

Configuring Secure Shell (SSH)

Contents

Contents	6-1
Overview	6-2
Terminology	6-4
Prerequisite for Using SSH	6-5
Public Key Formats	6-5
Steps for Configuring and Using SSH for Switch and Client Authentication	6-6
General Operating Rules and Notes	6-8
Configuring the Switch for SSH Operation	6-9
1. Assign Local Login (Operator) and Enable (Manager) Password .	6-9
2. Generate the Switch's Public and Private Key Pair	6-10
3. Provide the Switch's Public Key to Clients	6-12
4. Enable SSH on the Switch and Anticipate SSH Client Contact Behavior	6-15
5. Configure the Switch for SSH Authentication	6-18
6. Use an SSH Client To Access the Switch	6-21
Further Information on SSH Client Public-Key Authentication	6-21
Messages Related to SSH Operation	6-27

Overview

Feature	Default	Menu	CLI	Web
Generating a public/private key pair on the switch	No	n/a	page 6-10	n/a
Using the switch's public key	n/a	n/a	page 6-12	n/a
Enabling SSH	Disabled	n/a	page 6-15	n/a
Enabling client public-key authentication	Disabled	n/a	pages 6-19, 6-21	n/a
Enabling user authentication	Disabled	n/a	page 6-18	n/a

The ProCurve switches covered in this guide use Secure Shell version 1 or 2 (SSHv1 or SSHv2) to provide remote access to management functions on the switches via encrypted paths between the switch and management station clients capable of SSH operation.

SSH provides Telnet-like functions but, unlike Telnet, SSH provides encrypted, authenticated transactions. The authentication types include:

- Client public-key authentication
- Switch SSH and user password authentication

Client Public Key Authentication (Login/Operator Level) with User Password Authentication (Enable/Manager Level). This option uses one or more public keys (from clients) that must be stored on the switch. Only a client with a private key that matches a stored public key can gain access to the switch. (The same private key can be stored on one or more clients.)

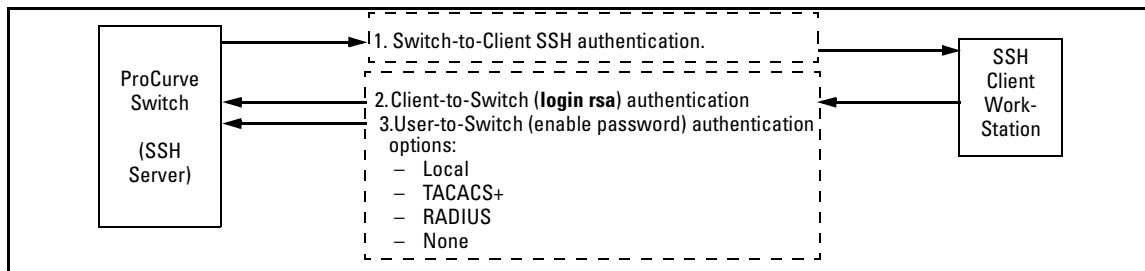


Figure 6-1. Client Public Key Authentication Model

Note

SSH in the ProCurve is based on the OpenSSH software toolkit. For more information on OpenSSH, visit <http://www.openssh.com>.

Switch SSH and User Password Authentication . This option is a subset of the client public-key authentication show in figure 6-1. It occurs if the switch has SSH enabled but does not have login access (**login public-key**) configured to authenticate the client's key. As in figure 6-1, the switch authenticates itself to SSH clients. Users on SSH clients then authenticate themselves to the switch (login and/or enable levels) by providing passwords stored locally on the switch or on a TACACS+ or RADIUS server. However, the client does not use a key to authenticate itself to the switch.

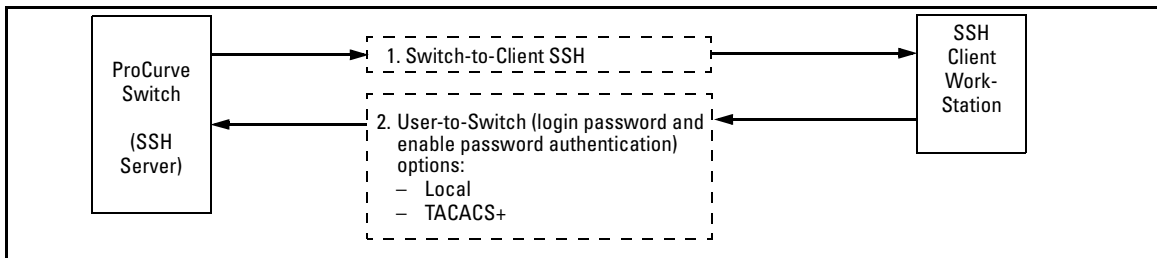


Figure 6-2. Switch/User Authentication

SSH on the ProCurve switches covered in this guide supports these data encryption methods:

- 3DES (168-bit)
- DES (56-bit)

Note

The ProCurve switches covered in this guide use the RSA algorithm for internally generated keys (v1/v2 shared host key & v1 server key). However, ProCurve switches support both RSA and DSA/DSS keys for client authentication. All references to either a public or private key mean keys generated using these algorithms unless otherwise noted

