RSA SecurID Software Token 1.1.2
for Android Administrator’s Guide
Contact Information
See the RSA corporate web site for regional Customer Support telephone and fax numbers: www.rsa.com

Trademarks
RSA, the RSA Logo and EMC are either registered trademarks or trademarks of EMC Corporation 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.rsa.com/legal/trademarks_list.pdf.

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

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 EMC software described in this publication requires an applicable software license.

EMC 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.” 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.
Contents

Preface ................................................................................................................................. 7
  About This Guide .................................................................................................................. 7
  Product Documentation ...................................................................................................... 7
  Related Documentation ...................................................................................................... 7
  Support and Service .......................................................................................................... 8
    Before You Call Customer Support ............................................................................. 8

Chapter 1: Planning for Deployment .................................................................................. 9
  About RSA SecurID Token for Android ......................................................................... 9
  System Requirements ......................................................................................................... 9
  Supported Authentication Servers ..................................................................................... 10
    User Self-Service Token Provisioning Solutions .......................................................... 10
  System Clock Settings ..................................................................................................... 11
    Synchronize Date, Time, and Time Zone Settings ....................................................... 11
  Token Configurations ....................................................................................................... 12
  Token Provisioning Mechanisms ....................................................................................... 13
    File-Based Provisioning ............................................................................................... 13
    Dynamic Seed Provisioning ......................................................................................... 13
  Token Security During Provisioning .................................................................................. 14
    Device Binding .............................................................................................................. 14
    Password Protection of File-Based Tokens ................................................................ 15
  Token Security on the Device .......................................................................................... 15
  Token Delivery Options .................................................................................................... 16
  RSA SecurID Authentication ............................................................................................ 17
    Supported Token Types ............................................................................................... 17
    Authentication with a PINPad-Style Software Token .................................................. 18
    Authentication with a Fob-Style Software Token ....................................................... 19
    Authentication with a PINless Software Token .......................................................... 20
  Language Support ............................................................................................................. 21
  Overview of Deployment Tasks ........................................................................................ 21

Chapter 2: Installing the Application ............................................................................... 23
  Product Package Contents ............................................................................................... 23
  Device Access Requirements ............................................................................................ 23
  Install the Application ....................................................................................................... 24
  User Documentation ......................................................................................................... 25
  Uninstall the Application .................................................................................................. 25

Chapter 3: Issuing Software Tokens Using RSA Authentication Manager 7.1 .................. 27
  Preparing to Issue Software Tokens .................................................................................. 27
  Determine the Token Provisioning Mechanism ............................................................... 27
  Review Supported Token Configurations ....................................................................... 27
Chapter 4: Using RSA Credential Manager Self-Service Provisioning

Preparing To Use Self-Service Provisioning ........................................................................ 37
Restrictions .......................................................................................................................... 37
Import the Android 1.x Device Definition File ................................................................. 37
Allow Device Binding to the Device IMEI/MEID (Recommended) ................................. 38
Issuing Dynamically Provisioned Tokens Using RSA Credential Manager ...................... 38
Step 1: Configure RSA Credential Manager Token Management Settings ................. 39
Step 2: Replace the Request Approval Notification E-Mail Template............................... 40
Step 3: Allow Users to Create a Self-Service Account ...................................................... 41
Step 4: Provide Instructions for Requesting a Software Token ........................................ 41
Step 5: Approve Software Token Requests ....................................................................... 43

Chapter 5: Issuing Software Tokens Using RSA Authentication Manager 6.1

Preparing to Issue Software Tokens ................................................................................ 45
Review Supported Token Configurations.......................................................................... 45
Review Device Binding Recommendations ...................................................................... 45
Obtaining the Device IMEI/MEID (Recommended) ........................................................ 46
Protect File-Based Tokens During Provisioning ............................................................... 47
Issuing Software Tokens in RSA Authentication Manager 6.1 ........................................ 47
Configure the Software Token Record ............................................................................. 48
Bind the Token Using Token Extension Data .................................................................... 50
Next Steps ......................................................................................................................... 50

Chapter 6: Delivering Software Tokens

Token Delivery Mechanisms ........................................................................................... 51
Delivering File-Based Tokens .......................................................................................... 51
Step 1: Issue the Software Token as an SDTID File ......................................................... 52
Step 2: Convert the Token File ......................................................................................... 53
Step 3: Compose and Send an E-mail Message ............................................................... 54
Step 4: Instruct the User on How To Import the Token ..................................................... 55
Delivering Dynamically Provisioned Tokens .................................................................... 56
Step 1: Issue a Dynamically Provisioned Token ............................................................... 56
Step 2: Construct a URL Link .......................................................................................... 56
Step 3: Compose and Send an E-mail Message............................................................. 58
Step 4: Instruct the User on How To Import the Token .............................................. 59
Step 5: Verify Successful Execution of the CT-KIP Protocol .................................... 60
Notification of Software Token Expiration ................................................................. 60

Chapter 7: Troubleshooting ......................................................................................... 61
Customer Support Information ...................................................................................... 61
Obtain the Device Firmware Version ........................................................................... 61
Application Installation Problems.................................................................................. 62
Token Import Problems ................................................................................................ 63
Authentication Problems............................................................................................... 65
Error Messages ............................................................................................................... 66

Index ............................................................................................................................. 69
Preface

About This Guide

This guide describes how to deploy RSA SecurID Software Token for Android in an enterprise environment and how to provision software tokens to users’ devices. It is intended for RSA Authentication Manager administrators. It assumes that these personnel have experience using RSA Authentication Manager. Do not make this guide available to the general user population.

Product Documentation

For more information about RSA SecurID Software Token for Android, see the following documentation:

*Release Notes.* Provides information about what is new and changed in this release, as well as workarounds for known issues. The latest version of the Release Notes is available from RSA SecurCare Online: [https://knowledge.rsasecurity.com](https://knowledge.rsasecurity.com).

*Quick Start.* Helps users install the application and import a software token. Also describes how to use a token to access resources protected by RSA SecurID.

*Help.* Describes procedures associated with the application screens. Users can access Help within the application by touching *Menu > Help.*

Related Documentation

*RSA Secured 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, including implementation guides and other information, go to [http://www.rsasecured.com](http://www.rsasecured.com).

*RSA Authentication Manager 7.1 Administrator’s Guide.* Provides information about how to administer users and security policy 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 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 Software Token Converter.** Command line utility for converting software token files (SDTID files) to Compressed Token Format (CTF) to allow the tokens to be delivered to mobile devices that do not support e-mail attachments. To download the Token Converter, go to [http://www.rsa.com/node.aspx?id=1313](http://www.rsa.com/node.aspx?id=1313).

---

**Support and Service**

<table>
<thead>
<tr>
<th>Service</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA SecurCare Online</td>
<td><a href="https://knowledge.rsasecurity.com">https://knowledge.rsasecurity.com</a></td>
</tr>
<tr>
<td>Customer Support Information</td>
<td><a href="http://www.rsa.com/support">www.rsa.com/support</a></td>
</tr>
<tr>
<td>RSA Secured Partner Solutions Directory</td>
<td><a href="http://www.rsasecured.com">www.rsasecured.com</a></td>
</tr>
</tbody>
</table>

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.

**Before You Call Customer Support**

Make sure you have access to the device running RSA SecurID Software Token for Android.

Please have the following information available when you call:

- Your RSA Customer/License ID.
- Product software version number.
- The model of the Android device on which the problem occurs.
- The Android OS version under which the problem occurs.
Planning for Deployment

This chapter introduces RSA SecurID Software Token for Android and describes the system requirements and the tasks that you need to complete before deploying the application and software tokens.

About RSA SecurID Token for Android

RSA SecurID Software Token for Android is authentication software that transforms an Android device into a network authentication device. The software consists of an application and a software-based security token.

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 a one-time password (OTP). Authorized Android users can use OTP values, along with other security information, to verify their identity when they attempt to access resources protected by SecurID, such as Virtual Private Networks (VPNs) and web applications.

System Requirements

RSA SecurID Software Token for Android requires the following hardware and software:

- An Android device running Android 2.1 or later. To view a current list of Android devices supported with the application, go to http://www.rsa.com/android.
- Access to a network protected by RSA Authentication Manager 6.1 or RSA Authentication Manager 7.1 SP3 or later
- 1 MB disk space available for the application.
Supported Authentication Servers

To issue software tokens for use with the application, you need one of the following authentication servers:

- RSA Authentication Manager 7.1 SP 3 or later. You may need to set up a proxy server for RSA Authentication Manager 7.1 if:
  - You are deploying software tokens using CT-KIP (Cryptographic Token Key Generation Protocol). You must configure a proxy server to connect to mobile devices through Secure Sockets Layer (SSL).
  - You want to restrict users from directly accessing RSA Authentication Manager for self-service token provisioning. You can configure a proxy server to accept RSA Self-Service Console requests and proxy to the Self-Service Console.

For more information, go to [http://www.rsa.com/node.aspx?id=2535](http://www.rsa.com/node.aspx?id=2535), and click the “Configuring a Self Service Proxy Server Technical Publication” link.

- RSA SecurID Appliance 3.0
- RSA Authentication Manager 6.1

User Self-Service Token Provisioning Solutions

For issuing tokens to a large number of Android device users, you can take advantage of the following web-based self-service provisioning solutions:

- **RSA Credential Manager.** Credential Manager enhances the capabilities of RSA Authentication Manager 7.1 by providing self-service software token provisioning. A configurable self-service console allows end users to request software tokens to use with specific mobile devices. Credential Manager is included in RSA Authentication Manager 7.1 Enterprise Edition.

  For instructions on configuring Credential Manager to issue software tokens for Android devices, see “Using RSA Credential Manager Self-Service Provisioning” on page 37. For more information, see [http://www.rsa.com/node.aspx?id=1180](http://www.rsa.com/node.aspx?id=1180) and the RSA Authentication Manager 7.1 Administrator’s Guide.

- **RSA Authentication Deployment Manager.** Formerly RSA SecurID Web Express, RSA Authentication Deployment Manager 1.3 offers end users a self-service platform for requesting tokens. Deployment Manager is available for use with RSA Authentication Manager 6.1. If you have a valid maintenance contract, go to [https://www.rsasecurity.com/go/form_ins.asp](https://www.rsasecurity.com/go/form_ins.asp) or contact your Account Manager. If you do not have a maintenance contract, contact your local RSA sales representative or reseller.
System Clock Settings

The SecurID algorithm uses Coordinated Universal Time (UTC) settings to calculate the current one-time password (OTP). Software tokens rely on the host device, in this case the Android 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 need only change the time zone to reflect the correct local time.

Synchronize Date, Time, and Time Zone Settings

To ensure that users are able to use the software token that you issue for the RSA SecurID application, instruct them to verify the clock settings on their devices before you provision software tokens.

Users can set the local time, date, and time zone manually, or automatically sync the date and time with their network.

Change Clock Settings Manually

To verify or change clock settings manually:
1. Touch Menu > Settings > Date & time.
2. Touch and configure the following options, as needed:
   - Set date
   - Select time zone
   - Set time

Automatically Sync Clock Settings

To automatically sync clock settings with your network:
1. Touch Menu > Settings > Date & time.
2. Touch Automatic.
   - A green check mark displayed next to the Automatic option signifies that the device is using network-provided values.
Token Configurations

The RSA SecurID application supports provisioning one software token per device.

