SunScreen SKIP Users Guide

Sun Microsystems, Inc.
901 San Antonio Road
Palo Alto, CA 94303-4900
U.S.A.

Part No: 806–5530
Aug 2, 2000

Adobe PostScript™

020531@3984

Contents

Tables

Figures

Preface

SunScreen™ SKIP is part of the family of SunScreen firewall products that provide a solution to security authentication, encryption, and privacy requirements giving companies a means of securing department networks connected to a public internetwork. This *SunScreen SKIP User's Guide, Release 1.5.1,For the Solaris Operating Environment* contains information for configuring and administering SunScreen SKIP 1.5.1 on your system. SunScreen SKIP enables secure, encrypted communication between a SunScreen 3.1 Administration Station and a SunScreen 3.1 Screen (or two or more Screens), and between a Screen and a remote SunScreen SKIP host running end-node SKIP.

Who Should Use This Guide

The *SunScreen SKIP User's Guide* is intended for SunScreen firewall system administrators responsible for the operation, support, and maintenance of network security. This guide assumed that you are familiar with UNIX system administration, Solaris® 2.6, Solaris 7, Trusted Solaris 7, or Solaris 8 operating environments, and TCP/IP networking concepts, and with your network topology.

For specific instructions on installing and configuring SunScreen SKIP as part of a SunScreen 3.1 Administration Station used to administer a remote Screen, see the *SunScreen 3.1 Installation Guide* and the *SunScreen 3.1 Administration Guide.*

Before You Read This Guide

This guide assumes that you are familiar with TCP/IP, networking, and public-key and shared-key cryptography.

How This Guide Is Organized

The *SunScreen SKIP User's Guide, Release 1.5.1, For the Solaris Operating System* is divided into the following chapters:

Chapter 1 describes how to install and configure the certificates for SKIP and how to protect your locally stored secrets with a passphrase.

Chapter 2 details how to create and install keys and certificates on your system. If you installed Unsigned Diffie-Hellman Certificates during installation, you can skip this chapter.

Chapter 3 describes how to use the `skiptool` graphical user interface (GUI) to monitor the network, how to configure SKIP, how to enable SKIP, how to verify SKIP installation and setup, how to view statistics, and how to manage keys.

Chapter 4 describes how to use the command-line interface as superuser or `root`.

Chapter 5 describes examples of using SunScreen SKIP in several network configurations.

Appendix A covers installing the SKIP binaries or adding the packages with `pkgadd`, and setting up IP-level encryption between two hosts.

Appendix B is an overview of what SKIP provides to users and how SunScreen SKIP fits in with other security products that use SKIP.

Appendix C contains instructions for troubleshooting SKIP and understanding SKIP error messages.

Glossary covers those terms that are specific or unique to Sun and the *SunScreen* line of products.

Typographic Conventions

The following table describes the typographic changes used in this book.

TABLE P–1 Typographic Conventions

Typeface or Symbol	Meaning	Example
`AaBbCc123`	The names of commands, files, and directories; on-screen computer output	Edit your `.login` file. Use `ls -a` to list all files. `machine_name% you have mail.`

`AaBbCc123`	What you type, contrasted with on-screen computer output	`machine_name%` **`su`**
		`Password:`

AaBbCc123	Command-line placeholder: replace with a real name or value	To delete a file, type **`rm`** *filename*.
AaBbCc123	Book titles, new words, or terms, or words to be emphasized.	Read Chapter 6 in *User's Guide*. These are called *class* options. You must be *root* to do this.

Shell Prompts in Command Examples

The following table shows the default system prompt and superuser prompt for the C shell, Bourne shell, and Korn shell.

TABLE P–2 Shell Prompts

Shell	Prompt
C shell prompt	`machine_name%`
C shell superuser prompt	`machine_name#`
Bourne shell and Korn shell prompt	`$`
Bourne shell and Korn shell superuser prompt	`#`

Related Books and Publications

You may want to refer to the following sources for background information on network security, cryptography, and SKIP.

- Schneier, Bruce, *Applied Cryptography*, John Wiley & Sons, 1996, 2nd edition, ISBN 0471128457
- Chapman, D. Brent, and Zwicky, Elizabeth D., *Building Internet Firewalls*, O'Reilly & Associates, 1995, ISBN 1565921240
- Walker, Kathryn M., and Cavanaugh, Linda Croswhite, *Computer Security Policies and SunScreen Firewalls*, Sun Microsystems Press, Prentice Hall, 1998, ISBN 0130960150
- Cheswick, Bill, and Bellovin, Steve, *Firewalls and Internet Security*, Addison-Wesley, 1994, ISBN 201633574
- Comer, Douglas E., *Internetworking with TCP/IP*, Volume 1, Prentice Hall, 1995, ISBN 0132169878
- Stallings, William, *Network and Internetwork Security Principles and Practice*, Institute of Electrical and Electronics, 1994, ISBN 078031108
- Garfinkel, Simson, and Spafford, Gene, *Practical UNIX and Internet Security*, O'Reilly & Associates, 1996, 2nd edition, ISBN 1565921488
- Stevens, W. Richard, *TCP/IP Illustrated*, Volume 1: The Protocols, Addison-Wesley, 1994, ISBN 0201633469
- Hunt, Craig, *TCP/IP Network Administration*, Addison Wesley, 1994, ISBN 020163469
- Kaufman, Charlie, Perlman, Radia, et al., *Network Security: Private Communication in a Public World*, Prentice Hall, 1995, ISBN 078816522.
- SKIP IP-Level Cryptography [`http://skip.incog.com/`]
- Sun Software and Networking Security [`http://www.sun.com/security/`]

Ordering Sun Documents

Fatbrain.com, an Internet professional bookstore, stocks select product documentation from Sun Microsystems, Inc.

For a list of documents and how to order them, visit the Sun Documentation Center on Fatbrain.com at `http://www1.fatbrain.com/documentation/sun`.

Accessing Sun Documentation Online

The docs.sun.com^{SM} Web site enables you to access Sun technical documentation online. You can browse the docs.sun.com archive or search for a specific book title or subject. The URL is `http://docs.sun.com`.

Getting Support for SunScreen Products

If you purchased this product from Sun Microsystems and require technical support, contact your Sun sales representative or Sun Authorized Reseller.

For information on contacting Sun, go to the URL: `http://www.sun.com/service/contacting/index.html`.

For information on Sun's Support services go to the URL: `http://www.sun.com/service/support/index.html.com/`.

Keys, Certificates, and Algorithms

Upgrade packages for U.S. Domestic (2048–bit and 4096–bit) and U.S. Export (2048–bit)keys, certificates, and algorithms from SunCA (Sun Microsystems' Certificate Authority) are intended to be used with SKIP.

U.S. customers and companies and some foreign customers and companies may order additional keys, certificates, and algorithms in stronger encryption strengths.

You can add new key pairs and local identities by using the SunScreen Key and Certificate diskettes that are available from Sun Microsystems Certificate Authority. Contact Sun using the email address CArequest@sun.com. These diskettes contain the private value, a signed certificate of the public value, and CA information. This type of key and certificate is known as an issued certificate.

CHAPTER 1

Installing and Configuring SunScreen SKIP

Note – While SunScreen SKIP is a part of a SunScreen Remote Administration solution, you should install and configure that particular configuration by using the SunScreen documentation: *SunScreen 3.1 Installation Guide* and *SunScreen3.1 Administration Guide*.

Overview of SunScreen

SunScreen is Sun Microsystems' implementation of Simple Key-Management for Internet Protocols (SKIP).

It is replacement software and upgrade software for any previous version of SKIP for the Solaris operating environment.

This chapter provides instructions for installing SunScreen on the Solaris 2.6. Solaris 7, or Solaris 8 operating environments for Sparc and Intel platforms and the Trusted Solaris 7 for the SPARC platform. Once SKIP is installed, configured, and enabled on the systems requiring its services, IP-layer encryption can begin. SKIP runs without further administration effort until new systems need to be added or certificate management is required. This chapter also describes how you can protect your locally stored secrets with a passphrase.

Hardware and Software Requirements

Supported Platforms

SunScreen is supported on the following platforms:

- Any Sun SPARC workstation running the Solaris 2.6, Solaris 7, or Solaris 8 operating environments.
- Any Intel-based PC that is compatible with and running the Solaris 2.6, Solaris 7, or Solaris 8 operating environments for the Intel Platform.

Note – The RC2-40 cryptor is restricted to use with the Solaris operating environment in 32-bit mode only.

Hardware Requirements

The hardware requirements are as follows:

- A minimum of 16–MB of RAM is required, 32–MB of RAM is recommended.
- A minimum of 6–MB of free disk space is required for installation, 3–MB of disk space is permanently used.
- One or more supported network interfaces.
- A CD-ROM drive.
- A floppy drive, if planning to install SunCA certificates.

Operating System Requirements

To run SunScreen, you must

1. **Install the Solaris SunCore® software group.**

   This software group contains the minimum software required to boot and run the Solaris operating environment. It includes some networking software and the drivers necessary to run the OpenWindows environment; it does not include the OpenWindows software.

2. **Additionally, install the following packages:**

system	SUNWadmr	System & Network Administration Root
system	SUNWcar	Core Architecture, (Root)
system	SUNWcsd	Core Solaris Devices
system	SUNWcsr	Core Solaris, (Root)
system	SUNWcsu	Core Solaris, (Usr)
system	SUNWdfb	Dumb Frame Buffer Device Drivers
system	SUNWesu	Extended System Utilities
system	SUNWkvm	Core Architecture, (Kvm)
system	SUNWlibC	SPARCompilers Bundled libC
system	SUNWlibms	SPARCompilers Bundled shared libm
system	SUNWtoo	Programming Tools
system	SUNWvolr	Volume Management, (Root)
system	SUNWvolu	Volume Management, (Usr)

3. **If you plan to use the** `skiptool` **GUI, install the packages for OpenWindows.**

   - `SUNWolrte`
   - `SUNWxwplt`
   - `SUNWolslb`

4. **If you are going to use certificates from a Certificate Authority, be aware that you must install the following operating system package:**

   ```
   system SUNWscpu Source Compatibility, (Usr)
   ```

   Otherwise the `install_skip_keys` command will fail.

Protocol Compatibility

SunScreen supports the following protocol versions:

- SKIP, Version 1, for SunScreen SPF-100/100G compatibility.
- Any platform that has implemented SKIP as described in the ICG Technical Reports, including the SunScreen product line, except SunScreen SPF-100, which only implements SKIP, Version 1 (see above).
- SunScreen, Release 1.5.1, is the upgrade for SunScreen SKIP, Release 1.5.

Installation Overview

Before installing SKIP, be sure that you have the CD-ROM for the base software and any encryption upgrade CD-ROMs or diskettes to which you are entitled.

Note – If you are an experienced SKIP user who just wants a quick installation overview, see Appendix A.

New Users

For the new user, this chapter tells about

- Installing SunScreen ("Installing the New Version" on page 25).
- Generating and installing an Unsigned Diffie-Hellman (UDH) key pair, if you are using UDH ("Installing SKIP Unsigned Diffie-Hellman (UDH) Certificates" on page 28).
- Installing SunScreen on your network interface ("Installing Your Network Interface" on page 32).
- Rebooting your system ("Rebooting Your System" on page 33).
- Protecting your locally stored secrets with a passphrase ("Activating Your Passphrase" on page 34).

Upgrade Users

For the user who is upgrading from any version of SKIP for the Solaris operating environment to this release, this chapter covers these additional topics (as well as the previously mentioned installation topics).

- Upgrading to SunScreen ("Upgrading From Earlier SKIP Versions" on page 23).
- Removing any old version of SKIP for the Solaris operating environment
- Preserving or removing previous configurations

Cryptography Upgrade Users

This chapter also contains information on how to add cryptography upgrade packages for those users who for example want to upgrade from a SKIP 512– bit version to a SKIP 2048– bit or 4096– bit version.

▼ Installing SKIP for the First Time

This section provides instructions for installing SKIP on the SPARC and Intel platforms running the Solaris 2.6, Solaris 7, or Solaris 8 operating environments.

To install and run the software, you must be able to become root on your local system and know the IP address of the machine on which SKIP is to be installed. Ask your systems administrator for the IP address of your machine. To install the software for the first time (or if you are installing it without saving the configurations), follow these steps:

1. **Open a terminal window and become `root`.**

2. **Mount the CD-ROM through the file manager by typing:**

   ```
   volcheck
   ```

   Note – If you are not using vold on your system, type `# mount -F hsfs -oro /dev/dsk/c0t6d0s0 /mnt`The device name or the mount point or both depends on your local system configuration.

   ---

3. **Go to the directory on the CD-ROM for your OS. (The examples assume a machine with only one CD-ROM.)**

   Solaris operating environment for the SPARC Platform:

   ```
   cd /cdrom/cdrom0/sparc
   ```

   Solaris operating environments for the Intel Platform:

   ```
   cd /cdrom/cdrom0/x86
   ```

   Note – If you have mounted the CD-ROM manually, replace `/cdrom/cdrom0` with `/mnt`.

   ---

4. **Type the standard Solaris operating environment `pkgadd` command to add all packages:**

   ```
   pkgadd  -d .
   ```

5. **You are prompted with the following menu of packages.**

   ```
   1 SUNW3des    SKIP 3DES
   Crypto Module
       (sparc) 1.5.1
   2 SUNW3desx    SKIP 3DES Crypto Module (64-bit
       (sparc) 1.5.1
   3 SUNWbdc    SKIP Bulk Data Crypt
       (sparc) 1.5.1
   ```

```
4 SUNWbdcx    SKIP Bulk Data Crypt (64-bit)
    (sparc) 1.5.1
5 SUNWdes    SKIP DES Crypto Module
    (sparc) 1.5.1
6 SUNWdesx    SKIP DES Crypto Module (64-bit)
    (sparc) 1.5.1
7 SUNWes    SKIP End System
    (sparc) 1.5.1
8 SUNWesx    SKIP End System (64-bit
    (sparc) 1.5.1
9 SUNWkdsup    SKIP D-Support module
    (sparc) 1.5.1
10 SUNWkeymg    SKIP Key Manager Tools
    (sparc) 1.5.1

... 8 more menu choices to follow;
<RETURN> for more choices, <CTRL-D> to stop display:

11 SUNWrc2    SKIP RC2 Crypto Module
    (sparc) 1.5.1
12 SUNWrc4    SKIP RC4 Crypto Module
    (sparc) 1.5.1
13 SUNWrc4s    SKIP RC4-128 Crypto Module
    (sparc) 1.5.1
14 SUNWrc4sx    SKIP RC4-128 Crypto Module (64-bit)
    (sparc) 1.5.1
15 SUNWrc4x    SKIP RC4 Crypto Module (64-bit)
    (sparc) 1.5.1
16 SUNWsafe    SKIP SAFER Crypto Module
    (sparc) 1.5.1
17 SUNWsafex    SKIP SAFER Crypto Module (64-bit
    (sparc) 1.5.1
18 SUNWsman    SKIP Man Pages
    (sparc) 1.5.1

Select package(s) you wish to process (or "all" to
process all packages). (default: all) [?,??,q]:
```

6. **Select `a` (all). As the prompts appear, answer questions with `Y` (yes) to add the package.**

7. **When you get back to the same menu of packages, type `q` to quit.**

8. **To eject the CD-ROM from the CD-ROM drive, type:**

   ```
   cd /
   eject cdrom0
   ```

   or eject the CD-ROM from the CD-ROM drive through the file manager.

Note – If you are not using `vold` on your system, unmount your CD-ROM by typing: **`#cd / #umount/mnt #eject cdrom0`**

---

9. **To add** /usr/sbin **to your PATH variable in the Bourne shell, type:**

   ```
   PATH=/usr/sbin:$PATH
   export PATH
   ```

10. **To add** /usr/share/man **to your MANPATH variable in the Bourne shell, type:**

    ```
    MANPATH=/usr/share/man:$MANPATH
    export MANPATH
    ```

11. **It will be helpful to add** /usr/sbin **to the PATH variable in your initialization file (such as:** .profile, .cshrc, **or** .login **file), and** /usr/share/man **to the MANPATH variable in the same file.**

    Now you are ready to complete the installation. The remaining steps include:

    - Generating and installing SKIP Unsigned Diffie-Hellman (UDH) certificates ("Installing SKIP Unsigned Diffie-Hellman (UDH) Certificates" on page 28) or installing SunCA certificates (Chapter 2). You can use SKIP Unsigned Diffie-Hellman certificates and SunCA keys and certificates at the same time on SunScreen.
    - Installing SunScreen on your network interface ("Installing Your Network Interface" on page 53).
    - Rebooting your system ("Rebooting Your System" on page 33).

Upgrading From Earlier SKIP Versions

To upgrade to SunScreen 1.5.1 from an earlier SKIP version, you must first remove the old version then install the new packages.

▼ Removing Versions Earlier than SKIP 1.5

To remove any version of SKIP for the Solaris operating environment earlier than 1.5, become root and use the pkginfo and pkgrm packages shown in the following steps.

1. **To list the SKIP packages that were installed, type:**

   ```
   pkginfo | grep SICG
   The list of packages is displayed:
   1 SICGbdcdr    SKIP Bulk Data Crypt 1.0.3-FCS Software
   2 SICGcrc2    SKIP RC2 Crypto Module 1.0.3-FCS Software
   3 SICGcrc4    SKIP RC4 Crypto Module 1.0.3-FCS Software
   4 SICGes    SKIP End System 1.0.3-FCS Software
   5 SICGkeymg    SKIP Key Manager Tools 1.0.3-FCS Software
   6 SICGkisup    SKIP I-Support module 1.0.3-FCS Software
   ```

```
        (sparc) 1.0.3-FCS
```

2. **To remove the packages, type:**

   `pkgrm` *package_names*

3. **Answer `Y` (yes:) to questions that the** `pkgrm` **program asks. The** `pkgrm` **program ends with the statement:**

   ```
   Removal of <SICGkisup> was successful.
   ```

   Note – This is valid only for this example. If moduli of other sizes were used, then the last package removed would be different.