Terminology

- **SSH Server:** A ProCurve switch with SSH enabled.
- **Key Pair:** A pair of keys generated by the switch or an SSH client application. Each pair includes a public key, that can be read by anyone and a private key, that is held internally in the switch or by a client.
- **PEM (Privacy Enhanced Mode):** Refers to an ASCII-formatted client public-key that has been encoded for portability and efficiency. SSHv2 client public-keys are typically stored in the PEM format. See figures 6-3 and 6-4 for examples of PEM-encoded ASCII and non-encoded ASCII keys.
- **Private Key:** An internally generated key used in the authentication process. A private key generated by the switch is not accessible for viewing or copying. A private key generated by an SSH client application is typically stored in a file on the client device and, together with its public key counterpart, can be copied and stored on multiple devices.
- **Public Key:** An internally generated counterpart to a private key. A device's public key is used to authenticate the device to other devices.
- **Enable Level:** Manager privileges on the switch.
- **Login Level:** Operator privileges on the switch.
- **Local password or username:** A Manager-level or Operator-level password configured in the switch.
- **SSH Enabled:** (1) A public/private key pair has been generated on the switch (**crypto key generate ssh [rsa]**) and (2) SSH is enabled (**ip ssh**). (You can generate a key pair without enabling SSH, but you cannot enable SSH without first generating a key pair. See “2. Generate the Switch's Public and Private Key Pair” on page 6-10 and “4. Enable SSH on the Switch and Anticipate SSH Client Contact Behavior” on page 6-15.)

Prerequisite for Using SSH

Before using the switch as an SSH server, you must install a publicly or commercially available SSH client application on the computer(s) you use for management access to the switch. If you want client public-key authentication (page 6-2), then the client program must have the capability to generate or import keys.

Public Key Formats

Any client application you use for client public-key authentication with the switch must have the capability export public keys. The switch can accept keys in the PEM-Encoded ASCII Format or in the Non-Encoded ASCII format.

```
"Pub Key Gen 21 Dec 2001 12:01"A1B3Hz1y2+orEHYL . . . Q8D8qDM1ozu1c="*** End of Pub Key ***"
```

Comment describing public key identity.

Beginning of actual SSHv2 public key in PEM-Encoded ASCII format.

Figure 6-3. Example of Public Key in PEM-Encoded ASCII Format Common for SSHv2 Clients

```
512 37 78193303392019545793321845914508115859448079486918367079008218589443776362026267. . .
```

Bit Size

Exponent <e>

Modulus <n>

Figure 6-4. Example of Public Key in Non-Encoded ASCII Format (Common for SSHv1 Client Applications)

Steps for Configuring and Using SSH for Switch and Client Authentication

For two-way authentication between the switch and an SSH client, you must use the login (Operator) level.

Table 6-1. SSH Options

Switch Access Level	Primary SSH Authentication	Authenticate Switch Public Key to SSH Clients?	Authenticate Client Public Key to the Switch?	Primary Switch Password Authentication	Secondary Switch Password Authentication
Operator (Login) Level	ssh login rsa	Yes	Yes¹	No¹	local or none
	ssh login Local	Yes	No	Yes	local or none
	ssh login TACACS	Yes	No	Yes	local or none
	ssh login RADIUS	Yes	No	Yes	local or none
Manager (Enable) Level	ssh enable local	Yes	No	Yes	local or none
	ssh enable tacacs	Yes	No	Yes	local or none
	ssh enable radius	Yes	No	Yes	local or none

¹ For **ssh login public-key**, the switch uses client public-key authentication instead of the switch password options for primary authentication.

The general steps for configuring SSH include:

A. Client Preparation

1. Install an SSH client application on a management station you want to use for access to the switch. (Refer to the documentation provided with your SSH client application.)
2. Optional—If you want the switch to authenticate a client public-key on the client:
 - a. Either generate a public/private key pair on the client computer (if your client application allows) or import a client key pair that you have generated using another SSH application.
 - b. Copy the client public key into an ASCII file on a TFTP server accessible to the switch and download the client public key file to the switch. (The client public key file can hold up to ten client keys.) This topic is covered under “To Create a Client-Public-Key Text File” on page 6-23.

B. Switch Preparation

1. Assign a login (Operator) and enable (Manager) password on the switch (page 6-9).
2. Generate a public/private key pair on the switch (page 6-10).

You need to do this only once. The key remains in the switch even if you reset the switch to its factory-default configuration. (You can remove or replace this key pair, if necessary.)
3. Copy the switch's public key to the SSH clients you want to access the switch (page 6-12).
4. Enable SSH on the switch (page 6-15).
5. Configure the primary and secondary authentication methods you want the switch to use. In all cases, the switch will use its host-public-key to authenticate itself when initiating an SSH session with a client.
 - SSH Login (Operator) options:
 - Option A:

Primary: Local, TACACS+, or RADIUS password
Secondary: Local password or none
 - Option B:

Primary: Client public-key authentication (**login public-key** — page 6-21)
Secondary: Local password or none

Note that if you want the switch to perform client public-key authentication, you must configure the switch with Option B.
 - SSH Enable (Manager) options:

Primary: Local, TACACS+, or RADIUS
Secondary: Local password or none
6. Use your SSH client to access the switch using the switch's IP address or DNS name (if allowed by your SSH client application). Refer to the documentation provided with the client application.

General Operating Rules and Notes

- Public keys generated on an SSH client must be exportable to the switch. The switch can only store ten keys client key pairs.
- The switch's own public/private key pair and the (optional) client public key file are stored in the switch's flash memory and are not affected by reboots or the **erase startup-config** command.
- Once you generate a key pair on the switch you should avoid re-generating the key pair without a compelling reason. Otherwise, you will have to re-introduce the switch's public key on all management stations (clients) you previously set up for SSH access to the switch. In some situations this can temporarily allow security breaches.
- On ProCurve switches that support stacking, when stacking is enabled, SSH provides security only between an SSH client and the stack manager. Communications between the stack commander and stack members is not secure.
- The switch does not support outbound SSH sessions. Thus, if you Telnet from an SSH-secure switch to another SSH-secure switch, *the session is not secure.*