The token configurations available with the RSA SecurID application depend on which version of Authentication Manager you use. In the following table, the configurations with a blue check mark are supported by the specified server. The configurations with a red X are not supported.

**Note:** The RSA SecurID application supports only 128-bit (AES) tokens. The application does not support 64-bit (SID) tokens.

<table>
<thead>
<tr>
<th>Token Attributes</th>
<th>RSA Authentication Manager 7.1</th>
<th>RSA Authentication Manager 6.1</th>
<th>RSA Credential Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>128-bit tokens</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Time-based</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>8-digit tokencode</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>6-digit tokencode</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>60-second tokencode duration</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>30-second tokencode duration</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>PINPad style tokens (PIN entry in the device)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Fob-style tokens (PIN entry in the protected resource)</td>
<td>✓</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Tokens that do not require a PIN (user authenticates with user name and tokencode)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Token file password (for importing file-based tokens to a device)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Options for binding the token to the Android device type GUID or to a specific Android device</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Token Provisioning Mechanisms

RSA SecurID supports the following token provisioning mechanisms:

- File-based provisioning, available in RSA Authentication Manager 6.1 and RSA Authentication Manager 7.1, including RSA SecurID Appliance 3.0
- Dynamic Seed Provisioning, available in RSA Authentication Manager 7.1, including RSA SecurID Appliance 3.0 and RSA Credential Manager 1.0

File-Based Provisioning

With file-based provisioning, an XML-format software token file (also called an SDTID file) is generated by Authentication Manager when you issue a software token to a user. This file contains the shared secret (“seed”) used by the SecurID algorithm, along with other metadata associated with the token (expiration date, serial number, number of digits in the tokencode, and so on).

To protect the seed against attack, the seed is encrypted using the AES encryption algorithm and an optional password. The seed is a string of random binary data, which when decrypted gives no hint that it is a valid secret. To gain access to the seed, an attacker would have to test each attempted decryption against the target Authentication Manager, which would cause the server to rapidly disable the software token that is under attack.

To deliver file-based tokens to Android devices, you must convert the data contained in the SDTID file into Compressed Token Format and send the converted data to the device as a link within an e-mail message. For more information, see “Delivering File-Based Tokens” on page 51.

Dynamic Seed Provisioning

Dynamic Seed Provisioning is a client-server protocol that enables secure, rapid setup of software tokens. A feature of RSA Authentication Manager 7.1, Dynamic Seed Provisioning uses the industry-standard Cryptographic Token Key Initialization Protocol (CT-KIP).

Dynamic Seed Provisioning eliminates the need for a token distribution file. Instead, the software token application running on the device (the client) and Authentication Manager (the server) use a four-pass CT-KIP protocol to exchange information that is used to dynamically generate a unique seed (shared secret) on the client and the server. In this way, the CT-KIP protocol protects against potential interception of the seed.

When you issue a dynamically provisioned token in Authentication Manager, a unique one-time token activation code is generated. To allow a user to import the token, you must supply the activation code and the URL of the CT-KIP server. You can automate the delivery of the activation code or require the user to enter the activation code manually to complete the token import. For more information, see “Delivering Dynamically Provisioned Tokens” on page 56.
To strengthen token security when provisioning tokens in Authentication Manager, RSA strongly recommends that you use:

- **Device binding.** Associates a token with a specific user’s Android device or with the Android device type. The administrator configures device binding when issuing a token in RSA Authentication Manager. For more information, see the following section, “Device Binding.”

- **Password protection of file-based tokens (SDTID files).** The administrator creates a token file password when issuing a token in Authentication Manager. For more information, see “Password Protection of File-Based Tokens” on page 15.

  **Note:** Password protection is not used with tokens issued using Dynamic Seed Provisioning.

### Device Binding

When configuring a software token record in Authentication Manager, you can bind the token by configuring a token extension attribute (DeviceSerialNumber). Binding the token ensures that it is installed only on a specific device or class of devices. RSA strongly recommends binding file-based software tokens.

With Dynamic Seed Provisioning, use of the one-time activation code helps alleviate some of the security issues associated with file-based tokens. For more security assurance, however, it is good practice to bind dynamically provisioned tokens.

You can bind software tokens intended for Android devices to one of the following:

- The “Android 1.x” device type GUID (globally unique identifier): a01c4380-fc01-4df0-b113-7fb98ec74694
  
  Binding a token to the GUID allows the user to import the token to any Android device that is supported by the RSA SecurID application. It prevents the token from being used on other types of mobile devices or computers running an RSA SecurID software token application.

- The user’s device IMEI (International Mobile Equipment Identity) or MEID (Mobile Equipment Identity).
  
  The IMEI/MEID is a unique number that identifies a specific device. It is used by the mobile network to identify valid devices. Binding a token to an IMEI/MEID prevents the user from importing the token to a device other than the device that is associated with the specific IMEI or MEID. The device IMEI/MEID is listed in the About screen of the installed RSA SecurID application. If the application has not been installed, you can obtain the IMEI/MEID from the device, as described in “Obtaining the Device IMEI/MEID (Recommended)” on page 28.
Determine Your Device Binding Mechanism

Use the information in the following table to help you decide which binding mechanism best suits your requirements.

<table>
<thead>
<tr>
<th>Binding Mechanism</th>
<th>Comments</th>
</tr>
</thead>
</table>
| Android 1.x device type GUID | • Provides less protection for file-based tokens than binding to a device IMEI/MEID because the token file can be installed on Android devices other than the user’s device.  
• Prevents installing the token on a computer or on a non-Android mobile device.  
• Easy to implement because the Authentication Manager administrator can bind all tokens to the same device type.  
• In RSA Authentication Manager 7.1, the device type GUID is the default binding entry, so the administrator does not need to configure a token extension attribute. In RSA Authentication Manager 6.1, the administrator must configure the extension attribute. |
| Device IMEI or MEID | • Provides a high level of protection because the token can be installed only on the device with the specified IMEI/MEID.  
• Requires the administrator to obtain the IMEI/MEID from the user before configuring the token record.  
• Simplifies importing a dynamically provisioned token because the IMEI/MEID can be used automatically as the activation code. |

Password Protection of File-Based Tokens

File-based software tokens can be protected during transit with a unique password that is sent to the user separately from the token, for example, using secure e-mail. The user must enter the password in the RSA SecurID application on the device to complete the token import.

The use of unique token passwords is designed to help protect against malicious users gaining access to the SDTID file and attempting to import the token on a different device. However, if the software token does not use device binding, the password mechanism does not prevent a user who has access to both the SDTID file and the password from installing the token on multiple devices. For this reason, RSA recommends using both device binding and password protection with file-based tokens.

Token Security on the Device

After a token has been imported to a device, it is stored in a token database and protected with a set of system attributes. When the application 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 tokencode or the application appears as not having a token. If the user obtains a new device, the software token must be reissued.
# Token Delivery Options

RSA provides several options for delivering tokens to Android devices. Use the information in the following table to help you decide which option best suits your requirements.

<table>
<thead>
<tr>
<th>Delivery Mechanism</th>
<th>Administrative Tasks</th>
<th>End User Actions</th>
<th>Notes and Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail with URL link containing converted token data</td>
<td>1. Issue a file-based token using standard Authentication Manager processes.</td>
<td>1. Open the e-mail message in the device e-mail, and touch the URL link or enter the token data in the Import Token screen.</td>
<td>• Used for delivering file-based tokens</td>
</tr>
<tr>
<td></td>
<td>2. Convert the token to Compressed Token Format.</td>
<td>2. Enter the token file password, if required.</td>
<td>• Required for tokens issued using RSA Authentication Manager 6.1</td>
</tr>
<tr>
<td></td>
<td>3. Send an e-mail message that contains a URL link with the converted token to the user’s device.</td>
<td></td>
<td>• “Delivering File-Based Tokens” on page 51</td>
</tr>
<tr>
<td></td>
<td>4. If the token is password protected, securely communicate the token password.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-mail with URL link containing CT-KIP data</td>
<td>1. Issue a dynamically provisioned token using standard Authentication Manager processes.</td>
<td>1. Open the e-mail message in the device e-mail, and touch the URL link or enter the token data in the Import Token screen.</td>
<td>• Used for delivering dynamically provisioned tokens</td>
</tr>
<tr>
<td></td>
<td>2. Send an e-mail message that contains a specially constructed URL link to the user’s device.</td>
<td>2. Enter the token activation code, if required.</td>
<td>• Requires RSA Authentication Manager 7.1</td>
</tr>
<tr>
<td></td>
<td>3. If the user needs to enter the activation code manually, securely communicate the activation code.</td>
<td></td>
<td>• “Delivering Dynamically Provisioned Tokens” on page 56</td>
</tr>
</tbody>
</table>
**RSA SecurID Authentication**

RSA SecurID uses two-factor authentication, a protocol in which two different factors are used in conjunction to verify a user’s identity. Two-factor authentication typically is a sign-on process where a user proves his or her identity with “something you know” and “something you have.” Using two factors delivers a higher level of authentication assurance than using one factor (for example, a password).

With RSA SecurID, something you have is a software token, and something you know is a PIN. The combination of the PIN and the current tokencode generated by the software token forms a passcode. The passcode is the one-time password (OTP) used for authentication. For every authentication, the user obtains a new OTP, which must be entered along with other security information, such as a user name.

The specific SecurID authentication procedure depends on the type of software token that you issue: PINPad-style, fob-style, or PINless.

**Supported Token Types**

**PIN-Enabled Tokens.** RSA Authentication Manager 7.1, including RSA Credential Manager, and RSA SecurID Appliance 3.0 support two types of PIN-enabled software tokens: PINPad-style and fob-style. Each type offers strong two-factor authentication assurance. The RSA SecurID application recognizes the token type that is installed on the Android device and displays customized screens accordingly. RSA Authentication Manager 6.1 supports PINPad-style software tokens, but does not support fob-style software tokens.

**PINless Tokens.** All supported versions of RSA Authentication Manager support issuing software tokens that do not require entering a PIN. If you issue PINless tokens, the user authenticates with the currently displayed tokencode (something you have). RSA strongly recommends that you use PINless tokens only when a system other than RSA SecurID is responsible for managing the second factor (something you know), such as an existing user name and password. In this scenario, the first factor (user name/password) is validated by the external system and the second factor (tokencode) is validated by Authentication Manager.
Authentication with a PINPad-Style Software Token

**Note:** RSA Authentication Manager 7.1, RSA SecurID Appliance 3.0, and RSA Authentication Manager 6.1 support issuance of PINPad-style software tokens.

With PINPad-style software tokens, the user enters his or her SecurID PIN on the Enter PIN screen on the device. This action generates the OTP (passcode). To complete the authentication, the user enters the OTP, along with other security information, into the protected resource. For example, with a VPN client application, the user typically enters a user name and the OTP.

This authentication experience is similar to using an RSA SecurID PINPad-style hardware token, such as the SD520, where the user enters the PIN on the token’s numeric keypad, and then enters the displayed OTP in the protected resource.

PINPad-style software tokens require a 4–8 digit numeric PIN. The PIN cannot begin with a zero (0).

