Abridged
RSA Authentication Manager 7.1 Administrator's Guide
for Security Domain Administrators

Iowa State University IT Services

May 2012

The Security Division of EMC

Contact Information

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

Trademarks

RSA and the RSA logo are registered trademarks of RSA Security Inc. in the United States and/or other countries. For the most up-to-date listing of RSA trademarks, go to **www.rsa.com/legal/trademarks_list.pdf**. EMC is a registered trademark of EMC Corporation. All other goods and/or services mentioned are trademarks of their respective companies.

License agreement

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

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

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

Third-party licenses

This product may include software developed by parties other than RSA. The text of the license agreements applicable to third-party software in this product may be viewed in the **thirdpartylicenses.html** files.

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

Limit distribution of this document to trusted personnel.

RSA notice

The RC5™ Block Encryption Algorithm With Data-Dependent Rotations is protected by U.S. Patent #5,724,428 and #5,835,600.

First printing: April 2008

3 Protecting Network Resources with RSA SecurID

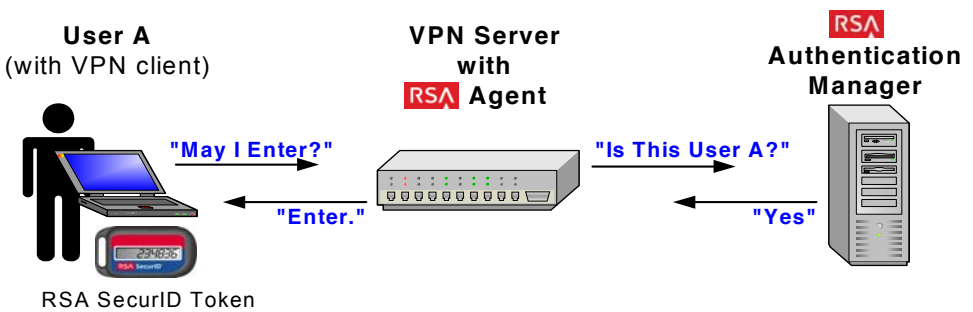
Overview of RSA SecurID Authentication

When a user successfully authenticates through RSA Authentication Manager, he or she is able to access a resource, a VPN server for example, that is protected by Authentication Manager. Authentication Manager uses authentication agents to protect network resources.

Authentication agents must be installed on each machine that you want to protect with Authentication Manager and RSA SecurID. You can either install an agent manually or use hardware that comes with preinstalled authentication agents. Authentication agents are software applications that securely pass authentication requests to and from Authentication Manager.

When a user attempts to gain access to a network resource, the agent receives the authentication request and submits it to Authentication Manager. The Authentication Manager then approves or denies the request, prompting the agent to allow or deny access to the user.

The following figure shows the flow of SecurID authentication:



Installing Authentication Agent Software on the Resource You Want to Protect

There are different types of authentication agents. The agent that you need depends on what type of resource you want to protect. For example, to protect an Apache Web server, download and install RSA Authentication Agent 5.3 for Web for Apache.

RSA provides the latest RSA Authentication Agent software for your platform at **http://www.rsa.com/node.asp?id=1174**. Included with the agent download package is an *Installation and Administration Guide* and a *Readme*. RSA recommends that you read these documents before installing the agent.

Important: For information about installing agent software, see your agent documentation.

You may also purchase products that contain embedded RSA Authentication Agent software. The software is embedded in a number of products, such as remote access servers, VPNs, firewalls, and web servers. For more information about products with embedded RSA Authentication Agents, go to **http://www.rsasecured.com**.

Creating an RSA Agent Record Using the RSA Security Console

After you install and configure authentication agent software on the machines that you want to protect, use the RSA Security Console to create an agent record in the Authentication Manager for each agent. This process is called registering the agent.

The agent record identifies the agent to the Authentication Manager and contains the following configuration information:

Hostname. The name of the machine where you installed the agent software. In most cases, the hostname must be a fully qualified domain name (machine name + domain). For example: 'mymachine.example.net'. However, if the machine is a member of a Windows workgroup, the hostname is the machine name only. For example, 'mymachine.'

If you add an authentication agent to a server node that is also running Authentication Manager, select the hostname from the list of existing server nodes.

IP Address. The IP address of the machine where you installed the agent software.

Protect IP Address. Select this option to prevent the agent auto-registration utility from reassigning the agent's IP address. For more information on agent auto-registration, see "Allowing Agents to Automatically Add Authentication Agent Records" on page 68.

Alternate IP Address. A secondary IP address for the machine where you installed the agent software. You can specify as many as necessary.

Agent Type. This can be Standard Agent or Web Agent. The default agent type is **Standard Agent**. Select **Web Agent** if you are adding an agent to a web server. Select **Standard Agent** for all other agents. This field is for informational use only, and is used primarily to simplify the task of searching for agents.

RADIUS Profile. Select a RADIUS profile for the agent.

Disabled. Select to disable the agent.

Agent May Be Accessed By. You can choose whether to allow all users to authenticate to a specific agent, or allow only users who are members of a user group that has been explicitly given permission to authenticate to the agent.

Agents that allow all users to authenticate are called unrestricted agents. Agents that require users to be members of user groups that are explicitly given permission to authenticate to the agent are called restricted agents. See "Using Authentication Agents to Restrict User Access" on page 73.

Authentication Manager Contact List. By default, authentication agents send authentication requests to the server node that responds first. That server node sends the agent an automatically maintained contact list informing the agent of other server nodes to communicate with if the original server node is offline.

You can override this default by manually assigning the agent a contact list. You should only choose this option if you have specific requirements for managing your authentication request traffic. See "Specifying Where Agents Send Authentication Requests" on page 72.

Trusted Realm Authentication. Enable the agent for trusted realm authentication. For more information on trusted realm authentication, see "Administering Trusted Realms" on page 145.

For instructions, see the Security Console Help topic "Add New Authentication Agents."

Another way to add agents is to duplicate an existing agent. You might do this when you are adding agents with settings similar to an existing agent. For instructions, see the Security Console Help topic "Duplicate Authentication Agents."

You may also configure the system so that agent records are added to the internal database automatically. See the following section, "Allowing Agents to Automatically Add Authentication Agent Records."

Note: To edit an agent record after you add it to the Authentication Manager internal database, see the Security Console Help topic "Edit Authentication Agents."

Allowing Agents to Automatically Add Authentication Agent Records

The Automated Agent Registration and Update utility (**sdadmreg.exe**), included with your RSA Authentication Agent software, enables new authentication agents to automatically add their agent record to the Authentication Manager internal database. This process is called registering the agent. Allowing authentication agents to automatically register themselves saves time and money by eliminating the need for an administrator to perform these tasks.

By default, the Automated Agent Registration and Update utility automatically runs whenever the agent host is started to allow any IP address changes to be registered in the internal database before the agent is started. This is useful for systems that use the Dynamic Host Configuration Protocol (DHCP) to assign IP addresses. If you use DHCP and do not enable this utility, you must manually update the IP addresses each time the agent host changes its IP address.

You can also run the Automated Agent Registration and Update utility manually whenever the IP address of an agent host changes to update the IP address in the internal database.

Note: The RSA Authentication Agent 6.1.2 for Microsoft Windows automatically updates the internal database with any IP address changes. If you are using this agent, you do not need to manually run the utility.

To allow agents to automatically register themselves in Authentication Manager, do the following:

- Install the Automated Agent Registration and Update utility on the agent hosts. Do this during agent installation.

  For utility installation instructions or for information about manually running the utility, see your RSA Authentication Agent documentation.

- Enable agent auto-registration on the Authentication Manager Settings page in the Security Console. This enables agent auto-registration on the Authentication Manager server.

  For configuration instructions, see the Security Console Help topic "Allowing Agents to Register Themselves with RSA Authentication Manager."

Default Agent Settings

When the Automated Agent Registration and Update utility is run on a newly installed, unregistered agent, a record for the agent is created in the internal database. By default, the agent has the following characteristics:

Disabled. The agent is unable to process authentication requests.

Unrestricted. All users are allowed to authenticate with this agent.

Has not been passed the node secret. The node secret is a shared secret known only to the authentication agent and the Authentication Manager.

IP address is unprotected. The agent IP address is not protected by default, so the agent auto-registration utility can reassign the IP address if the agent is inactive. Select this option if you want to prevent the agent auto-registration utility from reassigning the IP address to another agent.

If these default settings are not appropriate for the new agent, edit the agent record to change the settings. For instructions, see the Security Console Help topic "Edit Authentication Agents."

Agent Auto-Registration and Denial of Service (DOS)

It is important that you protect your critical IT infrastructure from potential Denial of Service (DOS) attacks. To reduce the vulnerability of your system:

- Disable agent auto-registration on critical machines such as e-mail and VPN servers.
- In your IT infrastructure, give critical agents static IP addresses.
- Protect IP addresses within Authentication Manager. To do this, select Protect IP Address on the Authentication Agent page in the Security Console. For more information, see "Creating an RSA Agent Record Using the RSA Security Console" on page 66.

Creating and Installing the RSA Authentication Manager Configuration File

The Authentication Manager configuration file contains the IP addresses of Authentication Manager server nodes with which an agent can communicate.

You must perform the following tasks for each agent in your deployment:

- Use the Security Console to generate a server configuration file. For instructions, see the Security Console Help topic "Generate the RSA Authentication Manager Configuration File."
- Install the server configuration file (**sdconf.rec**) on the machine where an authentication agent is installed, called the agent host. For instructions on installing the configuration file, see your agent documentation.

Authentication agents use the server node IP addresses in the configuration file to establish initial contact with the Authentication Manager. One of the IP addresses listed in the configuration file must be available for the first authentication.

After an agent makes initial contact with the Authentication Manager, the Authentication Manager provides the agent with a new list of server nodes, called the contact list, where the agent can direct authentication requests. See the following section, "Specifying Where Agents Send Authentication Requests."

If an agent cannot contact any of the server nodes in the contact list, the agent reverts to the Authentication Manager configuration file and uses one of the IP addresses in the configuration file to reconnect with the Authentication Manager.

The Authentication Manager automatically populates the Authentication Manager configuration file with a list of IP addresses, up to the maximum of 11, as follows:

- If you have only one instance in your deployment, an IP address for each server node is included until the list reaches the limit of 11.
- If you have multiple instances in your deployment, an IP address is included for one server node in each instance. IP addresses from each instance are added until the list reaches the limit of 11.

The configuration file also contains port numbers for the Authentication Service and the Agent Auto-Registration Service. You can edit these port numbers on the Authentication Manager Settings page in the Security Console. For instructions, see the Security Console Help topic "Configure RSA Authentication Manager."

Specifying Where Agents Send Authentication Requests

Depending on your license type, your Authentication Manager deployment can have a primary instance, as well as multiple replica instances, each of which may have multiple server nodes that process authentication requests. To increase the efficiency of your deployment, use contact lists to route authentication requests from agents to the server nodes that can respond the quickest.

Contact lists are ordered lists of server nodes available to accept authentication requests, and are created either automatically by the Authentication Manager, or manually by an administrator.

Automatic contact lists. An automatic contact list is assigned to each instance in your deployment. The list contains the IP addresses of each server node in the instance the contact list is assigned to, and the IP address of one server node from each other instance in your deployment, up to a limit of 11. Agents are sent automatic contact lists by default.

These lists are automatically maintained by the Authentication Manager, and are automatically updated each time a new server node is added to the deployment. When the list is updated, a time stamp associated with the list is also updated. Agents use this time stamp to determine when to request an updated list.

