For more information on backing up token data, see the *RSA SecurID Software Token for BlackBerry 10 Administrator's Guide*.
- For iOS devices, RSA strongly recommends having users perform an encrypted sync and a backup through iTunes or iCloud as soon as the app is installed and tokens have been imported. This will allow the user to recover tokens along with the app when the user performs a restore operation. For more information, go to **http://support.apple.com/kb/ht4946**.

Redistributing a Software Token to the Same Device

When you redistribute a software token, the authentication server generates a new token seed, which ensures that the user will have a different and unique seed for generating OTPs. RSA recommends redistributing software tokens if you have reason to believe your software tokens have been compromised or you need to change certain token attributes, such as the PIN type, tokencode length, or tokencode duration.

With few exceptions, RSA SecurID software token apps will reject a software token during import if its serial number matches a serial number stored in the token database on the device. Apps that support multiple tokens will not import a token if the app already contains the maximum number of tokens allowed. If the software token app allows the user to delete tokens, the user must delete the token and then import the replacement token. If the software token app does not provide an option to delete tokens (for example, RSA SecurID Software Token for Windows Phone), the user must remove the app to delete the token database. The user must then reinstall the app and import the replacement token.

Managing Software Token Expiration

Software tokens expire on the first second of their expiration day in GMT. RSA SecurID software token apps provide several features to help ensure that users always have a working software token installed.

- The user can view the token expiration date in the RSA SecurID software token app.
- The software token apps provide mechanisms to inform the user that a token has expired (for example, a “Token expired” message).
- On certain platforms, the software token apps display a token expiration notification starting 30 days before the token expiration date. The 30-day window gives the user ample time to request a new software token and import it to the device.

RSA SecurID Software Token for Windows and Mac OS X provide customized options associated with token expiration. You can set the expiration notification policy on Windows and Mac desktops to change the default number of days before the software token app displays a notification indicating that a token is nearing its expiration date. This policy, combined with the token renewal URL policy, adds a link within the expiration notification to a URL where the user can request a replacement token. For more information, see the appendix on customizing the software token app in the *RSA SecurID Software Token for Windows and Mac OS X Administrator’s Guide*.

4 Sample Software Token Deployment Scenarios

Enterprise Managed Windows PCs with Automatic CT-KIP

For enterprise-managed Windows PCs running RSA SecurID Software Token for Windows, you can automate the import of one software token to a PC using dynamic seed provisioning (CT-KIP). This requires customizing the RSA SecurID software token app by setting Windows registry policies. After you deploy the customized app and issue tokens using CT-KIP, the first time the user starts the desktop software token app, one token is automatically imported, as long as one of the following conditions is met:

- The user does not already have a software token.
- All of the tokens in the user's token database have expired.

Note: You cannot automatically import a token using CT-KIP in a Mac OS X implementation.

An automatic CT-KIP import requires setting the activation code in advance to a value known by the software token app: the Windows user SID. An automatic CT-KIP import should therefore be used only when the organization can ensure that the activation code cannot be spoofed by another device. Also, the organization must restrict access to the CT-KIP server endpoint to the authorized devices. This is designed to prevent a hacker with knowledge of the Windows user SID from using another CT-KIP client to request a token. If you are unable to ensure this level of security, you should have the user import the token manually. For more information, see the CT-KIP information in the *RSA SecurID Software Token for Windows Administrator's Guide*.

Tasks for Automating a CT-KIP Import

The tasks required to automate CT-KIP delivery to a Windows desktop or laptop include:

1. Customize the RSA SecurID software token app with Windows Group Policy before deploying the app to the user's PC. Customizing requires:
   - Installing an administrative policy template provided by RSA.
   - Creating Group Policy settings on a domain controller.
2. Deploy the customized software token app.
3. Issue the token in RSA Authentication Manager 8.x or RSA Authentication Manager 7.1, using the CT-KIP distribution option.
4. Bind the token to the user SID. (Enter the user SID in the **DeviceSerialNumber** field when configuring the token record.)
5. Select **DeviceSerialNumber** as the activation code option.

Configure Automatic CT-KIP Import

Setting Group Policy for RSA SecurID Software Token for Windows adds registry keys under **HKEY_LOCAL_MACHINE\Software\Policies\RSA\Software Token**. You create Group Policy settings on a domain controller using the Microsoft Management Console (MMC). The groups you want the policies to affect must exist in Active Directory.