The following figure shows user authentication to a VPN client application with a PINPad-style software token.
**Authentication with a Fob-Style Software Token**

**Note:** RSA Authentication Manager 7.1 and RSA SecurID Appliance 3.0 support issuance of fob-style software tokens. RSA Authentication Manager 6.1 and RSA Credential Manager 1.0 do not support issuance of fob-style software tokens.

Fob-style software tokens do not require you to enter the SecurID PIN on the device. Instead of displaying an Enter PIN screen, the device displays tokencodes. To authenticate to a VPN client application, for example, the user enters security information (typically, a user name) in the VPN client. The user then enters the PIN, followed by the current tokencode displayed on the device. The combination of the PIN and tokencode forms the OTP (passcode).

Fob-style software tokens used with the SecurID application require a 4–8 digit numeric PIN or a 4–8 character alphanumeric PIN. To require alphanumeric PINs, the administrator must configure the token policy in the RSA Security Console (Authentication > Policies > Token Policies).

The following figure shows user authentication to a VPN client application with a fob-style software token.
Authentication with a PINless Software Token

**Note:** RSA Authentication Manager 7.1, RSA SecurID Appliance 3.0, and RSA Authentication Manager 6.1 support issuance of PINless software tokens.

If you issue tokens that do not require a PIN, the user authenticates with the tokencode (OTP). For example, to authenticate to a VPN client application, the user obtains the current OTP from the Tokencode screen on the device. The user enters the OTP and other security information, such as a user name, in the protected resource.

The following figure shows user authentication to a VPN client application with a PINless software token.
Language Support

The RSA SecurID application is available in the following languages:

- English
- Chinese
- French
- German
- Italian
- Japanese
- Korean
- Portuguese
- Russian
- Spanish

The application is displayed in the language that the device is set to use. The default is English.

Overview of Deployment Tasks

Use the information in the following table to become familiar with the tasks required to deploy RSA SecurID Software Token for Android.

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Install the application on Android devices.</td>
<td>“Installing the Application” on page 23</td>
</tr>
<tr>
<td><strong>Important:</strong> The application must be installed before the user can import a token.</td>
<td></td>
</tr>
<tr>
<td>2. Issue software tokens using one of the following authentication servers:</td>
<td></td>
</tr>
<tr>
<td>• RSA Authentication Manager 7.1 SP3 or RSA SecurID Appliance 3.0</td>
<td>“Issuing Software Tokens Using Dynamic Seed Provisioning” on page 31</td>
</tr>
<tr>
<td>• RSA Credential Manager 1.0 (self-service provisioning)</td>
<td>“Using RSA Credential Manager Self-Service Provisioning” on page 37</td>
</tr>
<tr>
<td>• RSA Authentication Manager 6.1</td>
<td>“Issuing Software Tokens in RSA Authentication Manager 6.1” on page 47</td>
</tr>
<tr>
<td>3. Deliver software tokens to Android devices.</td>
<td>Chapter 6, “Delivering Software Tokens”</td>
</tr>
</tbody>
</table>
2: Installing the Application

This chapter describes options for installing and removing RSA SecurID Software Token for Android.

Product Package Contents

The RSA SecurID Software Token for Android product package, android111.zip, contains the following items:

- **util\Android-1.x-swt.xml**. A device definition file required for issuing tokens using RSA Authentication Manager 7.1 or later. For more information, see “Import the Device Definition File” on page 28.
- **doc\credmgr_email_template.txt**. Replacement e-mail template for RSA Credential Manager. For more information, see “Step 2: Replace the Request Approval Notification E-Mail Template” on page 40.
- **doc\SecurIDTokenAndroid111_admin.pdf**. This Administrator’s Guide.
- **doc\English\SecurIDTokenAndroid111_quickstart_en.pdf**. A Quick Start for end users. For more information, see “User Documentation” on page 25.

**Note:** Translations of the Quick Start are located in the doc folder under the language name, for example, doc\Spanish.

Device Access Requirements

The application requires access to the following on your Android device:

- **Network communication**. Full Internet access is required to provision a token to the device and allow the user to authenticate to resources that can be accessed from the device.
- **Phone calls**. Phone access is required to allow the software token API to obtain the phone’s identity. The application does not track phone use or contacts.
Install the Application

The RSA SecurID application can be downloaded for free. Tokens must be purchased separately.

Users must install the application from Android Market. Android Market is a centralized repository of applications developed for Android devices. RSA does not support other methods of installing the application.

To install the application from Android Market:

1. In your list of apps, touch the Market icon.
2. Do one of the following:
   • Touch Apps, and touch the Communication category. Scroll to RSA SecurID Software Token.
   • Touch Search, enter the keywords RSA SecurID, and touch the Search icon.
3. Touch RSA SecurID Software Token.
4. Touch Install.
   The device displays a list of functions to which the application will require access in order to run without encountering security exceptions.
5. Touch OK to start the download.
   The status bar displays the progress of the download. When the download is complete, a notification is displayed.

Next Steps:

After downloading the application, users must import a software token.

For information about issuing tokens in RSA Authentication Manager, see one of the following:

• “Issuing Software Tokens Using RSA Authentication Manager 7.1” on page 27
• “Issuing Software Tokens Using RSA Authentication Manager 6.1” on page 45

For information about methods for delivering tokens to user’s devices, see “Delivering Software Tokens” on page 51.
User Documentation

Deploying the RSA SecurID application affects the way that users access secure resources in the enterprise. RSA provides a *Quick Start* document in PDF format to help users install and use the application. The *Quick Start* contains instructions for:

- Downloading and installing the application
- Installing a software token
- Setting a PIN (if required) during the user’s first RSA SecurID authentication
- Using the application to log on to resources protected by RSA SecurID

The *Quick Start* is located in the `doc` folder of the `android111.zip` file. It is provided in multiple languages identified by subfolders. RSA recommends that you e-mail the *Quick Start* to users.

After launching the application, users can open a Help file from the application. For example, they can access the Help to learn how to use their software token to authenticate to their VPN client application.

Uninstall the Application

Uninstalling the application removes both the RSA SecurID Software Token for Android application and the token database.

**To uninstall the application:**

1. Touch *Menu > Settings*.
2. Touch *Applications > Manage applications*.
   A list of all applications that you have installed is displayed.
3. Locate the RSA SecurID application, and touch it to display the detailed screen about the application.
4. To remove the application from the device, touch *Uninstall*.
5. At the “Uninstall application?” prompt, touch *OK*.
   The application and the token database are removed.
3

Issuing Software Tokens Using RSA Authentication Manager 7.1

Provisioning software tokens consists of issuing tokens in RSA Authentication Manager and delivering them to Android devices. This chapter describes how to issue software tokens for the RSA SecurID application using RSA Authentication Manager 7.1 and RSA SecurID Appliance 3.0. To issue and deliver software tokens using RSA Credential Manager, see Chapter 4, “Using RSA Credential Manager Self-Service Provisioning.”

Chapter 6, “Delivering Software Tokens,” describes the methods available for delivering tokens issued using the full Authentication Manager to users’ devices.

For convenience, references to RSA Authentication Manager 7.1 also apply to the Appliance.

Preparing to Issue Software Tokens

Before you issue software tokens for the RSA SecurID application using RSA Authentication Manager 7.1 or Credential Manager, you must do the following (in any order):

• Determine which token provisioning mechanism you want to use
• Review the token configurations that are supported in RSA Authentication Manager 7.1
• Import the Android device definition file into Authentication Manager
• Review device binding recommendations (recommended)
• Obtain the user’s device binding information (recommended)

Determine the Token Provisioning Mechanism

RSA Authentication Manager 7.1 supports Dynamic Seed Provisioning and file-based provisioning.

To determine the token provisioning mechanism:

Review the information in “Token Provisioning Mechanisms” on page 13, and determine which mechanism is appropriate for your enterprise.

Review Supported Token Configurations

RSA Authentication Manager 7.1 supports standard token configurations, as well as newer configurations available only in version 7.1 and later.

To review the supported token configurations:

See “Token Configurations” on page 12.
Import the Device Definition File

Software tokens issued in RSA Authentication Manager 7.1 (or Credential Manager) must be associated with a device definition file. This is an XML file that defines the capabilities and attributes of tokens that can be used with the application. The file identifies the supported tokencode characteristics, the token type (PINPad, fob-style, or PINless), whether the token is CT-KIP capable, and the supported binding attributes.

RSA provides the “Android 1.x” device definition file for the Android application. Before you issue software tokens to use with the application, you must import this file into RSA Authentication Manager 7.1. Importing the device definition file allows you to configure the characteristics of the tokens and bind tokens.

To import the device definition file:
1. In the product kit /util folder, select Android-1.x-swtd.xml, and save it to a folder on the computer running the RSA Security Console.
2. Start the RSA Security Console.
3. Click Authentication > Software Token Device Types > Import Token Device Type.
4. Click Browse to locate the Android 1.x device definition file.
5. Select the file, and click Submit.

Review Device Binding Recommendations

RSA strongly recommends using device binding for file-based tokens (SDTID files) to help ensure the security of token files in transit. For added security, you can set a token file password that the user must enter to import the token. For more information, see “Issuing Software Tokens Using File-Based Provisioning” on page 35.

With Dynamic Seed Provisioning, use of the one-time activation code helps alleviate some of the security concerns associated with the delivery of file-based tokens. However, for more security assurance, it is good practice to bind dynamically provisioned tokens.

To review the device binding recommendations:
See “Determine Your Device Binding Mechanism” on page 15.

Obtaining the Device IMEI/MEID (Recommended)

Each mobile device is associated with an IMEI (International Mobile Equipment Identity) or MEID (Mobile Equipment Identity). This is a unique number that identifies a specific device. It is used by the mobile network to identify valid devices. Because the IMEI/MEID is unique to the device, using it to bind a token ensures that only the device with that number can import a token. Obtain the device IMEI/MEID from users before configuring token records.

Users can access the device IMEI/MEID in the following ways:
• From the device, if the user has not installed the RSA SecurID application
• From the installed application
Obtain the Device IMEI/MEID from the Device

Instruct users to use either of the following methods to obtain the IMEI/MEID from the device if the user has not installed the RSA SecurID application.

To display the IMEI/MEID on the device:

Do one of the following:

- Touch Menu > Settings > About Phone > Status, and scroll to the IMEI or MEID field. The name of the field depends on your device.
- On the device’s phone keypad, enter *#06#.

**Note:** Android devices from certain vendors do not support obtaining the IMEI/MEID from the phone keypad. For such devices, users must obtain the information from the installed application, as described in “Obtain the Device IMEI/MEID from the Installed Application.”

MEID on Samsung Galaxy Tab Device

The Samsung Galaxy Tab Status screen displays the MEID as both a decimal (Dec) and a hexadecimal (Hex) value, as shown in the following figure.

```
<table>
<thead>
<tr>
<th>MEID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dec: 268435459611792804</td>
</tr>
<tr>
<td>Hex: 0xA0000024B3F1A4</td>
</tr>
</tbody>
</table>
```

The MEID required for binding the Galaxy Tab is the Hex value without the leading “0x” characters. For example, the MEID in this example is A0000024B3F1A4.