The Super Admin can edit an automatic contact list on the Edit Authentication Manager Contact List page in the Security Console. Any edits that you make to an automatic contact list may be overwritten when a new server node is added to the deployment.

Manual contact lists. The Super Admin maintains manual contact lists. They must be updated manually to reflect the most recent list of server nodes. Manual lists can contain the IP address of any server node in the deployment, up to a limit of 11.

You create manual server lists on the Add New Authentication Manager Contact List page in the Security Console. You can edit a manual contact list on the Edit Authentication Manager Contact List page in the Security Console. For instructions, see the Security Console Help topics "Add a Manual Contact List" and "Edit Manual Contact Lists."

Authentication Manager uses contact lists to determine to which server node authentication requests are sent. Contact lists are sent to each agent by Authentication Manager after the initial contact between the agent and Authentication Manager.

Agents request new contact lists as a part of subsequent authentications. Periodically, the agent reviews all the server nodes listed in the contact list to determine where to send authentication requests. The agent uses metrics, such as the amount of time it takes the server node to respond to authentication requests, to determine where to send requests.

If none of the servers on the contact list respond to authentication requests, the agent reverts to the Authentication Manager configuration file and uses one of the IP addresses in the configuration file to reconnect with the Authentication Manager.

For many organizations, automatic contact lists are sufficient. However, you may choose to create a manual contact list if you have a specific way that you want to route authentication requests.

For example, suppose that you are an administrator at a company that has Boston, New York, and San Jose locations. The New York and San Jose locations are small and all authentications are routed to Authentication Manager replica instances at each site. The Boston location, however, is largest, and the primary instance at that location handles all of your Boston location users, as well as all VPN requests from external users. You may choose to create a manual contact list that routes authentication requests to all of your server nodes, except the database sever. This leaves the database server free to replicate data to your replica instances in New York and San Jose.

For instructions, see the Security Console Help topics, "Manage the RSA Authentication Manager Contact List," "Assign a Contact List to an Authentication Agent," and "Edit Manual Contact Lists."

Deploying Tokens to Users

Deploy tokens to users to allow them to authenticate using Authentication Manager.

A token is a device used to deliver a tokencode to the user. A tokencode is a pseudorandom number, usually six digits in length. A tokencode, combined with the user's PIN, is one way in which a user can authenticate through Authentication Manager.

Note: You can also deliver tokencodes using text message or e-mail, instead of assigning the user a token. For more information, see "Delivering Tokencodes Using Text Message or E-mail" on page 85.

Token Types

There are two kinds of SecurID tokens, hardware tokens and software tokens:

- Hardware tokens are usually key fobs or USB keys that display the tokencode.
- Software tokens and their accompanying application are installed on devices such as Palm Pilots and BlackBerries. Once installed in a device, the application can be used to display the tokencode.

While the two types of tokens perform the same function, the situations in which you use them can be very different.

For example, suppose your organization has internal users who must authenticate with a SecurID token when they log on to their desktop computer, as well as a remote sales force whose members must authenticate with a SecurID token when they log on to their laptop computers.

You might choose to distribute hardware tokens to your internal users. Because they generally log on at their desktop machine each day, the internal users are less likely to lose their tokens than someone who travels frequently. Many users choose to attach the key fob to their keychain, so that as long as they have their car keys, they have their token.

You might choose to distribute software tokens to your remote sales force. Your sales force is on the go constantly, and with a software token installed directly on a PDA or cell phone, they will be less likely to leave it at home, or lose it in an airport. As long as they have their PDA, they have their token.

Tokencode Delivery Methods

When a user authenticates with a token, Authentication Manager matches the tokencode entered by the user to the tokencode maintained within Authentication Manager. When the two tokencodes match, authentication is successful.

Hardware and software tokens deliver their tokencodes in one of two ways: time-based or event-based. The tokencode delivery dictates how Authentication Manager verifies the tokencode and authenticates the user:

Time-based. A time-based token displays a tokencode that automatically changes at a set interval, typically every 60 seconds.

For time-based tokens, the tokencodes are kept synchronized with Authentication Manager based on their internal "clocks" or time. So when the tokencode advances every 60 seconds, the corresponding tokencode in Authentication Manager advances as well. When a user authenticates, Authentication Manager matches the tokencodes based on time.

Event-based. An event-based token displays a tokencode only when initiated by the user. For example, an RSA SecurID Display Card only displays a tokencode when the user presses the appropriate button.

For event-based tokens, the tokencodes are kept synchronized with Authentication Manager based on tokencode count. For example, assume that you just assigned an event-based token to a user. To authenticate, the user presses the button on the token and receives the first tokencode. This advances the token's count to one. When the user attempts to authenticate with tokencode one, Authentication Manager matches the entered tokencode to its own tokencode one. The authentication is successful and the user gains access. At this point, Authentication Manager advances its tokencode count to two. The next time the user needs to authenticate, he or she presses the button on the token to get tokencode two, and the cycle continues.

Important: Because successful authentication attempts are based on count, it is very important that the user only advances the tokencode when they need to authenticate. Needless advancing can cause the token to become out-of-sync with Authentication Manager. For information on resynchronizing tokens, see "Resynchronizing Tokens" on page 103.

So a hardware token can be time-based or event-based, and a software token can be time-based or event-based. Regardless of the tokencode delivery method, each tokencode can only be used once.

Note: Event-based tokens cannot be used for offline authentication.

Deployment Steps

To successfully deploy hardware and software tokens to users, you must perform the following steps:

1. Import the hardware or software tokens to Authentication Manager using the Security Console.
2. Optional. Transfer token records to other security domains. You may want to do this for administrative reasons.
3. Assign the hardware or software tokens to users.
4. Distribute the tokens to users.
   - Hardware tokens - distribute the tokens to the users.
   - Software tokens - electronically deliver the software tokens to the assigned users in a token file or using remote token-key generation (CT-KIP).

Note: The deployment steps are the same for time-based and event-based tokens.

All of these steps are described in detail in the sections that follow.

Importing Hardware and Software Token Records

ITS Note: only if you purchased your own tokens.

Before you can assign tokens to users, use the Security Console to import the token records into the internal database.

Hardware tokens are shipped with associated token records stored as XML files. Software token records are shipped as XML files. You import token records on the Import SecurID Tokens Job page in the Security Console.

When you import token records, you must select the security domain where you want to import the token records. You can import token records into any security domain that is included in the scope of your administrative role. To administer the token records, administrators must have an administrative role that includes this security domain, and grants permission to administer tokens.

Token record XML files may be password protected when you receive them. Be sure to get the password from RSA before you try to import the token records.

Note: When importing tokens, you can choose to ignore or overwrite duplicate tokens. If you choose to overwrite duplicate tokens, there are certain cases when a duplicate will not get overwritten. For the complete list of exceptions, see the Security Console Help topic "Import Tokens."

For instructions, see the Security Console Help topic "Import Tokens."

After you import the token records, you can view them in the Security Console, and assign them to users.

Re-Importing Token Records

There are times when you might need to re-import a token record. For example, you might need to re-import a token record that was deleted. If you are re-importing token records for event-based tokens, you must resynchronize the tokens before reassigning them.

For more information on resynchronizing event-based tokens, see "Resynchronizing Tokens" on page 103.

Transferring Hardware and Software Token Records to Other Security Domains

ITS Note: most Security Domain Admins will not have authority over multiple domains and will not need to perform this action.

Depending upon how you have your organizational hierarchy configured, you may want to move tokens from one security domain to another. You can do this through the the Security Console.

Transferring token records allows you to move them in or out of an administrator's scope, or to move them to a security domain associated with the location where the tokens will be used.

By default, token records are imported into the top-level security domain. If you have more than one security domain in your deployment, you can transfer token records from one security domain to another.

For example, assume an organization has created security domains for each of its geographic locations—Boston, New York, and San Jose. Hardware tokens are shipped to each of the locations so that they can be assigned and distributed to users in each location by an on-site administrator. Because the scope of the administrators that assign tokens at each location is limited to their respective security domain, a Super Admin transfers token records to each of the security domains so that they can assign the tokens. After the token records are transferred, the on-site administrators can view the token records and assign the tokens to users.

Assigning and Unassigning Hardware and Software Tokens

Use the Security Console to assign hardware and software tokens to users. A token assigned to a user can be used by that user to authenticate.

Before you can assign tokens to users, you must:

- Import the token records from the XML file to the internal database.
- Make sure a user record exists in Authentication Manager for each user to whom you want to assign a token.

Note: A maximum of three tokens can be assigned to each user. If you attempt to assign more than three tokens at the same time, no tokens are assigned. For example, if a user has no assigned tokens, and you attempt to assign four tokens, no tokens are assigned to the user.

There are two ways to assign a token through the Security Console:

- Select **Assign More** or **Assign Next Available SecurID Token** in the user Context menu on the Users page.
- Select **Assign to User** in the token Context menu. This option only appears for unassigned tokens.

For complete instructions, see the Security Console Help topics "Assign Hardware Tokens" and "Assign Software Tokens."

After you assign a token to a user, distribute the token to the user. For hardware tokens, see "Distributing Hardware Tokens to Users" on page 80. For software tokens, see "Distributing Software Tokens to Users" on page 81.

Tokens Configured to Not Require PINs

Authentication Manager supports authentication with tokens that are configured so that they do not require a PIN. To authenticate, instead of entering the PIN followed by the tokencode, the user enters only the tokencode displayed on the token.

Note: Tokens that do not require PINs are not as secure as tokens that require PINs. RSA recommends that you configure all tokens to require a PIN.

Authenticating with just a tokencode is useful in situations such as:

- When a token is stored on a smart card and must be unlocked by the user with a PIN
- When a software token is on a desktop and must be unlocked with a password

In these situations, the resource is protected by two-factor authentication without the user having to enter two different PINs.

When assigning a token, you can configure both hardware and software tokens so that they do not require PINs. For instructions, see the Security Console Help topic "Authenticate without an RSA SecurID PIN."

Distributing Hardware Tokens to Users

Because hardware tokens are physical devices, you must deliver them to users before they can be used to authenticate.

If your organization has a single location, the fastest and most secure method is to have users pick up tokens at a central location.

If your organization has multiple locations, consider having administrative personnel at each site distribute the tokens. Alternatively, have your administrative staff travel to different locations at pre-announced times. The advantages of this method are the assurance that the hardware tokens are delivered to the right users and that they work when users receive them.

Another distribution method is to mail tokens to users. Mailing hardware tokens through interoffice mail, post, or overnight express, for example, might be more practical for your organization. However, this usually involves more up-front work, such as developing a process for generating mailing labels, and verifying that users receive their tokens, to ensure success.

RSA recommends that you only mail disabled tokens, which can be enabled after receipt by the correct user. Send information about how to enable tokens separately from the actual tokens or make it accessible only from a secure location. You may also want to consider grouping users so mailing can be accomplished in a controlled manner.

Ultimately, you may decide to use a combination of these delivery methods. For example, if you must distribute enabled tokens to assigned users, be sure to use secure channels, such as having them delivered in person by trusted staff members.

Distributing Software Tokens to Users

Distributing a software token is a different process than distributing a hardware token. Because a software token is installed on a device and cannot be mailed, distribution is electronic, and involves generating a token file and delivering the token file to the user.

Note: Before distributing the software token, make sure that you have imported the token records and assigned the token to a user.

There are four steps in the distribution process:

1. Make sure that the user has the token application. The token application is installed on the device and displays the tokencodes on the device screen. To get a token application, go to **http://www.rsa.com/node.asp?id=1313**.

   Installation instructions are included in the token application download kit.