Configuring the Switch for SSH Operation

SSH-Related Commands in This Section	Page
show ip ssh	6-17
show crypto client-public-key [keylist-str] [< babble fingerprint >]	6-24
show crypto host-public-key [< babble fingerprint >]	6-14
show authentication	6-21
crypto key < generate zeroize > ssh [rsa]	6-11
ip ssh	6-16
key-size < 512 768 1024 >	6-16
port < 1 - 65535 default >	6-16
timeout < 5 - 120 >	6-16
version < 1 2 1-or-2 >	6-16
aaa authentication ssh	
login < local tacacs radius public-key >	6-18, 6-20
< local none >	6-18
enable < tacacs radius local >	6-18
< local none >	6-18
copy tftp pub-key-file <tftp server IP> <public key file>	6-24
clear crypto client-public-key [keylist-str]	6-25

1. Assign Local Login (Operator) and Enable (Manager) Password

At a minimum, ProCurve recommends that you always assign at least a Manager password to the switch. Otherwise, under some circumstances, anyone with Telnet, web, or serial port access could modify the switch's configuration.

To Configure Local Passwords. You can configure both the Operator and Manager password with one command.

Syntax: password < manager | operator | all >

```
ProCurve(config)# password all
New password for Operator: *****
Please retype new password for Operator: *****
New password for Manager: *****
Please retype new password for Manager: *****
ProCurve(config)#
```

Figure 6-5. Example of Configuring Local Passwords

2. Generate the Switch's Public and Private Key Pair

You must generate a public and private host key pair on the switch. The switch uses this key pair, along with a dynamically generated session key pair to negotiate an encryption method and session with an SSH client trying to connect to the switch.

The host key pair is stored in the switch's flash memory, and only the public key in this pair is readable. The public key should be added to a "known hosts" file (for example, `$HOME/.ssh/known_hosts` on UNIX systems) on the SSH clients which should have access to the switch. Some SSH client applications automatically add the switch's public key to a "known hosts" file. Other SSH applications require you to manually create a known hosts file and place the switch's public key in the file. (Refer to the documentation for your SSH client application.)

(The session key pair mentioned above is not visible on the switch. It is a temporary, internally generated pair used for a particular switch/client session, and then discarded.)

Notes

When you generate a host key pair on the switch, the switch places the key pair in flash memory (and not in the running-config file). Also, the switch maintains the key pair across reboots, including power cycles. You should consider this key pair to be "permanent"; that is, avoid re-generating the key pair without a compelling reason. Otherwise, you will have to re-introduce the switch's public key on all management stations you have set up for SSH access to the switch using the earlier pair.

Removing (zeroing) the switch's public/private key pair renders the switch unable to engage in SSH operation and automatically disables IP SSH on the switch. (To verify whether SSH is enabled, execute **show ip ssh**.) However, any active SSH sessions will continue to run, unless explicitly terminated with the CLI **kill** command.

To Generate or Erase the Switch's Public/Private RSA Host Key Pair.

Because the host key pair is stored in flash instead of the running-config file, it is not necessary to use **write memory** to save the key pair. Erasing the key pair automatically disables SSH.

Syntax: `crypto key generate ssh [rsa]`

Generates a public/private key pair for the switch. If a switch key pair already exists, replaces it with a new key pair. (See the Note, above.)

`crypto key zeroize ssh [rsa]`

Erases the switch's public/private key pair and disables SSH operation.

`show crypto host-public-key`

Displays switch's public key. Displays the version 1 and version 2 views of the key.

[`babble`]

Displays hashes of the switch's public key in phonetic format. (See "Displaying the Public Key" on page 6-14.)

[`fingerprint`]

Displays fingerprints of the switch's public key in hexadecimal format. (See "Displaying the Public Key" on page 6-14.)

For example, to generate and display a new key:

```
ProCurve(config)# crypto key generate ssh rsa
Installing new RSA key.  If the key/entropy cache is
depleted, this could take up to a minute.
ProCurve(config)# show crypto host-public-key

-----
SSH host public key file
Version 1 format:

|896 35 3219295003103011452137203169501232714847265325085720757925409572738582167|
|49173126937413223781326827636154399173519641900117298772018339016754333892248319|
|41759125186557710233731689070801858880718460531164552600040416069890120011153581|
|9449254242176260739141950918771764467
Version 2 format:

ssh-rsa AAAAB3NzaClyc2EAAAABAwAAAEAnAAApdhq13Jynrs7j4lDUm8ivVm8ld2mZU5e+YZWp/T6|
QzP2RsDDMZLbAHHIBrxPLjW/bRogpYD0lWuVOhTojEVjqeVuXbwmdDnyOgBc06olePwdrbQ+FZevERiA|
JYG3C8NCzCRD/djXeI7FmRps8w==
-----
```

Figure 6-6. Example of Generating a Public/Private Host Key Pair for the Switch

The 'show crypto host-public-key' displays data in two different formats because your client may store it in either of these formats after learning the key. If you wish to compare the switch key to the key as stored in your client's known-hosts file, note that the formatting and comments need not match. For version 1 keys, the three numeric values bit size, exponent <e>, and modulus <n> must match; for PEM keys, only the PEM-encoded string itself must match.

Notes

"Zeroizing" the switch's key automatically disables SSH (sets **ip ssh** to **no**). Thus, if you zeroize the key and then generate a new key, you must also re-enable SSH with the **ip ssh** command before the switch can resume SSH operation.

3. Provide the Switch's Public Key to Clients

When an SSH client contacts the switch for the first time, the client will challenge the connection unless you have already copied the key into the client's "known host" file. Copying the switch's key in this way reduces the chance that an unauthorized device can pose as the switch to learn your access passwords. The most secure way to acquire the switch's public key for

distribution to clients is to use a direct, serial connection between the switch and a management device (laptop, PC, or UNIX workstation), as described below.