Obtain the Device IMEI/MEID from the Installed Application

Instruct users to obtain the IMEI/MEID from the installed application as follows.

To locate the IMEI/MEID in the installed application:

1. Start the RSA SecurID application.
2. Touch Menu > About.
   
   The About screen lists the device IMEI or MEID.
**Next Steps**

After you have imported the device definition file and obtained device IMEI or MEID (if required), you are ready to issue tokens. Use the following table to locate the information that you need.

<table>
<thead>
<tr>
<th>Provisioning Mechanism</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic Seed Provisioning (CT-KIP)</td>
<td>“Issuing Software Tokens Using Dynamic Seed Provisioning” on page 31</td>
</tr>
<tr>
<td>File-based provisioning</td>
<td>“Issuing Software Tokens Using File-Based Provisioning” on page 35</td>
</tr>
<tr>
<td>Self-service provisioning (CT-KIP)</td>
<td>“Issuing Dynamically Provisioned Tokens Using RSA Credential Manager” on page 38</td>
</tr>
</tbody>
</table>
Issuing Software Tokens Using Dynamic Seed Provisioning

With Dynamic Seed Provisioning (CT-KIP), you configure token records in RSA Authentication Manager 7.1 and select CT-KIP as the token distribution method. To allow users to import their token, you must provide them with two pieces of information:

- The URL of the CT-KIP server that is used to exchange information with the client to generate the token.
- A one-time token activation code of up to 30 characters.

Before You Begin

Before you can issue tokens in RSA Authentication Manager 7.1 using Dynamic Seed Provisioning, you must do the following in the RSA Security Console:

- Import tokens (Authentication > SecurID Tokens > Import Tokens Job > Add New)
- Add users (Identity > Users > Add New)
- Assign software tokens to users (Identity > Users > Manage Existing)

Note: Procedures for the tasks listed above are provided in the RSA Security Console Help.

After you assign software tokens to users, you must do the following:

- Determine the activation code option
- Configure the software token record
- Deliver the token to the user’s device, as described in Chapter 6, “Delivering Software Tokens”

Determining the Activation Code Option

When issuing a dynamically provisioned token, you must select an option to use as the token activation code, as described in “Step 7: Select an Activation Code Option” on page 35.

- You can use the IMEI/MEID as the activation code. To do so, you must first bind the token to the IMEI/MEID. For instructions, see “Step 4: (Recommended) Bind the Token” on page 33.
- You can use a system-generated activation code. You must use a system-generated activation code if you do not use device binding or if you use the default Android device type GUID as the binding attribute. For more information “Step 7: Select an Activation Code Option” on page 35.
Configure the Software Token Record

The following sections describe how to configure token records for the RSA SecurID application.

**Step 1: Access the Edit Token page**

1. Log on to the Security Console.
2. Click Authentication > SecurID Tokens > Manage Existing.
3. Locate the token that you want to edit, click the drop-down arrow next to its serial number, and select Edit.

**Step 2: Select the User Authentication Requirement**

In the SecurID PIN Management section, do one of the following:

- To require the user to authenticate with a passcode (PIN plus tokencode), select Require PIN during authentication.
- To require the user to authenticate with a tokencode only (no PIN), select Do not require PIN (only tokencode).

**Step 3: Select the Software Token Device Type**

From the Software Token Device Type drop-down list, select Android 1.x.

Selecting the device type displays the Device Specific Attributes section. By default, the DeviceSerialNumber attribute extension field is populated with the Android 1.x device type GUID, as shown in the following figure.
Step 4: (Recommended) Bind the Token

In the **Software Token Device Type** section, do one of the following:

- To bind the token to the Android 1.x device type, in the **DeviceSerialNumber** field, keep the default value shown in Step 3 on page 32.
- To bind the token to a device IMEI or MEID, clear the **DeviceSerialNumber** field, and enter the device IMEI/MEID string. The following figure shows binding to a device MEID.

![Software Token Device Type](image)

**Note:** The **Nickname** field is provided for applications that support assigning the token a user-friendly nickname. RSA SecurID Software Token for Android does not currently support token nicknames.

Step 5: Select the Distribution Method

In the **Basics** section, select **Generate CT-KIP Credentials for Web Download**.

![Basics](image)
Step 6: Select Software Token Settings

In the Software Token Settings section, select the settings for the token you are issuing. For example, if you want the user to authenticate with a PINPad-style software token (PIN plus tokencode), in the Displayed Value field, select Passcode. The following figure shows the settings supported by the Android application, and the table explains each setting.

<table>
<thead>
<tr>
<th>Software Token Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>The options enabled and the default choices are based on the selected device type.</td>
</tr>
</tbody>
</table>
| **Displayed Value:** | ○ Passcode (PIN incorporated into tokencode)  
○ Tokencode (PIN entered followed by tokencode during authentication) |
| **Tokencode Length:** | ○ 6 Digits  
○ 8 Digits |
| **Tokencode Type:** | ○ Time Based  
○ Event Based |
| **Tokencode Duration:** | ○ Display next tokencode every 30 seconds  
○ Display next tokencode every 60 seconds |

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Displayed Value**     | Displayed Value options are available if you selected Require PIN during authentication as the user authentication requirement.  
• Select Passcode (PIN incorporated into tokencode) to issue a PINPad-style software token.  
• Select Tokencode (PIN entered followed by tokencode during authentication) to issue a fob-style software token.  
If you selected Do not require PIN (only tokencode) as the user authentication requirement, the default displayed value is always set to “Tokencode.” The displayed value options do not affect the behavior of tokens that do not require a PIN. |
| **Tokencode Length**    | Select either 6 Digits or 8 Digits.                                           |
| **Tokencode Type**      | Time Based is automatically selected, indicating that the tokencode changes at a regular interval. The application does not support event-based tokens. |
| **Tokencode Duration**  | Select either Display next tokencode every 30 seconds or Display next tokencode every 60 seconds. |
Step 7: Select an Activation Code Option

1. In the Options section, from the CT-KIP Activation Code drop-down menu, select the option that you want to use for the activation code.
   - If you bound the token to the user’s device IMEI/MEID, and you want to use that binding option as the activation code, select the **DeviceSerialNumber** option, as shown in the following figure.
   - If you did not bind the token to the device IMEI/MEID, select **System Generated Code** to have Authentication Manager generate an activation code.

   ![Options menu](image)

2. Click **Next** to complete the issuance process.

Next Steps

For information on delivering tokens issued using Dynamic Seed Provisioning, see “**Delivering Dynamically Provisioned Tokens**” on page 56.

Issuing Software Tokens Using File-Based Provisioning

With file-based provisioning, you configure a token record and select **Issue Token File (SDTID)** as the token distribution method. This generates a software token distribution file (SDTID file) containing token data.

To issue SDTID files in RSA Authentication Manager 7.1, you must do the following:

- Import tokens (**Authentication > SecurID Tokens > Import Tokens Job > Add New**)
- Add users (**Identity > Users > Add New**)
- Assign software tokens to users (**Identity > Users > Manage Existing**)

**Note:** Procedures for the tasks listed above are provided in the RSA Security Console Help.

After you assign software tokens to users, you must do the following:

- Configure the software token record and, optionally, set a token file password
- Deliver the token to the user’s device, as described in Chapter 6, **“Delivering Software Tokens”**
Configure the Software Token Record and Set a Token File Password

You configure the software token record for a file-based token in the same way as for a dynamically provisioned token, except that you issue a token file. RSA recommends that you set a unique token file password to protect the token file in transit and securely communicate the password to the user.

The token file password can contain a maximum of 24 characters. It is case sensitive and can contain numbers, letters, spaces, punctuation, and symbols. The application prompts the user for the password during the token import. After the token has been imported, the password is not used again.

To configure the token record and set a token file password:

1. Follow steps 1–4 in “Configure the Software Token Record” on page 32.
2. In the Basics section, select Issue Token File (SDTID) as the distribution method.
   Once you select this distribution method, the page displays a Token File Options section.
3. In the Token File Options section, from the Password Protect options, select Password or another option for protecting the token distribution file. For information on password options, click Help on this page at the top of the screen.

   ![Token File Options Table]

   Note: The Enable copy protection field is automatically selected.

4. Enter and confirm the token file password, and click Next to display the results.

Next Steps

For information on delivering file-based tokens to users’ devices, see “Delivering File-Based Tokens” on page 51.
Using RSA Credential Manager Self-Service Provisioning

You can use RSA Credential Manager for Dynamic Seed Provisioning. RSA Credential Manager is the self-service and provisioning component of RSA Authentication Manager 7.1 Enterprise Edition. Users access the Self-Service Console to request a new software token or a replacement token.

Preparing To Use Self-Service Provisioning

You can allow users to request and import a dynamically provisioned token to their Android device using RSA Credential Manager self-service provisioning.

Before you allow users to set up a self-service account, instruct them to install the RSA SecurID application on their devices. The application is available from the Android Market.

To prepare to use Credential Manager, you must do the following:

- Import the Android 1.x device definition file.
- If you require device binding to the device IMEI/MEID, instruct users to have the information available.

Restrictions

RSA SecurID Software Token for Android does not support self-service provisioning of file-based tokens (SDTID files). For file-based provisioning, you must use the RSA Security Console. For more information, see “Issuing Software Tokens Using File-Based Provisioning” on page 35.

Credential Manager supports the default software token configuration: PINPad-style, 8-digit tokencode, and 60-second tokencode duration. If you want to issue tokens with alternative configurations (fob-style tokens, 6-digit tokencode, or 30-second tokencode duration), you must use the RSA Security Console.

Import the Android 1.x Device Definition File

Software tokens provisioned using RSA Credential Manager must be associated with a device definition file. This is an XML file that defines the capabilities and attributes of tokens that can be used with the RSA SecurID Software Token for Android application. You must import this file into Authentication Manager to allow self-service software token provisioning to Android devices.

To import the device definition file:

See “Import the Device Definition File” on page 28.
Allow Device Binding to the Device IMEI/MEID (Recommended)

RSA strongly recommends that you configure Credential Manager to allow users to enter a token binding attribute when requesting a software token for their Android device. If you want users to bind their token to a specific device, instruct them to enter their device IMEI/MEID data as the binding attribute.

To require device binding to the IMEI/MEID:

1. Configure Credential Manager to allow users to provide token attribute details, as described in “Step 1: Configure RSA Credential Manager Token Management Settings” on page 39.

2. Before you have users set up a self-service account, instruct them to have the device IMEI/MEID available. For information on obtaining the IMEI/MEID, see “Obtaining the Device IMEI/MEID (Recommended)” on page 28.

Issuing Dynamically Provisioned Tokens Using RSA Credential Manager

Issuing dynamically provisioned tokens using Credential Manager requires the following tasks, in order:

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Configure Credential Manager token management settings</td>
<td>“Step 1: Configure RSA Credential Manager Token Management Settings” on page 39</td>
</tr>
<tr>
<td>2. Replace the Request Approval Notification template</td>
<td>“Step 2: Replace the Request Approval Notification E-Mail Template” on page 40</td>
</tr>
<tr>
<td>3. Allow users to create a self-service account</td>
<td>“Step 3: Allow Users to Create a Self-Service Account” on page 41</td>
</tr>
<tr>
<td>4. Provide instructions for requesting a software token</td>
<td>“Step 4: Provide Instructions for Requesting a Software Token” on page 41</td>
</tr>
<tr>
<td>5. Approve software token requests</td>
<td>“Step 5: Approve Software Token Requests” on page 43</td>
</tr>
</tbody>
</table>
Step 1: Configure RSA Credential Manager Token Management Settings

The following figure shows sample RSA Credential Manager Console configuration settings for software tokens provisioned for Android devices. Use the following procedure to configure the software token management settings that you want.