2. Distribute the software token file. Software token files (.sdtid) are generated using the Security Console, and must be distributed to users and installed on desktops and handheld devices. These files can be distributed in two ways:

   - Token file (XML) - Save the software token to an XML file, and deliver it through secure e-mail or other electronic medium.
   - CT-KIP (Remote Token-Key Generation) - Use the Cryptographic Token-Key Initialization Protocol (CT-KIP). This option can only be used with CT-KIP-capable SecurID software tokens. A CT-KIP-capable SecurID software token is a 128 bit token.

3. Deliver the token file to the user through secure e-mail or other secure means. If using CT-KIP, provide the appropriate URL.

4. Instruct the user to install the software token on his or her device.

Instructions for distributing software tokens by file and CT-KIP are in the sections that follow.

Distributing Software Tokens By Token File (XML)

When you distribute software tokens by token file, you can e-mail the token file to the user who can then download the file to install the token. Token files are in XML.

You need the following token information when distributing software tokens by token file:

Note: Different token types require different sets of token information. Depending on the type of token you are distributing, you might not need all of the information described below.

Software Token Device Type. The type of device on which the token is being installed. An RSA SecurID Toolbar Token is an example of a software token device type. You have to select the device type and enter information for the device specific attributes.

You can add additional software token types to Authentication Manager. For more information, see "Adding Additional Software Token Device Types to Your Deployment" on page 120.

Device Nickname. The **Device Nickname** field allows a user to assign a user-friendly name to the software token. For example, a user might name software tokens "Office Token" or "Home Token" to differentiate between the tokens he or she uses at home and the office.

Binding a Software Token to a Device. RSA software tokens include a predefined field named **Device Serial Number**. When you issue the software token to a user, you can enter the serial number of the device in this field, which binds the issued token to the specific device with the corresponding serial number. A token that is bound to a specific device cannot be installed on any other device.

Software Token Selection Criteria. Know which tokens you want to distribute. You can search by security domain, token file format, serial number, and other token data.

Method for Issuing Software Tokens. You can select from the following methods for issuing software tokens:

- Multiple tokens per file. Authentication Manager packs up all token records into a single .sdtid file, and adds the .sdtid file to a .zip archive when it is downloaded.
- One token per file. One software token record is written to an .sdtid file.

Enabling Copy Protection. The Enable Copy Protection option ensures that the software token cannot be copied or moved from the directory in which it is installed on a user's computer or other device. By default, the Copy Protection option is enabled. RSA strongly recommends that you use copy protection.

Note: Copy protection creates a system fingerprint of the user's device and associates this information with the software token. When a device is repaired or upgraded, this information changes. Software tokens must be reissued if a user's computer hardware or device is repaired or upgraded.

Password Protection. When you issue software tokens, you can select from the following protection methods:

- **Password**. Enter a single password of your choice that applies to all software tokens that you issue.
- **User ID**. The user's default logon ID is used as the password.
- **Combination**. The user's default logon is appended to the password that you enter.

When users install the software token on their device, they are prompted for the User ID, password, or both. Passwords prevent unauthorized people from intercepting and using the software tokens. This password is only used when installing the software token.

RSA strongly recommends that you protect the software token files with passwords. You can assign passwords to the software token files as part of the issuing process. Software Token 3.0 passwords can be up to 24 characters. Software Token 2.0 passwords can be up to 8 characters.

Important: If you protect software tokens with a password, be sure to communicate the password to the user in a secure manner. For example, tell the user verbally, and do not write down the password.

Regenerating Software Tokens. Regenerating a software token changes the sequence of numbers generated by the token file.

When you regenerate the token, devices with the token already installed can no longer use it to authenticate.

Regenerating a token allows you to reuse the software token without fear that an old installation of the token will be used by an unauthorized person to authenticate.

Do this when you reissue a software token, move a software token from one device to another, or if a device containing a software token is lost.

You regenerate tokens as part of the issuing process on the Issue Software Tokens page in the Security Console.

When you distribute software token files, you can complete the operation for one or more software tokens at a time. Choose one of the following distribution methods:

- To distribute one or more software tokens at a time, use the **Distribute Software Tokens Job** option in the **Authentication > SecurID Tokens** menu. From there, select **Add New > Issue Software Token Files**.
- You can distribute software tokens individually from the Edit Tokens page. Click **Save and Distribute** to follow the process.

After choosing individual or multiple distribution, do the following:

- E-mail the token file to the user.
- Instruct the user to download the token file to his or her device.
- Instruct the user to download the token application. The token application is installed on the device, and displays the tokencodes on the device screen. Token applications are available from the following URL: http://www.rsa.com/node.asp?id=1313.

  Installation instructions are included in the token application download kit.

Distributing Software Tokens Using Remote Token-Key Generation (CT-KIP)

Note: Before distributing the software token, make sure that you have imported the token records and assigned a CT-KIP-capable token to the user.

When you assign a CT-KIP-capable software token to a user, you can optionally select to use remote token-key generation (CT-KIP) to deploy a token on user devices.

CT-KIP is more secure than other delivery methods because it enables Authentication Manager and the device that hosts the software token, such as a web browser, to simultaneously and securely generate the same token file on a device and the Authentication Manager.

This allows you to put a token file on a user's device without actually sending the token file through e-mail or putting it on external electronic media. This greatly decreases the chances that the token file will be intercepted by an unauthorized person.

You need the following information when distributing software tokens using CT-KIP:

Software Token Device Type. The type of device on which the token is being installed. An RSA SecurID Toolbar Token is an example of a software token device type. You have to select the device type and enter information for the device specific attributes.

You can add additional software token types to Authentication Manager. For more information, see "Adding Additional Software Token Device Types to Your Deployment" on page 120.

Device Nickname. The **Device Nickname** field allows a user to assign a user-friendly name to the software token. For example, a user might name software tokens "Office Token" or "Home Token" to differentiate between the tokens he or she uses at home and the office.

Binding a Software Token to a Device. RSA software tokens include a predefined field named **Device Serial Number**. When you issue the software token to a user, you can enter the serial number of the device in this field, which binds the issued token to the specific device with the corresponding serial number. A token that is bound to a specific device cannot be installed on any other device.

CT-KIP Activation Code. Choose the format of the CT-KIP activation code. The code can be system generated, or you can choose to use a device-specific attribute as the activation code.

Software Token Selection Criteria. Know which tokens you want to distribute. You can search by security domain, token file format, serial number, and other token data.

When you distribute software token files, you can complete the operation for one or more software tokens at a time. Choose one of the following distribution methods:

- To distribute one or more software tokens at a time, use the **Distribute Software Tokens Job** option in the **Authentication > SecurID Tokens** menu. From there, select the **Add New > Generate Token CT-KIP Credentials** option.
- You can distribute software tokens individually from the Edit Tokens page. Click **Save and Distribute** to follow the process.

Important: When you select the RSA SecurID Toolbar Token from the **Software Token Type** menu, be sure to enter the correct serial number in the **Device Serial Number** field. If you enter the serial number incorrectly, the token does not load properly. If you are unsure of the serial number, leave this field blank.

After choosing individual or multiple distribution, do the following:

- Distribute the token-key generation URL to the assigned user through secure e-mail or other secure means.
- Instruct the user to click the URL or to paste it into a browser window running on the user's device. This step generates a token file and loads it on the device.
- Instruct the user to download the token application. The token application is installed on the device, and displays the tokencodes on the device screen. Token applications are available from the following URL: **http://www.rsa.com/node.asp?id=1313**.

  Installation instructions are included in the token application download kit.

Preventing and Handling User Authentication Problems

This section describes educational measures you can take to facilitate administration of your Authentication Manager deployment.

Educating Users About Security Responsibilities

A critical part of implementing a secure system is educating users about their security responsibilities. No security product can fully protect your system if users do not take their security responsibilities seriously.

Authentication Manager can offer no protection against an intruder who has obtained both a user's PIN and SecurID token. Therefore, it is essential to make sure that users are aware of the following obligations. Users must:

- Notify an administrator immediately if a PIN is compromised.
- Notify an administrator immediately if a token is missing.
- Protect tokens from physical abuse.
- Advance event-based tokens only when they need a tokencode for authentication.
- Lock unattended workstations.
- Log off of secure applications and sites when finished, and close open web browsers.

You use the Security Console to disable tokens and clear PINs. For instructions, see the Security Console Help topics "Disable Tokens" and "Clear an RSA SecurID PIN."

4 Administering Users

Note: See Appendix C, "Managing RSA SecurID Tokens with the Microsoft Management Console (MMC)," if you are using the Microsoft Management Console for token-related tasks.

Enabling and Disabling Users

As an administrator, one of your tasks is enabling and disabling users for authentication. Enabled and disabled are terms used to describe the user's authentication status. An enabled user can authenticate using RSA Authentication Manager, but a disabled user cannot.

Users who are added to Authentication Manager, whether added manually or by linking to an identity source, are automatically enabled. You can assign RSA SecurID tokens to enabled users so that they can gain access to the resources protected by Authentication Manager.

You may choose to disable a user if you know that the user does not need to authenticate for an extended period of time, such as during a short-term or long-term leave.

Note: When a user is disabled, any tokens belonging to that user remain enabled. Disabling tokens is a separate function. See "Enabling and Disabling Tokens" on page 102.

For example, assume that one of your users is taking a one-time leave of absence. Although the user will be out of the office for one month, the user will need the ability to authenticate upon returning to work. Since the user's account is going to be inactive for one month, you disable the user's account during that time period. When the user returns to work, you enable the user's account so that the user can authenticate and access the resources protected by Authentication Manager.

Important: A disabled user is different than a user who has been locked out of the system. Disabling is done manually, by the administrator, and means that the user's account has been turned off. Lockout occurs when the system locks the user's account for violating the lockout policy. For more information on assisting users with locked accounts, see the following section, "Assisting Users Who Have Been Locked Out of the System."

Before enabling and disabling users, note the following:

- Enable and disable users on the Edit User page in the RSA Security Console.
- Administrators can only enable and disable users within their scope. For example, the administrator of the Greenley security domain can enable and disable users in Greenley and all of Greenley's lower-level security domains.
- Disabling a user does not remove the user from the identity source.
- Authentication Manager verifies the identity source enable/disable setting at each authentication. Authentication Manager accounts for external identity source enable/disable settings. For example, if you use Active Directory, and the user is disabled in Active Directory, then that user cannot authenticate.

For instructions, see the Security Console Help topics "Enable Users" and "Disable Users."

Assisting Users Who Have Been Locked Out of the System

Each user is governed by the lockout policy of the security domain to which the user is assigned. The lockout policy specifies the number of failed authentication attempts allowed before the system locks a user's account.

Note: A user's account gets locked, not the user's assigned token.

Lockout policies are designed to protect your company's resources from unauthorized individuals who attempt to authenticate by posing as authorized users and guessing passcodes until they find the correct one. It is not uncommon, however, for authorized users to be locked out of the system for exceeding the number of failed authentication attempts. This usually happens when the user incorrectly enters the PIN or tokencode.

When users violate the lockout policy, their accounts are locked and they can no longer authenticate. You can manually unlock the users so that they can authenticate.

Note: Lockout policies can be created so that user accounts are automatically unlocked after a specified period of time. These accounts can also be unlocked manually.

For example, one of your users calls the Help Desk because he has made four authentication attempts and cannot gain access to the system. Because the default lockout policy only allows three failed authentication attempts, you realize that the user's account has been locked. Since the lockout policy also specifies that the account must be unlocked by an administrator, you must unlock the account.

You can manually unlock the user on the Edit User page in the Security Console.