4. **To remove the** `/etc/opt/SUNWicg/skip` **directory and any configurations that were installed, type:**

   ```
   rm -rf /etc/opt/SUNWicg/skip
   ```

   Caution – If you want to preserve previous configurations (including certificates, and the key manager configuration file), do *not* remove the `/etc/opt/SUNWicg/skip` directory.

5. **To reboot the machine, type:**

   ```
   init 6
   ```

▼ Removing SunScreen SKIP 1.5 or 1.5B

To remove SunScreen SKIP. Release 1.5 or Release 1.5B, for the Solaris operating environment, become root and use the `pkginfo` and `pkgrm` packages shown in the following steps.

1. **To list the SKIP packages that were installed, type:**

   ```
   pkginfo | grep -i skip
   The list of packages is displayed:
   application SUNW3des       SKIP 3DES Crypto Module
   application SUNW3desx      SKIP 3DES Crypto Module (64-bit)
   application SUNWbdc        SKIP Bulk Data Crypt
   application SUNWbdcx       SKIP Bulk Data Crypt (64-bit)
   application SUNWdes        SKIP DES Crypto Module
   application SUNWdesx       SKIP DES Crypto Module (64-bit)
   application SUNWes         SKIP End System
   application SUNWesx        SKIP End System (64-bit)
   application SUNWkdsup      SKIP D-Support module
   ```

```
application SUNWkeymg      SKIP Key Manager Tools
application SUNWkusup      SKIP U-Support module
application SUNWrc2        SKIP RC2 Crypto Module
application SUNWrc4        SKIP RC4 Crypto Module
application SUNWrc4s       SKIP RC4-128 Crypto Module
application SUNWrc4sx      SKIP RC4-128 Crypto Module (64-bit)
application SUNWrc4x       SKIP RC4 Crypto Module (64-bit)
application SUNWsafe       SKIP SAFER Crypto Module
application SUNWsafex      SKIP SAFER Crypto Module (64-bit)
application SUNWsman       SKIP Man Pages
```

2. **To remove the packages, type**

   `pkgrm` *package_names*

3. **Answer `Y` (yes:) to questions that the `pkgrm` program asks. The `pkgrm` program ends with the statement:**

   ```
   Removal of <SUNWsman> was successful.
   ```

   Note – This is valid only for this example. If moduli of other sizes were used, then the last package removed would be different.

   ---

4. **To remove the `/etc/opt/SUNWicg/skip` directory and any configurations that were installed, type:**

   ```
   rm -rf /etc/opt/SUNWicg/skip
   ```

   Caution – If you want to preserve previous configurations (including certificates, and the key manager configuration file), do *not* remove the `/etc/opt/SUNWicg/skip` directory.

   ---

5. **To reboot the machine, type:**

   ```
   init 6
   ```

Installing the New Version

Follow these steps:

1. **Open a terminal window and become `root`.**

2. **Mount the CD-ROM through the file manager or by typing:**

   ```
   volcheck
   ```

Note – If you are not using vold on your system, type # mount -F hsfs -oro /dev/dsk/c0t6d0s0/mntThe device name or the mount point or both depends on your local system configuration.

---

3. **Go to the directory on the CD-ROM for your OS:**

   Solaris operating environment for the SPARC Platform:

   ```
   cd /cdrom/cdrom0/sparc
   ```

   Solaris operating environment for the Intel Platform:

   ```
   cd /cdrom/cdrom0/x86
   ```

Note – If you have mounted the CD-ROM manually, replace /cdrom/cdrom0 with /mnt.

---

4. **To use the standard Solaris operating environment** pkgadd **command to add all packages, type:**

   ```
   pkgadd  -d .
   ```

5. **You are prompted with the following menu of packages:**

   ```
   1 SUNW3des    SKIP 3DES
   Crypto Module
       (sparc) 1.5.1
   2 SUNW3desx    SKIP 3DES Crypto Module (64-bit
       (sparc) 1.5.1
   3 SUNWbdc    SKIP Bulk Data Crypt
       (sparc) 1.5.1
   4 SUNWbdcx    SKIP Bulk Data Crypt (64-bit)
       (sparc) 1.5.1
   5 SUNWdes    SKIP DES Crypto Module
       (sparc) 1.5.1
   6 SUNWdesx    SKIP DES Crypto Module (64-bit)
       (sparc) 1.5.1
   7 SUNWes    SKIP End System
       (sparc) 1.5.1
   8 SUNWesx    SKIP End System (64-bit
       (sparc) 1.5.1
   9 SUNWkdsup    SKIP D-Support module
       (sparc) 1.5.1
   10 SUNWkeymg    SKIP Key Manager Tools
       (sparc) 1.5.1

   ... 8 more menu choices to follow;
   <RETURN> for more choices, <CTRL-D> to stop display:

   11 SUNWrc2    SKIP RC2 Crypto Module
   ```

```
    (sparc) 1.5.1
12 SUNWrc4    SKIP RC4 Crypto Module
    (sparc) 1.5.1
13 SUNWrc4s    SKIP RC4-128 Crypto Module
    (sparc) 1.5.1
14 SUNWrc4sx    SKIP RC4-128 Crypto Module (64-bit)
    (sparc) 1.5.1
15 SUNWrc4x    SKIP RC4 Crypto Module (64-bit)
    (sparc) 1.5.1
16 SUNWsafe    SKIP SAFER Crypto Module
    (sparc) 1.5.1
17 SUNWsafex    SKIP SAFER Crypto Module (64-bit)
    (sparc) 1.5.1
18 SUNWsman    SKIP Man Pages sparc) 1.5.1

Select package(s) you wish to process (or "all" to
process all packages). (default: all) [?,??,q]:
Select a (all). As the prompts appear, answer questions with Y (yes)
followed with a <Return> if you wish to add the package.
```

6. **Select `a` (all) or the number of the package. As the prompts appear, answer questions with `Y` (yes), if you wish to add the package.**

7. **When you get back to the same menu of packages, type `q` to quit.**

8. **If you want to use certificates, and the key manager configuration file from an earlier version of SKIP, type:**

```
cp /etc/opt/SUNWicg/skip/* /etc/skip
```

Note – 1.x ACLs cannot be used in version 1.5.1

9. **To eject the CD-ROM from the CD-ROM drive, type:**

```
cd /
eject cdrom0
```

or eject the CD-ROM through the file manager.

Note – If you are not using vold on your system, unmount your CD-ROM by typing: **`#cd / #umount/mnt #eject cdrom0`**

10. **To add** `/usr/sbin` **to your PATH variable in the Bourne shell, type:**

```
PATH=/usr/sbin:$PATH

export PATH
```

11. **To add** `/usr/share/man` **to your MANPATH variable in the Bourne shell, type:**

```
MANPATH=/usr/share/man:$MANPATH
export MANPATH
```

12. **It will be helpful to add** `/usr/sbin` **to the PATH variable in your initialization file (such as:** `.profile, .cshrc,` **or** `.login` **file), and** `/usr/share/man` **to the MANPATH variable in the same file.**

    Now you are ready to generate and install SKIP Unsigned Diffie-Hellman (UDH) certificates (if you are going to use them). You may use SKIP UDH certificates and SunCA keys and certificates at the same time on SunScreen.

    You are also ready to install SKIP on any new or different network interface (if you need to). Generate and install the SKIP UDH certificates ("Installing SKIP Unsigned Diffie-Hellman (UDH) Certificates" on page 28) and install SunScreen on the network interface ("Installing Your Network Interface" on page 32) before you reboot your system.

Note – If you are going to use the same keys, certificates and network interface that you used in SKIP for the Solaris operating environment, Release 1.0, you only need to reboot your system and restore any ACL files that you use. This is only true if you did not remove the `/etc/opt/SUNWicg/skip` directory and you copied over your old files.

Installing SKIP Unsigned Diffie-Hellman (UDH) Certificates

Once SKIP has been installed, you must install at least one local identity (public-private key pair) for your host. The following procedure creates a SKIP UDH certificate, which is the one you will most likely use. For a more detailed discussion of SKIP UDH certificates, see Appendix B.

Chapter 2 discusses keys, certificates, and hashes in greater detail. If you are installing other kinds of keys and certificates, see the documentation that is supplied with them or contact the vendor. If you are installing keys and certificates from Sun Microsystems, see Chapter 2 "".

The `skiplocal` command creates and manages all local key types, including UDH certificates, on your system. You can have more than one UDH certificate on your system. Your local identities can also be of different lengths (moduli), depending on the version of SunScreen that you have. The default will always be the largest modulus you can generate.

Note – Local secret is the term used for an encryption certificate and key.

▼ Initialize SKIP Directories

- **On a first-time SKIP installation, you must initialize the SKIP directories before you create any certificates. Issue the following command to initialize the SKIP directories:**

  ```
  skiplocal -i
  ```

▼ Generating a UDH Keypair

- **To generate an UDH key pair locally, type:**

  ```
  skiplocal -k
  ```

Note – If you have local identities of different strengths, such as 512 bits, 1024 bits), and 2048 bits or 4096 bits), use the argument *-m* followed immediately with the bit size of the modulus without an intervening space as in the following figure.

When generating an unsigned certificate, no authority exists to certify the identities. This means that each party must verify the name of the certificate over the telephone or some other trusted channel. Without verification through a secure channel, you have no way of knowing if the certificate belongs to the correct party or not.

In thefollowing figure,the `skiplocal -k` command was used to generate a local key pair, in this case with a 512–bit modulus.

EXAMPLE 1–1 512-bit Modulus

```
# skiplocal -k -m 512
generating local secret with 512 modulus size
It would help the quality of the random numbers if you would
type 50-100 random keys on the keyboard. Hit return when
you are done.
100
Format: Hashed Public Key (MD5)
Name/Hash: 9e 23 db 35 a2 c2 d8 17 20 19 21 99 3d c9 06 e1
Not valid Before: Sun Aug 25 17:00:00 1996
Not valid After: Sat Aug 25 17:00:00 2001
g: 2
p: f52aff3ce1b1294018118d7c84a70a72d676c40319c807297aca950cd9969fabd00a509b0246
d3083d66a45d419f9c7cbd894b221926baaba25eca55e92a055f
public key: 0b5522b769b3d2b8098e69312a941ce7e6de9e1635ca09dd780b328db71141739e9bb46a3
```

EXAMPLE 1–1 512-bit Modulus *(Continued)*

```
d0d183372d98d7c2a0d850b70fad05edaaaa865ae5dddf618cadbff
Added local identity slot 0
```

Printing out Local Information

- **To print out local information in a shareable form, type:**

`skiplocal -x`

In the following figure, the `skiplocal -x` command prints out the local system's current information in a form that can be sent (for example, via e-mail) to other users who wish to communicate with you.

Caution – The defaults proposed by `skiplocal -x` work well if you and the party with whom you wish to communicate have one key and one network interface. If you have some other configuration, you should not use `skiplocal -x`.

A safer solution than using `skiplocal -x` is to have each user run `skiptool` and then call each other on the telephone and type the other person's key ID in the Remote Key ID field in the add window (See Chapter 3).

In the following example, the first command shows you the local information. The next command redirects that information to a mail message sent to a machine that wishes to communicate with you using SKIP. Upon receiving the message, the user would copy the information and paste it onto their own command line which adds an ACL entry for your host.

EXAMPLE 1–2 Sending and Loading an ACL Entry

On local machine (mysun) display ACL entry in export format

```
# skiplocal -x
skiphost -a mysun -R 0x24be59e388dadfa6814885d1e5f79de9 -r 8
-s 8 -k des-ede-k3 -t des-cbc -m md5
```

Mail above text to the username@host

```
# skiplocal -x| mail username@host
```

On peer machine (host) execute skiphost command from mail message sent by mysun

```
# skiphost -a mysun -R 0x24be59e388dadfa6814885d1e5f79de9 -r 8
-s 8 -k des-ede-k3 -t des-cbc -m md5
```

Result:

EXAMPLE 1–2 Sending and Loading an ACL Entry *(Continued)*

```
Adding mysun:                                SKIP params:
    IP mode:                                tunneling
    Tunnel address                                mysun
    Kij alg:                                DES-EDE-K3
    Crypt alg:                                DES-CBC
    MAC alg:                                MD5
    Receiver NSID                                MD5 (DH Pub. Value)
    Receiver key id                                0x24be59e388dadfa6814885d1e5f79de9
    Sender NSID                                MD5 (DH Pub. Value)

                                                done.
```

Caution – Even when using `skiplocal -x`, make sure you both verify the key ID over the telephone with the other party to make sure no one is impersonating them.

▼ Listing the Current Local Identities

- **To list the current local identities, type:**

`skiplocal -l`

In the following figure, the `skiplocal -l` command is used to list the current local identities.

EXAMPLE 1–3 Listing All Local Identities

```
# skiplocal -l
Local ID Slot Name: 0    Type: Software Slot
    NSID: 8 MKID (name): 24be59e388dadfa6814885d1e5f79de9
    Not Valid Before: Tue Aug 6 17:00:00 1996
    Not Valid After: Mon Aug 6 17:00:00 2001
    Modulus size: 2048 bits

Local ID Slot Name: 1    Type: Software Slot
    NSID: 8 MKID (name): 8ace505b602127f38e08f74f13d0c915
    Not Valid Before: Sun Aug 25 17:00:00 1996
    Not Valid After: Sat Aug 25 17:00:00 2001
    Modulus size: 2048 bits

Local ID Slot Name: 2    Type: Software Slot
    NSID: 8 MKID (name): 9e23db35a2c2d817201921993dc906e1
    Not Valid Before: Sun Aug 25 17:00:00 1996
    Not Valid After: Sat Aug 25 17:00:00 2001
    Modulus size: 512 bits
```

EXAMPLE 1–3 Listing All Local Identities *(Continued)*

```
#
```

For more information on the `skiplocal` command, refer to Chapter 4 and to the `man` pages for SunScreen.

Note – If you installed an UDH certificate during installation, the information in Chapter 2 will not apply to you unless you also plan to install SunCA keys and certificates. You may use SKIP UDH certificates and SunCA keys and certificates at the same time on SunScreen.

Installing Your Network Interface

▼ Installing on One Interface

The `skipif` command is used to install SKIP on a network interface.

- **If you are adding SunScreen to a machine with only one interface, make sure that you are root and type:**

```
skipif -a
```

▼ Installing on Multiple Interfaces

- **If you are adding SunScreen to a machine with multiple interfaces, make sure that you are root and type:**

```
skipif -i networkinterface -a
```

Note – Replace *networkinterface* with the interface that you wish to specify. If you do not specify the network interface, it attaches to the first network interface that it finds.

You can add SKIP on more than one interface. In that case, you need to run the `skipif -a -i` *interface* command for each interface on which you want to use SKIP.

▼ Installing On All Interfaces

- **If you want to use SKIP on all the network interfaces present in the system, type:**

```
skipif -a -i all
```

Rebooting Your System

After you have installed the software, generated and installed the local identities, and installed the network interface, you must reboot your system.

- **To reboot the machine, type:**

```
init 6
```

Security Issues

This section describes how you can secure your SKIP software with an administrative password (passphrase) and information on why you should secure your core files and backup files.

Passphrase Protection

SKIP includes a feature that allows you to protect your locally stored secrets with a *passphrase*. A passphrase differs from a password in that it is longer and capitalization counts. This passphrase is used to encrypt all of your SKIP secret values. Your passphrase should be one that you can remember, but that is hard to guess. You can change the passphrase or delete it at any time. After you set, change, or delete your passphrase, you should run `skipd_restart` to reinitialize your key manager.

Note – Once you have protected your secret values with a passphrase, each time that you reboot you will *not* be able to run SKIP-encrypted connections because your system cannot get to your locally stored secrets with the passphrase. You must run `skipd_restart` which will then prompt you for your passphrase.

Caution – If you forget your passphrase, there is no way to discover it or recover it. Your protected locally stored secrets will no longer be available. If you do not know the passphrase and you want to reinstall or upgrade the software, you must first remove the old software and its locally stored secrets. See "Upgrading From Earlier SKIP Versions" on page 23. The old locally stored secrets will remain encrypted with the old passphrase and will be unavailable.

Once you set a passphrase, you will be prompted for it each time you add a new local identity (through skiplocal -a) or generate a new key (through skiplocal -k).

▼ Activating Your Passphrase

To activate your passphrase, use the following procedure:

1. **Type:**

```
skiplocal -P
```

2. **You are prompted as follows:**

```
You are now assigning
a global passphrase which will be used to encrypt all of your SKIP
secret values. Please choose a passphrase which you will remember,
but will be hard for someone else to guess
New global passphrase:    <type a new passphrase>
again: <type the new passphrase>
```

3. **To reinitialize your key manager, type:**

```
skipd_restart
```

▼ Changing Your Passphrase

To change your passphrase, use the following procedure:

1. **Type:**

```
skiplocal -P
```

2. **You are prompted as follows:**

```
You are now changing
the global passphrase which is used to encrypt your SKIP secrets
Global passphrase:    <type a old passphrase>
New Passphrase:    <type a new passphrase>
again:    <type the new passphrase>
```

3. **To reinitialize your key manager, type**

```
skipd_restart
```

▼ Removing Your Passphrase

To remove your passphrase, use the following procedure:

1. **Type:**

   ```
   skiplocal -R
   ```

2. **You are prompted as follows:**

   ```
   You are now removing
   the global passphrase which will be used to encrypt all of your
   SKIP secrets.
   Global passphrase:    <type your passphrase>
   ```

   If it matches, all locally stored secrets are decrypted and stored and the passphrase feature is disabled.