To configure Credential Manager settings:

1. From the RSA Security Console, click **Setup > Component Configuration > Credential Manager**.

2. On the Credential Manager Configuration Home page, under **Token Provisioning**, click **Manage Tokens**.

3. On the Manage Tokens page, under **Software Token Types Available for Request**, click **Allow users to request Android 1.x software tokens**. The **Display Name**, **Image location**, and **Description** fields are automatically populated with the application name, device image, and application description that will be displayed to the user in the Self-Service Console. Credential Manager does not support changing the default image.

4. In the **Require User to Authenticate With** field, do one of the following:
   - Click **Passcode (PIN + tokencode)** to require passcode authentication.
   - Click **Tokencode only (PIN-less)** to require tokencode authentication (no PIN entry).
5. In the **Supported Token Distribution Methods** field, click **Generate CT-KIP Credentials for Web Download**.

6. Leave the **Client Application URL** field blank.

7. (Optional) Use the **Device Help Document URL** field to point to a URL where users can download the *Quick Start* document. Otherwise, leave the field blank.

8. (Optional) In the **Make Default** field, click **Make this token type the default option for token requests**.

9. (Optional) If you want the user to bind the token, in the **Attribute Details** field, select **Allow users to edit token attribute details**.

10. At the bottom of the screen, click **Save**.

**Step 2: Replace the Request Approval Notification E-Mail Template**

When you approve a request for a software token, Credential Manager generates an e-mail with instructions for importing the token. The default e-mail template provided by Credential Manager contains some supplementary information that is not needed to import the token.

To ensure an optimal user experience, RSA recommends that you replace the default Request Approval Notification e-mail template with a simplified e-mail template, `credmgr_email_template.txt`. You can find this file in the `doc` folder of the RSA SecurID product kit. The replacement template allows the user to automatically import a dynamically provisioned token by touching a URL link in the e-mail notification. The link contains all of the required token data, including the one-time activation code.

To replace the e-mail notification template:

1. From the RSA Security Console, click **Setup > Component Configuration > Credential Manager**.

2. On the Credential Manager Configuration Home page, under **Token Provisioning**, click **Define e-mail settings**.

3. In the **Email Notification Templates** section, next to **Request Approval Notification SW Tokens**, click **View & manage template details**.

4. In the **Body** field, delete the existing code and replace it with the code in the replacement template, `credmgr_email_template.txt`. For a sample of the e-mail that the user receives, see “Software Token Request Approval E-Mail” on page 43.

5. Click **Save**.
Step 3: Allow Users to Create a Self-Service Account

To allow a user to create a self-service account:

1. Provide a URL link to the Self-Service Console.
2. Instruct the user to request an account.
3. Approve the account request.

Step 4: Provide Instructions for Requesting a Software Token

The following procedure describes the steps a user must take to request a software token for an Android device using the RSA Self-Service Console. Use this information to guide first-time users through the process.

To request a software token using the RSA Self-Service Console:

1. Log on to the Self-Service Console URL.
2. In the My SecurID Tokens section, click Request a Token.
3. From the Request a Token drop-down menu, select I need a specific software token.
   The Token Type section is displayed.
4. Scroll to and select Android 1.x.
5. Under **Provide Your Token Details**, in the **DeviceSerialNumber** field, do one of the following, as instructed by your administrator:

- Leave the default value.
- Delete the default value, and enter your device IMEI/MEID. The following figure shows a sample MEID value.

<table>
<thead>
<tr>
<th>Provide Your Token Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeviceSerialNumber:</strong></td>
</tr>
<tr>
<td><strong>Nickname:</strong></td>
</tr>
</tbody>
</table>

Leave the **Nickname** field blank. This field is provided for applications that support assigning the token a user-friendly nickname. RSA SecurID Software Token for Android does not currently support token nicknames.

6. If the **Create Your PIN** section is displayed, create and confirm a PIN using your administrator’s guidelines.

The PIN cannot begin with a zero (0).

<table>
<thead>
<tr>
<th>Create Your PIN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Create PIN:</strong></td>
</tr>
<tr>
<td><strong>Confirm PIN:</strong></td>
</tr>
</tbody>
</table>

**Important:** Memorize your PIN. If you forget your PIN, you will need to access the Self-Service Console to reset it before you can continue using your token.

7. In the **Reason for Token Request** field, enter the reason that you need a token, for example, “to access the corporate VPN client.”

8. Click **Submit Request**.

   A page opens with a request confirmation number.

9. Print the confirmation page for your records, and then click **OK** to return to the My SecurID Tokens page.

   The **My Requests** section lists your pending request. When your administrator approves the request, you will receive an e-mail notification.
Step 5: Approve Software Token Requests

To complete the provisioning process, you must approve users’ software token requests.

To approve the request:

1. In the RSA Security Console, click **Administration > Provisioning**.
2. Select the token requests that you want to approve.
3. In the **Comment to User** field, enter any comments that you want included in the notification e-mail sent to the user.
   You must enter some text in this field.
4. Click **Approve Requests**.

Software Token Request Approval E-Mail

When you approve the user’s request for a software token, an e-mail approval is sent automatically to the user’s computer and Android device. The following sample e-mail assumes that you replaced the default e-mail template with the **credmgr_email_template.txt** template file provided in the product kit.

As shown in the following figure, the user is instructed to import the token by touching the link in the **Import Your Software Token** field. This link contains all of the data required to import the token, including the activation code.

---

John, your new or additional software token request has been approved with the following comments from your admin:

Admin: Enjoy your new token.

Please ensure that the RSA SecurID application is installed on your device before attempting to import your software token.

How To Import Your Software Token

1. Please view this e-mail on your Android device, and touch the link below.
2. When prompted to **Complete action using**, touch **RSA SecurID**.

Import Your Software Token: [http://127.0.0.1/securid/ckip?scheme=https&amp;url=ctkserver123.yourco.com/ctkipservice](http://127.0.0.1/securid/ckip?scheme=https&amp;url=ctkserver123.yourco.com/ctkipservice)

This request was initiated by: John Smith [jsmith6]
Confirmation #: ba2dc35f4fdd640a01c171a035bca11e
Approval Date: 8/1/11 11:45:29 AM EDT

If you did not initiate this request, please contact your administrator with the information in this e-mail.
Chapter 5: Issuing Software Tokens Using RSA Authentication Manager 6.1

Provisioning software tokens involves issuing tokens in RSA Authentication Manager and delivering them to Android devices. This chapter describes how to issue software tokens for the RSA SecurID application using RSA Authentication Manager 6.1. Chapter 6, “Delivering Software Tokens,” describes the methods available for delivering the tokens to users’ devices.

**Note:** RSA Authentication Manager 6.1 supports file-based token provisioning only. To use Dynamic Seed Provisioning (CT-KIP), you must use RSA Authentication Manager 7.1.

### Preparing to Issue Software Tokens

Before you issue software tokens for the RSA SecurID application using RSA Authentication Manager 6.1, you must do the following (in any order):

- Review the token configurations that are supported in RSA Authentication Manager 6.1
- Review device binding recommendations (recommended)
- Obtain the user’s device binding information (recommended)
- Review recommendations for protecting token files in transit

### Review Supported Token Configurations

RSA Authentication Manager 6.1 supports only standard token configurations. If you want to issue tokens with alternative configurations (fob-style tokens, 6-digit tokencode, 30-second tokencode duration), you must upgrade to RSA Authentication Manager 7.1.

**To review the supported token configurations:**

See “Token Configurations” on page 12.
Review Device Binding Recommendations

RSA strongly recommends using device binding for file-based tokens (SDTID files) to help ensure the security of token files in transit.

To review the device binding recommendations:

See “Determine Your Device Binding Mechanism” on page 15.

Obtaining the Device IMEI/MEID (Recommended)

Each mobile device is associated with an IMEI (International Mobile Equipment Identity) or MEID (Mobile Equipment Identity). This is a unique number that identifies a specific device. It is used by the mobile network to identify valid devices. Because the IMEI/MEID is unique to the device, using it to bind a token ensures that only the device with that number can import a token. Obtain the device IMEI/MEID from users before configuring token records.

Users can access the device IMEI/MEID in the following ways:

- From the device, if the user has not installed the RSA SecurID application
- From the installed application

Obtain the Device IMEI/MEID from the Device

Instruct users to use either of the following procedures to obtain the IMEI/MEID from the device if the user has not installed the RSA SecurID application.

To display the IMEI/MEID on the device screen:

Do either of the following:

- Touch Menu > Settings > About Phone > Status, and scroll to the IMEI or MEID field. The name of the field depends on your device.
- On the device’s phone keypad, enter *#06#.

**Note:** Certain Android devices do not support obtaining the IMEI/MEID from the phone keypad. For such devices, users must obtain the information from the installed application, as described in “Obtain the Device IMEI/MEID from the Installed Application” on page 47.

MEID on Samsung Galaxy Tab Device

The Samsung Galaxy Tab Status screen displays the MEID as both a decimal (Dec) and a hexadecimal (Hex) value, as shown in the following figure.

```
MEID
Dec:268435459611792804
Hex:0xA000024B3F1A4
```

The MEID required for binding the Galaxy Tab is the Hex value without the leading “0x” characters. For example, the MEID in this example is A000024B3F1A4.
Obtain the Device IMEI/MEID from the Installed Application

Instruct users to obtain the IMEI/MEID from the installed application as follows.

To locate the IMEI/MEID in the installed application:
1. Start the application.
2. Touch Menu > About.
   The About screen lists the device IMEI or MEID.

Protect File-Based Tokens During Provisioning

RSA recommends protecting file-based tokens in transit by assigning a unique token file password and transmitting it using a secure mechanism. This can help prevent malicious users from gaining access to the SDTID file. When a token file is imported to the user’s device, the user must enter the token file password to complete the import. After the token has been imported, the password is not used again.

For added security, use device binding in conjunction with password protection. For example, by binding the token to the user’s device IMEI/MEID, you can prevent an intruder who gains access to both the SDTID file and the password from installing the token on multiple computers or devices.

To set a token file password:
See “Configure the Software Token Record” on page 48.

To bind a token to a device attribute:
See “Bind the Token Using Token Extension Data” on page 50.

Issuing Software Tokens in RSA Authentication Manager 6.1

To issue file-based software tokens in RSA Authentication Manager 6.1, you must do the following:

- Add token records to the database
- Add a user and assign a token
- Add an Agent Host
- Activate the user on an Agent Host

Note: Instructions for the tasks listed above are provided in the Database Administration application Help.

After you activate the user, you must do the following:

- Configure the software token record
- Bind the token using token extension data (recommended)
- Deliver the token to the user’s device, as described in Chapter 6, “Delivering Software Tokens”
Configure the Software Token Record

**Note:** Issue only one token per token distribution file. The application imports the first token in a multiple-token file and ignores the remaining tokens.