The public key generated by the switch consists of three parts, separated by one blank space each:

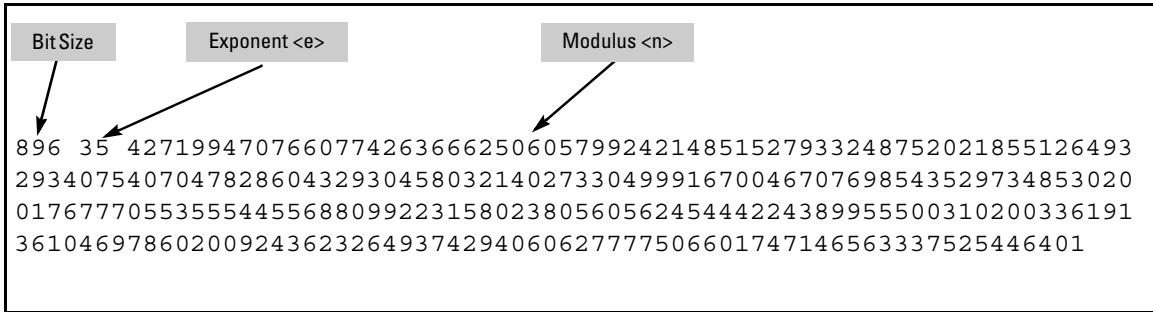


Figure 6-7. Example of a Public Key Generated by the Switch

(The generated public key on the switch is always 896 bits.)

With a direct serial connection from a management station to the switch:

1. Use a terminal application such as HyperTerminal to display the switch's public key with the **show crypto host-public-key** command (figure 6-6).
2. Bring up the SSH client's "known host" file in a text editor such as Notepad as straight ASCII text, and copy the switch's public key into the file.
3. Ensure that there are no changes in breaks in the text string. (A public key must be an unbroken ASCII string. Line breaks are not allowed. Changes in the line breaks will corrupt the Key.) For example, if you are using Windows® Notepad, ensure that **Word Wrap** (in the **Edit** menu) is disabled, and that the key text appears on a single line.

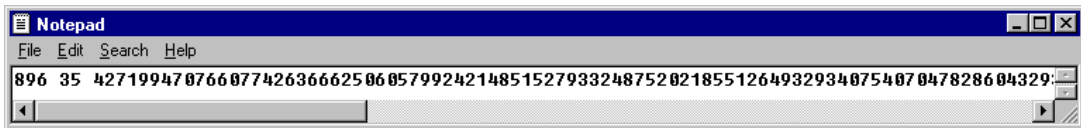


Figure 6-8. Example of a Correctly Formatted Public Key

4. Add any data required by your SSH client application. For example Before saving the key to an SSH client's "known hosts" file you may have to insert the switch's IP address:

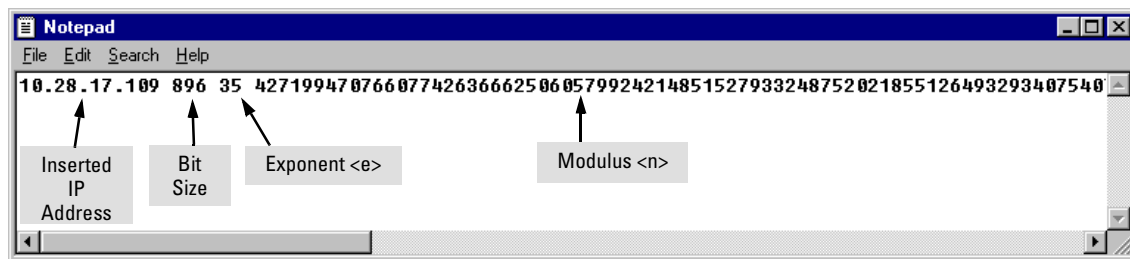


Figure 6-9. Example of a Switch Public Key Edited To Include the Switch's IP Address

For more on this topic, refer to the documentation provided with your SSH client application.

Displaying the Public Key. The switch provides three options for displaying its public key. This is helpful if you need to visually verify that the public key the switch is using for authenticating itself to a client matches the copy of this key in the client's "known hosts" file:

- **Non-encoded ASCII numeric string:** Requires a client ability to display the keys in the "known hosts" file in the ASCII format. This method is tedious and error-prone due to the length of the keys. (See figure 6-8 on page 6-13.)
- **Phonetic hash:** Outputs the key as a relatively short series of alphabetic character groups. Requires a client ability to convert the key to this format.
- **Hexadecimal hash:** Outputs the key as a relatively short series of hexadecimal numbers. Requires a parallel client ability.

For example, on the switch, you would generate the phonetic and hexadecimal versions of the switch's public key in figure 6-8 as follows:

```
ProCurve(config)# show crypto host-public-key babble
896 xozik-kobaf-daroh-fygas-byveb-bymiz-nupap-povaz-cesin-kafec-rixux
   host_sshl
896 xefes-hikot-kyher-cukuz-balah-gezos-gumym-rezif-horib-cicyp-poxyx
   host_ssh2.pub
ProCurve(config)# show crypto host-public-key fingerprint
896 53:c0:14:75:72:84:90:cc:c8:ba:5e:ca:92:fc:c7:5c host_sshl
896 bf:fb:8a:d0:10:5a:48:57:61:f9:8a:6a:61:13:8a:fb host_ssh2.pub
```

Phonetic "Hash" of Switch's Public Key

Hexadecimal "Fingerprints" of the Same Switch

Figure 6-10. Examples of Visual Phonetic and Hexadecimal Conversions of the Switch's Public Key

The two commands shown in figure 6-10 convert the displayed format of the switch's (host) public key for easier visual comparison of the switch's public key to a copy of the key in a client's "known host" file. The switch has only one RSA host key. The 'babble' and 'fingerprint' options produce two hashes for the key—one that corresponds to the challenge hash you will see if connecting with a v1 client, and the other corresponding to the hash you will see if connecting with a v2 client. These hashes do not correspond to different keys, but differ only because of the way v1 and v2 clients compute the hash of the same RSA key. The switch always uses ASCII version (without babble or fingerprint conversion) of its public key for file storage and default display format.