3. **To reinitialize your key manager, type:**

   ```
   skipd_restart
   ```

Upgrading Cryptography Modules

The following table contains information about the packages you need if you want to add additional cryptography modules to your configuration. For example, SunScreen 3.1 ships with the 512– bit version of SKIP which only contains the RC2 and RC4(x) Crypto modules. To add additional modules, for example DES, you must take some care to install only the packages you need.

Note – Do not add the End System SKIP modules (SUNWes and SUNWesx) to a SunScreen EFS 3.0 Screen or a SunScreen 3.1 Screen.

TABLE 1–1 SKIP Cryptography Upgrades

If you have the 512–bit version...	Add these packages to upgrade to the 1024– bit version...	Add these packages to upgrade to the 2048– or 4096–bit version...
	SUNWkusup SKIP U-Support module	SUNWkdsup SKIP D-Support module

TABLE 1–1 SKIP Cryptography Upgrades *(Continued)*

If you have the 512–bit version...	Add these packages to upgrade to the 1024– bit version...	Add these packages to upgrade to the 2048– or 4096–bit version...
	SUNWdes SKIP DES Crypto Module	SUNWdes SKIP DES Crypto Module
	SUNWdesx SKIP DES Crypto Module (64-bit)	SUNWdesx SKIP DES Crypto Module (64-bit)
		SUNW3des SKIP 3DES Crypto Module
		SUNW3desx SKIP 3DES Crypto Module (64-bit)
		SUNWrc4s SKIP RC4-128 Crypto Module
		SUNWrc4sx SKIP RC4-128 Crypto Module (64-bit)
		SUNWsafe SKIP SAFER Crypto Module
		SUNWsafex SKIP SAFER Crypto Module (64-bit)

Security Concerns

Core Files and Security

You should be aware that a saved core file contains your local secret(s). While it would be difficult for someone to discern or discover the secrets from this file, it is possible. You should, therefore, protect a core file as carefully as any of your other local secrets. Remember, if you send your core file out-of-house for analysis, you are giving your local secret to the analyst.

Any system backups made while such a core file exists may contain the core file as well and so must be considered a possible means of discovering your local secret(s). These backups must be kept in a secure location.

Expired Certificates and Security

Two systems can still communicate even after one of the systems's certificate has expired; communication between two peers persists until you issue a `skipd_restart` command. The key manager daemon or commands check against certificate expiration upon identities addition or daemon restart. There is no checking against certificate expiration when the ACL and the corresponding key management information have been passed to the kernel.

CHAPTER **2**

Installing Keys and Certificates

This chapter tells you how to install keys and certificates on your system.

Note – If you installed an UDH certificate during installation, the information in this chapter will not apply to you unless you also plan to install SunCA keys and certificates.

There are two kinds of certificates that you can use with SunScreen:

- UDH
- SunCA

Which certificates you choose to use is determined by the security policy of your company.

At the end of the configuration process in Chapter 1 you created a SKIP UDH certificate using the `skiplocal` command.

You can use the `install_skip_keys` command to install SunCA keys and certificates on SunScreen at the same time. This section shows you how to install certificates signed by the SunCA.

Note – You must be root to use the command-line commands.

Keys and Certificates

Keys

Traditional cryptography relies on the sender and receiver of a message knowing and using the same secret key. When both sender and receiver use the same secret key, the system is referred to as a symmetric or single-key crypto system. The problems with using the same secret key are: how is one selected, how do the parties inform each other of the secret key if they are not physically in the same location, how do they change keys from time to time, and how is the secret key kept secure.

Public-key cryptography was proposed as a solution to the problems found in traditional, symmetric key cryptography. In public-key cryptography, each person, host, or network participating in a coded exchange, receives a pair of keys: one public and one private. The private key is kept a secret and the public key is published so that anyone who wishes to communicate confidentially with a person or an entity can do so by encoding their message using the public key. The confidential message can then only be decoded by the private key, which is kept in the sole possession of the intended recipient.

SKIP is a public-key, certificate-based, key-management scheme. It uses certified Diffie-Hellman public values to eliminate the need for prior communications between two entities wishing to exchange encrypted data.

There are times when it is useful to allow a system to have more than one pair of public-private keys. For example, different key sizes may be required when communicating with subsidiaries in other countries because of U.S. or local regulations. To meet these user requirements, SunScreen's implementation permits a system to possess as many local keys as required. Public-private key pairs like UDH keys can be used for authentication.

Certificates

To ensure that a public key is authentic (that is, it has not been tampered with by an unauthorized user and does indeed belong to the claimant), the public key is normally signed by a Certification Authority (CA). The result, a digital document called a certificate, can be freely passed around the network. Its authenticity can be verified by anyone holding the CA's signature information; that is, the CA's public key.

Before any form of encrypted communication can begin, the parties involved in the transaction must exchange certificates. This is a manual procedure in that the certificate and possibly the key are provided by the certifying agency on physical media: tape, diskette, or CD-ROM. The user must load them into the system through a command-line interface.

Key and Certificate Management

Secure key management is a necessary requirement for any cryptographic product. Users must be able to obtain keys as required for their security needs, have a method of looking up other's public keys, publicize their own keys, and determine that a key is valid. Certificates are used for this purpose.

Certificates must be unforgeable, obtainable in a secure manner, and processed in such a way that an unauthorized user cannot misuse them. This means that the network manager must handle the following issues:

- Loss or compromise of a private key
- Verifiable signature after key expiration
- Expiration dates
- Secure storage of private keys

Note – Two systems can still communicate even after one of the systems's certificate has expired; communication between two peers persists until you issue a `skipd_restart` command.

Adding Certificates or Local Identities with `install_skip_keys`

The `install_skip_keys` command is used to install key packages that have been received from a key server or from one of the SunCAs. If used with `-icg,` it means that the SunCA or the SunCAglobal CA certified the keys. The SunCA certifies 1024–bit and 2048–bit modulus certificates, and the SunCAglobal certifies 512–bit certificates.

To communicate with a *SunScreen SPF-100* or *SunScreen SPF-100 G,* you need to use SunCA or SunCAglobal certificates.

Requirements

If you are going to use certificates from a Certificate Authority, be aware that you must install the following operating system package:

system SUNWscpu Source Compatibility, (Usr)

Otherwise the `install_skip_keys` command will fail.

Note – The `install_skip_keys` command is not used to add someone else's certificate. It is only used to install local identities for CA key packages.

The **Failed Cross Reference Format** shows installing a SunCAglobal key and certificate from diskettes. After installing the key and certificate, because you have added a new local identity, you must either run the `skipd_restart` command or reboot your system to initialize the key manager.

EXAMPLE 2–1 Installing a SunCA Global Key and Certificate from Diskette

```
# install_skip_keys -icg /floppy/unnamed_floppy
Added CA certificate as ca-slot 0

Added local identity slot 3

added 0a1030cc to database
/usr/sbin/install_skip_keys: you should now reboot the machine to initialize SKIP.
```

For more information on `install_skip_keys`, see Chapter 4 and the `man` pages.

CHAPTER **3**

Using the skiptool GUI

This chapter tells you:

- How to start and use skiptool (“skiptool Overview” on page 43)
- How to configure SKIP (“Configuring SunScreen” on page 47)
- How to enable SKIP (“Enabling SKIP” on page 62)
- How to tell if SKIP is working (“Is SKIP Working?” on page 65)
- How to view statistics (“Viewing SunScreen Statistics ” on page 66)
- How to manage keys (“Key Management with skiptool” on page 76)

skiptool Overview

Once you install SunScreen and the local keys on your machine, you must set it up so that it can communicate with other systems using SKIP.

SKIP provides two ways to configure and manage SunScreen: skiptool (the GUI) and skiphost (the command-line interface discussed in Chapter 4) The easiest way to set up your ACLs is through skiptool. Using skiptool you can:

- Enable and disable access to your machine
- Set the type of encryption used for hosts or network connections to your system (encrypted or unencrypted [clear])
- Determine how to deal with unauthorized hosts that try to connect to your system.
- View the following statistics including:
  - Network Interface Statistics
  - SKIP Header Statistics
  - Key Statistics
  - Encryption Statistics (for Versions 1 and 2)

- Authentication Statistics

Note – If you are managing a large amount of certificates. you will probably find it easier to use the CLI as the `skiptool` GUI displays the MKID as truncated.

`skiptool` Requirements

To run `skiptool`, you must have root privileges on your system.

Note – Enable access for any client to the *X* server for Solaris 2.*x* operating environments by entering the `xhost +` *localhost* command before you become root.

▼ Starting `skiptool`

Use the following steps to start `skiptool`:

● **Become root, and type:**

```
# skiptool&
```

Note – If you are configuring a system with multiple network interfaces, you can specify the interface following the command; for example, `skiptool le1`.

The main window of `skiptool` is shown in the following figure.

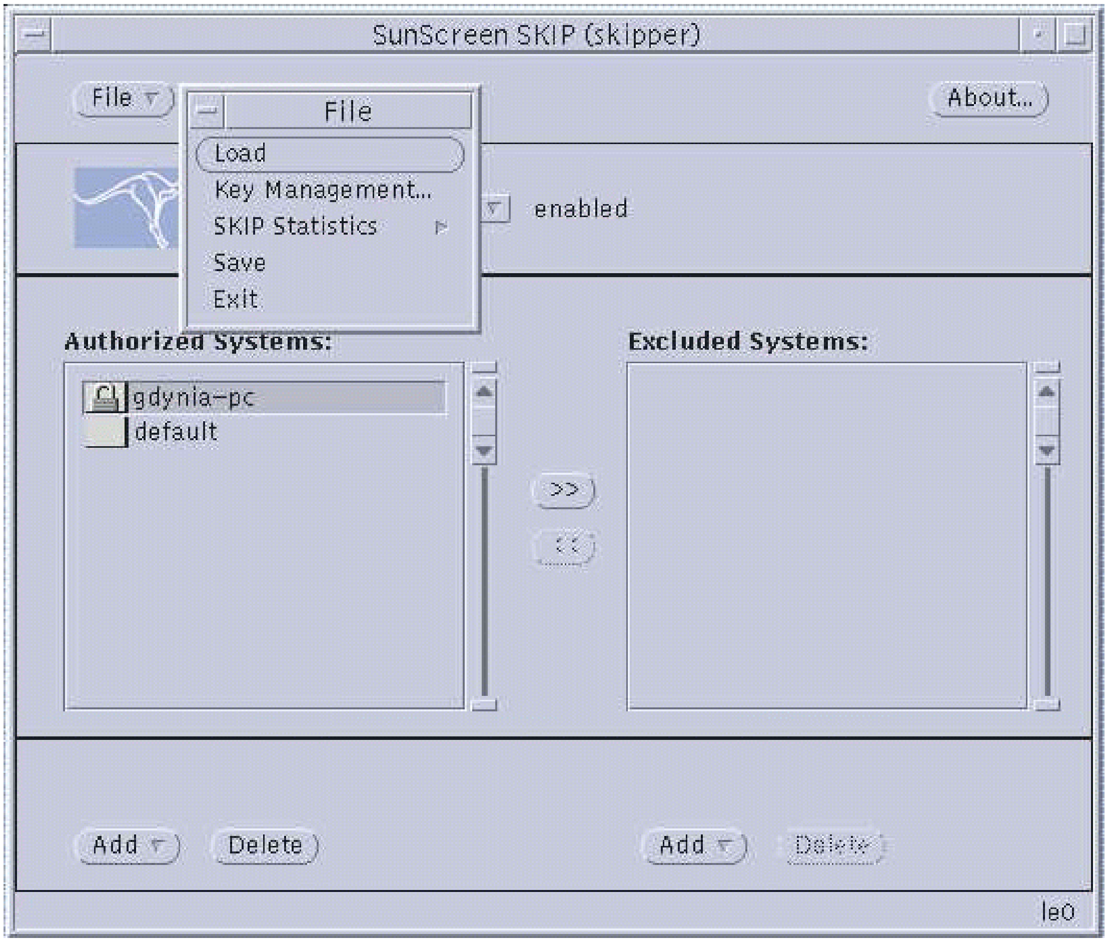


FIGURE 3–1 skiptool Main Window

The skiptool Main Window

The skiptool main window has several important features:

- The File button
- The Access Control buttons
- The Authorized and Excluded Systems lists
- The Add and Delete (Management) buttons

Note – If you have other windows layered on top of the `skiptool` main window, you need to click on the Title Bar to bring the main window to the front.

File Menu

The file menu has five submenus:

Load—Loads the current ACL from the kernel. This feature is useful if you have modified the ACL through other tools and want to update the configuration in `skiptool`.

Key Management—Defines the parameters for key usage, including when to delete an unused key (in seconds) and how much data to transmit per key (in Kbytes).

SKIP Statistics—Brings up one of six statistics windows: (Network Interface Stats, SKIP Header Stats, Encryption Stats (Version 1), Encryption Stats (Version 2), Key Stats, or Authentication Stats.

Save—Makes the configuration permanent. Before saving, it prompts you to add any systems that are in use, that have access, and that are not currently on the authorized list. The next time that you reboot this configuration is used. Quitting and restarting `skiptool` will not affect either saved or unsaved changes in the configuration.

Note – If you do not save the changes in the configuration, they will only remain in effect until the next time you reboot your machine.

Exit—Closes all open windows and quits `skiptool`. The Statistics window will not close when you quit `skiptool`.

Access Control Buttons

Access Control button—This button toggles to enable or disable SKIP. When SKIP is enabled, the ACL rules apply.

For example, you could have only the *default* entry in the authorized systems list and some entries in the excluded systems list. In this case, any host except those that are in the excluded systems list could connect. When SKIP is disabled, any system can connect, if the *default* entry is configured in the clear.

Authorized Systems/Excluded Systems Lists

Authorized Systems—A list of systems that are authorized to access to host. System types are *host*, *network*, or *nomadic*. Secure systems appear with a a padlock or the Sun Microsystems' logo next to the system name (depending on the type of security being used.)

Excluded Systems—A list of systems that are specifically denied access to your system. When you move or add a system to the excluded list, it is immediately excluded.

`skiptool` allows you to move systems from the list of authorized systems to the list of excluded systems and vice versa with the arrows between the two lists.

Management Buttons

These buttons enable you to add or delete a system from the access list. The buttons are available for both authorized and excluded systems.

Add—Brings up the *Add* pop-up menu where you select the system type to be added to the ACL:

Host—Adds an individual host, either with or without security.

Network—*Adds a network, either with or without security.*

Nomadic—Adds a nomadic identity, with SKIP Version 1 or SKIP Version 2 security.

Delete—Deletes the selected system from the list. When an item is deleted, the deletion occurs immediately and cannot be undone.

You may also move ACL entries from one list to another with the arrow buttons. These arrow buttons make it easy to add or delete systems when troubleshooting.

Caution – If you add, delete, or move ACL entries from one list to another, the action takes effect immediately.

Configuring SunScreen

You can configure only one network interface at a time using `skiptool`. If you have more than one network interface, you must configure each one separately.

▼ Configuring SKIP

Configuring SunScreen requires completing several simple steps:

1. **Adding authorized systems**
2. **Adding any excluded systems**
3. **Setting up the behavior for unauthorized systems**
4. **Enabling SKIP**
5. **Verifying the installation and set up**

Adding Authorized Systems

Any remote host with which you want to communicate must be configured using the Add pop-up window.

An authorized host may or may not be using encryption. The *Add* pop-up window provides three options:

- Off (not using encryption)
- Using SKIP encryption
- Using SKIP Version 1 encryption

You add hosts to the authorized systems list using the *Add* button. The valid types of remote hosts that you can add to your ACL are

- Host
- Network
- Nomadic

Caution – When setting up SunScreen, be sure to include any NFS servers and NIS or DNS name servers on the authorized systems list, otherwise your system may hang. To avoid problems such as this, a safe approach at the beginning is to add the clear “default” entry. Once you become more comfortable with SKIP configuration, you can remove it.

To determine the servers your system communicates with, use the following commands:

- For NFS servers, type **`mount`**

- For NIS servers, type **`ypwhich`**
- For DNS servers, consult your system administrator
- Verify the current routing entries used by the local system. To verify the current routing entries, type **`netstat -rn`** and add specific network ACL entries.

If you do not specify a system that you currently have in use when you enable access control, a menu will come up and ask if you want to add the system. It also checks for multicast routers that are being used for others and adds them to the proposed list of systems to add.

Regardless of the type of system that you are adding to the ACL, you must implement the same policy on both your machine and the entity with which you wish to communicate securely. If you do not configure both systems properly, the packets are silently dropped and it appears as if that particular host does not exist. `skiplog` is useful in diagnosing this situation.