**To configure a token record:**

1. Open the Database Administration application, and select **Tokens > Issue Software Tokens**.

   ![Issue RSA SecurID Software Tokens](image)

2. Accept the default algorithm (**AES SDTID 3.0**).

3. Under **Options**, leave **Enable Copy Protection** selected, and select **Edit Extension Data**.

   ![Options](image)

4. Accept the default algorithm (**AES SDTID 3.0**).

5. Under **Options**, leave **Enable Copy Protection** selected, and select **Edit Extension Data**.
4. To protect the token distribution file during provisioning, select **Password Protect**, and then enter and confirm a static password. The token file password can contain up to 24 characters. It is case sensitive and can contain numbers, letters, spaces, punctuation, and symbols. You can optionally select a different password protection option. For more information, click the **Help** button at the bottom right of the screen.

**Note:** The user must enter the token file password when importing the token. The password is not used again.

5. Under **Output**, in the **Target Directory** field, browse to the directory on your system to which you want the token file to be exported.

6. Under **Output**, select **One Token Per File**.

7. Click **Next**, and select **One user**.

8. Click **Next**, and select the user for whom you want to issue the token. Click **OK**, and click **Next**.

9. Do one of the following:
   - To require passcode authentication, leave **Do not change** selected or select **User authenticates with passcode**.
   - To issue a token that does not require a PIN, select **User authenticates with tokencode only**.

10. Click **Next**. When asked if you want to continue, click **Yes**. The Edit Token Extension Data screen is displayed. Use the instructions in the following sections to bind the token to a device attribute and set a token nickname.
Bind the Token Using Token Extension Data

To bind a token in RSA Authentication Manager 6.1, you must edit token extension data.

**To bind a token using token extension data:**

1. On the Edit Token Extension Data page, in the **Key** field, enter **DeviceSerialNumber**.
2. In the **Data** field, enter one of the following:
   - The Android 1.x device type GUID: a01c4380-fc01-4df0-b113-7fb98ec74694
   - The device IMEI/MEID that you obtained from the user.
3. Click **Save**.

The following figure shows the new token binding attribute as it is displayed in the Edit Token Extension Data dialog box. In this example, the token identified as serial number 000005000008 is bound to the user’s device IMEI.

![Edit Token Extension Data](image)

**Next Steps**

For information on delivering file-based tokens to users’ devices, see “Delivering File-Based Tokens” on page 51.
Delivering Software Tokens

This chapter describes the methods that you can use to deliver tokens issued in RSA Authentication Manager to Android device users.

Token Delivery Mechanisms

RSA provides the mechanisms listed in the following table to deliver software tokens to Android devices. For information to help you determine which mechanism best suits your requirements, see “Token Delivery Options” on page 16.

<table>
<thead>
<tr>
<th>Delivery Mechanism</th>
<th>Supported Provisioning Mechanism</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail with URL link containing converted token data</td>
<td>File-based (SDTID file)</td>
<td>“Delivering File-Based Tokens” on page 51</td>
</tr>
<tr>
<td>E-mail with URL link containing CT-KIP token data</td>
<td>Dynamic Seed Provisioning (CT-KIP)</td>
<td>“Delivering Dynamically Provisioned Tokens” on page 56</td>
</tr>
</tbody>
</table>

Delivering File-Based Tokens

To deliver tokens issued as token files (SDTID files) to Android devices, you must convert the data contained in the SDTID file into Compressed Token Format and send the converted token data appended to a specially constructed URL to the user’s device in an e-mail.

Delivering file-based tokens requires the following tasks, in order:

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Issue the software token as an SDTID file.</td>
<td>“Step 1: Issue the Software Token as an SDTID File” on page 52</td>
</tr>
<tr>
<td>2. Convert the token file to a compressed format that can be imported by the device.</td>
<td>“Step 2: Convert the Token File” on page 53</td>
</tr>
<tr>
<td>3. Compose an e-mail message containing the converted token data, and send it to the user’s device.</td>
<td>“Step 3: Compose and Send an E-mail Message” on page 54</td>
</tr>
<tr>
<td>4. Instruct the user on how to import the token.</td>
<td>“Step 4: Instruct the User on How To Import the Token” on page 55</td>
</tr>
</tbody>
</table>
The following figure shows the steps for converting file-based tokens to Compressed Token Format and delivering them to devices.

**Note:** If you password protect the token in Authentication Manager, the user must enter the token file password to complete the process. After the user correctly enters the token file password, the password is not used again.

**Step 1: Issue the Software Token as an SDTID File**

Issue the software token as an SDTID file, if you have not already done so.

**To issue the software token as an SDTID file:**

- For RSA Authentication Manager 6.1, see Chapter 5, “Issuing Software Tokens Using RSA Authentication Manager 6.1.”
- For RSA Authentication Manager 7.1, see “Issuing Software Tokens Using File-Based Provisioning” on page 35.
Step 2: Convert the Token File

After issuing the token file (SDTID file), you must use the RSA SecurID Software Token Converter to convert the token data. The Token Converter is a free command line utility that converts a software token that has been issued as an SDTID file to a Compressed Token Format consisting of an 81-digit string.

To convert a token file (SDTID file):

2. Follow the instructions in the Token Converter Readme, observing the following requirements:
   • Convert only one token file at a time and only one token per token file.
   • Use the -android option to specify that the output of the Token Converter will be a text string containing converted token data appended to a specially constructed URL that can be imported to a supported Android device.
   • Use the -o filename option to output the string containing the converted token to a text file that you can send to the user. If you do not use the -o filename option, the output is written to the screen.
   • Use the -p password option if you password-protected the token file in Authentication Manager.
   • Do not use the -f option or the -v option. The -android option ignores these options.
   • If you bound the token in Authentication Manager, the binding information will be transmitted as part of the token data.

Token Converter Output

When you convert a token using the -android and -o filename options, the output is a text file that contains token data appended to a specially constructed URL, as shown in the following example

http://127.0.0.1/securid/ctf?ctfData=200000000300572630760165772657732067311631716424243040224110520050357522716747153
Step 3: Compose and Send an E-mail Message

Using Microsoft Outlook with Microsoft Exchange Client, compose and send an e-mail message that contains the converted token URL link.

Before You Begin

- The RSA SecurID application must already be installed on the Android device. If the application has not been installed, the user will not be able to import the token.
- The device must have an Internet connection.
- Securely communicate the token file password to the user, if you assigned one.
- On some Android devices, the user may need to import the token by copying token data from the e-mail and pasting it into the Import Token screen. Consider providing instructions on copying text from the device e-mail application.

To compose and send an e-mail with a link containing token data:

1. Compose an e-mail message similar to the following:

   Touch the following link to import your software token.

   http://127.0.0.1/securid/ctf?ctfData=200000000300572630760165772657732067311631716424243040224110520050357522716747153

2. Deliver the e-mail to the user's device.
Step 4: Instruct the User on How To Import the Token

RSA recommends that you distribute the end user Quick Start document, which contains instructions for importing tokens. Users can also touch Menu > Help within the RSA SecurID application. The following procedures are provided for reference.

Import a Token—URL Link in E-mail

Use this procedure to import a software token from a URL link in an e-mail message.

To import a token from a URL link in e-mail:

1. On your device, open the e-mail, and touch the link in the e-mail message.
2. When prompted to select which application to open (Complete action using), touch RSA SecurID.
3. If prompted, enter the token password, and touch OK.
4. After you have imported the token, delete the e-mail containing the token data.

Import a Token—Import Token Screen

Use this procedure to import a software token if the link in the e-mail does not work on your device.

To import a token using the Import Token screen:

1. On your device, open the e-mail.
2. Open your device’s Copy function (for example, by using a long press), and copy the entire string containing the token data.
3. Verify that you did not leave out characters or copy extra spaces or characters.
4. Touch the RSA SecurID icon to open the application.
5. Touch Menu > Import Token.
6. On the Import Token screen, touch Menu > Paste to paste the data into the Enter Token Data field.
7. Touch Import.
8. If prompted, enter the token password, and touch OK.

Important: If a progress indicator is displayed, wait for the import to finish. Do not cancel the import.

9. After you have imported the token, delete the e-mail containing the token data.
Delivering Dynamically Provisioned Tokens

To deliver dynamically provisioned tokens to Android devices, you must construct a URL link containing the token data and send this link in an e-mail to the user’s device. The following tasks are required, in order:

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Issue a dynamically provisioned token.</td>
<td>“Step 1: Issue a Dynamically Provisioned Token” on page 56</td>
</tr>
<tr>
<td>2. Construct a URL link.</td>
<td>“Step 2: Construct a URL Link” on page 56</td>
</tr>
<tr>
<td>3. Compose an e-mail message containing the link, and send it to the user’s device.</td>
<td>“Step 3: Compose and Send an E-mail Message” on page 58</td>
</tr>
<tr>
<td>4. Instruct the user on how to import the token.</td>
<td>“Step 4: Instruct the User on How To Import the Token” on page 59</td>
</tr>
<tr>
<td>5. Verify successful execution of the CT-KIP protocol</td>
<td>“Step 5: Verify Successful Execution of the CT-KIP Protocol” on page 60</td>
</tr>
</tbody>
</table>

**Step 1: Issue a Dynamically Provisioned Token**

Configure the token record and issue a dynamically provisioned token, if you have not already done so.

**To configure a token record and issue a dynamically provisioned token:**
See “Issuing Software Tokens Using Dynamic Seed Provisioning” on page 31.

**Step 2: Construct a URL Link**

When you issue a token in Authentication Manager using Dynamic Seed Provisioning, the Distribute Software Token page displays the CT-KIP server URL link in the **Service Address** field, and the token activation code in the **Activation Code** field, as shown in the following figure:

![Distribute Software Token](https://example.com/distribute_token.png)

<table>
<thead>
<tr>
<th>CT-KIP Credentials</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Token Key Generation URL:</td>
<td><a href="https://ctkserver123.yourco.com/ctkip/trigger.jsp?authcode=">https://ctkserver123.yourco.com/ctkip/trigger.jsp?authcode=</a></td>
</tr>
<tr>
<td>Activation Code:</td>
<td>00108310</td>
</tr>
<tr>
<td>Service Address:</td>
<td><a href="https://ctkserver123.yourco.com/ctkip/services/ctkipService">https://ctkserver123.yourco.com/ctkip/services/ctkipService</a></td>
</tr>
</tbody>
</table>
To allow the device to import the token data, you must construct a larger URL link consisting of prefixed text combined with three separate parameters: Your CT-KIP server URL scheme (http or https), the URL body, and the activation code.

**To construct the URL link:**

Construct the URL link as shown below. The created link must be URL encoded.

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

The URL link must start with the prefix text

```
http://127.0.0.1/securid/ctkip?
```

This format allows the device to communicate directly with Authentication Manager. The device will not communicate with rsa.com. Instead, this specific URL causes the device to launch the SecurID app and begin the CT-KIP process.

- Insert your CT-KIP server URL scheme (http or https) immediately after “scheme=.” Do not include the colon and slash characters (://).
- Replace `<service address>` with the CT-KIP server URL from the Service Address field. Use the same URL for each new token that you issue.
- Replace `<activation code>` with the one-time token activation code from the Activation Code field. Each new token that you issue will have a different activation code.