RSA SecurID policies are applied on a per computer (per-machine) basis. That is, the policies you set apply to all users of a particular computer rather than to individual users.

The following instructions assume you have added the RSA administrative template to the MMC.

To configure automatic CT-KIP import:

1. Start the Microsoft Management Console.
2. Navigate to the RSA administrative template, **RSASecurIDToken.adm**.
3. In the right pane, double-click one of the following policies, and change the setting to **Enabled**. Do the same for the other policy.
   - Policy name - **Specify a CT-KIP URL to use for downloading software tokens**
   - Policy name - **User SID for CT-KIP activation code**

For full instructions on implementing automatic CT-KIP delivery on Windows PCs, see the appendix on customizing the software token app in the *RSA SecurID Software Token for Windows Administrator's Guide*.

Enterprise-Managed Windows PCs with Token Files

If you use Microsoft Systems Management Server (SMS) or another third-party deployment tool for enterprise-wide deployment of RSA SecurID Software Token for Windows, you can include token files (SDTID files) in your deployment package and import them automatically when the user starts the software token app.

Note: Use this option only if you provision software tokens using RSA Authentication Manager 6.1. If you use RSA Authentication Manager 8.x or RSA Authentication Manager 7.1, you should first consider using dynamic seed provisioning (CT-KIP). The use of one-time activation codes with dynamically provisioned tokens is designed to help prevent some of the security issues associated with delivering file-based tokens.

Tasks for Including Token Files in the Deployment Package

1. When provisioning software tokens in RSA Authentication Manager:
   - Bind each token to a device-specific identifier, either the Windows user SID or the serial number of the device (for example, the unique serial number of the local hard drive on the user's computer). This is to ensure that the tokens are imported only to the intended device.
   - Password protect each token, making sure you communicate the required password to the user separately from the deployment.
2. Prior to deployment, store the token files in a secure location where access is restricted.
3. When the token files are ready for deployment, place them in the same directory as the MSI file. Configure the SMS package so that tokens will be installed to **Desktop** or **My Documents**. This way, tokens will be imported automatically when a user starts the software token app.

   Note: If you password protect each token, the user is prompted to enter the password before the token is imported. The password is not used again.

4. When you create the SMS package, use a specific script so that each user receives a unique token. For example, use a script containing logic such as "ifsystemresource.name=LAPTOP-LAP, copy username.sdtid c:\." Otherwise, the SMS package will copy every token to every target computer.

Android, BlackBerry 10, iOS, and Windows Phone Devices with CT-KIP

Enterprises can use dynamic seed provisioning (CT-KIP) to efficiently and securely deploy software tokens to unmanaged or personally owned mobile devices such as Android, iOS, BlackBerry 10, and Windows Phone devices.

Tasks for CT-KIP Provisioning

1. Instruct users to download and install the software token app, as follows:
   - Google Play (Android devices)
   - Apple App Store (iOS devices)
   - BlackBerry World (BlackBerry 10 devices)
   - Windows Phone Store (Windows Phone devices)
2. Configure the token record in RSA Authentication Manager 8.x or RSA Authentication Manager 7.1, making sure to bind the token to the user's device.
3. Select the CT-KIP distribution option.
4. Ensure that the custom CTF URL for your CT-KIP server can be reached by the mobile devices.
5. Record the activation codes generated by RSA Authentication Manager 8.x or RSA Authentication Manager 7.1.
6. If you deploy large numbers of tokens, you can use the RSA Authentication Manager 8.x or RSA Authentication Manager 7.1 batch job functionality. Also consider using tools available from RSA Professional Services. For more information, see "RSA Authentication Manager Prime Deployment Tools" on page 27.
7. RSA Authentication Manager 8.x is designed to automatically construct the correct custom CT-KIP URL to allow the user to import the CT-KIP token data into the RSA SecurID app. If you use RSA Authentication Manager 7.1, you must construct a custom CT-KIP URL yourself. You can optionally include the activation code in the URL for either RSA Authentication Manager server. The CT-KIP URL is common to all users. The activation code, if included, is unique to each user.

   Important: For security reasons, RSA recommends delivering the activation code using a separate, secure channel. This has the additional advantage of allowing you to send the same custom CT-KIP URL to every user. When you deliver the activation code separately, the user is prompted to enter it to launch the CT-KIP process. If you deploy large numbers of tokens, you can develop scripts to automate the email delivery based on the unique activation codes.