4. Enable SSH on the Switch and Anticipate SSH Client Contact Behavior

The **ip ssh** command enables or disables SSH on the switch and modifies parameters the switch uses for transactions with clients. After you enable SSH, the switch can authenticate itself to SSH clients.

Note

Before enabling SSH on the switch you must generate the switch's public/private key pair. If you have not already done so, refer to "2. Generate the Switch's Public and Private Key Pair" on page 6-10.

When configured for SSH, the switch uses its host public-key to authenticate itself to SSH clients. If you also want SSH clients to authenticate themselves to the switch you must configure SSH on the switch for client public-key authentication at the login (Operator) level. To enhance security, you should also configure local, TACACS+, or RADIUS authentication at the enable (Manager) level.

Refer to "5. Configure the Switch for SSH Authentication" on page 6-18.

SSH Client Contact Behavior. At the first contact between the switch and an SSH client, if you have not copied the switch's public key into the client, your client's first connection to the switch will question the connection and, for security reasons, give you the option of accepting or refusing. As long as you are confident that an unauthorized device is not using the switch's IP address in an attempt to gain access to your data or network, you can accept the connection. (As a more secure alternative, you can directly connect the client to the switch's serial port and copy the switch's public key into the client. See the following Note.)

Note

When an SSH client connects to the switch for the first time, it is possible for a "man-in-the-middle" attack; that is, for an unauthorized device to pose undetected as the switch, and learn the usernames and passwords controlling access to the switch. You can remove this possibility by directly connecting the management station to the switch's serial port, using a **show** command to display the switch's public key, and copying the key from the display into a file. This requires a knowledge of where your client stores public keys, plus the knowledge of what key editing and file format might be required by your client application. However, if your first contact attempt between a client and the switch does not pose a security problem, this is unnecessary.

To enable SSH on the switch.

1. Generate a public/private key pair if you have not already done so. (Refer to "2. Generate the Switch's Public and Private Key Pair" on page 6-10.)
2. Execute the **ip ssh** command.

To disable SSH on the switch, do either of the following:

- Execute **no ip ssh**.
- Zeroize the switch's existing key pair. (page 6-11).

Syntax: [no] ip ssh

Enables or disables SSH on the switch.

[key-size < 512 | 768 | 1024 >] Version 1 only

The size of the internal, automatically generated key the switch uses for negotiations with an SSH client. A larger key provides greater security; a smaller key results in faster authentication (default: 512 bits).

[port < 1-65535 | default >]

*The TCP port number for SSH connections (default: 22). **Important:** See “Note on Port Number” on page 6-17.*

[timeout < 5 - 120 >]

The SSH login timeout value (default: 120 seconds).

[version < 1 | 2 | 1-or-2 >]

The version of SSH to accept connections from. (default: 1-or-2)

The **ip ssh key-size** command affects only a per-session, internal server key the switch creates, uses, and discards. This key is not accessible from the user interface. The switch’s public (host) key is a separate, accessible key that is always 896 bits.

Note on Port Number

ProCurve recommends using the default TCP port number (22). However, you can use **ip ssh port** to specify any TCP port for SSH connections except those reserved for other purposes. Examples of reserved IP ports are 23 (Telnet) and 80 (http). Some other reserved TCP ports on the ProCurve switches are 49, 80, 1506, and 1513.

```
ProCurve(config)# ip ssh
ProCurve(config)# show ip ssh
```

SSH Enabled	: Yes
SSH Version	: 1-or-2
IP Port Number	: 22
Timeout (sec)	: 120
Server Key Size (bits)	: 512

```
Ses Type      | Protocol  Source IP and Port
-----+-----
1  console   |
2  telnet    |
3  ssh       | SSH v2    12.255.255.255:18
4  inactive  |
```

Enables SSH on the switch.

Lists the current SSH configuration and status.

The switch uses these five settings internally for transactions with clients. See the **Caution** on page 6-18.

With SSH running, the switch allows one console session and up to three other sessions (SSH and/or Telnet). Web browser sessions are also allowed, but do not appear in the **show ip ssh**

Figure 6-11. Example of Enabling IP SSH and Listing the SSH Configuration and Status

Caution

Protect your private key file from access by anyone other than yourself. If someone can access your private key file, they can then penetrate SSH security on the switch by appearing to be you.

SSH does not protect the switch from unauthorized access via the web interface, Telnet, SNMP, or the serial port. While web and Telnet access can be restricted by the use of passwords local to the switch, if you are unsure of the security this provides, you may want to disable web-based and/or Telnet access (**no web-management** and **no telnet**). If you need to increase SNMP security, you should use SNMP version 3 only. If you need to increase the security of your web interface, refer to chapter 7, “Configuring Secure Socket Layer (SSL)”. Another security measure is to use the Authorized IP Managers feature described in the switch’s *Management and Configuration Guide*. To protect against unauthorized access to the serial port (and the Clear button, which removes local password protection), keep physical access to the switch restricted to authorized personnel.

5. Configure the Switch for SSH Authentication

Note that all methods in this section result in authentication of the switch’s public key by an SSH client. However, only Option B (page 6-19) results in the switch also authenticating the client’s public key. Also, for a more detailed discussion of the topics in this section, refer to “Further Information on SSH Client Public-Key Authentication” on page 6-21

Note

ProCurve recommends that you always assign a Manager-Level (enable) password to the switch. Without this level of protection, any user with Telnet, web, or serial port access to the switch can change the switch’s configuration. *Also, if you configure only an Operator password, entering the Operator password through telnet, web, SSH or serial port access enables full manager privileges.* See “1. Assign Local Login (Operator) and Enable (Manager) Password” on page 6-9.