For instructions, see the Security Console Help topic "Enable Users."

Note: Users can also use RSA Credential Manager to unlock their accounts.

Assisting Users Whose Tokens Are Lost, Stolen, Damaged, or Expired

You may occasionally encounter users who are unable to use their tokens because the tokens are either damaged, lost, temporarily misplaced, stolen, or expired.

Important: Encourage your users to report lost or stolen tokens as soon as possible.

When a token is unavailable or expired, the user may need a new token, or require temporary access to Authentication Manager. To assist the user, you can:

- Provide temporary access.

  A user might need to authenticate despite the lost or destroyed token, or while waiting for the arrival of the replacement token. Even with a missing token, two-factor authentication is still possible with the use of an Emergency Access Tokencode, a temporary tokencode generated by Authentication Manager and used for access to the protected resources. For more information, see "Providing Users with Temporary Emergency Access" on page 96.

  Important: If the user has an expired token, replace the token and then provide temporary access. An Emergency Access Tokencode cannot be assigned to an expired token.

- Replace the token.

  Permanently lost, stolen, damaged, or expired tokens must be replaced. For more information on replacing tokens, see "Replacing Tokens" on page 101.

  Note: Users whose tokens are temporarily unavailable (the token was left at home, for example), but known to be in a safe place, do not require replacement tokens. However, these users may require temporary access. See "Providing Users with Temporary Emergency Access" on page 96.

Note: Users can also use Credential Manager to request replacement tokens or to request temporary access to Authentication Manager.

Providing Users with Temporary Emergency Access

Users may occasionally require temporary emergency access to Authentication Manager if their token is temporarily unavailable. For example, users may require temporary access if they leave their token at the office while traveling for business, or if the token has been temporarily misplaced. Users with lost, stolen, damaged, or expired tokens may also require temporary emergency access while waiting for their replacement tokens.

You can provide temporary emergency access to Authentication Manager for the following two scenarios:

- Online authentication.

  Provide emergency access for users with misplaced, lost, stolen, or damaged tokens. Temporary emergency access is available using an Online Emergency Access Tokencode. There are two types of Online Emergency Access Tokencodes:

  - Temporary Fixed Tokencode. A temporary tokencode used in conjunction with the user's PIN. You can configure the expiration date.
  - One-Time Tokencode set. A set of tokencodes. Each tokencode can be used only once, and is used with the user's PIN.

- Offline authentication.

  Provide emergency access for RSA SecurID for Windows users who require emergency access while authenticating offline. These are users with lost or stolen tokens, or users who have forgotten their PIN. Temporary emergency access can be provided in one of two ways:

  - Offline Emergency Access Tokencode. Use this option if the user has a temporarily misplaced, lost, or stolen token. The Offline Emergency Access Tokencode is used with the user's PIN.
  - Offline Emergency Passcode. Use this option if the user has forgotten his or her PIN. The Offline Emergency Passcode is used in place of the user's PIN and tokencode.

Important: If the user has an expired token, replace the token, and then provide temporary access. An Emergency Access Tokencode cannot be assigned to an expired token. See "Replacing Tokens" on page 101.

These scenarios are described in detail in the following sections.

Note: Users can also use Credential Manager to request temporary access to Authentication Manager.

Providing Temporary Emergency Access for Online Authentication

Even with a missing token, two-factor authentication is still possible with the use of an Online Emergency Access Tokencode. The Online Emergency Access Tokencode is an 8-character alphanumeric code generated by Authentication Manager and used for online access to the protected resources. Similar to the tokencode, the Online Emergency Access Tokencode is combined with the user's PIN to create a passcode.

There are two types of Online Emergency Access Tokencodes: Temporary Fixed Tokencodes and One-Time Tokencode sets. A Temporary Fixed Tokencode is a tokencode that can used more than once. You can configure the expiration date and other Temporary Fixed Tokencode attributes. A One-Time Tokencode set is a set of tokencodes, each of which can be used only once. You can specify how many tokencodes are in the set. Both tokencode types work in the same way, using the user's PIN, and are described below.

Note: The format of the Online Emergency Access Tokencode (Temporary Fixed Tokencodes and One-Time Tokencode sets) is determined by the token policy of the security domain to which it belongs. For example, if the token policy is set to allow special characters, the Online Emergency Access Tokencode can include special characters.

For example, assume that a user has lost his or her token. Despite having lost the token, the user needs to authenticate immediately. You assign a set of One-Time Tokencodes for the user to use with his or her PIN until you can replace the lost token.

Important: If the user has an expired token, replace the token, and then provide temporary access. An Online Emergency Access Tokencode cannot be assigned to an expired token. See "Replacing Tokens" on page 101.

To use the Security Console to generate an Online Emergency Access Tokencode for online authentication, select the **Manage Emergency Access Tokencodes** option in the token Context menu (use the Context menu belonging to the missing/lost token). On this page, you can control the use and security of the Online Emergency Access Tokencode.

In the Online Emergency Access section of the Manage Emergency Access Tokencodes page, you can configure the following attributes:

- Select the type of Online Emergency Access Tokencode:
  - Temporary Fixed Tokencode
  - One Time-Tokencode set
- Select the number of tokencodes in the set (One-Time Tokencode sets only).
- Set the Online Emergency Access Tokencode lifetime.

  For security reasons, you may want to limit the length of time the Online Emergency Access Tokencode can be used. Because the Online Emergency Access Tokencode is a fixed code, it is not as secure as the pseudorandom number generated by the token.
- Specify what happens if the missing token is recovered (if the user finds the lost token, for example). You have the following options:
  - Deny authentication with token

    Use this option if you do not want the token to be used for authentication if recovered.

    Important: If the token is permanently lost or stolen, use this option. This safeguards the protected resources in the event the token is found by an unauthorized individual who attempts to authenticate.
  - Allow authentication with token at any time and disable online emergency tokencode

    Use this option if the token is temporarily misplaced (the user left the token at home, for example). When the user recovers the token, he or she can immediately resume using the token for authentication. The Online Emergency Access Tokencode is disabled as soon as the recovered token is used.
  - Allow authentication with token only after the emergency code lifetime has expired and disable online emergency tokencode

    You can also use this option for temporarily misplaced tokens, however when the missing token is recovered, it cannot be used for authentication until the Online Emergency Access Tokencode expires.

Note: You cannot assign an Online Emergency Access Tokencode (Temporary Fixed Tokencode or One-Time Tokencode set) to a disabled token.

For example, a user calls because he or she left his or her SecurID token at the office. The user is currently at home and needs to authenticate immediately. Although the token is not lost, the user still requires temporary access. In this situation, you can generate an Temporary Fixed Tokencode for the user.

Because you know that the user will have the token the following day, you can set a lifetime for the Temporary Fixed Tokencode. You may also choose to specify that the Temporary Fixed Tokencode is automatically disabled when the user attempts to authenticate with his or her token.

Important: You may also encounter a situation where the token is permanently lost. For example, assume one of your users calls to tell you that his or her SecurID token has been stolen. In this situation, you can grant temporary access by generating an Online Emergency Access Tokencode for the user. When granting temporary access, it is extremely important that you choose to deny authentication with the token if it is recovered. This protects your resources in the event an unauthorized individual attempts to authenticate.

For instructions, see the Security Console Help topic "Generate An Online Emergency Access Tokencode."

Note: If the user is certain that the token is permanently lost, destroyed, or expired, you must replace the token. See "Replacing Tokens" on page 101.

Providing Temporary Emergency Access for Offline Authentication

RSA SecurID for Windows users may need temporary emergency access so that they can authenticate while working offline. Temporary emergency access is necessary for users with misplaced, lost, or stolen tokens, or users who have forgotten their PIN.

Note: Temporary emergency access for offline authentication cannot be provided for event-based tokens. Users cannot use event-based tokens for offline authentication.

Users who are authenticating offline can gain temporary emergency access using one of the following options:

- Offline Emergency Access Tokencode. Use this option if the user has a misplaced, lost, or stolen token. The Offline Emergency Access Tokencode is used with the user's PIN and allows two-factor authentication.
- Offline Emergency Passcode. Use this option if the user has forgotten his or her PIN. The Offline Emergency Passcode is used in place of the user's PIN and tokencode.

Important: If the user has an expired token, replace the token, and provide temporary access. An Offline Emergency Access Tokencode cannot be assigned to an expired token. See "Replacing Tokens" on page 101.

Both types of temporary emergency access are described in the sections that follow.

Assigning a Temporary Tokencode for Offline Authentication

To use the Security Console to view and configure an Offline Emergency Access Tokencode for offline authentication, select the **Manage Emergency Access Tokencodes** option in the token Context menu (use the Context menu belonging to the missing/lost token).

In the Offline Emergency Access section of the Manage Emergency Access Tokencodes page, you can:

- View the Offline Emergency Access Tokencode.
- View the Offline Emergency Tokencode expiration date.
- Reset the Offline Emergency Tokencode.
- Allow the user to use the Offline Emergency Access Tokencode for emergency online access.

Note: Offline Emergency Access Tokencodes can only be issued if the user has used the token to authenticate, at least once, to an agent that can provide offline data for SecurID for Windows users.

For instructions, see the Security Console Help topic "Assign an Offline Emergency Access Tokencode."

Assigning a Temporary Passcode for Offline Authentication

To use the Security Console to view and configure an Offline Emergency Passcode for offline authentication, select the **Manage Emergency Offline Access** option in the user Context menu.

On the Manage Offline Emergency Access page, you can:

- View the Offline Emergency Access Passcode.
- View the Offline Emergency Passcode expiration date.
- Reset the Offline Emergency Passcode.

For instructions, see the Security Console Help topic "Assign an Offline Emergency Passcode."

Replacing Tokens

Sometimes you must assign a new token to a user. For example, you must assign a new token if a user's token has been permanently lost or destroyed, or if the current token has expired.

Note: Use the View option in the token Context menu to check the token expiration date.

In the Security Console, there are two ways to assign a replacement token:

- Select the **Replace with Next Available SecurID Token** option on the token Context menu if you want the system to automatically assign the next available token to the user.
- Select the **Replace SecurID Tokens** option on the token action menu if you want to choose the replacement token. If your company has different locations, select a replacement token from the user's office location. This makes distribution more efficient.

  For example, assume that you are the administrator for the New York security domain and one of your users has permanently destroyed his token. You select the **Replace SecurID Tokens** option to assign a new token to the user. Since you have multiple office locations, you select a token that also belongs to the New York security domain.

  In the above example, if all of the available tokens belong to a different security domain, San Jose, for example, you can assign the token as long as both security domains are included in your administrative scope. When you assign the token, you can leave it in the San Jose security domain, or you can transfer it to the New York security domain so that it belongs to the same security domain as the user to which it is assigned. When you assign the token, disable it to keep it secure during the transit to New York. Tell the user to notify you as soon as the token arrives so that you can enable it for authentication.

If you must mail a replacement token, disable the token after you assign it. Once you have confirmation that the user has received the replacement token, re-enable it so that it can be used for authentication. This safeguards the system in the event the token is lost in the mail.

For instructions, see the Security Console Help topic "Replace a Token."

Note: Users can also use Credential Manager to request replacement tokens.

The user may need access to Authentication Manager while waiting for the replacement token. In this case, you can give the user temporary access by generating an Emergency Access Tokencode for the existing token. The user can use the Emergency Access Tokencode to authenticate until the replacement token arrives. For more information on providing temporary access to Authentication Manager, see "Providing Users with Temporary Emergency Access" on page 96.