Construct a Custom CT-KIP URL

If you provision custom CT-KIP URLs using RSA Authentication Manager 7.1, you must manually construct the custom CT-KIP URL. This is necessary to allow the device to communicate directly with RSA Authentication Manager 7.1 and initiate the CT-KIP process.

Important: You do not need to manually construct a custom CT-KIP URL if you use RSA Authentication Manager 8.x. The generated custom CT-KIP URL causes the device to launch the RSA SecurID software token app and initiate the CT-KIP process.

Android Custom CT-KIP URL

For the RSA SecurID software token app for Android, the custom CT-KIP URL must be constructed as follows and must be URL encoded.

```
http://127.0.0.1/securid/ctkip?scheme=<http or https>
&url=<service address>&activationCode=<activation code>
```

The following example shows a properly constructed custom CT-KIP URL with an activation code:

```
http://127.0.0.1/securid/ctkip?scheme=https&url=ctk
server123.yourco.com/ctkip/services/CtkipService&activation
Code=00108310
```

The following example shows a properly constructed CT-KIP URL without an activation code:

```
http://127.0.0.1/securid/ctkip?scheme=https&url=ctk
server123.yourco.com/ctkip/services/CtkipService
```

BlackBerry 10 Custom CT-KIP URL

For the RSA SecurID software token app for BlackBerry 10, the custom CT-KIP URL must be constructed as follows and must be URL encoded.

```
com.rsa.securid://ctkip?url=https://customer_ctkip_server_ur
l&activationCode=activation_code
```

The following example shows a properly constructed custom CT-KIP URL with an activation code:

```
com.rsa.securid://ctkip?url=https://ctk-server123.yourco.com
/ctkip/services/CtkipService&activationCode=00108310
```

The following example shows a properly constructed custom CT-KIP URL without an activation code:

```
com.rsa.securid://ctkip?url=https://ctk-server123.yourco.com
/ctkip/services/CtkipService
```

iOS Custom CT-KIP URL

For the RSA SecurID software token app for iOS, the custom CT-KIP URL must be constructed as follows:

```
com.rsa.securid://ctkip?url=https://customer_ctkip_server
_url&activationCode=activation_code
```

The following example shows a properly constructed custom CT-KIP URL with an activation code:

```
com.rsa.securid://ctkip?url=https://ctk-server123.yourco.com
/ctkip/services/CtkipService&activationCode=00108310
```

The following example shows a properly constructed custom CT-KIP URL without an activation code:

```
com.rsa.securid://ctkip?url=https://ctk-server123.yourco.com
/ctkip/services/CtkipService
```

Windows Phone URL String

For the RSA SecurID software token app for Windows Phone, the custom CT-KIP URL must be constructed as follows:

```
com.rsa.securid://ctkip?scheme=<http or
https>&url=<ctkipserver url>&activationCode=<activationCode>
```

The following example shows a properly constructed custom CT-KIP URL with an activation code:

```
com.rsa.securid://ctkip?scheme=https&url=ctk-server123.yourc
o.com/ctkip/services/CtkipService&activationCode=00108310
```

The following example shows a properly constructed custom CT-KIP string without an activation code:

```
com.rsa.securid://ctkip?scheme=https&url=ctk-server123.yourc
o.com/ctkip/services/CtkipService
```

Android, BlackBerry 10, iOS, and Windows Phone Devices with Token Files or CTF

File-based provisioning is recommend for customers whose version of RSA Authentication Manager does not support CT-KIP (for example, RSA Authentication Manager 6.1). They can also use file-based provisioning if their CT-KIP server is not reachable from mobile devices on the Internet.

Tasks for File-Based Provisioning

The tasks required to provision token files to Android, BlackBerry 10, iOS, and Windows Phone devices are:

1. Instruct users to download and install the software token app from:
   - Google Play (Android devices)
   - BlackBerry World (BlackBerry 10 devices)
   - Apple App Store (iOS devices)
   - Windows Phone Store (Windows Phone devices)
2. Configure the token record in RSA Authentication Manager, making sure to:
   - Bind the token to the device ID provided by RSA.
   - Password protect the token with a strong, unique password. Communicate the password to the user separately using a secure channel.
3. Generate the token file using the SDTID distribution option.
4. Deliver the token using one of the methods described in the following sections.