Option A: Configuring SSH Access for Password-Only SSH Authentication. When configured with this option, the switch uses its public key to authenticate itself to a client, but uses only passwords for client authentication.

Syntax: `aaa authentication ssh login < local | tacacs | radius >[< local | none >]`

*Configures a password method for the primary and secondary login (Operator) access. If you do not specify an optional secondary method, it defaults to **none**.*

`aaa authentication ssh enable < local | tacacs | radius>[< local | none >]`

*Configures a password method for the primary and secondary enable (Manager) access. If you do not specify an optional secondary method, it defaults to **none**.*

Option B: Configuring the Switch for Client Public-Key SSH

Authentication. If configured with this option, the switch uses its public key to authenticate itself to a client, but the client must also provide a client public-key for the switch to authenticate. This option requires the additional step of copying a client public-key file from a TFTP server into the switch. This means that before you can use this option, you must:

1. Create a key pair on an SSH client.
2. Copy the client's public key into a public-key file (which can contain up to ten client public-keys).
3. Copy the public-key file into a TFTP server accessible to the switch and download the file to the switch.

(For more on these topics, refer to “Further Information on SSH Client Public-Key Authentication” on page 6-21.)

With steps 1 - 3, above, completed and SSH properly configured on the switch, if an SSH client contacts the switch, login authentication automatically occurs first, using the switch and client public-keys. After the client gains login access, the switch controls client access to the manager level by requiring the passwords configured earlier by the **aaa authentication ssh enable** command.

Syntax: `copy tftp pub-key-file < ip-address > < filename >`

Copies a public key file into the switch.

`aaa authentication ssh login public-key`

*Configures the switch to authenticate a client public-key at the login level with an optional secondary password method (Default: **none**).*

Caution

To allow SSH access *only* to clients having the correct public key, you *must* configure the secondary (password) method for **login public-key** to **none**. Otherwise a client without the correct public key can still gain entry by submitting a correct local login password.

Syntax: `aaa authentication ssh enable < local | tacacs | radius > < local | none >`

*Configures a password method for the primary and secondary enable (Manager) access. If you do not specify an optional secondary method, it defaults to **none**.*

For example, assume that you have a client public-key file named Client-Keys.pub (on a TFTP server at 10.33.18.117) ready for downloading to the switch. For SSH access to the switch you want to allow only clients having a private key that matches a public key found in Client-Keys.pub. For Manager-level (enable) access for successful SSH clients you want to use TACACS+ for primary password authentication and **local** for secondary password authentication, with a Manager username of "leader" and a password of "m0ns00n". To set up this operation you would configure the switch in a manner similar to the following:

```
ProCurve(config)# password manager user-name leader
New password for Manager: *****
Please retype new password for Manager: *****
ProCurve(config)# [aaa authentication ssh login public-key none]
ProCurve(config)# [aaa authentication ssh enable tacacs local]
ProCurve(config)# [copy tftp pub-key-file 12.255.255.255 Client-key.pub]
ProCurve(config)# write memory
```

Configures Manager user-name and password.

Configures the switch to allow SSH access only a client whose public key matches one of the keys in the public key file

Copies a public key file named "Client-Keys.pub" into the switch.

Configures the primary and secondary password methods for Manager (enable) access. (Becomes available after SSH access is granted)

Figure 6-12. Configuring for SSH Access Requiring a Client Public-Key Match and Manager Passwords

Figure 6-13 shows how to check the results of the above commands.

```
ProCurve (config)# show authentication
```

Status and Counters - Authentication Information

Login Attempts : 3

Access Task	Login Primary	Login Secondary	Enable Primary	Enable Secondary
Console	Local	None	Local	None
Telnet	Local	None	Local	None
Port-Access	Local			
SSH	PublicKey	None	Tacacs	Local

ProCurve (config)# show crypto client-public-key

```
[0] "Maden name [1024-bit rsa, Local_crypto @Localcrypto, Thu Nov 07 2002 21:25:42]" ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGCz9oNfGxMHUFEC6frSulSa4UhlEFznFhQqmgP29HXYP6NR/1QOUmACtrFU+QD1lEtM/YM9FrN/XvZH/kIxTdEc5exFX/S10tcKaFYzI9UjK80dBMqvBGKB
IyVEbCVvlqdaqbkaEX3d/WaPS2xArLCFHsT2hmCvQTZDOGABlfrlcw==
[1] "[768-bit rsa, Local_crypto @Localcrypto, Mon Dec 16 2002 23:01:51]" ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQYQD0tmzA32JBgeuFJN0iXI3bfooPKZ09JKPCQcXEVk7N+eKf9MOX
vnmfFuEpw/fpqhlvsE66n8FDu7W/B2tKH/tgQLFqx7GiVcxNGhLiNO/pg5AuEym8Enc1Gu/LgAM9daM=
```

Figure 6-13. SSH Configuration and Client-Public-Key Listing From Figure 6-12

6. Use an SSH Client To Access the Switch

Test the SSH configuration on the switch to ensure that you have achieved the level of SSH operation you want for the switch. If you have problems, refer to "RADIUS-Related Problems" in the Troubleshooting chapter of the *Management and Configuration Guide* for your switch.

Further Information on SSH Client Public-Key Authentication

The section titled "5. Configure the Switch for SSH Authentication" on page 6-18 lists the steps for configuring SSH authentication on the switch. However, if you are new to SSH or need more details on client public-key authentication, this section may be helpful.