Enabling and Disabling Tokens

As an administrator, one of your tasks is enabling and disabling tokens so that they can be assigned to users and used for authentication. Enabled and disabled are terms that describe the token's authentication status. An enabled token can be used for authentication, but a disabled token cannot.

After Authentication Manager is installed, tokens must be imported into the system. All imported tokens are automatically disabled. This is a security feature that protects the system in the event that the tokens are lost or stolen.

Note: A disabled token does not refer to a token belonging to a user who has been locked out of the system. Disabling a token is done manually, by the administrator, and means that the token cannot be used for authentication. Lockout applies to a user's account, not a user's token.

You can manually enable and disable tokens on the Edit Token page in the Security Console. You must enable a token before it can be used for authentication.

Important: Tokens are automatically enabled when first assigned to a user.

In these situations, you should disable a token after it has been assigned to a user:

- When it is going to be mailed or delivered to a user. Re-enable the token when you know that it has been successfully delivered to the user to whom it has been assigned.
- If you know that the user to whom the token is assigned does not need to authenticate for some period of time. For example, you may want to disable a token belonging to a user who is going away on short-term leave or extended vacation. Once you disable the token, that user cannot authenticate with the token until the token is re-enabled.

Note: Disabling a token does not remove it from the system. Disabled tokens can be viewed using the Security Console.

For example, assume that one of your users is taking a one-time leave of absence. Although the user will be out of the office for one month, the user will need the ability to authenticate upon returning to work. Since the user's account is going to be inactive for one month, you disable the user's token and the user's account during that time period. When the user returns to work, you enable the user's account and the user's token so that the user can authenticate and access the resources protected by Authentication Manager.

Note: You can only enable and disable tokens in security domains that are included in your administrative scope.

For instructions, see the Security Console Help topics "Enable Tokens" and "Disable Tokens."

Resynchronizing Tokens

You can use the Security Console to resynchronize tokens that have become unsynchronized with Authentication Manager. A token needs to be resynchronized when the following occurs:

- For time-based tokens, resynchronization is necessary when the token clock and the Authentication Manager system clock do not match. When the clocks do not match, the tokencodes are not the same. If the tokencodes are different, authentication attempts fail.
- For event-based tokens, resynchronization is necessary when the token's tokencode count and the Authentication Manager tokencode count are not the same. When the tokencode counts are different, authentication attempts fail.

Important: You must resynchronize event-based tokens if you have re-imported them.

When a token becomes unsynchronized with the system, when the user attempts to authenticate, the system prompts the user to enter the tokencode. If the tokencode is correct, the system prompts the user to enter the next tokencode. This behavior can be confusing to users, as they are used to entering only one tokencode to authenticate. In this case, resynchronize the user's token so that he or she is not prompted for a second tokencode when authenticating.

To use the Security Console to resynchronize the token, select the **Resynchronize Token** option in the token Context menu to launch the Resynchronize Token page.

If you want to resynchronize multiple event-based tokens at the same time, you can enable Database Recovery Mode. When Authentication Manager is in database recovery mode, the system resynchronizes the event-based tokens at the first post-disaster logon.

For instructions, see the Security Console Help topic "Resynchronize a Token."

For more information on database recovery mode for event-based tokens, see the Security Console Help topic "Enable Database Recovery Mode."

Note: Users can also use Credential Manager to resynchronize their tokens.

Clearing PINs

You need to clear a user's PIN if the user has forgotten it. When you clear a PIN, the current PIN is deleted so that the user can create a new one.

When a PIN has been cleared, the user is prompted to create a new PIN on the next authentication attempt. Similar to what happens to users who are authenticating for the first time, the user initially enters their current tokencode only. Upon successfully entering the tokencode, the user is prompted to create and then confirm a new PIN. The new PIN is then associated with the token.

Note: Encourage users to create PINs containing both letters and numbers, as they are more secure. You can also set PIN requirements in the token policy. See "Setting Token Usage Requirements" on page 51.

For example, assume that you are a system administrator and one of your users calls. It has been months since the user has made an authentication attempt, and she has since forgotten her PIN. The user asks you to clear her PIN so that she can create a new one. After verifying the user's identity, you clear the PIN. Tell the user to enter her tokencode when prompted for her passcode on the next authentication attempt. After entering the tokencode, the user is prompted to create a new PIN.

To use the Security Console to clear a PIN, select the **Clear SecurID PIN** option on the token Context menu.

For instructions, see the Security Console Help topic "Clearing an RSA SecurID PIN."

Note: Users can also use Credential Manager to reset their PIN.

RSA SecurID SID800 Authenticators

Users with SID800 Smart Cards need a PIN Unlocking Key to access their token if they have forgotten their PIN. You can view the PIN Unlocking Key on the Token Properties page in the Security Console.

For more information, see the Security Console Help topic "Obtain the PIN-Unlocking Key for a SID800 Smart Card."

Note: You must load the SID800 Smart Card data into Authentication Manager before you can view it. To load the data, use the "Import PIN Unlocking Key Utility" on page 269.

Requiring Users to Change Their PINs

You can force users to change their PINs if there is concern that the PIN has been compromised. A compromised PIN puts the resources protected by Authentication Manager at risk.

Important: Instruct users to report compromised PINs as soon as possible, as they pose a significant security risk.

Forcing a user to change a PIN assumes that the user knows the current PIN. When you force a PIN change, on the next authentication attempt the user authenticates as he or she normally would, using the existing PIN and tokencode. After successfully authenticating, the user is immediately prompted to create a new PIN. The user creates a new PIN, confirms the new PIN, and then the PIN is associated with the token.

For example, assume that you are a system administrator and one of your users calls, concerned that her PIN has been compromised. She was using her computer at a local coffee shop and she is worried that someone may have seen her type her PIN. Because she knows the PIN, it is not necessary to clear the PIN. Instead, require her to create a new PIN on her next authentication attempt.

Depending on the token policy, the user may be required to use a system-generated PIN instead of creating one. In this case, on the next authentication attempt, the system provides the user with the new, system-generated PIN. The user then authenticates again using the new, system-generated PIN.

Note: Encourage users to create PINs containing both letters and numbers, as they are more secure. You can also enforce PIN requirements using a token policy. See "Setting Token Usage Requirements" on page 51.

In the Security Console, select the **Require SecurID PIN Change on Next Logon** option in the token Context menu to force users to change their PINs on the next authentication attempt.

For instructions, see the Security Console Help topic "Force RSA SecurID PIN Changes."

Note: Because this feature requires knowledge of the current PIN, you cannot use it for users who have forgotten their PIN. For more information on how to help users who have forgotten their PINs, see the preceding section "Clearing PINs."

Note: Users can also use Credential Manager to reset their PIN.

Providing Users with Fixed Passcodes

You can assign a fixed passcode to users, which allows them to authenticate without an RSA SecurID PIN and tokencode. Instead, users enter their fixed passcode to gain access to the resources protected by Authentication Manager.

Important: Fixed passcodes are essentially passwords, and are not recommended as they eliminate all of the benefits of two-factor authentication. Use fixed passcodes only in test environments and in situations when users are authenticating to authentication agents within the corporate firewall.

To use the Security Console to set a fixed passcode, use the **Fixed Passcode** field on the Authentication Settings page. The Authentication Settings page is accessed through the user Context menu.

For instructions, see the Security Console Help topic "Managing Fixed Passcodes."

Clearing Incorrect Passcodes

The system counts each time the assigned user enters an incorrect passcode, clearing this count automatically with each correct passcode. If a user enters more incorrect passcodes than are allowed by the SecurID Token policy and then enters a correct passcode, the user is prompted for his or her next tokencode. If you do not want a user to be prompted for the next tokencode, you can use the Security Console to clear incorrect passcodes. Select **Clear Incorrect Passcodes** on the Authentication Settings page (this page is accessed through the user Context menu).

When you select this checkbox, the user is not prompted for the next tokencode on his or her next authentication attempt. Keep in mind, however, that if the user exceeds the number of failed logon attempts allowed by the lockout policy, the user is locked out of the system.

This operation only clears the existing count. To clear future counts, you must perform the procedure again.

For instructions, see the Security Console Help topic "Manage User Authentication Attributes."

Designating a Default Shell for UNIX Users

The default shell is the shell the user logs on to when accessing a UNIX machine.

To use the Security Console to assign a default shell, use the **Default Shell** field on the Authentication Settings page. The Authentication Settings page is accessed through the user Context menu.

For instructions, see the Security Console Help topic "Manage User Authentication Attributes."

Assigning Logon Aliases

Logon aliases allow for situations where users are able to log on with their own user ID and a user group ID. The user group ID is associated with a user group that has access to a restricted agent.

For example, users may be able to use an account name based on their first initial and last name as well as an administrative account with a specific name, such as "root." If a logon alias has been set up, Authentication Manager verifies the authentication using the user's passcode, regardless of the account name the user used to log on to the operating system. For backward compatibility, a shell value is also maintained by the system.

You can assign logon aliases on the Authentication Settings page in the Security Console. The Authentication Settings page is accessed through the user Context menu.

For instructions, see the Security Console Help topic "Manage User Authentication Attributes."

The Security Division of EMC

9 Logging and Reporting

Using the Activity Monitor

An Activity Monitor allows you to view log messages in real time. You can use these real-time messages to see what is happening in the system, or you can use them for troubleshooting. For example, you might need to assist a user who is having trouble authenticating. Looking at the log messages for the user's activities can help you figure out why the user is having trouble.

There are three different Activity Monitors in Authentication Manager, each of which contains different types of log messages:

Authentication Activity Monitor. Displays authentication-specific events such as authentication requests and restricted agent access checks.

System Activity Monitor. Displays system events such as the replication of data.

Administrator Activity Monitor. Displays administrator activities such as creating and updating system administrators.

Note: The Activity Monitor opens in a new browser window and there is no limit to the number of windows that you can open. For example, you can simultaneously monitor a specific administrator, an entire user group, and an entire security domain.

Once the Activity Monitor is launched, you can filter the log events using the available filter criteria. For example, you can use the criteria to filter the data so that you can view the activity of a single administrator, a single authentication agent, or an entire security domain.

You can also configure the display attributes of the log events. For example, you can set the number of messages that the Activity Monitor displays at any given time, up to 1,500 messages. You can also configure the type of messages that the monitor displays. For example, you can view:

- Successful events
- Warning events
- Failure events
- A combination of any of the above event types

New messages are added at the top of the Activity Monitor display. As you reach the message limit that you specified, older messages are removed from the display. For example, if you configured the monitor to display only 50 messages, each message after 50 is added to the top of the display and the oldest message is removed from the display. Click **Clear Monitor** to clear all of the messages from the display.

You can also pause the Activity Monitor display. This allows you to take as much time as you want to view specific log messages. When you resume monitoring, all of log messages that were generated while the monitor was paused are added at the top of the Activity Monitor. If the number of new messages exceeds the number of messages you chose to display, only the most recent messages are displayed.

The Activity Monitor adds a message to the display whenever you pause or resume the monitor. This allows you to keep track of where the pauses occurred within the set of log messages.

Note: You cannot filter log messages while the Activity Monitor is paused.

You can view more details of an event by clicking on it. This launches a pop-up window that displays detailed information on the event. For example, a system log event includes information on the instance, client and node IP addresses, component, and other important system-related data.

To use the Security Console to launch the Activity Monitors, select the **Real-time Activity Monitor** option in the Reporting menu.

For instructions, see the Security Console Help topic "Manage the Activity Monitor."