Token Delivery Options

As described in "Delivering Software Tokens" on page 24, you can deliver tokens to users' devices in one of the following ways:

- Send an SDTID file as an attachment to an email message. (Not supported on the RSA SecurID app for Windows Phone.)
- Use the Token Converter to convert an SDTID file to a custom CTF URL and send the custom CTF URL link in an email message.
- Alternatively, use the Token Converter to generate a custom CTF URL embedded in a QR Code, and scan the QR Code into the app.

Note: The use of QR Codes is currently supported only with release 2.0 of the Android and iOS apps. For the Android 2.0 app, you can use Token Converter 3.0 or 3.1. For the iOS 2.0 app, you must use Token Converter 3.1, and iOS devices must be running iOS 7 or later.

Mixed Deployments

If you plan to support a mixed deployment, such as a deployment of the RSA SecurID Software Token for Android and the RSA SecurID Software Token for iOS, refer to the following table to understand your delivery options:

Device Platform	Supported Token Delivery Method
Android	SDTID, CT-KIP, CTF, QR Code
iOS	SDTID, CT-KIP, CTF, QR Code **Note:** QR Code delivery is supported on iOS devices running iOS 7 or later.
BlackBerry 10	SDTID, CT-KIP
Windows Phone	CT-KIP, CTF

For more information on delivery options for Android, BlackBerry 10, iOS, or Windows Phone devices, see the associated software token *Administrator's Guide*.

Android, BlackBerry 10, iOS, and Windows Phone Devices with Self-Service

This section provides an overview of using RSA Self-Service, RSA Credential Manager, or RSA Authentication Deployment Manager to provision software tokens.

Note: For more details on the Self-Service features available in different versions of Authentication Manager, see "Self-Service Provisioning" on page 23.

Self-Service and RSA Authentication Manager Version	Software Token Delivery Support
RSA Self-Service with RSA Authentication Manager 8.x	• Custom CT-KIP URL link delivered in an email. • SDTID file delivered as an email attachment. • Custom CTF URL link delivered in an email.
RSA Credential Manager with RSA Authentication Manager 7.1	• Custom CT-KIP URL link delivered in an email. • SDTID file delivered as an email attachment.
RSA Authentication Deployment Manager with RSA Authentication Manager 6.1	• SDTID file

Provisioning Tasks in RSA Self-Service

set up the RSA Self-Service console in RSA Authentication Manager 8.x by doing the following:

1. Add a software token profile through the **Authentication** > **Software Token Profiles** option in the RSA Security Console.
2. Configure an email server to distribute the software tokens to users, as described in the RSA Authentication Manager Security Console Help.
3. Decide if you want to allow emergency access for users to access emergency tokens in case they lose their mobile device or leave it at home. You set the Emergency Access Tokencode Setting when you configure the Self-Service settings through the RSA Security Console.

Once you complete those tasks, do the following to allow users to access RSA Self-Service:

1. Define the settings for the RSA Self-Service through the **Setup** > **Self-Service Settings** option in the RSA Security Console.
2. Provide information for users to request software tokens.
3. Approve the software token request.

For more information on configuring RSA Self-Service, see the software token *Administrator's Guide* for Android, BlackBerry 10, iOS, or Windows Phone. For information on the general configuration of Self-Service products, see the RSA Authentication Manager documentation.

Provisioning Tasks in RSA Credential Manager

To allow Self-Service software token provisioning of Android, BlackBerry 10, iOS, or Windows Phone devices, the administrator must first configure an email server to distribute the software tokens to users as described in the *RSA Authentication Manager 7.1 Administrator's Guide*.

Once the server is set up, you can do the following to allow users to access Self-Service:

1. Configure RSA Credential Manager through the **Token Provisioning** > **Manage Tokens** option through the RSA Credential Manager home page.
2. Replace the default email template that generates an email with instructions for the user.
3. Approve the software token request.

For more information on configuring RSA Credential Manager to provision software tokens, see the *Administrator's Guide* for the specific software token platform. For information on the general configuration of RSA Credential Manager software, including workflow approval processes, see the RSA Authentication Manager 7.1 documentation.

Provisioning Tasks in RSA Authentication Deployment Manager

For information on configuring RSA Authentication Deployment Manager to provision software tokens, see the *RSA Authentication Deployment Manager (RSA SecurID Web Express 1.3) Planning Guide* available at RSA SecurCare Online. If you have a valid maintenance contract, contact your account manager.

A Additional Provisioning Information

Supported Device Binding Attributes

The following table lists supported device binding attributes, by platform. For more information, see "Protecting Software Tokens During Provisioning" on page 10.