Configuring Secure Shell (SSH)

Further Information on SSH Client Public-Key Authentication

When configured for SSH operation, the switch automatically attempts to use its own host public-key to authenticate itself to SSH clients. To provide the optional, opposite service—client public-key authentication to the switch—you can configure the switch to store up to ten RSA or DSA public keys for authenticating clients. This requires storing an ASCII version of each client's public key (without babble conversion, or fingerprint conversion) in a client public-key file that you create and TFTP-copy to the switch. In this case, only clients that have a private key corresponding to one of the stored public keys can gain access to the switch using SSH. *That is, if you use this feature, only the clients whose public keys are in the client public-key file you store on the switch will have SSH access to the switch over the network.* If you do not allow secondary SSH login (Operator) access via local password, then the switch will refuse other SSH clients.

SSH clients that support client public-key authentication normally provide a utility to generate a key pair. The private key is usually stored in a password-protected file on the local host; the public key is stored in another file and is not protected.

(Note that even without using client public-key authentication, you can still require authentication from whoever attempts to access the switch from an SSH client— by employing the local username/password, TACACS+, or RADIUS features. Refer to “5. Configure the Switch for SSH Authentication” on page 6-18.)

If you enable client public-key authentication, the following events occur when a client tries to access the switch using SSH:

1. The client sends its public key to the switch with a request for authentication.
2. The switch compares the client's public key to those stored in the switch's client-public-key file. (As a prerequisite, you must use the switch's **copy tftp** command to download this file to flash.)
3. If there is not a match, and you have not configured the switch to accept a login password as a secondary authentication method, the switch denies SSH access to the client.
4. If there is a match, the switch:
 - a. Generates a random sequence of bytes.
 - b. Uses the client's public key to encrypt this sequence.
 - c. Send these encrypted bytes to the client.
5. The client uses its private key to decrypt the byte sequence.
6. The client then:

- a. Combines the decrypted byte sequence with specific session data.
 - b. Uses a secure hash algorithm to create a hash version of this information.
 - c. Returns the hash version to the switch.
7. The switch computes its own hash version of the data in step 6 and compares it to the client's hash version. If they match, then the client is authenticated. Otherwise, the client is denied access.

Using client public-key authentication requires these steps:

1. Generate a public/private key pair for each client you want to have SSH access to the switch. This can be a separate key for each client or the same key copied to several clients.
2. Copy the public key for each client into a client-public-key text file.
3. Use **copy tftp** to copy the client-public-key file into the switch. Note that the switch can hold 10 keys. The new key is appended to the client public-key file
4. Use the **aaa authentication ssh** command to enable client public-key authentication.

To Create a Client-Public-Key Text File. These steps describe how to copy client-public-keys into the switch for RSA challenge-response authentication, and require an understanding of how to use your SSH client application.

Bit Size	Exponent <e>	Modulus <n>	Comment
↓	↙	↓	↘
<pre>1024 35 11407406661701446907963803652840180539127043745111482882509285550110168603082 6038959146896306569035982041222025542543282764329943344032963504381021098947647460564 5572227682031607648603664020534703408371002884293231503492265409355321119922465153140 745413543765609589968291386053556814705585051025488575846923smith@support.cairns.com</pre>			

Figure 6-14. Example of a Client Public Key

Notes

Comments in public key files, such as **smith@support.cairns.com** in figure 6-14, may appear in a SSH client application's generated public key. While such comments may help to distinguish one key from another, they do not pose any restriction on the use of a key by multiple clients and/or users.

Public key illustrations such as the key shown in figure 6-14 usually include line breaks as a method for showing the whole key. However, in practice, line breaks in a public key will cause errors resulting in authentication failure.

Configuring Secure Shell (SSH)

Further Information on SSH Client Public-Key Authentication

1. Use your SSH client application to create a public/private key pair. Refer to the documentation provided with your SSH client application for details. The switch supports the following client-public-key properties:

Property	Supported Value	Comments
Key Format	ASCII	See figure 6-8 on page 6-13. The key must be one unbroken ASCII string. If you add more than one client-public-key to a file, terminate each key (except the last one) with a <CR><LF>. Spaces are allowed within the key to delimit the key's components. Note that, unlike the use of the switch's public key in an SSH client application, the format of a client-public-key used by the switch does not include the client's IP address.
Key Type	RSA only	
Maximum Supported Public Key Length	3072 bits	Shorter key lengths allow faster operation, but also mean diminished security.
Maximum Key Size	1024 characters	Includes the bit size, public index, modulus, any comments, <CR>, <LF>, and all blank spaces. If necessary, you can use an editor application to verify the size of a key. For example, placing a client-public-key into a Word for Windows text file and clicking on File Properties Statistics , lets you view the number of characters in the file, including spaces.

2. Copy the client's public key into a text file (*filename.txt*). (For example, you can use the Notepad editor included with the Microsoft® Windows® software. If you want several clients to use client public-key authentication, copy a public key for each of these clients (up to ten) into the file. Each key should be separated from the preceding key by a <CR><LF>.
3. Copy the client-public-key file into a TFTP server accessible to the switch.

Copying a client-public-key into the switch requires the following:

- One or more client-generated public keys. Refer to the documentation provided with your SSH client application.
- A copy of each client public key (up to ten) stored in a single text file or individual on a TFTP server to which the switch has access. Terminate all client public-keys in the file except the last one with a <CR><LF>.

Note on Public Keys

The actual content of a public key entry in a public key file is determined by the SSH client application generating the key. (Although you can manually add or edit any comments the client application adds to the end of the key, such as the **smith@fellow** at the end of the key in figure 6-14 on page 6-23.)