G Troubleshooting

This appendix contains the following information on some common RSA Authentication Manager issues and their corresponding solutions:

Common Problems and Resolutions

The following table lists common problems, their possible causes, and the corresponding resolutions. Topics are broken down into these categories:

- User and token-related
- System-related
- Identity source or LDAP
- RSA Credential Manager
- Microsoft Management Console (MMC).

Problem	Possible Cause	Resolution
User and Token-Related		
User cannot authenticate or user is getting an access denied message.	User is locked out of Authentication Manager for violating the lockout policy.	Assisting Users Who Have Been Locked Out of the System on page 94.
	User did not violate the Authentication Manager lockout policy, but did violate the external identity source lockout policy (for example, an Active Directory lockout policy).	Check the identity source policy, and unlock the user in the identity source if necessary.
	Token is out of sync with Authentication Manager.	Resynchronizing Tokens on page 103.

Problem	Possible Cause	Resolution
	Token has expired. **Note:** If the token has expired, you see a log message in the audit log.	Assign a new token, and provide emergency access if necessary. See Providing Emergency Access on page 417. **Note:** To avoid having users with expired tokens, schedule a recurring report that shows tokens close to expiration. Be proactive and replace tokens before they expire.
	The user ID is too long.	Do not create a user ID longer than 48 characters.
	The firewall is not configured properly or the appropriate ports are not open.	Assessing the Impact of Firewalls on RSA Authentication Manager on page 412.
	IP name resolution or agent host name is entered incorrectly.	Name and IP Address Resolution in RSA Authentication Manager on page 420.
	The agent configuration file is corrupt or invalid.	Updating an Agent Configuration File on page 422.
	If the user is trying to access a restricted agent, and your deployment uses an Active Directory forest or Global Catalog for authentication, the default group type must be set to Universal.	Specify the default group type on the Add Identity Source page in the Operations Console. For more information, see "Adding an Identity Source in RSA Authentication Manager" on page 25. For more information on using Active Directory and Global Catalogs, see Appendix A, "Integrating Active Directory Forests."

Problem	Possible Cause	Resolution
	If you have configured Authentication Manager to allow system-generated PINs, and your deployment includes RSA RADIUS, you must configure RADIUS to allow system-generated PINs.	By default, RADIUS does not allow system-generated PINs. Edit the RADIUS configuration file, **securid.ini**, to allow system-generated PINs. For more information, see "Using System-Generated PINs with RSA RADIUS" on page 48.
	Authentication Manager is out of sync with Coordinated Universal Time (UTC). **Note:** If Authentication Manager is out of sync with UTC, all of the users are unable to authenticate.	Resynchronizing RSA Authentication Manager with Coordinated Universal Time on page 421.
User is being prompted to enter a second tokencode.	User has violated the token policy and incorrect passcodes must be cleared.	Clearing Incorrect Passcodes on page 106.
	Token is out of sync with Authentication Manager.	Resynchronizing Tokens on page 103.
	Authentication Manager is out of sync with Coordinated Universal Time (UTC). **Note:** If Authentication Manager is out of sync with UTC, all of the users are prompted to enter a second tokencode or are unable to authenticate. The behavior they experience is based on the time discrepancy.	Resynchronizing RSA Authentication Manager with Coordinated Universal Time on page 421.
User is being prompted to create a new PIN.	PIN has been cleared.	Instruct user how to create a new PIN.
	User has a new token.	Instruct user how to create a PIN.

Problem	Possible Cause	Resolution
An existing user (user exists in Active Directory) gets an error message saying that he or she cannot enroll in RSA Credential Manager.	The user account is disabled, locked, expired, or set to require users to change their password during the next logon in the Active Directory.	If Active Directory has the "change password during next logon" option set, then users cannot enroll in Credential Manager. Clear this option so that users can enroll in Credential Manager. Resolve disabled, locked, or expired users in Active Directory to allow users to enroll in Credential Manager.
When administering RSA Credential Manager, if you approve a request with two approval steps and then click the Submit and Continue button, an error message appears stating that the request is completed.	If a custom extension is added to a request with two approval steps to automatically approve the second approval step, there is a time delay that causes the error message to appear.	Wait and then refresh the RSA Security Console.

User and Token-Related Resolutions

As an administrator, you receive telephone calls from users who need assistance. For example, users may call because they cannot authenticate, or because they have lost or damaged their token. This section provides information on the following user-related situations that you may encounter as an administrator:

- Unlocking a User
- Assisting Users with Lost, Stolen, Damaged or Expired Tokens
- Providing Emergency Access
- Clearing PINs
- Forcing PIN Changes
- Clearing Incorrect Passcodes
- Resynchronizing a Token

Unlocking a User

Users are locked out of Authentication Manager for the following reasons:

- The user violated the lockout policy specified by the security domain to which he or she belongs.

  For this situation, see Assisting Users Who Have Been Locked Out of the System on page 94.

- The user violated the lockout policy as specified by the external identity source to which he or she belongs. Some identity sources, Microsoft Active Directory and Sun Java System Directory Server, for example, have their own lockout policies.

  If a user violates the identity source lockout policy, the user profile in the Security Console does not indicate that the user is locked out, but the user is unable to authenticate. Check your identity source to see if the user has violated the identity source lockout policy. If so, unlock the user.

Assisting Users with Lost, Stolen, Damaged or Expired Tokens

You may occasionally encounter users who are unable to use their tokens because the tokens are either damaged, lost, temporarily misplaced, stolen, or expired. In these situations, replace the token (if applicable) and provide temporary emergency access if necessary.

Important: Encourage your users to report lost or stolen tokens as soon as possible.

See "Assisting Users Whose Tokens Are Lost, Stolen, Damaged, or Expired" on page 95.

Providing Emergency Access

Users may occasionally require temporary emergency access to Authentication Manager if their token is temporarily unavailable, or if they are waiting for a replacement for their lost, stolen, damaged, or expired token.

You can provide temporary emergency access to Authentication Manager for the following scenarios:

Online authentication. Provides emergency access for users with misplaced, lost, stolen, or damaged tokens. Temporary emergency access is available using an Online Emergency Access Tokencode.

Offline authentication. Provides emergency access for RSA SecurID for Windows users who require emergency access while authenticating offline. Temporary emergency access is available using an Offline Emergency Access Tokencode or an Offline Emergency Passcode.

See "Providing Users with Temporary Emergency Access" on page 96.

Clearing PINs

You need to clear a user's PIN if the user has forgotten it. When you clear a PIN, the current PIN is deleted so that the user can create a new one.

See "Clearing PINs" on page 104.

Forcing PIN Changes

You can force users to change their PINs if there is concern that the PIN has been compromised. A compromised PIN puts the resources protected by Authentication Manager at risk.

Important: Instruct users to report compromised PINs as soon as possible, as they pose a significant security risk.

See "Requiring Users to Change Their PINs" on page 105.

Clearing Incorrect Passcodes

The system counts each time the assigned user enters an incorrect passcode, clearing this count automatically with each correct passcode. If a user enters more incorrect passcodes than allowed by the token policy, and then enters a correct passcode, the user is prompted for his or her next tokencode.

If you do not want a user to be prompted for the next tokencode, you can clear the incorrect passcodes so the user does not violate the token policy.

Note: If the user has violated both the lockout policy and the token policy, you must unlock the user account after clearing the incorrect passcodes. For more information, see "Unlocking a User" on page 416.

See "Clearing Incorrect Passcodes" on page 106.

Resynchronizing a Token

A token needs to be resynchronized when the following occurs:

- For time-based tokens, resynchronization is necessary when the token clock and the Authentication Manager system clock do not match. When the clocks do not match, the tokencodes are not the same. If the tokencodes are not the same, authentication attempts fail.
- For event-based tokens, resynchronization is necessary when the token's tokencode count and the Authentication Manager tokencode count are not the same. When the tokencode counts are different, authentication attempts fail.

See "Resynchronizing Tokens" on page 103.

Glossary

Term	Definition
Active Directory	The directory service that is included with Microsoft Windows Server 2003 and Microsoft Windows 2000 Server.
Active Directory forest	A federation of identity servers for Windows Server environments. All identity servers share a common schema, configuration, and Global Catalog.
AD	See Active Directory.
adjudicator	A component that defends Authentication Manager against replay attacks in which an intruder attempts to reuse an old passcode or acquires the current passcode for a token and sets the system clock back to use the captured passcode.
administrative command	A command other than a system-generated command.
administrative role	A collection of permissions and the scope within which those permissions apply.
administrator	Any user with one or more administrative roles that grants administrative permission to manage administrative resources.
Advanced Encryption Standard (AES)	The current cryptographic standard, adopted by the National Institute of Standards and Technology (NIST) in November, 2001. AES replaces Data Encryption Standard (DES) because it is considered to be more secure.
AES	See Advanced Encryption Standard.
agent	A software application installed on a device, such as a domain server, web server, or desktop computer, that enables authentication communication with Authentication Manager on the network server.
agent auto-registration utility	A utility included in the RSA Authentication Agent software that enables you to automatically register new authentication agents in the internal database, and updates the IP addresses for existing agents.
agent host	The machine on which an agent is installed.

Term	Definition
Agent Protocol Server	The Authentication Manager component that manages the ACE protocol packet traffic to and from agents. The inbound request packets are routed to the appropriate message handler. The response packets are sent to the originating agent.
approver	A Request Approver or an administrator with approver permissions.
attribute	A characteristic that defines the state, appearance, value, or setting of something. In Authentication Manager, attributes are values associated with users and user groups. For example, each user group has three standard attributes called Name, Identity Source, and Security Domain.
attribute mapping	The process of relating a user or user group attribute, such as User ID or Last Name, to one or more identity sources linked to a given realm. No attribute mapping is required in a deployment where the internal database is the primary identity source.
audit information	Data found in the audit log representing a history of system events or activity including changes to policy or configuration, authentications, authorizations, and so on.
audit log	A system-generated file that is a record of system events or activity. The system includes four such files, called the Trace, Administrative, Runtime Audit, and System logs.
authentication	The process of reliably determining the identity of a user or process.
authentication authority	The central entry point for authentication services.
authentication broker	A component that handles the authentication process and issuance of authentication tickets.
authentication method	The type of procedure required for obtaining authentication, such as a one-step procedure, a multiple-option procedure (user name and password), or a chained procedure.
authentication policy	A collection of rules that specify the authentication requirements. An authentication policy may be associated with one or more resources.
authentication protocol	The convention used to transfer credentials of a user during authentication. For example, HTTP-BASIC/DIGEST, NTLM, Kerberos, and SPNEGO.

Term	Definition
Authentication Server	An Authentication Manager component made up of services that handle authentication requests, database operations, and connections to the RSA Security Console.
authenticator	A device used to verify a user's identity to Authentication Manager. This can be a hardware token (for example, a key fob) or a software token.
authorization	The process of determining if a user is allowed to perform an operation on a resource.
authorization data	Information defined by the provisioning server, which is necessary to complete the provisioning of a CT-KIP-enabled token. Authorization data includes the appropriate serial number and places the new token credentials in the Authentication Manager internal database.
auto-registration	A setting which, if enabled, permits unregistered users to become registered upon a successful authentication to a system-managed resource. If auto-registration is disabled, only an administrative action can register users. Also see registered user and unregistered user.
Base Server license	Authentication Manager license that allows one primary instance and one replica instance. (Multiple replica instances and server nodes are not allowed.) Includes RSA Credential Manager self-service. Credential Manager provisioning can be added.
Business Continuity option	Authentication Manager option that allows you to temporarily increase the number of users allowed into your system and the number of users allowed to use on-demand authentication.
certificate	An asymmetric public key that corresponds with a private key. It is either self-signed or signed with the private key of another certificate.
certificate DN	The distinguished name of the certificate issued to the user for authentication.
chained authentication	The process of creating a strong form of authentication by combining two weaker forms. For example, the user is required to use a PIN and a tokencode.
client time-out	The amount of time (in seconds) that the user's desktop can be inactive before reauthentication is required.
CLU	See command line utility.