Note: To bind a token to a device-specific identifier, the administrator must have advance knowledge of the device-specific identifier. For details on obtaining the device-specific identifier for a particular platform, see the *Administrator's Guide* for the platform.

Platform	Device Class (GUID) Identifier	Device-Specific Identifier
Android	a01c4380-fc01-4df0-b113-7fb98ec74694	Device ID generated by RSA. App provides a button to email the device ID.
iOS	556f1985-33dd-442c-9155-3a0e994f21b1	Binding ID generated by RSA. (Formerly called device ID.) App provides a button to email the binding ID.
BlackBerry 10	b77a1d06-d505-4200-90d3-1bb397748704	BlackBerry 10 device PIN. App provides a button to email the device ID.
Windows Phone	c483b592-63f0-4f19-b4cb-a6bce8e57159	Windows Phone device ID. App provides a button to email the device ID.
Windows and Mac OS X laptops and desktops	• Windows hard drive GUID: 8f94b226-d362-4204-ac52-3b21fa333b6f • Mac OS X hard drive GUID: d0955a53-569b-4ecc-9cf7-6c2a59d4e775	• Windows user SID. Obtained using a utility such as PsGetSid (part of the Microsoft PS Tools suite). • Device serial number of hard drive plug-in. Displayed in Token Storage Devices screen in installed software token app.
Toolbar in web browser	N/A	Binding ID. Available from **About RSA SecurID Toolbar > Binding ID** field in installed RSA Toolbar.

Platform	Device Class (GUID) Identifier	Device-Specific Identifier
Active X control in a web application (Web SDK)	N/A	N/A

Number of Tokens Supported Per App

This section lists the number of tokens supported by each RSA SecurID software token app. Apps that support multiple tokens import the tokens one at a time.

Note: Software token apps that support more than one token also support assigning token nicknames (user-friendly names) in RSA Authentication Manager or SAE.

Platform	Maximum Number of Supported Tokens
Android	10 per user
iOS	10 per user
BlackBerry 10	10 in workspace, 10 in personal space
Windows Phone	10 per user
Windows and Mac OS X laptops and desktops	20 per user; more with the software token API
Toolbar in Internet Explorer web browser	20 per user
Active X control in a web application (Web SDK)	1 per company

Activation Code Information

The following table lists information by platform about one-time activation codes used with dynamic seed provisioning (CT-KIP).

Note: Activation codes generated by RSA Authentication Manager 8.x and RSA Authentication Manager 7.1 (system-generated activation codes) are numeric. Alphanumeric activation codes are also supported and are case sensitive.

Platform	Maximum Characters Allowed	Device-Specific Identifiers That Can Be Used as Activation Codes
Android	40	Device ID generated by RSA
iOS	40	Binding ID (formerly called device ID) generated by RSA
BlackBerry 10	40	BlackBerry 10 device PIN
Windows Phone	40	Windows Phone device ID
Windows and Mac OS X laptops and desktops	• 25 through the software token app UI • 50 (approx.) if a device-specific identifier is used	• Windows User SID (requires setting custom policy) • Device serial number
Toolbar in Internet Explorer web browser	1000	N/A
Active X control in a web application (Web SDK)	50	RSA recommends using a system-generated activation code.

Token Expiration Information

The following table shows the information displayed on the platforms notifying users their tokens are about to expire or have expired.

Platform	Token Expiration Information
Android	• Expiration date displayed by accessing the Token List screen • Warning displayed on Tokencode screen 30 days before expiration • “Token Expired” screen
iOS	• Expiration date displayed on Tokencode, Passcode, Next Code, and My Tokens screens • Warning displayed on above screens 30 days before expiration • Displays “expired “once the token has expired • “Token Expired” screen
BlackBerry 10	• Expiration date displayed on Tokens screen • Displays token name, serial number, and date of the expiration when a user attempts to use an expired token • “Token Expired” screen
Windows Phone	Expiration date on Information screen
Windows and Mac OS X laptops and desktops	• Expiration warning displayed 30 days before a token expires. • Policy can be set to change the number of days before the notice is displayed. • Expiration date can be accessed from the software token app options.
Toolbar in web browser	• Displays “expired” instead of tokencode when a token has expired. No message prior to expiration. • Expiration date can be accessed from the software token app options.
Active X control in a web application (Web SDK)	N/A