Including the activation code in the URL is optional. You can omit the activation code if you want to send the same URL to every user. If you send the activation code separately, the user is prompted to enter it before the CT-KIP process starts.

The following example shows a properly constructed URL link with the activation code appended.

```
http://127.0.0.1/securid/ctkip?scheme=https&url=
ctkserver12.yourco.com/ctkip/services/CtkipService&activationCode=00108310
```

The following example shows a properly constructed URL link without the activation code appended:

```
http://127.0.0.1/securid/ctkip?scheme=https&url=
ctkserver123.yourco.com/ctkip/services/CtkipService
```
Step 3: Compose and Send an E-mail Message

Using Microsoft Outlook with Microsoft Exchange Client, create an e-mail message that contains the URL link that you constructed.

Before You Begin

- The RSA SecurID application must already be installed on the Android device. If the application has not been installed, the user will not be able to import the token.
- The Android device must have network connectivity.
- Securely communicate the activation code to the user if you did not include it in the constructed URL link.
- On a few devices, the user may need to import the token by copying token data from the e-mail and pasting it into the Import Token screen. Consider providing instructions on copying text from the device e-mail application.
- If you use a self-signed certificate in your Authentication Manager CT-KIP deployment, the application displays a warning that the certificate is not trusted and prompts the user to accept or reject the certificate. Instruct the user to accept the certificate.

To compose and send an e-mail with a link containing token data:

1. Compose an e-mail message similar to the following:

   Touch the following link to import your software token.
   
   http://127.0.0.1/securid/ctkip?scheme=https&url=ctkserver123.yourco.com/ctkip/services/CtkipService&activationCode=00108310

2. Deliver the e-mail to the user's device.
Step 4: Instruct the User on How To Import the Token

RSA recommends that you distribute the end user *Quick Start* document, which contains instructions on importing tokens. Users can also touch **Menu > Help** within the RSA SecurID application. The following procedures are provided for reference.

**Import a Token—URL Link in E-mail**
Use this procedure to import a software token from a URL link in an e-mail message.

**To import a token from a URL link in e-mail:**
1. On your device, open the e-mail, and touch the link in the e-mail message.
2. When prompted to select which application to open (**Complete action using**), touch **RSA SecurID**.
3. If prompted, enter the token activation code, and touch **OK**.
4. After you have imported the token, delete the e-mail containing the token data.

**Import a Token—Import Token Screen**
Use this procedure to import a software token if the link in the e-mail does not work on your device.

**To import a token using the Import Token screen:**
1. On your device, open the e-mail.
2. Open your device’s Copy function (for example, by using a long press), and copy the entire string containing the token data, making sure not to omit characters or copy extra spaces or characters.
3. Touch the RSA SecurID icon to open the app.
4. Touch **Menu > Import Token**.
5. On the Import Token screen, touch **Menu > Paste** to paste the data into the **Enter Token Data** field.
6. Touch **Import**.
7. If prompted, enter the token activation code, and touch **OK**.

**Important:** If a progress indicator is displayed, wait for the import to finish. Do not cancel the import.

8. After you have imported the token, delete the e-mail containing the token data.
Step 5: Verify Successful Execution of the CT-KIP Protocol

To verify successful communication between the CT-KIP server and the CT-KIP client:

Access the Administration Activity Monitor in the RSA Security Console (Reporting > Realtime Activity Monitors > Administration Activity Monitor).

The following example shows that a CT-KIP key for the token was delivered to the device and that the activation code was deleted from the security domain.

<table>
<thead>
<tr>
<th>Time</th>
<th>Activity Key</th>
<th>Description</th>
<th>Result</th>
<th>Administrator User ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010-05-07</td>
<td>Generate CTKIP Key</td>
<td>Administrator &quot;SYSTEM&quot; attempted to generate CTKIP key for CTKIP data &quot;Token&quot; stored in identity source &quot;SYSTEM&quot; managed in security domain &quot;SystemDomain&quot;</td>
<td>Success</td>
<td>SYSTEM</td>
</tr>
<tr>
<td>2010-05-07</td>
<td>Delete CTKIP Authcode</td>
<td>Administrator &quot;SYSTEM&quot; attempted to delete CTKIP authcode &quot;Token&quot; managed in security domain &quot;SystemDomain&quot;</td>
<td>Success</td>
<td>SYSTEM</td>
</tr>
</tbody>
</table>

These results typically indicate that the token was successfully imported to the application. However, in some cases the Activity Monitor might report successful execution of the CT-KIP protocol, but other factors might prevent a successful import. For example, the token would not be imported if the Android device has network connectivity problems.

If the Activity Monitor does not display any activity, the problem is probably a connectivity issue involving the Android device and the CT-KIP service.

Notification of Software Token Expiration

Software tokens used with Android devices expire at 00:00:00 GMT of the token’s expiration date. The expiration date is displayed in the About screen of the RSA SecurID application. To ensure that the user always has a working software token installed, the application displays a token expiration notification starting 30 days before the token expiration date. For example, if a token is set to expire on July 31, the user receives the first notification on July 1. The 30-day window gives the user ample time to request a new software token and import it to the device.
7: Troubleshooting

This chapter describes issues that might occur with RSA SecurID Software Token for Android, and their corresponding solutions.

Customer Support Information

If you need to contact RSA Customer Support in order to resolve an issue, have the following information available:

- The date and time set on the device
- The information presented on the About screen
- Device model
- Cellular network or wireless provider
- Device firmware version

Include a detailed description of the problem that can be used to form the basis for steps to reproduce the issue.

Obtain the Device Firmware Version

To obtain the firmware version:

On the device, touch Menu > Settings > About Phone.

Depending on the device, the firmware version is listed as Android version or Firmware version.
# Application Installation Problems

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

<table>
<thead>
<tr>
<th>Problem</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>The application cannot be found in the Android Market.</td>
<td>The user has an unsupported device. For a list of hardware requirements, see “System Requirements” on page 9.</td>
</tr>
<tr>
<td>The user cannot install the application.</td>
<td>The device does not have network connectivity, or a network failure occurred. Instruct the user to establish a network connection.</td>
</tr>
<tr>
<td>The Android device does not have enough space to install the application.</td>
<td>Instruct the user to free up space on the device. See “System Requirements” on page 9.</td>
</tr>
</tbody>
</table>
## Token Import Problems

This section describes problems that users might encounter when they attempt to import a software token, and provides workarounds.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Error</strong></td>
<td></td>
</tr>
<tr>
<td>The user cannot import a token because the application has not been installed on the device.</td>
<td>Instruct the user to download and install the application before attempting to import a token.</td>
</tr>
<tr>
<td>In a file-based import, the user forgot the token file password or entered an incorrect token file password.</td>
<td>The user must retry with the correct password or contact the administrator for the password.</td>
</tr>
<tr>
<td>The user attempted to import a dynamically provisioned token, but the import failed because the device does not have network connectivity.</td>
<td>The user must retry after establishing a network connection.</td>
</tr>
<tr>
<td><strong>Administrator Error</strong></td>
<td></td>
</tr>
<tr>
<td>One of the following errors occurred in issuing the token in RSA Authentication Manager:</td>
<td></td>
</tr>
<tr>
<td>• The token is not intended for an Android device.</td>
<td>If you issue tokens in RSA Authentication Manager 7.1, verify that you installed the Android 1.x device definition file and that you selected <strong>Android 1.x</strong> as the device type when issuing tokens.</td>
</tr>
<tr>
<td>• The token device binding is incorrect. For example, the administrator may have entered an incorrect IMEI/MEID when binding the token to a device.</td>
<td>Correct the token device binding and reissue the token.</td>
</tr>
<tr>
<td>• The token type is not supported, for example, 64-bit SID.</td>
<td>Issue a 128-bit (AES) token.</td>
</tr>
<tr>
<td>• The death date of the token lifetime configured in Authentication Manager has passed.</td>
<td>Issue a new token.</td>
</tr>
</tbody>
</table>
### Problem Workaround

<table>
<thead>
<tr>
<th>Problem</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CT-KIP Errors</strong></td>
<td>The user cannot import a token because of an error in the URL link containing the token data. The most likely cause is that the administrator used an older, unsupported link format.</td>
</tr>
</tbody>
</table>
|  | Correct the URL link format, and reissue the token. The URL link must start with the following prefix text:  
http://127.0.0.1/securid/ctkip?scheme=  
For examples, see “Step 2: Construct a URL Link” on page 56. |
|  | The e-mail 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, resend the e-mail to the user’s device. |
| **Other** | The user cannot launch the application or import a software token, because the application cannot not retrieve device information. |
|  | • 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. |
|  | • This problem can also occur if you install the application on a WiFi-only device. The application does not support WiFi-only devices. |
| **Compressed Token Errors** | The user cannot import a token because of an error in the URL link containing the token data. The most likely cause is that the administrator used an older, unsupported link format. |
|  | Correct the URL link format, and reissue the token. The URL link created by Token Converter 2.6 starts with the following prefix text:  
http://127.0.0.1/securid/ctf?ctfData=  
For an example, see “Step 2: Convert the Token File” on page 53. |
<p>|  | The SDTID file was not converted properly with the Token Converter utility. For example, the -p password option was not specified for converting a password-protected token file. |
|  | Review the instructions in “Delivering File-Based Tokens” on page 51. Also see the RSA SecurID Software Token Converter Readme. |</p>
<table>
<thead>
<tr>
<th>Problem</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Token Converter could not convert the SDTID file because the file contained double-byte characters in the UserFirstName, UserLastName, or UserLogin fields.</td>
<td>See “Known Issues” in the RSA SecurID Software Token Converter Readme.</td>
</tr>
<tr>
<td>When issuing a token, the administrator chose to export multiple tokens in a single SDTID file. The Token Converter properly converted the first token in the file, but that token was intended for a different user, so the user’s attempt to authenticate failed.</td>
<td>Reissue the token, making sure that each SDTID file contains only one token.</td>
</tr>
</tbody>
</table>

**Import from E-Mail Errors**

<table>
<thead>
<tr>
<th>Problem</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td>The device e-mail application does not correctly handle the link containing token data.</td>
<td>Send the token data link to the user in plain text, and instruct the user to manually enter it on the RSA SecurID application’s Import Token screen.</td>
</tr>
</tbody>
</table>
## Authentication Problems

This section describes problems that users might encounter when attempting to authenticate, and provides workarounds.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Workaround</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User Error</strong></td>
<td></td>
</tr>
</tbody>
</table>
| 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 subsequent tokencode displayed on the device. After you obtain the pair of tokencodes, resynchronize the token with the Authentication Manager server.  
*Note:* Instruct users with PIN-enabled tokens to touch the **Enter** button to display the tokencode. No PIN is required. |
| The user’s token requires a PIN, but the user attempted to authenticate before setting a PIN. | Instruct the user to follow the instructions in the end-user **Quick Start** or application Help to set a PIN. |
| The user entered an incorrect PIN or entered the PIN in the wrong location. For example, 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. Provide the end-user **Quick Start** or remind the user to access the application Help. |
| **Other** | |
| The time on the Android device may be out of synchronization with the clock settings in Authentication Manager. | Instruct the user to access the About screen in the RSA SecurID application and to read you the time shown in the **GMT** field. |
| The user’s token has expired. | The user must contact the administrator to request a replacement token. |
## Error Messages

The following table describes error messages that users might encounter, including possible reasons for the errors and corrective actions.