Term	Definition
cluster	An instance consisting of a database server and one or more server nodes.
command line utility (CLU)	A utility that provides a command line user interface.
connection pool	A named group of identical connections to a data store.
contact list	A list of server nodes provided by the Authentication Manager to the agent, to which the agent can direct authentication requests.
context-based authentication	An authentication sequence in which the system presents the user with only the authentication options that are appropriate for the User ID entered. The options are based on policy requirements and the authenticators that the user owns.
core attributes	The fixed set of attributes commonly used by all RSA products to create a user. These attributes are always part of the primary user record, whether the deployment is in an LDAP or RDBMS environment. You cannot exclude core attributes from a view, but they are available for delegation.
Credential Manager Provisioning	An option that automates the token deployment process and provides user self-service options.
cryptographic algorithm	A mathematical function that uses plain text as the input and produces cipher text as the output and vice-versa. It is used for encryption and decryption.
CT-KIP	Cryptographic Token-Key Initialization Protocol.
CT-KIP-capable token	A token that is capable of storing the authorization data and seed generated as a result of CT-KIP operations between a CT-KIP 1.0 client and an Authentication Manager CT-KIP server.
CT-KIP client	A program that implements the CT-KIP client-side protocol and interacts with a CT-KIP server for the secure initialization of CT-KIP-capable tokens.
CT-KIP server	A software component of Authentication Manager that implements the CT-KIP server-side protocol and interacts with a CT-KIP client application for the secure initialization of CT-KIP-capable tokens.
CT-KIP toolkit	An implementation of the CT-KIP client-server protocol. It provides the API for creating CT-KIP server or client applications.

Term	Definition
customer name	The name of the enterprise to which the license is issued.
data encryption standard (DES)	The cryptographic standard prior to November 2001, when the National Institute of Standards and Technology (NIST) adopted the Advanced Encryption Standard (AES).
data store	A data source such as a relational database (Oracle or DB2) or directory server (Sun Java System Directory Server or Microsoft Active Directory). Each type of data source manages and accesses data differently.
data transfer object	Simple object used to pass data between tiers. It does not contain business logic.
database server	The server where the database is installed.
delegated administration	A scheme for defining the scope and responsibilities of a set of administrators. It permits administrators to delegate a portion of their responsibilities to another administrator.
denial of service	The process of making a system or application unavailable. For example, the result of barraging a server with requests that consume all the available system resources, or of passing malformed input data that can cause the system to stop responding.
delivery address	The e-mail address or the cell phone number where the on-demand tokencodes will be delivered.
deployment	The arrangement of Authentication Manager instances into appropriate locations in a network to perform authentication.
DES	See data encryption standard.
distribution file	A shared secret between a hardware or software authenticator and an authentication server. The authenticator, sometimes called a token, and the server work together in a time synchronous, or time dependent mode to provide a one-time passcode that the token holder enters at logon.
distribution file password	A password used to protect the distribution file when the distribution file is sent by e-mail to the user.
distributor	A Token Distributor or an administrator with distributor permissions.
DTO	See data transfer object.

Term	Definition
dump	An RSA ACE/Server format used to back up, restore, and merge database information. A dump file is a binary data file that contains all database tables and columns in table-dependency order.
EAP	See extensible authentication protocol.
EAP-POTP	An RSA-proposed IETF (Internet Engineering Task Force) standard that defines the method for one-time password (RSA SecurID) authentication. It provides capabilities, such as end-to-end protection of one-time passwords and support for token exception cases (New PIN, Next Tokencode, and others).
EAP-POTP client	Client that supports the EAP-POTP method.
e-mail notifications	Contain status information about requests for user enrollment, tokens, and user group membership are sent to users who initiated the request. For token requests, e-mail notifications also contain information about how to download and activate tokens. Request Approvers and Token Distributors receive e-mail notifications about requests that require their action. See e-mail templates.
e-mail templates	Templates that administrators can use to customize e-mail notifications about user requests for user enrollment, tokens, user group membership, or the on-demand tokencode service. See e-mail notifications.
emergency access	The process for enabling a token for a user whose token is not available or is not functioning. Used in connection with offline authentication access.
emergency access passcode	A complete authentication code that, if enabled, can be used by a user to perform an offline authentication without an authenticator or PIN.
emergency access tokencode	A partial authentication code that, if enabled, can be used by a user to perform an offline authentication without an authenticator. The user is required to provide his or her PIN.
Enterprise Server license	Authentication Manager license that allows a primary instance, multiple replica instances, and multiple server nodes.
Evaluation license	Authorizes an evaluation copy of the product at a customer site.
event-based token	A hardware token that displays a tokencode whenever the user presses the button on the token.

Term	Definition
excluded words dictionary	A dictionary containing a record of words that users cannot use as passwords. It includes several thousand commonly used words that are likely to be included as part of any dictionary attacks on the system, for example, "password." The excluded words dictionary prevents users from using common, and therefore, easily guessed words as passwords.
extensible authentication protocol (EAP)	An authentication framework that supports multiple authentication methods.
failover mode	The state in which the connection pool management service has to use the secondary connection pools for serving the connection requests, because the primary connection pools are not available due to the failed primary data servers.
four-pass CT-KIP	The exchange of two protocol data units (PDUs) between the client and server.
Global Catalog	A read-only, replicated repository of a subset of the attributes of all entries in an Active Directory forest.
graded authentication	A mechanism for noting the relative strengths of authentication methods (either individually or as combinations). For example, an RSA SecurID token is stronger than a user name and password. Equivalently ranked methods may be used interchangeably.
group membership	See user group.
hardware token	A physical device, such as an RSA SecurID standard card, key fob, or PINPad that displays a tokencode.
high-water mark	The highest numbered interval used by a user to authenticate.
identity attribute definition	Customer-defined attributes that are mapped to an existing customer-defined schema element. They are always stored in the same physical repository as the user's or user group's core attribute data. You can search, query, and report on these attributes. Each identity attribute definition must map to an existing attribute in the LDAP or RDBMS.
Identity Management Services	The set of shared components, toolkits, and services used to build RSA products, for example, Authentication Manager.
identity source	A data store containing user and user group data. The data store can be the internal database or an external directory server, such as Sun Java System Directory Server or Microsoft Active Directory.

Term	Definition
IMS	See Identity Management Services.
initial time-out	The wait time, in seconds, before the initial remote access prompt appears. (The term is used in relation to remote RSA SecurID authentication.)
instance	One single database server, or a database server and one or more server nodes, acting as a single cohesive processing unit. An instance does not have to be a cluster, but a cluster is an instance.
instance ID	This ID identifies a single logical installation of a product or component. For example, in a non-clustered environment, it identifies the database server. In a clustered environment, it identifies the database server and the entire cluster of server nodes. Likewise for web agents, a single agent may have a unique instance ID or an entire server cluster may share a single instance ID.
instance name	The name assigned to an instance. It is either the hostname where a single server node is installed or the cluster name where the clustered instance is installed.
interval	A value used to represent a specific time-based PRN code being generated by an authenticator.
internal database	The Authentication Manager proprietary data source.
J2EE	See Java 2 Enterprise Edition.
Java 2 Enterprise Edition	A framework for building enterprise applications using Java technology.
Java Cryptographic Architecture (JCA)	The set of APIs provided by the Java 2 platform that establishes the architecture and encapsulates limited cryptographic functionality from various cryptographic providers.
Java Cryptographic Extensions (JCE)	The set of APIs provided by the Java 2 platform that encapsulates additional cryptographic functionality from various cryptographic providers.
Java keystore (JKS)	The Java 2 platform implementation of a keystore provided by Sun Microsystems.
Java Management Extensions (JMX)	The set of APIs provided by the Java 2 platform that enables building distributed, web-based, dynamic, and modular solutions for managing and monitoring devices, applications, and service-driven networks.

Term	Definition
Java Messaging Service (JMS)	A standard Java interface for interacting with message queues and topics.
Java Server Pages (JSP)	A commonly used technology for dynamic web content.
JCA	See Java Cryptographic Architecture.
JCE	See Java Cryptographic Extensions.
JKS	See Java keystore.
JMS	See Java Messaging Service.
JMX	See Java Management Extensions.
JSP	See Java Server Pages.
keystore	The Java 2 platform facility for storing keys and certificates.
Key Management services	The management of the generation, use, storage, security, exchange, and replacement of cryptographic keys.
Key Management encryption key	The key used for encryption or decryption operations of keys managed by Key Management services.
license	A verifiable piece of information that represents permission from RSA to use Authentication Manager, its features, or both. A license is a component of the License Management Service.
license category	A way of grouping different types of licenses. The license categories for Authentication Manager are Base Server, Enterprise Server, and Evaluation.
license creation date	The date when the license file is created.
license deployment	Specifies either a server or floating license.
license file	An XML file containing license data that is common across all IMS-based products. The categories of data are: client, product, and feature. A license file is a component of LMS.
license file version	The version of the license schema to which the generated license conforms.
license ID	An internal identifier associated with the license. RSA Manufacturing assigns the license ID.
License Management Service (LMS)	A service responsible for managing and validating product licenses.

Term	Definition
license.rec	A license record file containing the database key needed to extract critical information from the dump file.
LMS	See License Management Service.
local authentication client component	An RSA Authentication Agent component that requires users to enter valid RSA SecurID passcodes to access their Microsoft Windows desktops.
locked license	A license limited to a specific server instance. See server license.
lockout policy	A set of conditions specifying when an account will be locked and whether the account must be unlocked by an administrator or will unlock on its own after a designated amount of time. Lockout policies are applied to security domains. Each realm has a default lockout policy.
log archival	Creates a backup copy of the log for noncurrent, permanent storage.
logging service	A component responsible for recording system, audit, and trace events.
lower-level security domain	In a security domain hierarchy, a security domain that is nested within another security domain.
Management Information Base (MIB)	A type of virtual database used to manage the devices (switches and routers, for example) in a communication network. For example, SNMP uses MIB to specify the data in a device subsystem.
MD5	An algorithm that produces a 128-bit message digest.
member user	A user who is a member of a member user group.
member user group	A user group that is a member of another user group. For example, an organization might define a Sales Managers user group within a North America user group. All member user groups must belong to the same identity source as the parent group, with one exception: any user group from any identity source can be assigned to a parent group that is stored in the internal database.
MIB	See Management Information Base.
Microsoft Management Console (MMC)	A user interface through which system administrators can configure and monitor the system.
MMC	See Microsoft Management Console.

Term	Definition
namespace	A set of names. A namespace defines a scope for a collection of names.
Network Management System (NMS)	Software used to manage and administer a network. The NMS uses SNMP to monitor networked devices and is responsible for polling and receiving SNMP traps from agents in the network.
NMS	See Network Management System.
NMS administrator	The person monitoring the network (through the NMS) for significant events. Also known as a network administrator.
node secret	A long-lived symmetric key that the agent uses to encrypt the data in the authentication request. Authentication Manager generates the authentication request when a user makes a successful authentication attempt. The node secret is known only to the Authentication Manager and the agent.
offline emergency tokencode	Provides emergency access for RSA SecurID for Windows users who require emergency access while authenticating offline. Use this option if the user has a temporarily misplaced, lost, or stolen token. The Offline Emergency Access Tokencode is used with the user's PIN.
offline emergency passcode	Provides emergency access for RSA SecurID for Windows users who require emergency access while authenticating offline. Use this option if the user has forgotten his or her PIN. The Offline Emergency Passcode is used in place of the user's PIN and tokencode.
object	Describes the following: security domains, identity sources, attributes, users, user groups, administrative roles, and policies.
offset	A value used to represent the amount of time an authenticator's internal clock has drifted over time.
on-demand tokencode	Tokencodes delivered by SMS or SMTP. They require the user to enter a PIN to achieve two-factor authentication. On-demand tokencodes are user-initiated, as Authentication Manager only sends a tokencode to the user when it receives a user request. An on-demand tokencode can only be used once, and you configure the lifetime of an on-demand tokencode. See on-demand tokencode service.