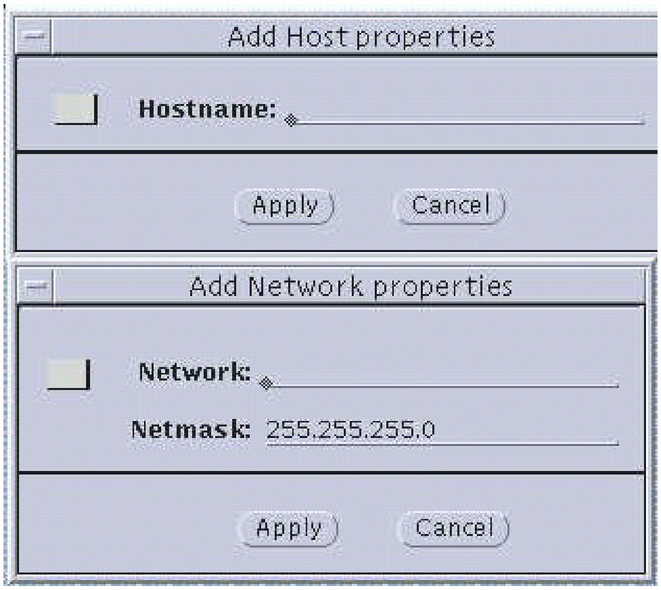
When you click on the *Add* button, the *Add* pop-up window appears. From the menu in this window, you select the type of connection: Host, Network, or Nomadic. Next, use the pull-right menu to set the security level. After you have selected the level of security, the appropriate *Properties* window becomes available. The *Add System Properties* window is used to set up the options for the type of encryption used by the host, network, or nomadic system being authorized. The procedures in the sections following the table detail how to set up each encryption option.

▼ Adding a Host or Network with No Encryption

This procedure is used to allow a host or network access to your system without using any encryption.

1. **Click and hold the Add button at the bottom of the authorized systems list on the `skiptool` main window.**
2. **Select the type of connection being authorized: Host or Network. (*Nomadic* does not offer this option.)**
3. **Pull right on the type of connection and select Off.**

   The Add Host properties or Add Network properties dialog box, shown in the following figure, in appears.



Netmask: 255.255.255.0

FIGURE 3–2 Add Host/Properties—No Encryption

4. **In the Add Host or Network properties window, enter the name or IP address of the host system to be added to your ACL.**

   In the case of a network, you must define the network with the IP address and the netmask.

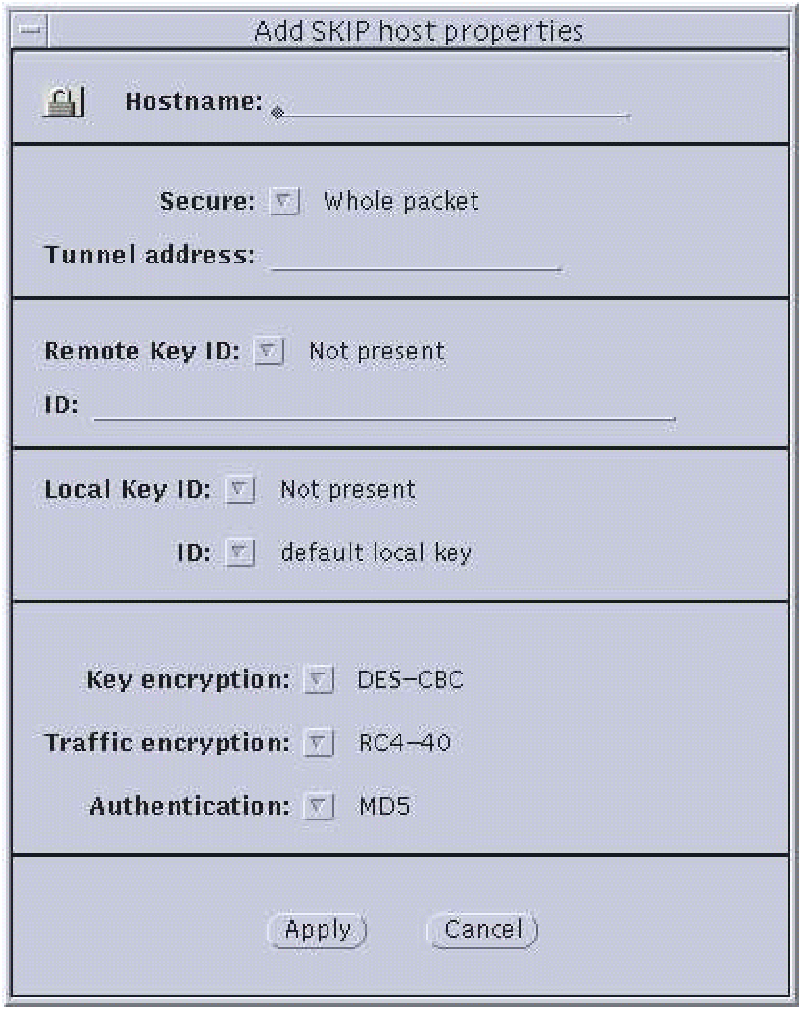
5. **Click the Apply button.**

Setting Up Security for a Host, Network, or Nomadic System

The procedures in this section enable a host, network, or nomadic system access to your system according to the encryption rules you set up. Remember, both your system and the other system need to use the same rules in order to communicate.

Dialog Box Parameters

The following section provides some background on the choices available to you from the `skiptool` dialog boxes. This material will be useful to use the procedures that follow. The two encryption dialog boxes (SKIP and SKIP Version 1) use common set-up parameters, as you can see in Figures 3-3 through Figure 3–5. Explanations of the parameters follow the figures.



Hostname:

Secure: Whole packet

Tunnel address:

Remote Key ID: Not present

ID:

Local Key ID: Not present

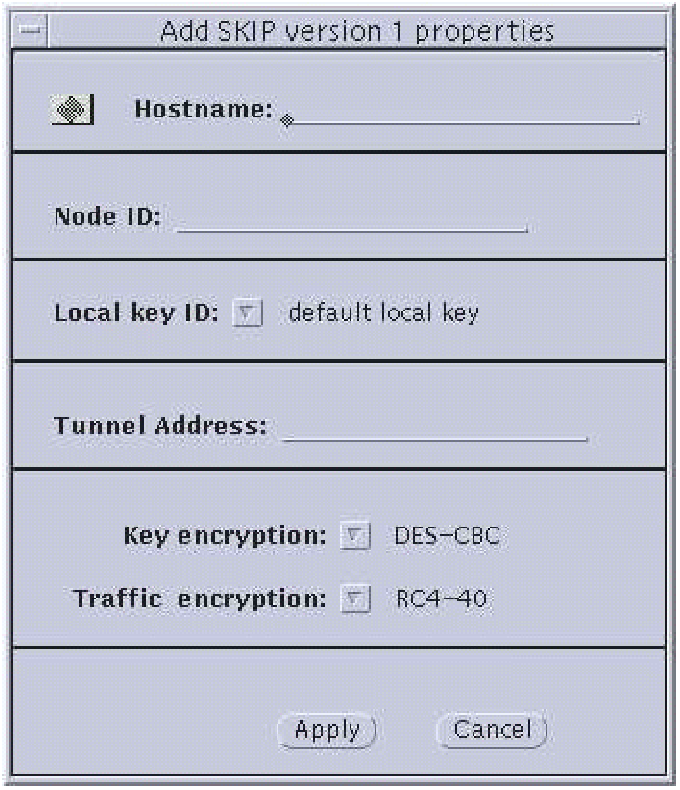
ID: default local key

Key encryption: DES-CBC

Traffic encryption: RC4-40

Authentication: MD5

FIGURE 3–3 Host—Add SKIP Host Properties and SKIP Version 1 Properties

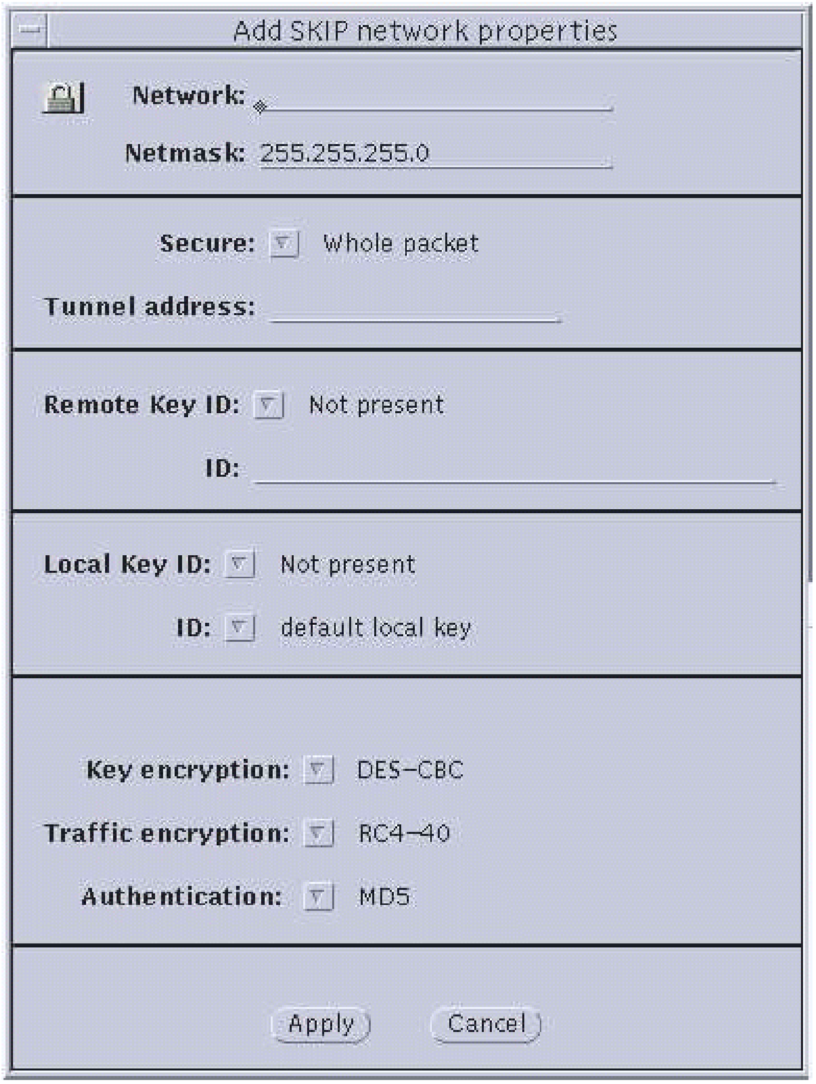


Local key ID: default local key

Key encryption: DES-CBC

Traffic encryption: RC4-40

FIGURE 3–3 Host—Add SKIP Host Properties and SKIP Version 1 Properties



Add SKIP network properties

Network:

Netmask: 255.255.255.0

Secure: Whole packet

Tunnel address:

Remote Key ID: Not present

ID:

Local Key ID: Not present

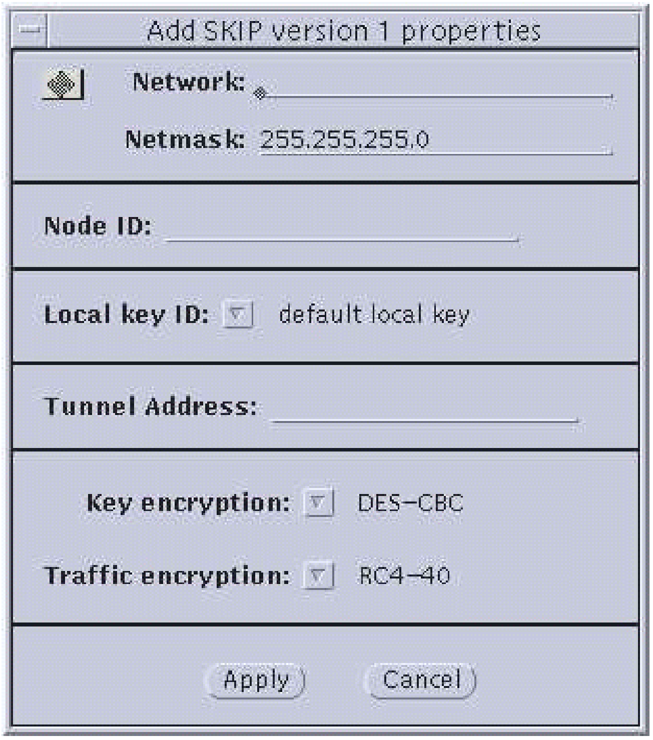
ID: default local key

Key encryption: DES-CBC

Traffic encryption: RC4-40

Authentication: MD5

FIGURE 3–4 Network—Add SKIP Network Properties and SKIP Version 1 Properties



Add SKIP version 1 properties

Network:

Netmask: 255.255.255.0

Node ID:

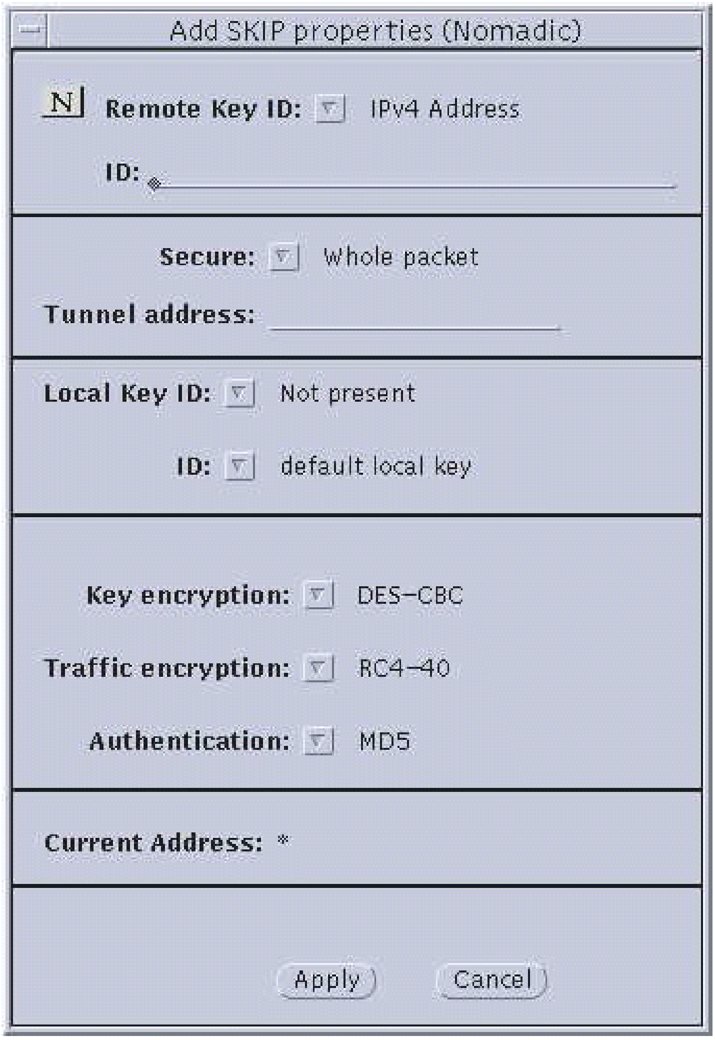
Local key ID: default local key

Tunnel Address:

Key encryption: DES−CBC

Traffic encryption: RC4−40

FIGURE 3–4 Network—Add SKIP Network Properties and SKIP Version 1 Properties



Add SKIP properties (Nomadic)

Remote Key ID: IPv4 Address

ID:

Secure: Whole packet

Tunnel address:

Local Key ID: Not present

ID: default local key

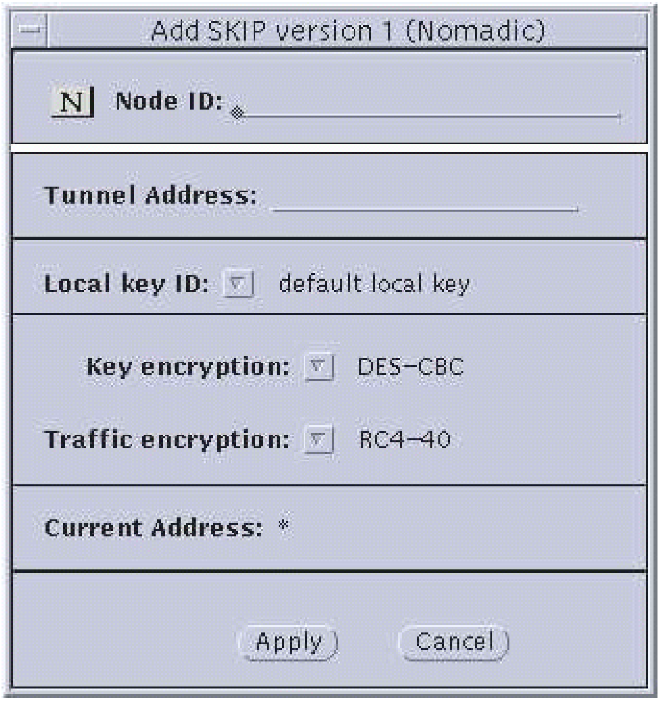
Key encryption: DES-CBC

Traffic encryption: RC4-40

Authentication: MD5

Current Address: *

FIGURE 3–5 Nomadic—Add SKIP Properties and Add SKIP Version 1



Add SKIP version 1 (Nomadic)

N Node ID:

Tunnel Address:

Local key ID: default local key

Key encryption: DES-CBC

Traffic encryption: RC4-40

Current Address: *

FIGURE 3–5 Nomadic—Add SKIP Properties and Add SKIP Version 1

Parameter Explanations

- *Hostname/Network/Node ID*. Enter the name of the host or nomadic system, or the IP address of the host or network.
- *Netmask*. (network only) Enter the netmask of the network. The default (255.255.255.0) is already entered.
- *Secure button*. (SKIP only) Set to either *Whole packet* (tunnel mode) or *Data only* (transport mode). Whole packet is recommended because it offers a greater degree of security.
- *Node ID*. (SKIP Version 1 only) This is the IPv4 key ID.
- Tunnel Address. Use the tunnel address as the destination IP address. Tunnel address is generally used for clients of encrypted gateways where the IP address of the host entered here serves as the intermediary for any or all hosts on a network whose topography must remain unknown or hidden from the rest of the world. This is called topology hiding. This field is not available if you select *Data only*.
- Remote *Key ID button*. (SKIP only) Select whether you want the remote system's key ID included in SKIP packets and, if so, the namespace that key ID occupies. Selecting *Not Present* means that the receiver key ID will not be sent.