Syntax: copy tftp pub-key-file <ip-address> <filename>

Copies a public key file from a TFTP server into flash memory in the switch.

show crypto client-public-key [babble | fingerprint]

Displays the client public key(s) in the switch's current client-public-key file.

*The **babble** option converts the key data to phonetic hashes that are easier for visual comparisons.*

*The **fingerprint** option converts the key data to phonetic hashes that are for the same purpose.*

For example, if you wanted to copy a client public-key file named **clientkeys.txt** from a TFTP server at 10.38.252.195 and then display the file contents:

```
ProCurve(config)# copy tftp pub-key-file 10.38.252.195 Clientkeys.txt
ProCurve(config)# show crypto client-public-key
0."Maden name [1024-bit rsa, Jamie_wilson@Jamiewilson, Thu Nov 07 2002 21:25:42]" ssh-rsa AAAAB3NzaClyc2EAAAADAQABAAQgQCz9oNfqxMHUFEC6frSulSa4Uh1EFznFhQqmgP29HXyp6NR/1QOUmACtrFU+QD11EtM/YM9FrN/XvZH/kIxTdEc5exFX/$10tcRaFYzI9UjK80dBMqvBGKB
IyVEbCVwlqdAqbkaEX3d/WaPS2xArLCFHsTZhnCvQTZDOGAB1frlcw==
1."[768-bit rsa, Jamie_wilson@Jamiewilson, Mon Dec 16 2002 23:01:51]" ssh-rsa
AAAAB3NzaClyc2EAAAADAQABAAQgYQD0tmzA32JBgeuFJN0iXI3bfooPKZ09JKCPQcXEVk7N+eKf9MOX
vnmfFuEpw/fpqhlvsE66n8FDu7W/B2tKH/tqQLFqx7GiVcxNGhLiN0/pq5AuEym8Enc1Gu/LgAM9daM=
```

Key Index Number

Figure 6-15. Example of Copying and Displaying a Client Public-Key File Containing Two Client Public Keys

Replacing or Clearing the Public Key File. The client public-key file remains in the switch's flash memory even if you erase the startup-config file, reset the switch, or reboot the switch.

- You can remove the existing client public-key file or specific keys by executing the **clear crypto public-key** command.

Syntax:clear crypto public-key

Deletes the client-public-key file from the switch.

Syntax:clear crypto public-key 3

Deletes the entry with an index of 3 from the client-public-key file on the switch.

Enabling Client Public-Key Authentication. After you TFTP a client-public-key file into the switch (described above), you can configure the switch to allow one of the following:

- If an SSH client's public key matches the switch's client-public-key file, allow that client access to the switch. If there is not a public-key match, then deny access to that client.
- If an SSH client's public key does not have a match in the switch's client-public-key file, allow the client access if the user can enter the switch's login (Operator) password. (If the switch does not have an Operator password, then deny access to that client.)

Syntax: `aaa authentication ssh login public-key none`

Allows SSH client access only if the switch detects a match between the client's public key and an entry in the client-public-key file most recently copied into the switch.

`aaa authentication ssh login public-key local`

Allows SSH client access if there is a public key match (see above) or if the client's user enters the switch's login (Operator) password.

With **login public-key local** configured, if the switch does not have an Operator-level password, it blocks client public-key access to SSH clients whose private keys do not match a public key in the switch's client-public-key file.

Caution

To enable client public-key authentication to block SSH clients whose public keys are not in the client-public-key file copied into the switch, you must configure the Login Secondary as **none**. Otherwise, the switch allows such clients to attempt access using the switch's Operator password.

Messages Related to SSH Operation

Message	Meaning
00000K Peer unreachable.	Indicates an error in communicating with the tftp server or not finding the file to download. Causes include such factors as: <ul style="list-style-type: none">• Incorrect IP configuration on the switch• Incorrect IP address in the command• Case (upper/lower) error in the filename used in the command• Incorrect configuration on the TFTP server• The file is not in the expected location.• Network misconfiguration• No cable connection to the network
00000K Transport error.	Indicates the switch experienced a problem when trying to copy tftp the requested file. The file may not be in the expected directory, the filename may be misspelled in the command, or the file permissions may be wrong.
Cannot bind reserved TCP port <port-number>.	The ip ssh port command has attempted to configure a reserved TCP port. Use the default or select another port number. See “Note on Port Number” on page 6-17.
Client public key file corrupt or not found. Use 'copy tftp pub-key-file <ip-addr> <filename>' to download new file.	The client key does not exist in the switch. Use copy tftp to download the key from a TFTP server.
Download failed: overlenght key in key file.	The public key file you are trying to download has one of the following problems: <ul style="list-style-type: none">• A key in the file is too long. The maximum key length is 1024 characters, including spaces. This could also mean that two or more keys are merged together instead of being separated by a <CR><LF>.• There are more than ten public keys in the key file and switch total. Delete some keys from the switch or file. The switch does not detect duplicate keys.• One or more keys in the file is corrupted or is not a valid rsa public key.
Download failed: too many keys in key file.	
Download failed: one or more keys is not a valid public key.	Refer to “To Create a Client-Public-Key Text File” on page 23 for information on client-public-key properties.
Error: Requested keyfile does not exist.	The client key does not exist in the switch. Use copy tftp to download the key from a TFTP server.

Configuring Secure Shell (SSH)
Messages Related to SSH Operation

Message	Meaning
Generating new RSA host key. If the cache is depleted, this could take up to two minutes.	After you execute the <code>crypto key generate ssh [rsa]</code> command, the switch displays this message while it is generating the key.
Host RSA key file corrupt or not found. Use 'crypto key generate ssh rsa' to create new host key.	The switch's key is missing or corrupt. Use the crypto key generate ssh [rsa] command to generate a new key for the switch.