Term	Definition
on-demand tokencode service	A service that allows users to request on-demand tokencodes delivered by text message or e-mail, instead of tokens. You configure the on-demand tokencode service for requests using the Security Console. Users must be enabled to receive on-demand tokencodes before they can request them.
one-time tokencode set	Used for online emergency access. A set of tokencodes, each of which can be used only once, and is used with the user's PIN to create a passcode. The administrator can specify how many tokencodes are in the set.
PAM	See Pluggable Authentication Modules.
passcode	A code entered by a user to authenticate. The passcode is a combination of a PIN and a tokencode.
password-based encryption	The process of obscuring information so that it is unreadable without knowledge of the password.
password policy	A set of specifications that define what constitutes a valid password and the conditions under which the password expires. Password policies are applied to security domains.
PDU	See Protocol Data Unit.
permissions	Specifies which tasks an administrator is allowed to perform.
Pluggable Authentication Modules (PAM)	Mechanisms that allow the integration of new authentication methods into an API, independent of the existing API authentication scheme.
primary connection pool	Refers to the connection pools containing the connections to the primary instance database server.
primary instance	The machine with the installation of Authentication Manager at which authentication and all administrative actions occur.
private key	In asymmetric key cryptography, the cryptographic key that corresponds to the public key. The private key is usually protected by some external mechanism (for example, smart card, password encrypted, and so on).
PRN	See pseudorandom number.
Protocol Data Unit	A packet of data exchanged between two application programs across a network.
provisioning	See token provisioning.

Term	Definition
provisioning data	The provisioning server-defined data. This is a container of information necessary to complete the provisioning of a token device. Its format is not specified by CT-KIP because it is outside the realm of CT-KIP, but it is necessary for provisioning.
pseudorandom number (PRN)	A random number or sequence of numbers derived from a single seed value.
public key	In asymmetric key cryptography, the cryptographic key that corresponds with the private key. The public key is usually encapsulated within a certificate.
RADIUS	See Remote Authentication Dial-In User Service.
realm	An entire security domain hierarchy consisting of a top-level security domain and all of its lower-level security domains. A realm includes all of the objects managed within the security domain hierarchy (users, tokens, and password policies, for example). Each realm manages users and user groups in one or more identity sources.
regular time-out	The number of seconds before remote access prompts time out. The term is used in relation to remote RSA SecurID authentication.
Remote Authentication Dial-In User Service (RADIUS)	A UDP-based protocol for administering and securing remote access to a network.
remote EAP (extensible authentication protocol)	A remote authentication feature that requires users to submit RSA SecurID passcodes in order to open remote connections to the network. EAP has a graphical user interface and enhanced security and is supported in both Point-to-Point Protocol (PPP) authentication environments and non-PPP authentication environments, including Point-to-Point Tunneling Protocol (PPTP) VPN connections, 802.1x wired, and 802.11 wireless connections, and other specialized network media.
remote post-dial	Refers to the dial-in Point-to-Point Protocol (PPP) authentication support. With a post-dial terminal-based connection, when remote users dial in, a terminal-like character interface presents a simple user name and passcode prompt. If the right passcode is entered, the PPP connection is established. If the wrong passcode is entered, the dial-up connection is severed.

Term	Definition
replica instance	The machine with the installation of Authentication Manager at which authentication occurs and at which an administrator can view the administrative data. No administrative actions are performed on the replica instance. All administrative actions are performed on the primary instance.
requests	Allows users to enroll, as well as request tokens, the on-demand tokencode service, and user group membership.
Request Approver	A predefined administrative role that grants permission to approve requests from users for user enrollment, tokens, or user group membership.
RSA Credential Manager	A component of Authentication Manager that allows users to request, maintain, and troubleshoot tokens.
RSA EAP	The RSA Security implementation of the EAP 15 authentication protocol that facilitates RSA SecurID authentication to networks in PPP, PPTP (VPN), and 802.1x (wireless or port access) environments.
RSA Operations Console	An administrative user interface through which the user configures and sets up Authentication Manager, for example, adding and managing identity sources, adding and managing instances, and disaster recovery.
RSA Protected OTP	The RSA implementation of the EAP 32 authentication protocol that facilitates RSA SecurID authentication to networks in PPP, PPTP (VPN), and 802.1x (wireless or port access) environments.
RSA Security Console	An administrative user interface through which the user performs most of the day-to-day administrative activities.
RSA Self-Service Console	A user interface through which the user requests, maintains, and troubleshoots tokens.
runtime	Describes automated processing behavior—behavior that occurs without direct administrator interaction.
runtime command	A logon or logoff command.
runtime identity source	The runtime representation of the identity source. Runtime identity sources are used during runtime operations, such as authentication and group membership resolution instead of the corresponding administrative source, which is used for all other operations. This is an integral part of Active Directory forest support, which uses the Global Catalog during runtime operations.

Term	Definition
scope	In a realm, the security domain or domains within which a role's permissions apply.
secondary connection pool	The connection pools containing the connections to the secondary data stores.
Secure Sockets Layer (SSL)	A protocol that uses cryptography to enable secure communication over the Internet. SSL is widely supported by leading web browsers and web servers.
security domain	A container that defines an area of administrative management responsibility, typically in terms of business units, departments, partners, and so on. Security domains establish ownership and namespaces for objects (users, roles, permissions, and so on) within the system. They are hierarchical.
security questions	A way of allowing users to authenticate without using their standard method. To use this service, a user must answer a number of security questions. To authenticate using this service, the user must correctly answer all or a subset of the original questions. The answers to security questions are case sensitive.
self-service	Allows users to perform maintenance tasks and troubleshoot tokens themselves, instead of calling the Help Desk. See also Token Provisioning.
Self-Service Console	See RSA Self-Service Console.
self-service requests	See requests.
self-service troubleshooting policy	Provides an emergency form of authentication that allows users to log on to the RSA Self-Service Console to perform troubleshooting tasks.
server node	An installation of Authentication Manager on a single server host. Each instance has one server node that contains the internal database. You can add additional server nodes to an instance, if your license allows. The additional server nodes cannot operate alone because they do not contain the internal database.
session	An encounter between a user and a software application that contains data pertaining to the user's interaction with the application. A session begins when the user logs on to the software application and ends when the user logs off of the software application.

Term	Definition
session policy	A set of specifications designating the restrictions on overall session lifetime and multiple session handling. Session policies are applied to an instance.
SHA1	A secure hash algorithm function that produces a 160-bit hash result.
shipping address	An address used by distributors to distribute hardware tokens.
Short Message Service (SMS)	A mechanism of delivery of short messages over mobile networks. It is often called text messaging. In Authentication Manager, it is a means of sending tokencodes to a cell phone. Tokencodes delivered by SMS are called on-demand tokencodes.
Simple Mail Transfer Protocol (SMTP)	A TCP/IP protocol used in sending and receiving e-mail. In Authentication Manager, it is a means of sending tokencodes to e-mail accounts. Tokencodes delivered by SMTP are called on-demand tokencodes.
Simple Network Management Protocol (SNMP)	A protocol for exchanging information about networked devices and processes. SNMP uses MIBs to specify the management data, and then uses the User Datagram Protocol (UDP) to pass the data between SNMP management stations and the SNMP agents.
single sign-on (SSO)	The process of requiring only a single user authentication event in order to access multiple applications and resources.
SMS	See Short Message Service.
SMTP	See Simple Mail Transfer Protocol.
snap-in	A software program designed to function as a modular component of another software application. For example, the MMC has a variety of snap-ins that offer different functionality (for example, Device Manager).
SNMP	See Simple Network Management Protocol.
SNMP agent	Software module that performs the network management functions requested by network management stations.
SNMP trap	An asynchronous event that is generated by the agent to tell the NMS that a significant event has occurred. SNMP traps are designed to capture errors and reveal their locations.
SSL	See Secure Sockets Layer.
SSO	See single sign-on.

Term	Definition
Super Admin	An administrator who has all permissions within the system. A Super Admin: • Can create and delete realms • Can link identity sources to realms • Has full permissions within any realm • Can assign administrative roles within any realm
symmetric key	A key that allows the same key value for the encryption and decryption of data.
system event	System-generated information related to nonfunctional system events such as server startup and shutdown, failover events, replication events, and so on.
system log	Persistable store for recording system events.
TACACS+	See Terminal Access Controller Access Control System+.
temporary fixed tokencode	Used for online emergency access. This temporary tokencode is used in conjunction with the user's PIN to create a passcode. The user can use this tokencode more than once. The administrator can configure the expiration date and other Temporary Fixed Tokencode attributes.
Terminal Access Controller Access Control System+ (TACACS+)	A remote authentication protocol that is used to communicate with an authentication server. Allows a remote access server to communicate with an authentication server to determine if a user has access to the network.
time-based token	A hardware token that always displays a tokencode and the tokencode changes automatically every 60 seconds.
token	A hardware device or software program that generates a pseudorandom number that is used in authentication procedures to verify a user's identity.
Token Distributor	A predefined administrative role that grants permission to act upon requests from users for tokens. Distributors record how they plan to deliver tokens to users and close requests.
token provisioning	The automation of all the steps required to provide enrollment, user group membership, RSA SecurID tokens, and the on-demand tokencode service to users. See also self-service.
tokencode	The random number displayed on the front of a user's RSA SecurID token. Tokencodes change at a specified time interval, typically every 60 seconds.

Term	Definition
top-level security domain	The top-level security domain is the first security domain in the security domain hierarchy (realm). The top-level security domain is unique in that it links to the identity source or sources and manages password, locking, and authentication policy for the entire realm.
trace log	Persistable store for trace information.
trusted realm	A trusted realm is a realm that meets these criteria: • It is located in a different deployment than your realm. • It has exchanged configuration settings with your realm. The settings are in an XML file called a trust package.
trust package	An XML file that contains configuration information about the realm.
two-factor authentication	An authentication protocol requiring two different ways of establishing and proving identity, for example, something you have (such as an authenticator) and something you know (such as a PIN).
two-pass CT-KIP	The exchange of one protocol data unit (PDU) between the client and server.
UDP	See User Datagram Protocol.
user	An account managed by the system that is usually a person, but may be a computer or a web service.
User Datagram Protocol (UDP)	A protocol that allows programs on networked computers to communicate with one another by sending short messages called datagrams.
user group	A collection of users, other user groups, or both. Members of the user group must belong to the same identity source. User group membership determines access permission in some applications.
User ID	A character string that the system uses to identify a user attempting to authenticate. Typically a User ID is the user's first initial followed by the last name. For example, Jane Doe's User ID might be *jdoe*.
workflow	The movement of information or tasks through a work or business process. A workflow can consist of one or two approval steps and a distribution step for different requests from users.

Term	Definition
workflow participant	Either approvers or distributors. Approvers review, approve, or defer user requests. Distributors determine the distribution method for token requests and record the method for each request. See also workflow.