The following namespaces are listed in this menu:

- Not Present
- IPv4 Address
- MD5 (DH Public Value)

  Not Present is the default. It uses the IP address of the remote system to identify its certificate. If a remote system has a key ID other than that identified by its IP address, set the namespace and indicate the remote system's key ID in the ID field.

Note – When you add a host by entering the hostname and you change the Remote key Id menu to *IPV4 address* from *Not present* the cursor does not appear in the ID field even if you click there. You must click on the Local key Id menu to move the cursor and regain focus.

- *Remote Key ID field*. (SKIP only) The namespace indicated in the Remote Key ID field is determined by the type of certificate, shown in the following table, that you are using or have obtained for this system:

  TABLE 3–1 Remote Key ID Field

Certificate Type	Remote Key ID Field
CA (Sun or other)	IPv4
Self-generated unsigned key	MD5 (DH Public Value)

  If the Remote Key ID field has been set to other than *Not Present*, enter the key ID in hexadecimal format in the ID field (such as 0x0a000000). It must contain the appropriate key ID for the system being authorized based upon the selection made with the Remote Key ID button. Depending on the type of certificate, this information may be obtained from the master key ID on the diskette or from the local key ID field of the other host.

- *Local Key ID and ID buttons*. Use the Local Key ID button to indicate whether you want your local system to send its key ID in the SKIP packet and, if so, the namespace that key occupies. If you select *Not Present*, the sender's key ID is not sent in the packet and the remote system uses the local system's IP address to decide what key to use.

Note – If you have installed new local keys after you have started `skiptool`, `skiptool` will not list them. You must restart the key manager with the `skipd_restart` command to list them and rerun `skiptool`.

All the local-key times installed for this host are listed. Select the namespace for the local key that is to be used for communication with the above host. Once you have selected the namespace, click on the ID field to select the key to be used, in hexadecimal, for communication with this host.

- *Key Encryption button*. Selecting this button lists the available key encryption algorithms. The algorithms available are determined by the system type and the selected encryption method selected.

TABLE 3–2 Available Key and Traffic Encryption Algorithms

	512 bit	1024 bit	2048 and 4096 bit
Key Encryption	DES-CBC	DES-CBC	DES-CBC
	RC2-40 (32-bit mode only)	RC2-40 (32-bit mode only)	RC2-40 (32-bit mode only)
			3DES
			SAFER
Traffic Encryption	RC2-40 (32-bit mode only)	RC2-40 (32-bit mode only)	RC2-40 (32-bit mode only)
	RC4-40	RC4-40	RC4-40
		3DES	3DES
			SAFER
			RC4-128

Note – The RC2 cryptor is supported in 32-bit mode only. If you inadvertently select this cryptor while in 64-bit mode, the console reports an error message and communication with the system using this cryptor stops (no information is transferred in the clear).

- *Traffic Encryption button.* Select the algorithm for encrypting the traffic between your system and the remote system. The algorithms available are determined by the system type, the version of SunScreen, and the method of encryption selected. Please see the previous table for complete information.
- Authentication button. Use the authentication button to select the type of authentication for the packets. Currently, SunScreen supports two types of authentication—MD5 and MD5-NAT. You can also select *None* for no authentication.

Repeat Steps 1 though 8 for all encrypted hosts. Remember that your policy options for each system entered on your ACL must be the same as those entered on the system entity with which you wish to communicate through encrypted channels. If the configuration on your system does not match that of the party with which you wish to communicate, the packets are silently dropped. It will simply appear as though that host no longer exists.

Using Default System Entry

The default system entry is used when no other more specific ACL entry matches a host. Often, this entry is set to clear to allow hosts that are not listed in the ACL to communicate in the clear. It may, however, be used to create a default encryption rule.

Communicating In the Clear (Off)

Typically, the NIS and DNS servers to which your systems have access are set up as communicating with your system in the *clear* or *unencrypted*. In addition, any host that does not use an encryption package must be set up to communicate with you in the clear.

Communicating Using SKIP Version 1

Complete the following steps to set these fields for encrypted traffic between your server and the system to be authorized.

1. **After selecting the type of system and setting the security to SKIP, enter the Hostname.**

2. **Enter the Node ID.**

   This is the IPv4 key ID.

3. **Local Key ID and ID buttons.**

   Use the Local Key ID button to indicate whether you want your local system to send its key ID in the SKIP packet.

4. **Set the Tunnel Address, if you are using topology hiding.**

   Tunnel addressing is generally used for clients of encrypted gateways where the IP address of the host entered here serves as the intermediary for any or all hosts on a network whose topography is to remain unknown or hidden from the rest of the world.

5. **Select the appropriate key and traffic algorithms for the Key and Traffic encryption buttons.**

The available key and traffic encryption algorithms depend on the which version of SKIP you are using: 512 bitl, 1024 bit, 2048 bit, 4096 bit. To see which options are available, please see Table 2-2, "Available Key and Traffic Encryption Algorithms," on page 67.

Communicating Using SKIP

Complete the following steps to set these fields for encrypted traffic between your server and the system to be authorized.

1. **After selecting the type of system and setting the security to SKIP, enter the Hostname.**

2. **Set the Secure button to either Whole packet (tunnel mode) or Data only (transport mode).**

   Whole packet is recommended because it offers a greater degree of security.

3. **Set the Tunnel address, if you are using topology hiding.**

   Tunnel addressing is generally used for clients of encrypted gateways where the IP address of the host entered here serves as the intermediary for any or all hosts on a network whose topography is to remain unknown or hidden from the rest of the world.

4. **Use the Remote Key ID button to select whether you would like the remote system's keyID included in SKIP packets.**

   If so, what namespace does that key occupy. By selecting *Not Present*, the receiver key ID is not sent.

   Not Present is the default. It uses the IP address of the remote system to identify its certificate. If a remote system has a key ID other than identified by its IP address, set the namespaces and indicate the remote system's key ID in the ID Field. The namespace indicated in the Remote Key ID field is determined by the type of certificate that is used or obtained for this system. The type of certificate and the Remote Key ID field for that certificate is shown in the following table:

TABLE 3–3 Type of Certificate and Remote Key ID Field

Certificate Type	Remote Key ID Field
CA (Sun or other)	IPv4
Self-generated unsigned key	MD5 (DH Public Value)

5. **The following namespaces are used in this menu:**

Not present	IPv4 Address	MD5 (DH public Value)

6. **If the Remote Key ID field has been set to something other than Not Present, enter the key ID in hexadecimal format in the ID field (0x0a000000).**

   It must contain the appropriate key ID for the system that is being authorized based upon the selection made in the *Remote Key ID* field. Depending on the type of certificate, this information may be obtained from the master keyID on the diskette or from the *Local key ID* field of the other host.

7. **Select the appropriate key and traffic algorithms for the Key and Traffic encryption buttons.**

   The available options that appear depend on which version of SKIP you are using. versions Available options include: 512 bit, 1024 bit, 2048 bit, or 4096 bit. See Table 2-2, "Available Key and Traffic Encryption Algorithms," on page 67 for more information.

8. **Authentication button.**

   Use the authentication button to select the type of authentication for the packets. Currently, SunScreen supports MD5 and MD5-NAT. You can also select *None* for no authentication.

Excluding Systems

If the *default* entry remains on the authorized systems list, any remote host with which you want to exclude must be configured using the *Add* button located under the *excluded systems* list. When setting up an excluded system, you only need to enter the hostname for hosts and network number for networks. For nomadic systems you need to specify the key IDs.

If the state of the host or network changes to an authorized system, you must delete the system from the excluded systems list and add it to the authorized systems list.

The easiest way to exclude a system is to move it from the authorized systems list with the arrow button to the excluded systems list. The arrow buttons make it easy to add or delete systems when troubleshooting and the host is already present in the authorized systems list. If the host does not already exist on one of the lists, it is simpler to add it directly on the excluded systems list so that you can move it easily with the arrow button when you wish to add it to the authorized systems list.

Note – If you move an encrypted host from the authorized systems list to the excluded systems list with the arrow button, SunScreen retains the encryption parameters so that if you later move this host back to the authorized systems list, its parameters are restored.

▼ Adding Excluded Systems

Complete the following steps to exclude a system:

1. **Click on the Add button at the bottom of the excluded systems list on skiptool's main window.**
2. **Select the system type: Host, Network, or Nomadic.**
3. **In the Hostname field on the Exclude System window, enter the name or IP address of the host system that you want to deny access to your system.**

   If you are excluding a nomadic system, also enter the key ID.
4. **Click Apply on the Exclude System window.**

Caution – If you add or delete ACL entries from one list to another, the addition or deletion takes effect immediately.

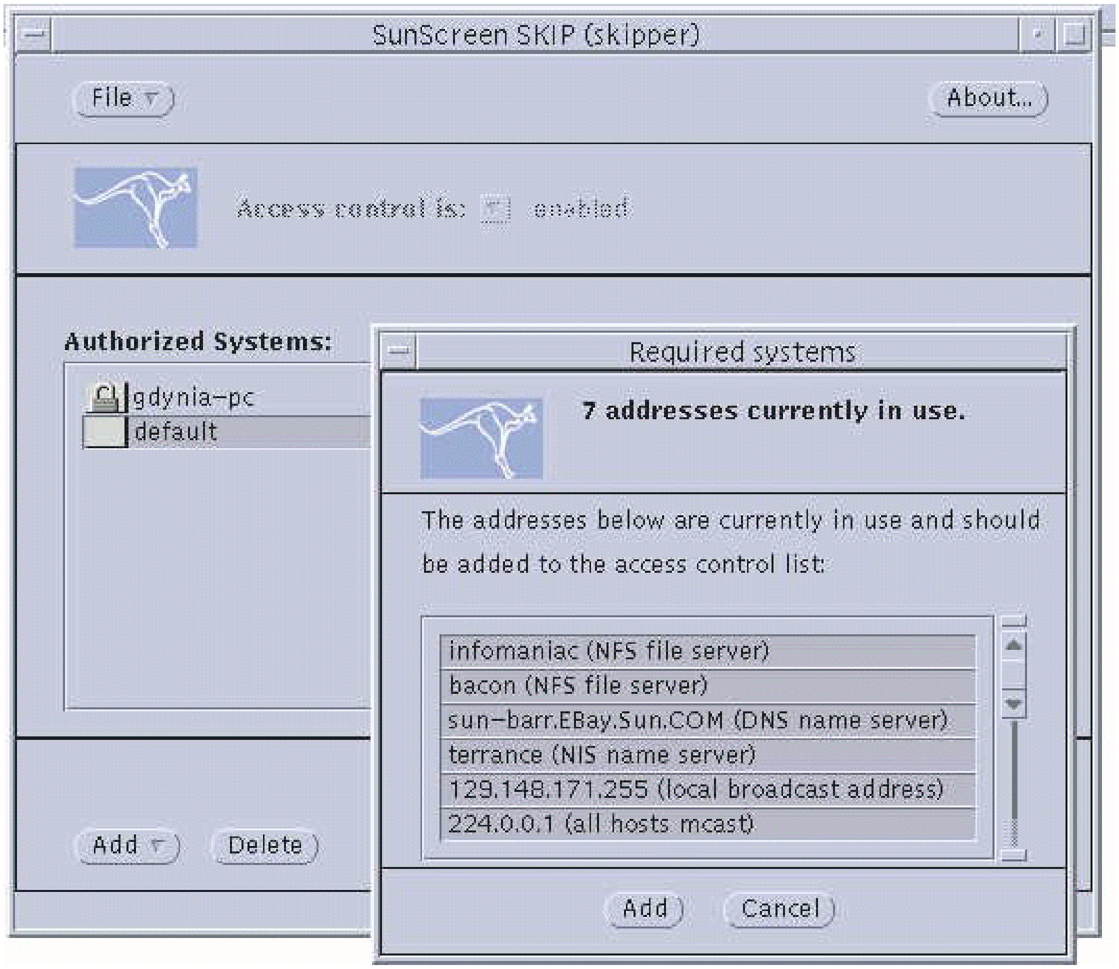
Enabling SKIP

The last step in setting up SunScreen is to enable access control for the system. You may also wish to understand the different symbols in the Authorized Systems list and know how to iconify `skiptool`.

▼ Enabling Access Control

● **Enable SunScreen by selecting enabled from the Access Control button on the main window.**

When SKIP is enabled for the first time, it checks for all systems with which you are talking in the clear. It detects the NFS, X Windows, NIS, and DNS servers with which you are communicating and offers the possibility of adding the systems automatically to the ACL when you select Add from the Required Systems window, shown in the following figure. Choosing Cancel can hang your system or prevent your access to the system or network the next time you try to log in because certain necessary servers may not have been added. To prevent this, select disable after canceling.



7 addresses currently in use.

The addresses below are currently in use and should be added to the access control list:

infomaniac (NFS file server)
bacon (NFS file server)
sun-barr.EBay.Sun.COM (DNS name server)
terrance (NIS name server)
129.148.171.255 (local broadcast address)
224.0.0.1 (all hosts mcast)

FIGURE 3–6 Enabling SKIP

Understanding the Symbols in the Authorized Systems List

The authorized systems area lists all the hosts that are allowed access. The excluded systems area shows all those known hosts that are explicitly denied access. The graphic preceding the host name or IP address depicts what type of security is being used with that host.

- A blank box preceding the host name indicates no encryption (Security = Off).

- A box with a lock in it indicates that the system is using SKIP as the encryption method (Security = SKIP).
- A box with the Sun Microsystems' logo in it indicates that the system is using SKIP Version 1 (Security = SKIP version 1).
- A box with an N indicates a system that is Nomadic (that is, it is identified by its key ID not its IP address) and that it is using either SKIP or SKIP Version 1 as the security method.

Iconify SunScreen

Once you have enabled SunScreen, it is no longer necessary to keep the window open. At this time, you can iconify the main window. The `skiptool` icon, which is shown in the following figure, shows SKIP's status. If you have set unauthorized systems to *No Access*, you can quit `skiptool`.

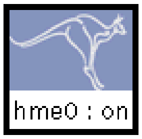


FIGURE 3–7 SKIP Icon Showing Both the Enabled and Disabled States

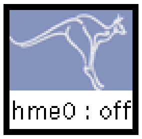


FIGURE 3–7 SKIP Icon Showing Both the Enabled and Disabled States

If you quit the application, SKIP stays in whatever mode it was last in (enabled or disabled).

Unauthorized Systems automatically changes to *No Access*, since there is no longer any way to notify you if an unauthorized system attempts to gain access.

Is SKIP Working?

Once you have configured and enabled SKIP, you may want.to determine that it is working properly. If the configurations on the systems do not match (that is, the encryption algorithms used), it will appear as if the other part of the communication equation does not exist. SKIP silently drops the packets. `skiplog` will log this event.

To verify that SunScreen is operating properly on your system, complete one or more of the following procedures:

1. **Ping the remote system.**

   The remote system must have SunScreen enabled and be using the same key and traffic encryption algorithms as your system.

   If you have the remote site's certificate, you can immediately start sending encrypted IP. Otherwise, SKIP will need to fetch the remote machine's certificate. By default, this is done by asking the remote site for its certificate over a clear channel. If you have configured other hosts to act as key servers, they will be asked for the certificate. See the `man` pages for `skipd` and `skipd.conf` for details. If there are no problems at the remote site, you receive replies when you ping.

   Note – The initial ping can fail because the key manager's computation may exceed the time-out value of some of the IP protocols, such as ping.

2. **Run `snoop` on your local system or a sniffer to see that packets are being encrypted.**

   If encryption is not taking place between your system and a system on your authorized systems list or you cannot connect to that system, check the following items.

   - Is SKIP enabled? Check the Access Control button. Set it to *enabled*.
   - Verify that a certificate exists for each system you wish to communicate with on your authorized systems list. Use the `skipdb` command to check for the certificate of the remote system by dumping the database to the screen. Try to restart the key manager by using the `skipd_restart` command.
   - Verify that SKIP is installed, configured, enabled, and has the certificate of the remote system.
   - Verify the key ID of the remote system in the log file `/var/log/skipd.log` to see if the key manager has set the key ID to what you think it should be. If it is not the correct key ID, get certificates for the correct key ID.
   - Verify that both machines have the same key encryption, traffic encryption, and authentication algorithms. You can check which ACL entry will be used when communicating with a remote host by using `skiphost <hostname/IP`

address> command. This command will check default entries, as well as network entries.

- Certificate Discovery works by sending UDP requests to port 1640 of the server. If you are connecting through a firewall, check with your system administrator that UDP messages are allowed to pass on port 1640. These ports are required for the certificate discovery protocol (CDP). As a workaround, you can manually distribute keys. Also, make sure that the SKIP protocol 57 (decimal number) and the SKIP Version 1 protocol 79 (decimal number) are permitted to pass through the firewall.
- Some routers also filter packets. Check on the router and its configuration.
- Verify that the CDP server specified in skipd.conf is correct and has been authorized in skiptool. If the cdp_server entry is = or @, it is specifying the tunnel address or host address, respectively.
- SKIP requires that machine clocks be synchronized within one hour. Make sure they are synchronized. Messages in /var/log/skipd.log will indicate this situation. You can use the UNIX command rdate (1M) to synchronize the clocks.
- If the skiplocal -x command has been used to communicate key IDs when one or both of the systems have multiple keys or multiple network interfaces, the key ID may have been bound to the wrong network interface or local key ID. Use skiptool or skiphost to add the remote host after verifying key IDs over the telephone.
- Use skiplog to verify configuration mismatches.