<table>
<thead>
<tr>
<th>Message</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The PIN must either be blank or be 4-8 digits long.</td>
<td>The PIN contains fewer than 4 digits or more than 8 digits.</td>
<td>Enter a PIN that meets the PIN requirements.</td>
</tr>
<tr>
<td>No network connection.</td>
<td>The device does not have an Internet connection.</td>
<td>Establish an Internet connection on the device.</td>
</tr>
<tr>
<td>Incorrect password.</td>
<td>The user entered an incorrect token file password.</td>
<td>Retry with the correct password.</td>
</tr>
<tr>
<td>The certificate issuer for this site is untrusted or unknown. Do you wish to proceed? (Yes/No)</td>
<td>This message may be displayed during a CT-KIP import for a variety of reasons, for example, if the RSA Authentication Manager CT-KIP implementation uses a self-signed certificate.</td>
<td>Select Yes to accept the digital certificate.</td>
</tr>
</tbody>
</table>
| Token import failed. Invalid URL format. Contact your administrator.                              | The CT-KIP or CTF URL format is incorrect.                                                               | • If you use CT-KIP, the URL link must start with the following prefix text:  
  http://127.0.0.1/securid/ctkip?scheme=  
  • If you use the Token Converter utility, verify that you used version 2.6, which creates a URL link that starts with the following prefix text:  
  http://127.0.0.1/securid/ctf?ctfData= |
| Token import failed. Error communicating with server. (Retry/Cancel)                              | The cause may be one of the following:  
  • The CT-KIP server URL is invalid.  
  • The CT-KIP server is not operating.  
  • The network is not available.     | Do one of the following, as needed:  
  • Verify the CT-KIP server URL.  
  • Establish an Internet connection on the device.  
  • Retry later.                      |
| Token import failed. Contact your administrator.                                                  | General token import failure.                                                                           | Reissue the token, and instruct the user to retry the token import process.                        |
| Token import failed. Invalid token data. Contact your administrator.                              | Format of token data was incorrect.                                                                       | • If using CT-KIP, instruct the user to retype the CT-KIP server URL, making sure that the URL begins with http:// or https://.  
  • If using Compressed Token Format, rerun the Token Converter with the proper options. |
<table>
<thead>
<tr>
<th>Message</th>
<th>Cause</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Token import failed. Verify the activation code or contact your</td>
<td>Incorrect CT-KIP server URL or activation code was used.</td>
<td>Verify the URL that was used, reissue the token, and instruct the user to retry the token import.</td>
</tr>
<tr>
<td>administrator. (Retry/Cancel)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Token import failed. Unsupported token. Contact your administrator.</td>
<td>An invalid token was issued, for example, a 64-bit token.</td>
<td>Issue a valid 128-bit token. The application does not support 64-bit tokens.</td>
</tr>
<tr>
<td>Token import failed. Token not intended for this device. Contact your</td>
<td>The token sent to the user was bound to a different device.</td>
<td>Issue a token bound to the correct device.</td>
</tr>
<tr>
<td>administrator.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Token import failed. Token expired. Contact your administrator.</td>
<td>The user attempted to import a token that has already reached its</td>
<td>Issue a replacement token.</td>
</tr>
<tr>
<td></td>
<td>expiration date.</td>
<td></td>
</tr>
<tr>
<td>RSA application data is inaccessible. Contact your administrator.</td>
<td>The token database has become corrupted or is otherwise unreadable.</td>
<td>Instruct the user to uninstall the application and then reinstall it from Android Market. You must then reprovision the token.</td>
</tr>
<tr>
<td>RSA SecurID cannot retrieve device information. Please verify your</td>
<td>• The device was turned on in airplane mode or in a location that did</td>
<td>• Disable airplane mode, if enabled, and restart the device in a location that has cellular service.</td>
</tr>
<tr>
<td>network connection and restart your device.</td>
<td>not have cellular service.</td>
<td>• The application does not support WiFi-only devices.</td>
</tr>
<tr>
<td></td>
<td>• The user installed the application on a WiFi-only device.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Index

A
- activation code, 13
  - maximum length, 31
  - selecting option for, 35
  - setting to IMEI/MEID, 35
- Administration Activity Monitor, 60
- Android device type, 14
- Android Market, 23
- application deployment overview, 21
- application, installation requirements, 23
- attributes, token, 12
- authentication
  - with fob-style token, 19
  - with PINPad-style token, 18
  - without a PIN, 20
- authentication issues, troubleshooting, 65
- authentication requirement
  - setting in RSA Authentication Manager 6.1, 49
  - setting in RSA Authentication Manager 7.1, 32
  - setting in RSA Credential Manager, 39
- authentication server documentation, 7
- authentication servers, 10
- automated token provisioning products, 10

B
- binding a token
  - in RSA Credential Manager, 40
  - in RSA Authentication Manager 6.1, 50
  - in RSA Authentication Manager 7.1, 33
  - options for, 15
- binding mechanism
  - Android device type, 15
  - device IMEI/MEID, 15

C
- clock settings, 11
- configuring
  - file-based token in RSA Authentication Manager 7.1, 36
  - RSA Credential Manager token management, 39
  - token record in RSA Authentication Manager 6.1, 48
  - token record in RSA Authentication Manager 7.1, 32
- converting a token file, 53
- Coordinated Universal Time, 11
- Cryptographic Token Key Initialization Protocol (CT-KIP), 13
- CT-KIP, 13, 56
  - provisioning, in RSA Credential Manager, 40
- CT-KIP activation code
  - maximum length, 31
- CT-KIP protocol, verifying in RSA Authentication Manager 7.1, 60
- Customer Support, 8

D
- Database Administration application, 48
- database encryption key, 15
- delivering tokens
  - CT-KIP import, 56
  - CT-KIP requirements, 58
  - options for, 51
  - using Compressed Token Format, 51
- device binding
  - overview, 14
  - recommendations, 28, 46
  - using Self-Service Console, 38
- device binding information, locating, 28, 46
- device binding mechanisms, comparing, 15
- device definition file, 28, 37
  - importing, 28
- device requirements, 9
- device type, 14, 32
- device type GUID, 32
- device-specific attributes, 32
- Displayed Value option, 34
- documentation, 7
  - authentication servers, 7
  - end user, 24
- Dynamic Seed Provisioning, 13, 31
- dynamically provisioned tokens, requirements for delivering, 58

E
- error messages, 66
- expired token notification, 60

F
- file-based provisioning, 13
  - in RSA Authentication Manager 6.1, 47
  - in RSA Authentication Manager 7.1, 35
fob-style software token, 19

G
getting support, 8

I
IMEI (International Mobile Equipment Identity), 14
IMEI/MEID, 28, 46
    as activation code, 31
    obtaining from RSA SecurID application, 47
    obtaining from the device, 46
importing device definition file, 28
installation package, 23
troubleshooting, 62
installing application, 23
issuing software tokens
    in RSA Credential Manager, 38
    in RSA Authentication Manager 6.1, 45, 47
    in RSA Authentication Manager 7.1, 27

M
MEID (Mobile Equipment Identity), 14

N
nickname, 33, 42
no PIN tokens, 20

O
one-time password (OTP), 9, 18, 19, 20
options for delivering tokens, 16, 51

P
passcode, 9, 18
    authentication requirement in RSA Authentication Manager 6.1, 49
    authentication requirement in RSA Authentication Manager 7.1, 32
    authentication requirement in RSA Credential Manager, 39
Passcode option, 34
password protection, 15, 47
PIN, 19
    creating in RSA Credential Manager, 42
PIN requirement, 18
PINPad-style software token, 18, 34
product package contents, 23

R
request approval notification template, 40
requesting a token using RSA Self-Service Console, 41
requirements, device, 9
RSA Authentication Deployment Manager, 10
RSA Credential Manager, 10, 37, 38
    configuring token management settings, 39
    editing the request approval notification template, 40
    e-mail notification, 43
RSA SecurID application installing, 23
RSA SecurID authentication overview, 17
RSA SecurID PIN, 18, 19
RSA Security Console, 28
RSA Self-Service Console, 41
RSA SecurID Software Token Converter, 53

S
SDTID files, 13
    issuing in RSA Authentication Manager 6.1, 47
    issuing in RSA Authentication Manager 7.1, 36
    password protecting, 15, 47
selecting user authentication requirement
    in RSA Authentication Manager 6.1, 49
    in RSA Authentication Manager 7.1, 32
    in RSA Credential Manager, 39
self-service token provisioning, 37, 38
service, getting support, 8
software token
    device type, 32
    fob-style, 19
    no PIN, 20
    PINPad style, 18, 34
software token attributes, 12
    supported in RSA Authentication Manager 7.1, 34
    supported in RSA Authentication Manager 6.1, 45
software token record
configuring in RSA Authentication Manager 6.1, 48
configuring in RSA Authentication Manager 7.1, 32
software token request approval notification, 43
software token settings, selecting in RSA Authentication Manager 7.1, 34
software tokens
expiration notification, 60
issuing in RSA Credential Manager, 38
issuing in RSA Authentication Manager 6.1, 45
issuing in RSA Authentication Manager 7.1, 27
support and service, 8
synchronizing clock settings, 11
system clock settings, 11
system requirements, 9

T
time-based tokens, 34
token attributes, 12
  supported in RSA Authentication Manager 7.1, 34
token configurations, 12
  supported with RSA Authentication Manager 7.1, 27
  supported with RSA Authentication Manager 6.1, 45
Token Converter
  command line options, 53
token database, 15, 25
token delivery mechanisms, 51
token extension data, 50
token file, 13
  converting, 53
token file password, 47
  assigning in RSA Authentication Manager 6.1, 49
  assigning in RSA Authentication Manager 7.1, 36
  maximum characters, 49
token installation, troubleshooting, 63
token provisioning
  Dynamic Seed Provisioning, 13
  file-based, 13
  mechanisms, 13, 27
token record
configuring in RSA Authentication Manager 6.1, 48
configuring in RSA Authentication Manager 7.1, 32
token security
during provisioning, 14
on the device, 15
tokencode, 9, 19, 20
  authentication requirement in RSA Authentication Manager 6.1, 49
  in RSA Authentication Manager 7.1, 34
  authentication requirement in RSA Credential Manager, 39
Tokencode Duration option, 34
Tokencode Length option, 34
tokencode only authentication requirement in RSA Authentication Manager 6.1, 49
in RSA Authentication Manager 7.1, 32
Tokencode option, 34
Tokencode Type option, 34
tokens
  128-bit, 12
  binding in RSA Authentication Manager 6.1, 50
  binding in RSA Authentication Manager 7.1, 33
delivery options, 16
deployment overview, 21
issuing, 27, 45
security during provisioning, 14
self-service provisioning, 37
 supported configurations, 12
troubleshooting, 61

U
uninstalling application, 25
URL link
  and Compressed Token Format, 54
  and CT-KIP, 56
user authentication requirement
  in RSA Authentication Manager 7.1, 32
  in RSA Authentication Manager 6.1, 49
  in RSA Credential Manager, 39