Viewing SunScreen Statistics

SunScreen provides two methods of viewing statistics: skiptool and skipstat (the command-line interface for viewing SKIP statistics and is discussed in Chapter 4.) The method you choose is a matter of personal preference since both interfaces provide the same data. The skiptool display has the word UPDATED in front of fields whose values have changed since the last “sampling.” This feature is not available through skipstat.

The following statistics are available in SunScreen:

- Network Interface Statistics
- SKIP Header Statistics
- Key Statistics
- Encryption Statistics (for Versions 1 and 2)
- Authentication Statistics

The Statistics Window

You can view the Network Interface, SKIP Header, Key, Encryption (Versions 1 and 2), and Authentication statistics in real-time by selecting SKIP Statistics from the File menu (File —> SKIP Statistics) on the skiptool main window, shown in the following figure.

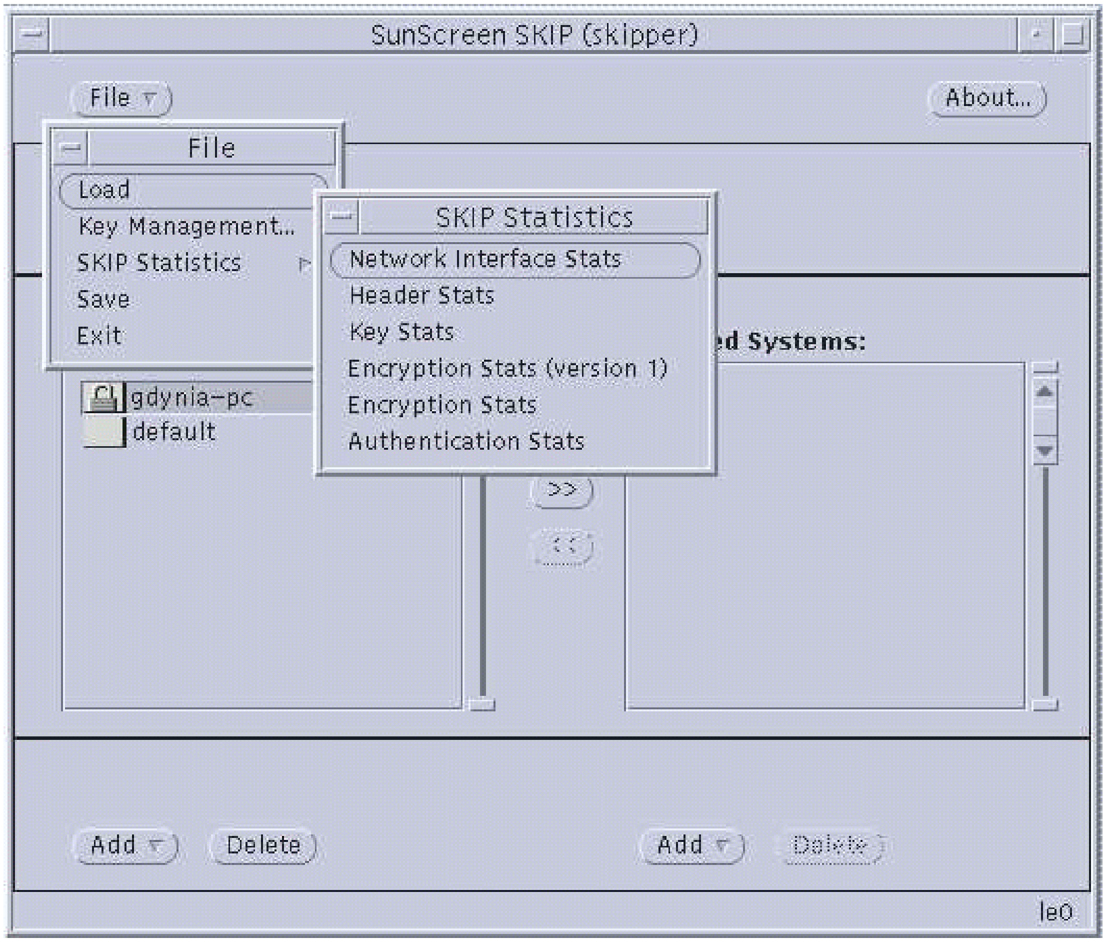


FIGURE 3–8 Bringing Up a Statistics Window

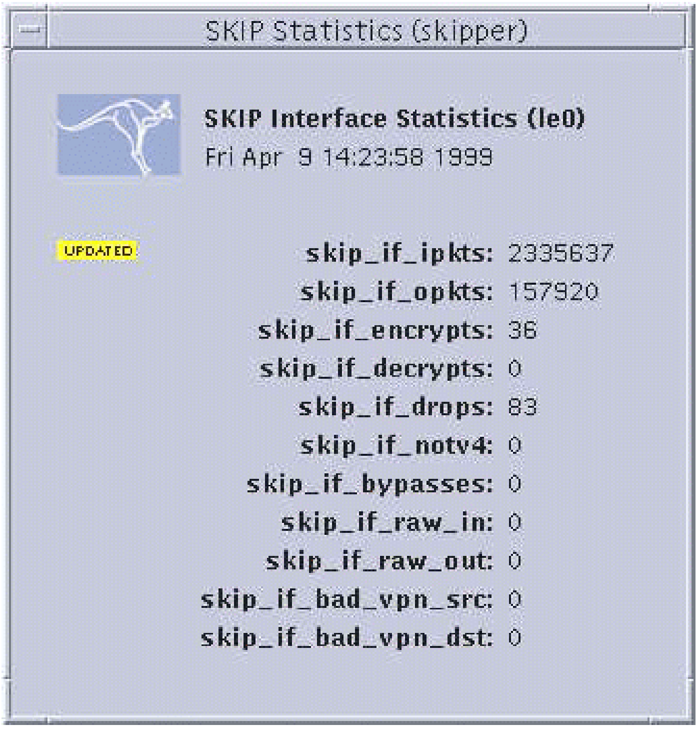
Each of the statistics available for SunScreen is described on the following pages. Sample data with field descriptions illustrate the information available for monitoring SunScreen's performance. The fields on the statistics screens are updated approximately every 3 seconds. A status change is indicated with the word `UPDATED` next to the fieldname.

SKIP Statistics

Interface Statistics

Selecting File —> SKIP Statistics —> Network Interface Stats displays the SKIP Interface Statistics window, shown in the following figure.

.



SKIP Interface Statistics (le0)
Fri Apr 9 14:23:58 1999

```
     skip_if_ipkts: 2335637
     skip_if_opkts: 157920
  skip_if_encrypts: 36
  skip_if_decrypts: 0
     skip_if_drops: 83
     skip_if_notv4: 0
  skip_if_bypasses: 0
    skip_if_raw_in: 0
   skip_if_raw_out: 0
skip_if_bad_vpn_src: 0
skip_if_bad_vpn_dst: 0
```

FIGURE 3–9 SKIP Interface Statistics Window

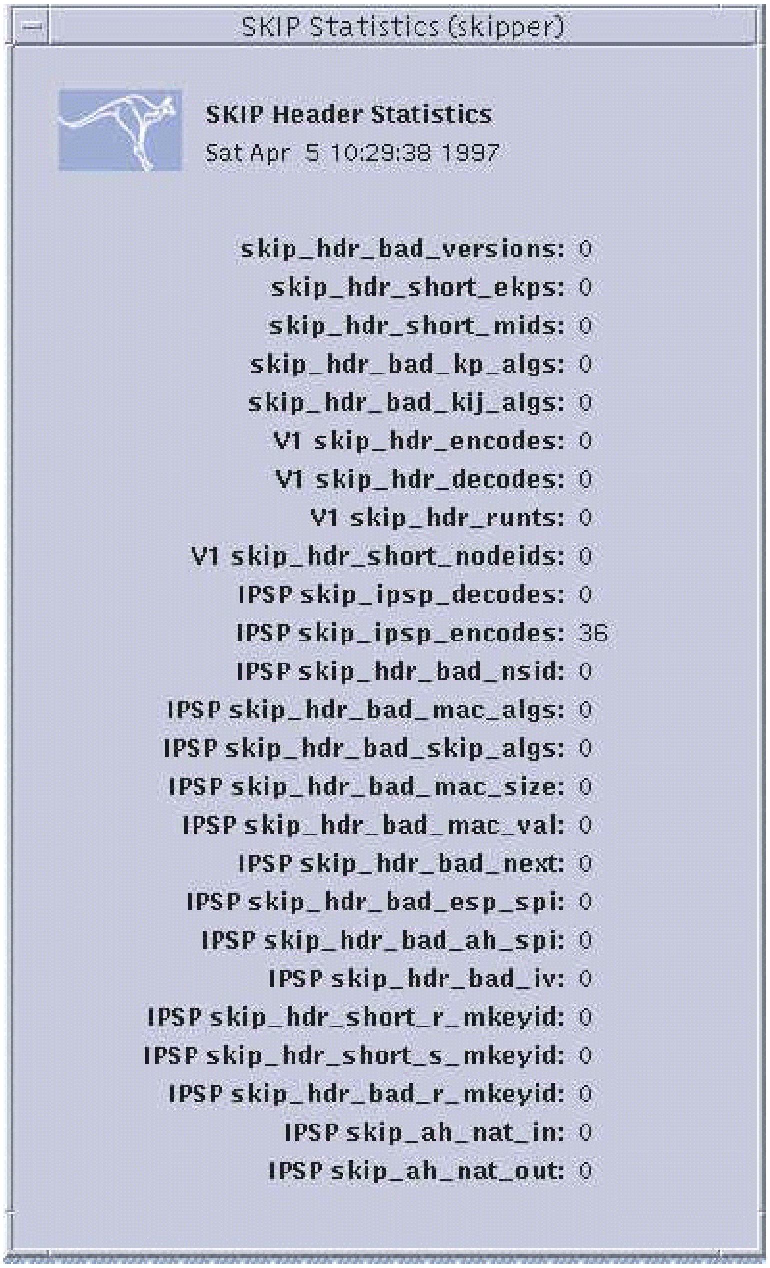
A brief description of each field is given below:

`skip_if_ipkts`	Packets received by the interface.
`skip_if_opkts`	Packets sent by the interface.
`skip_if_encrypts`	Packets encrypted.
`skip_if_decrypts`	Packets decrypted.

`skip_if_drops`	Packets dropped.
`skip_if_notv4`	Packets that are not IPv4 packets.
`skip_if_bypasses`	Number of certificate packets.
`skip_raw_in`	Number of non-SKIP IPSEC packets received.
`skip_raw_out`	Number of non-SKIP IPSEC packets sent.
`skip_if_bad_vpn_src`	Number of incorrect source tunnel addresses.
`skip_if_bad_vpn_dst`	Number of incorrect destination tunnel addresses.

Header Statistics

Selecting File —> SKIP Statistics —> Header Stats displays the Header Statistics window, shown in the following figure. In the field descriptions below, V1 refers to SKIP Version 1.



SKIP Statistics (skipper)

SKIP Header Statistics

Sat Apr 5 10:29:38 1997

```
         skip_hdr_bad_versions: 0
          skip_hdr_short_ekps: 0
          skip_hdr_short_mids: 0
         skip_hdr_bad_kp_algs: 0
        skip_hdr_bad_kij_algs: 0
         V1 skip_hdr_encodes: 0
         V1 skip_hdr_decodes: 0
            V1 skip_hdr_runts: 0
   V1 skip_hdr_short_nodeids: 0
     IPSP skip_ipsp_decodes: 0
     IPSP skip_ipsp_encodes: 36
     IPSP skip_hdr_bad_nsid: 0
IPSP skip_hdr_bad_mac_algs: 0
IPSP skip_hdr_bad_skip_algs: 0
IPSP skip_hdr_bad_mac_size: 0
  IPSP skip_hdr_bad_mac_val: 0
     IPSP skip_hdr_bad_next: 0
 IPSP skip_hdr_bad_esp_spi: 0
   IPSP skip_hdr_bad_ah_spi: 0
       IPSP skip_hdr_bad_iv: 0
IPSP skip_hdr_short_r_mkeyid: 0
IPSP skip_hdr_short_s_mkeyid: 0
IPSP skip_hdr_bad_r_mkeyid: 0
        IPSP skip_ah_nat_in: 0
       IPSP skip_ah_nat_out: 0
```

FIGURE 3–10 SKIP Header Statistics Window

A brief description of each field in SKIP Header Statistics window is given below:

skip_hdr_bad_versions	The number of headers with invalid protocol versions.
skip_hdr_short_ekps	The number of headers with short ekp fields.
skip_hdr_short_mids	The number of headers with short MID fields.
skip_hdr_bad_kp_algs	The number of headers with unknown cryptographic algorithms.
skip_hdr_bad_kij_algs	The number of headers with unknown key encryption algorithms
V1 skip_hdr_encodes	The number of SKIP V1 headers encoded.
V1 skip_hdr_decodes	The number of SKIP V1 headers decoded.
V1 skip_hdr_runts	The number of headers with short SKIP V1 packets.
V1 skip_hdr_short_nodeids	The number of headers with short SKIP V1 key ID.
IPSP skip_ipsp_decodes	The number of SKIP headers decoded.
IPSP skip_ipsp_encodes	The number of SKIP headers encoded.
IPSP skip_hdr_bad_nsid	The number of headers with a bad SKIP name-space ID.
IPSP skip_hdr_bad_mac_algs	The number of headers with unknown or bad authentication algorithms.
IPSP skip_hdr_bad_skip_algs	The number of bad SKIP algorithms.
IPSP skip_hdr_bad_mac_size	The number of headers with an authentication error in the MAC size.
IPSP skip_hdr_bad_mac_val	The number of headers with an authentication error in the MAC value.
IPSP skip_hdr_bad_next	The number of headers with a bad SKIP next protocol field.
IPSP skip_hdr_bad_esp_spi	The number of headers with a bad SKIP SPI field.
IPSP skip_hdr_bad_ah_spi_	The number of bad AH/SPI headers (manual keying).
IPSP skip_hdr_bad_iv	The number of headers with a bad SKIP initialization vector.
IPSP skip_hdr_short_r_mkeyid	The number of headers with a short SKIP receiver key ID.

`IPSP skip_hdr_short_s_mkeyid` The number of headers with a short SKIP sender key ID.

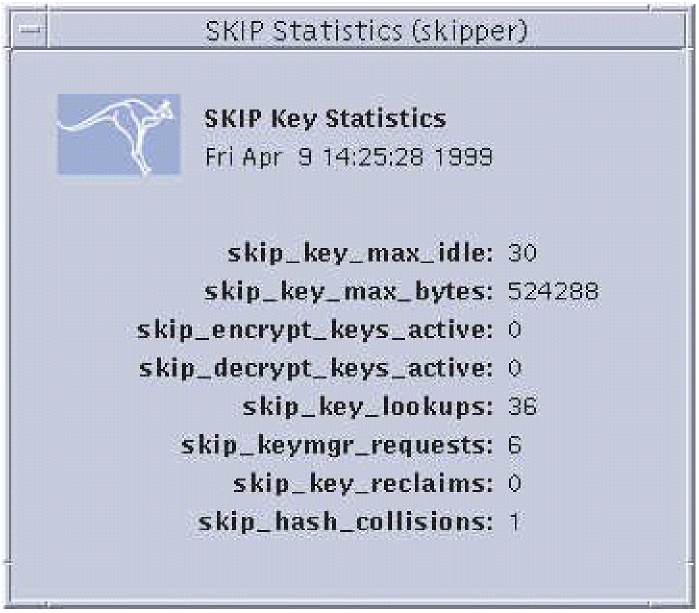
`IPSP skip_hdr_bad_r_mkeyid` The number of headers with a bad SKIP receiver key ID.

`skip_ah_nat_in` MD5-NAT packets received.

`skip_ah_nat_out` MD5-NAT packets sent.

Key Statistics

Selecting File —> SKIP Statistics —> Key Stats displays the Key Statistics window, shown in the following figure.



SKIP Statistics (skipper)

SKIP Key Statistics

Fri Apr 9 14:25:28 1999

```
      skip_key_max_idle: 30
     skip_key_max_bytes: 524288
skip_encrypt_keys_active: 0
skip_decrypt_keys_active: 0
        skip_key_lookups: 36
    skip_keymgr_requests: 6
       skip_key_reclaims: 0
     skip_hash_collisions: 1
```

FIGURE 3–11 SKIP Key Statistics Window

A brief description of each field on the Key Statistics window is given below:

`skip_key_max_idle` The time, in seconds, until an unused key is reclaimed.

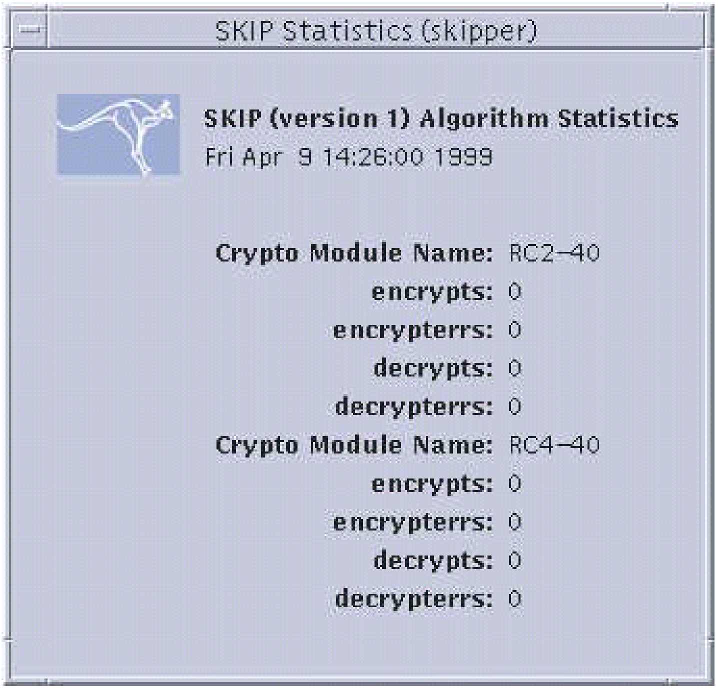
`skip_key_max_bytes` Maximum number of bytes to encrypt before discarding a key.

`skip_encrypt_keys_active`	Number of encryption keys in the cache.
`skip_decrypt_keys_active`	Number of decryption keys in the cache.
`skip_key_lookups`	The total number of key cache lookups.
`skip_keymgr_requests`	The total number of key cache misses (key not found).
`skip_key_reclaims`	The total number of key entries reclaimed.
`skip_hash_collisions`	The total number of table collisions.

Algorithm Statistics

Selecting File —> SKIP Statistics —> Encryption Stats (Version 1) displays the Algorithm Statistics window for SKIP Version 1 as is shown in the following figure.

Selecting File —> SKIP Statistics —> Encryption Stats displays the standard Algorithm Statistics window, as is shown in the following figure.

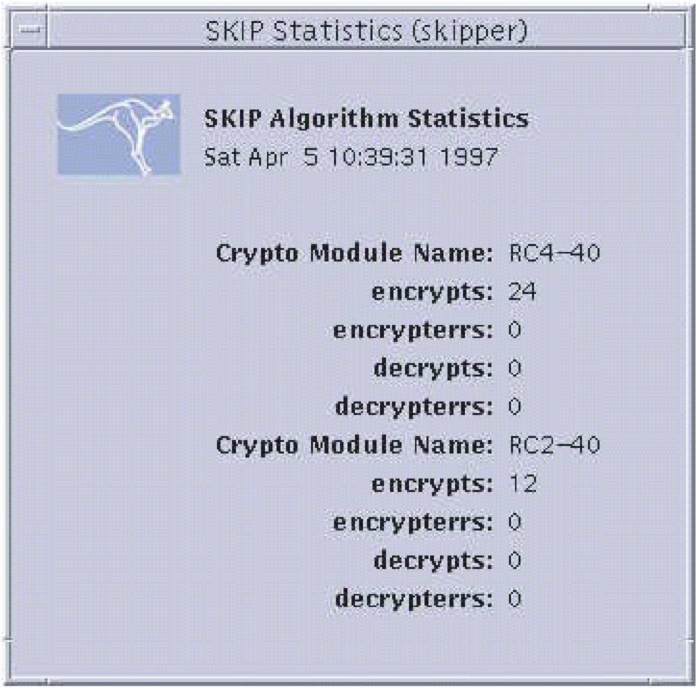


SKIP Statistics (skipper)

SKIP (version 1) Algorithm Statistics
Fri Apr 9 14:26:00 1999

```
Crypto Module Name: RC2-40
          encrypts: 0
       encrypterrs: 0
          decrypts: 0
       decrypterrs: 0
Crypto Module Name: RC4-40
          encrypts: 0
       encrypterrs: 0
          decrypts: 0
       decrypterrs: 0
```

FIGURE 3–12 Encryption Statistics Window—SKIP Version 1 and Standard SKIP


SKIP Statistics (skipper)

SKIP Algorithm Statistics
Sat Apr 5 10:39:31 1997

```
Crypto Module Name: RC4-40
          encrypts: 24
       encrypterrs: 0
          decrypts: 0
       decrypterrs: 0
Crypto Module Name: RC2-40
          encrypts: 12
       encrypterrs: 0
          decrypts: 0
       decrypterrs: 0
```

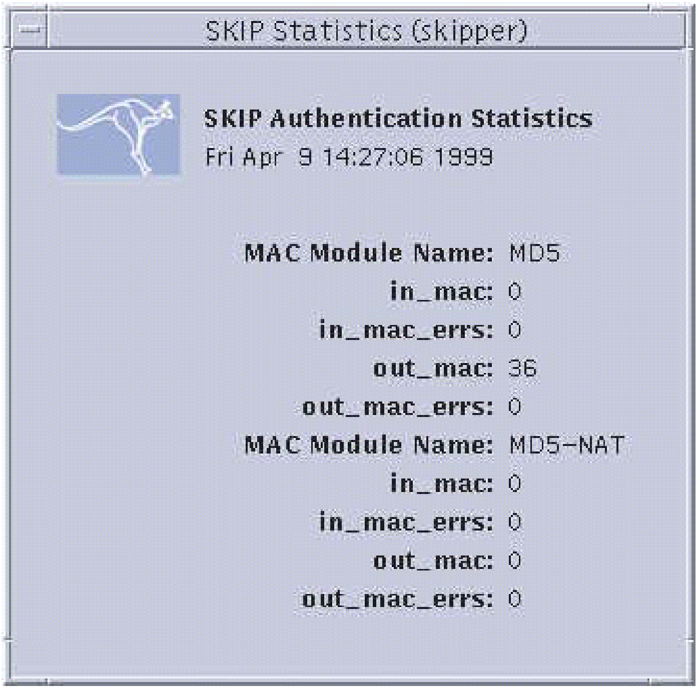
FIGURE 3–12 Encryption Statistics Window—SKIP Version 1 and Standard SKIP

One set of statistics is displayed for each different traffic and key encryption module. A brief description of each field is give below:

`Crypto Module Name`	The name of the cryptographic module for which the statistics are being displayed.
`encrypts`	Number of successful encryptions.
`encrypterrs`	Number of failed encryptions.
`decrypts`	Number of successful decryptions.
`decrypterrs`	Number of failed decryptions.

Authentication Statistics

Selecting File —> SKIP Statistics —> Authentication Stats displays the Authentication Statistics window, shown in the following figure, which provides information on MACs (Message Authentication Code).



SKIP Statistics (skipper)

SKIP Authentication Statistics
Fri Apr 9 14:27:06 1999

MAC Module Name:	MD5
in_mac:	0
in_mac_errs:	0
out_mac:	36
out_mac_errs:	0
MAC Module Name:	MD5-NAT
in_mac:	0
in_mac_errs:	0
out_mac:	0
out_mac_errs:	0

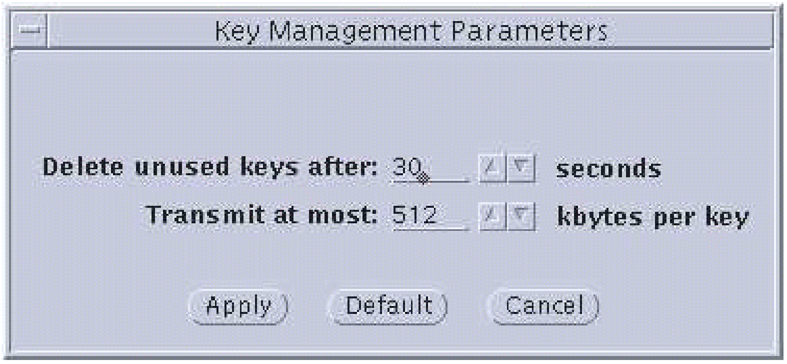
FIGURE 3–13 Authentication Statistics Window

A brief description of each field on the Authentication Stats window is given below:

`MAC_Module_Name`	MAC method used for authentication.
`in_mac`	Number of received MAC calculations that succeeded.
`in_mac_errs`	Number of received MAC calculations that failed.
`out_mac`	Number of sent MAC calculations that succeeded.
`out_mac_errs`	Number of sent MAC calculations that failed.

Key Management with `skiptool`

The Key Management Parameters window, The following figure is displayed by selecting File —> Key Management. Key management parameters are global; that is, one set of key management parameters governs the activity of all keys on a particular system. They determine when a key is deleted based upon use and the maximum number of bytes transmitted per encrypt key.



Delete unused keys after: 30 seconds

Transmit at most: 512 kbytes per key

FIGURE 3–14 Key Management Parameters Window

The *Key Management Parameters* window has four major components.

Change transmit keys every: The system uses the delete unused key parameter to decide when to change active encrypt keys.

*Delete unused keys after:*This button sets the number of seconds an unused traffic key is kept before it is deleted. The number may be changed by either typing in a new number or clicking on the up and down arrows until the desired number is reached. Default value = 30 seconds. Valid range: 5 seconds to 10,000 seconds.

Transmit at most: This button sets the maximum amount of information that can be transmitted using a particular key. When the set amount is reached, the key is changed. The number can be changed by either typing in a new number or clicking on the up and down arrows until the desired number is reached. Default value = 512 Kbytes per key. Valid range: 1 Kbyte per key to 10,000 Kbytes per key.

Management Buttons: These three buttons enable you to apply the new values, return to the default values, or dismiss the window without changes:

- *Apply*—Makes the changes made in the window active.
- *Default*—Returns the values in the window to the default values (30 seconds and 512 Kbytes).

- *Cancel*—Dismisses the window without changing anything.

CHAPTER 4

Using the Command-Line Interface

This chapter describes how to use the command-line interface.

Note – To use the command-line interface, you must be logged in as `root`.

SKIP Command-Line Interface

The SunScreen command-line interface commands follow, including a brief description of what they do. Many of these commands duplicate what can also be done using the GUI, while others are enabling commands for other commands. For a more complete discussion of the command-line interface, refer to the man pages for SunScreen.

`print_cert`	Prints a certificate to standard output.
`certreq`	Requests and retrieves a certificate from a key server or other host.
`install_skip_keys`	Installs a private key and certificate received from a key server or from the SunCA.
`skipca`	Manages the SKIP Certificate Authorities Database. It is used to add, delete, or list CAs.

`skipd`
It is not a user command, but a system process not normally start by the user.The skipd daemon is started at system boot, and restarted when necessary with the skipd_restart command. Only one key manager may be running at a time. The key manager must be started by root.

`skipd.conf`
This is not a command but the SKIP Key Manager configuration file.

`skip_conf`
Changes the `skipd.conf` configuration parameters.

`skipd_restart`
Kills the existing running SKIP key-management daemon (`skipd`) and starts a new one. It is used after any changes in key configurations to make them permanent.

`skipdb`
Administers the SKIP database of certificates. SKIP stores the long-term certificates in the database so that the key manager can have access to them.

`skiphost`
Lists, adds, or deletes host, network, or nomadic (mobile) system information from SKIP's ACL. `skiphost` can be also used to enable or disable SKIP.

`skipif`
Adds or removes SKIP from the network interfaces. It is also used to save ACL status.

`skiplocal`
Used to manage the SKIP local keys for the workstation. It is used to add, delete, or print local keys.

`skiplog`
Displays security events for the local system.

`skipstat`
Displays statistical information about the use of SKIP on the local system.

`skipvar`
Allows you to displays and edit SKIP internal keystore variables.

Using the Command-Line Interface

print_cert: Printing a Certificate to Standard Output

print_cert prints the contents of the certificate found in the certificate file specified. You can specify the type of certificate—the types of certificates supported are X.509 and UDH. The default is X.509.

Syntax

```
print_cert -[V|t]
```

Options

-V	Prints the output in a machine readable form.certificate-type.
-t	Specifies the type of the certificate provided. Supported types x509 (1) and udh (4).

certreq: Retrieving a Certificate From a Key Server

certreq is a maintenance command. It requests and retrieves a certificate from a key server or other host. You must specify the key ID and key server. This command is a debugging tool and is not meant for general use. The interface is cryptic and there is no way to specify a host name or IP address instead of the key ID, even if the key ID is identical to the IP address.

Syntax

```
certreq [-d] [-n
NSID] [-h server] [keyid]
```

Options

`-d`	Requests that the received certificate be decoded prior to output. Without the -d option, the raw certificate, suitable for saving into a file, is written to standard out.
`-n NSID`	By default NSID 1 is used for retrieves. Any NSID number may be specified with the -n option.
`-h server`	Specifies the keyserver.
`keyid`	The Master Key ID specified in hex.

`install_skip_keys`: Installing Keys and Certificates From a Certificate Authority

`install_skip_keys` installs keys received from a key server (default) or from the SunCA (if `-icg` is specified). If you are installing a key package from a key server, the filename specifies the name of that package. The key file is a pretty good privacy (PGP) or an encoded file containing: a Diffie-Hellman private key, a Diffie-Hellman signed public key, the common Diffie-Hellman parameters used by the certificate issuer, the certificate issuer's signed public key, and a MD5 checksum of the other four files. The filename is an encoded `tar` file usually received from a key server or other certificate issuer.

If you are installing a SunCA certificate, the filename is the name of the directory that contains the files. This is usually a diskette, so the path will often be similar to

```
/floppy/floppy0
```

`install_skip_keys` verifies the MD5 checksums of the individual files with the checksum file. If they match, the files are copied into place.

The key manager must be restarted (see `skipd_restart`) in order for it to recognize the new keys.

Currently, the name of the certificate is hard coded into the code. Certificates are expected to come from the SKIP experimental Zero Assurance Certificate Issuer or the SunCA. Even if they do not, the certificate will have to be called `ZeroAssurance_Cert`. This release does not support multiple certificate issuers.

Syntax

```
install_skip_keys [-icg] filename
```

Options

`-icg filename`	The filename is the name of the directory that contains the files.

`skipca`: Setting Up Trusted CAs

Certificates are the digital documents that testify to the binding of a public key to an individual or other entity for the purpose of preventing someone else from impersonating you. In order for two hosts running a security package to communicate, they must exchange certificates. The `skipca` command-line interface is used to designate a CA as trusted and to manage that database. `skipca` options are add, extract, init, list, delete, create, and revoke CA certificates.

You must restart the key manager with `skipd_restart` before any changes will take effect.

This command has broad security implications. By designating a CA, you are trusting the identity of *all* certificates signed by that CA. Since root CA certificates are self-signed, there is no automated way to verify that a CA certificate actually comes from that CA. Before adding a CA certificate, you *must* be absolutely certain that the certificate is valid. Validity may be checked by having the CA publish the hash of its certificate publicly and comparing that hash with the hash obtained from the certificate.

Syntax

```
skipca -[a|r|l|i|e|R|U] [...]
```

Options

`-a [-c ca-file]`	The add option places new certificates into the trusted Certificate Authority database. The ca-file is an X509 certificate which is either self-signed or signed by an existing trusted CA in this CA database. Note: The add option does not copy over a CA certificate if it already exists in the CA database.

`-e [-s ca-slot]`

The extract command writes the CA certificate in the specified slot-number to the standard output. If the output is redirected to a file, the file is suitable for the skipca -a command.

`-i [-qo]`

Prior to use, the CA database must be initialized. The init option creates the database. The init option does NOT delete any of the CA certificates present when issued for an existing database. Use the init option with the -o operand to forcibly reinitialize the data base, destroying any existing certificates. The init option with the -q operand tells init to be as quiet as possible about initialization.

`-l [-VvxL] [-s ca-slot]`

The list option provides a listing of all the certificates in the CA database by slot number, Issuer, and Subject. If a slot number is specified, only the CA Certificate for that slot is printed. The -L flag enables printing of the Certificate validity periods. -v enables a verbose display of the entire certificate. If -V is specified, the output is displayed in a machine parseable manner. If -x is specified the manual revocation list for that CA is display.

`-R [-s ca-slot] [-S serialnumber]`

Each CA maintains a list of certificates which have been revoked by the user. This is different from a traditional CRL as it is not distributed by the CA and is manually maintained. The revoke command allows the user to add certificates to the per-CA list of revoked certificates. ca-slot specifies which CA to operate on. The ca-slot may be obtained through the skipca -l command. serial-number is the serial number of the certificate which you wish revoke. Each X509 Certificate produced by a CA is numbered uniquely with a serial number.

`-U [-s ca-slot] [-S serialnumber]`

The unrevoke command removes hosts from the per CA revocation list. ca-slot and serial-number are the same as the arguments for the revoke command.

`-r[-s ca-slot]`

The rm option deletes the CA certificate in the specified slot number.

`skipdb`: Managing Keys and Certificates

`skipdb` is used to manage certificates. Long-term certificates are stored in a database for access by the key manager. The `skipdb` command allows the manual administration of the certificate database.

X.509 certificates without proper signatures will not be added to the `skipdb` database. The CA's certificate must be added to the CA certificate database using the `skipca` command before adding certificates signed by that CA to the `skipdb` database.

Unsigned public keys will be added with the appropriate hash of the contents as the name.

Syntax

```
skipdb -[a|r|l|i|e|C]
 [action specific arguments]
```

Options

`-a [-t certtype] [-n nsid] [-c filename]`

Adds certificates to SKIP certificate database. The certtype argument sets the type of the certificate to be added. Certificate types are X509 and udh (unsigned Diffie-Hellman). The nsid argument is a decimal number which corresponds to the namespace of the certificate. Common NSID values are 1 (ipv4) and 8 (udh). Filename is the certificate file you wish to add to the database.

`-e [-n nsid] [-k keyid]`

Extracts a certificate to the standard output. The first certificate which matches nsid and keyid will be written. The extracted form is suitable for addition to a database using the skipdb -a command. This subcommand writes only one certificate to the standard output, even if there are multiple certificates which match the nsid, keyid pair.

`-i [-qo]`

Prior to being used, the certificate database must be initialized through the init subcommand. If the database exists, the -o option will delete the contents of the database. The -q option suppresses warning messages.

`-l [-VvL] [-n nsid] [-k keyid]`

Lists the certificates in the Certificate database. -V switches the output to a format more easily parsed by machines. -L lists expiration times along with the Name Space and Master KeyId. -v switches the output to a verbose mode where the entire certificate is printed. -n and -k limit the listing to certificates whose name matches the specified keyid and nsid.

`-r -n nsid -k keyid`

Deletes certificates in the certificate database. Certificates with the specified namespace identifier (NSID) and keyid will be deleted.

`-C`	Checks existence of the certificate database. Returns true upon existence.

`skipd_restart`: Activating the Changes

`skipd_restart` reinitializes the SKIP key manager in order for the changes that you made though `skipca`, `skipdb`, and `skiplocal` to take effect. Any options supplied are passed through to the `skipd` daemon.

Syntax

```
skipd_restart [options]
```

`skiphost`: Setting Up the ACL

The functionality of `skiphost` is the same as the `skiptool` GUI.

Use `skiphost` to list, add, and delete host, network, or nomadic (mobile) systems from the ACL, as well as to enable and disable SKIP. Without arguments, it lists the state of the SKIP interface and authorized or unauthorized hosts, networks, and nomadic systems for the default interface.

The ACL allows the user to configure which remote systems can obtain access to the local host and the type of access granted. Access control is usually based on the IP address of the remote host or by the remote system’s key ID.

Remote systems can be specified either as individual hosts, networks, or nomadic systems.

Hosts are specified by their host name or IP address. Networks of subnetworks are specified by a network address plus a mask similar to that used in subnetworking. Nomadic systems can be specified in SKIP and in SKIP Version 1. They are specified by a key identifier (that is, any IP address with the key ID “*x*”).

The order of processing ACL entries is as follows. A search is made for an ACL entry specifying the remote host. If one exists, it will be used.If no entry containing the IP address can be found, then a search is made for a nomadic ACL entry containing the sender’s key ID in the SKIP protocol header. If one is found and the packet is correctly authenticated, then the sender’s IP address is stored for future reference.

If no corresponding ACL entry can be found for a remote system, the default is used. The default may be configured to allow access or to deny access. This method is similar to the method used by the IP when it is deciding how to route a packet to a destination (that is, host routes take precedence over network routes, and, in the absence of anything better, the default route is used).

When applying access control, the system treats the lists of authorized and excluded systems as a global list and always selects the best match.

A default entry can be specified to indicate all other hosts not specifically covered by other access-control entries.

Note – Before you enable SKIP, any hosts needed for operation of the local system must be present in the ACL. Verify that any NFS file servers, NIS servers, or any local broadcast addresses for your network are on the ACL.

Syntax

```
skiphost -[i|h|o|P|V|f|d|x|a][hostname/IP address][option
specific arguments...]
```

Options

`-i` The -i option takes the interface name as an argument and is used with the -o option to enable or disable SKIP for a particular interface. If this option is not specified skiphost operates on the system's pr mary network interface.

`-f` This option is used to remove (flush) all ACL entries from a given network interface. This option will automatically disable SKIP.

`-h` This option is used to display the SKIP statistics for a given network interface.

`-o` This option enables and disables SKIP. To enable SKIP, use -o on, to disable SKIP use -o off.

`-P` Adding this option to skiphost prints the current access control list in a format which is suitable for execution in a shell script.

`-V` Adding this option to skiphost prints the current access control list in a name=value verbose format.

`hostname/IP address` Takes the -M mask argument. skiphost used without any options, checks if the system hostname or network exists in the access control list and displays its parameters.

`-a`

Adds the hostname or network (specified using the hostname/address -M mask argument) to the access control list and enables traffic between the hosts in the clear. To add hostname or network and enable encrypted and/or authenticated traffic to the host, use the -k, -m and/or -t options. For more arguments, see the description of *.

`-d`

Removes hostname, network or nomadic system from the access control list. Also takes hostname/IP address/* -M mask as well as other option specific arguments. For more arguments, see the description of *.

`-x`

Excludes hostname, network or nomadic system from the access control list.Also takes hostname/IP address/* -M mask as well as other option specific arguments. For more arguments, see the description of *.

`-a ’*’`

This option is used to specify a nomadic system. It must be used in conjunction with the authentication and receiver key ID options. To encrypt and/or authenticate communications with a remote system the following options should be used:

`-k key algorithm`

Specifies the key encryption algorithm or encrypting keys. A list of supported algorithms is available using the skipstat(1M) command.

`-t crypt algorithm`

Specifies the traffic encryption algorithm for encrypting traffic (bulk data).

`-m mac algorithm`

Specifies the authentication algorithm.

`-c comp algorithm`

Specifies the compression algorithm. Not currently implemented.

`-r receiver NSID -R Receiver key ID -s sender NSID -S Sender key ID`

The Key Name Space Identifier (NSID) options (-r and -s) are used to control the identification of keying information in the SKIP protocol. They take numeric values from 0 to 11. The remote key id option (-R) and local key id option (-S) take a hexadecimal value of different lengths, depending on the name space being used. The default NSID values (0, "Not Present") are normally acceptable for most applications. Currently only name spaces 0 ("Not Present"), 1 ("IPv4 address") and 8 ("MD5 DH public values") are supported.

`-v SKIP version`

SKIP can use an old version of the protocol to communicate with SunScreen(tm) SPF-100 and Sun Screen SPF-100G systems. To use this mode, specify the -v 1 option. If no version is specified, skiphost will use SKIP version 2 by default.

`-A tunnel address`

This option is used in tunneling mode to replace the destination address in outgoing packets with the supplied value. This permits hiding of network topology. By default, the tunnel address is set to the destination address.

`-T`

Encrypt or authenticate only the data part of the IP packet. By default, SKIP uses tunneling mode and protects the whole packet.

See the `man` pages for more detail.

`skipif`: Managing Network Interfaces

`skipif` is used to add or delete SKIP from network interfaces. `skipif` is also used to save SKIP’s ACL for a given network interface so that it is permanent across system reboots. In addition, `skipif` is used to list the network interfaces present in the system and optionally to print the current access control configuration for each network interface.

SKIP's ACL for each network interface is stored as a text file (as a series of skiphost commands to be executed during SKIP start-up). SKIP's ACL files are under the /etc/skip directory and the ACL file name for a given interface is acl.<interface name> (for example, acl.le0, acl.hme0, and acl.qe1). If an incorrect or incomplete ACL prevents the system from operating, it may be necessary to modify the file manually or remove the appropriate file. Some non-LAN interfaces (PPP, for example) will not be configured at boot time even if an ACL exists for these interfaces. It is the responsibility of the user in the interface configuration procedure to use the SKIP configuration file for this interface.

skipif notifies the user if it is necessary to reboot the system so that any changes will take effect.

Syntax

```
skipif -[i <ifname|all>|a|d|s|l|h]
```

Options

-i [interface]	The -i option is used to specify the name of the inter face for which the command is applicable. If this option is not specified skipif operates on the system's primary network interface. If the interface name "all" is used, the command will be applied to all the network interfaces present in the system. The loopback inter face "lo0" is excluded.
-a	This option is used to add SKIP to a network interface. The access control list is initialized as empty with SKIP present on the interface but disabled (off).
-s	This option is used to make the current access control list permanent across system reboots. This option must be used with care as an incorrect or incomplete access control list can stop the system from functioning correctly.
-d	Deletes SKIP from a network interface. The network interface is returned to normal non protected operation.

`-l` This option lists all the network interfaces present in the system. If the interface has SKIP added it will be tagged "[skip]". If the access control list for the interface has been modified but not saved, the inter face will be tagged with “[ACL not saved]”. Using the “-v” option will cause skipif to also print the access control list for each interface along with its status.

`-h` This option displays the skipif usage message.

See the `man` pages for more detail.

`skiplocal`: Managing Local Identities

`skiplocal` is the utility for managing SKIP identities on a workstation. A host may wish to have multiple identities if it must interoperate with other hosts that have incompatible Diffie-Hellman parameters (for instance, a U.S. host may wish to communicate with other U.S. hosts with a 1024–bit modulus, but must also communicate with a host outside the U.S. that is limited to a 512–bit modulus). Each local identity has a secret, a certificate, and a unique name. The name is extracted from the certificate and used as a local identity. `skiplocal` is the primary tool for administering local identities. With `skiplocal`, you can create, delete, and list local identities based on the command option specified. When you create a new certificate, its creation date will be assigned as the day before you actually created it. This is a product feature.

You can use `skiploc`al to set or remove a passphrase that is used to encrypt SKIP locally stored secrets. See the -P and -R sections of the command description for more information.

Caution – Beware of electronically transmitting access control commands to remote hosts. For complete security, the receiving system shouldverify the remote key ID out of band.

Note – After adding a local ID, the key manager must be restarted using `skipd_restart`, in order for any changes to take effect.

Caution – `skiplocal -x` does not work well for communicating with multiple keys. Since the local system does not know which key on the remote system should be used, incorrect bindings can occur. Therefore, it is recommended that the `skiplocal -x` command be used carefully.

Syntax

```
skiplocal -[a|r|l|i|e|k|x|P|R]
 [subcommand specific arguments]...
```

Options

Note:

The -d directory specifies an alternate directory to store or retrieve localid information. The default directory is /etc/skip/localid. (This option applies to all the subcommands below.)

`-a [-T slot type] [-t cert type] [-n nsid] [-Z secretfile] [-c certfile]`

The add command is used to add local identities to the trusted Certificate Authority database. All parameters above are required. -T specifies the type of slot. Currently, only "soft", for a software slot, is implemented. -t specifies a certificate type. Currently, x509 and udh are implemented. -n specifies the name space in which the certificate’s name lives. -Z specifies the file containing the Diffie-Hellman private key. -c specifies the certificate used to establish identity.

When a local ID is added, the certificate is checked for validity. Therefore, the local certificate’s CA must have been previously added to the CA database with the skipca command.

If a password has been assigned for encryption of secrets using the skiplocal passwd subcommand, the user will be prompted for that password prior to adding any local ids.

`-R`

The rmpasswd subcommand removes the password which is used to encrypt locally stored secrets. The user is prompted for the old password, and if it matches, all secrets are decrypted and stored and the password feature is disabled.

`-x [-s slot] [-n nsid]`

Creates an “exportable” skiphost command line which could be used to add an access control entry for the local host on a remote system (that is, in the remote /etc/skip/acl.interface file).

By default, -x will choose first slot in the local identity database. A slot may be specified with the -s option. If the -n option is provided, the first slot with an identity in the given namespace will be used.

An attempt is made to determine the local hostname for inclusion in the generated skiphost command. This hostname may be overridden by setting the SKIPLOCAL_EXPORT_HOST environment variable.

The default arguments provided for skiphost specify des for key and traffic encryption, and MD5 for authentication. These arguments may be overridden setting the environment variable SKIPLOCAL_EXPORT_ARGS.

`-r [-v] [-s slot- number]`

Deletes the LocalID in the specified slot number. The control, secret and certificate files are all deleted.

`-l [-Vv] [-s slot-number]`

The list command lists the local ids present on the system. By default, slot number, slot type, NSID, MKID (name) and validity periods are printed. The -v options specifies that the local certificate for that slot should be printed, as well. -V produces output more easily machine parseable.

`-k [-m mod_size] [-E exponent_size] [-L lifetime] [-f] [-V] [-M]`

Generate a new secret key and a UDH (unsigned) certificate and adds them as a new slot to the set of local identities. -V produces output more easily machine parseable.

The -m option specifies the modulus size in bits. Modulus sizes of 512, 1024, 2048, and 4096 bits are supported in the US domestic release. The highest number of bits allowed by the export control limitations of the software is the default. The -L option specifies the lifetime of the udh certificate, in days. The default is 5 years. The -f option suppresses the prompt for keyboard input to obtain better random numbers.

The -E option specifies how large of a random exponent will be generated. The default is 256 bits. The -M option simply reports the modulus of the key that would have been generated. (No key is actually generated.)

If a password has been assigned for encryption of secrets using the skiplocal passwd subcommand, the user will be prompted for that password prior to adding any local ids.

`-e [-s slotnumber]`

The extract command writes the certificate in the specified slot-number to the standard output. If the output is redirected to a file, the file is suitable for the skipdb command.

`-i [-qo]`

Prior to use, the Local ID database must be initialized. The init command creates the database. By default, if the database exists, the init command will NOT delete any of the Local Identities present. The user may force reinitialization the database and destruction of all identities by specifying the -o option. -q tells init to be as quiet as possible about initialization.

`-P`

Assigns or changes the password which is used to encrypt locally stored secrets. If no password as present, you will be prompted for a new one. If a password already exists, you will be prompted for the old password prior to the new one.

See the `man` pages for more detail.

`skiplog`: Viewing Security Events

`skiplog` displays security events for the local system. It displays the types of events presented below. In all cases, the date and time of the event, as well as the IP address information, are logged.

Unknown Source—A packet was received from a system that is not currently in the ACL. The packet is dropped.

Unknown Destination—The local system sent a packet to a system that is not currently in the ACL. The packet is dropped.

E*xcluded Source*—A packet was received from a system explicitly excluded by the ACL. The packet is dropped.

Excluded Destination —The local system sent a packet to a system that was explicitly excluded by the ACL. The packet is dropped.

*Bad Parameter*s—A packet was received that contained security parameters that were incompatible with the ACL entry.

Syntax

```
skiplog [-i interface]
```

Options

`-i [interface]`	Display events for the specified network interface. By default, skiplog displays events for the system's principal network interface.

See the `man` pages for more detail.

`skipstat`: Viewing SunScreen Statistics

`skipstat` is the command-line interface for viewing SKIP statistics. Because `skipstat` is a command-line interface, the information that is displayed does not update on screen with the results of the latest sampling as `skiptool` does.

The following statistics are available in SunScreen:

- SKIP Network Interface Statistics
- SKIP Header Statistics
- SKIP Key Statistics
- SKIP Encryption Statistics (for Versions 1 and 2)
- SKIP Authentication Statistics

Syntax

```
skiplog -[a|C|c|m|k|K|h] [option
specific arguments]
```

Options

`-a`	Displays all information available.
`-C`	Display cryptographic algorithms supported by the local system. Each algorithm is listed with its module identifier and name.
`-c [version]`	Displays cryptographic algorithm statistics for SKIP version; 1= SKIP V1, 2=SKIP
`-m`	Displays MAC algorithms statistics.
`-k`	Displays SKIP key statistics.
`-K`	Displays local key information.
`-h`	Displays SKIP header statistics.

See the `man` pages for more detail.

The following is a breakdown of `skipstat` output for each of the main options:

SKIP Network Interface Statistics

Note – The `skipstat -i` command is no longer supported.

New Command: `skiphost -h`

SKIP interface (le0) statistics:

`skip_if_ipkts:`	number of packets received by interface
`skip_if_opkts:`	number of packets sent by interface
`skip_if_encrypts:`	number of packets encrypted
`skip_if_decrypts:`	number of packets decrypted
`skip_if_drops:`	number of packets dropped
`skip_if_notv4:`	number of non-IPV4 packets
`skip_if_bypasses:`	number of certificate packets
`skip_if_raw_in:`	number of raw packets received

`skip_if_raw_out:`	number of raw packets sent

SKIP Header Statistics:

Command: `skipstat -h`

Note – In the description below, V1 refers to SKIP's *SunScreen SPF-100* and *SPF-100G* compatibility mode (based on an earlier version of the SKIP protocol).

`skip_hdr_encodes:`	number of SKIP V1 headers encoded
`skip_hdr_decodes:`	number of SKIP V1 headers decoded
`skip_ipsp_encodes:`	number of SKIP V2 headers encoded
`skip_ipsp_decodes:`	number of SKIP V2 headers decoded

Header decode error statistics:

`skip_hdr_bad_versions:`	invalid protocol version
`skip_hdr_short_ekps:`	short eKp fields
`skip_hdr_short_mids:`	short MID fields
`skip_hdr_bad_kp_algs:`	unknown crypto algorithms
`skip_hdr_bad_kij_algs:`	unknown key encryption algorithms
`skip_hdr_runts:`	short SKIP V1 packets
`skip_hdr_short_nodeids:`	short SKIP V1 node ids
`skip_hdr_bad_nsid:`	bad V2 namespace ID
`skip_hdr_bad_mac_alg:`	bad MAC algorithm
`skip_hdr_bad_mac_size:`	bad MAC data size
`skip_hdr_bad_mac_val:`	bad MAC value
`skip_hdr_bad_next:`	bad V2 next protocol field
`skip_hdr_bad_esp_spi:`	bad V2 encryption SPI field
`skip_hdr_bad_ah_spi:`	bad V2 MAC SPI field
`skip_hdr_bad_iv:`	bad V2 initialization vector
`skip_hdr_short_r_mkeyid:`	short V2 receiver key ID

`skip_hdr_short_s_mkeyid:`	short V2 sender key ID
`skip_hdr_bad_r_mkeyid:`	bad V2 receiver key ID
`skip_ah_nat_in:`	# MD5-NAT packets received
`skip_ah_nat_out:`	# MD5-NAT packets sent

Key Statistics

Command: `skipstat -k`

`skip_key_max_idle:`	unused key time-out
`skip_key_max_bytes:`	maximum bytes to encrypt
`skip_encrypt_keys_active:`	encrypt keys in cache
`skip_decrypt_keys_active:`	decrypt keys in cache
`skip_key_lookups:`	key cache lookups
`skip_keymgr_requests:`	key cache misses
`skip_key_reclaims:`	cache entries reclaimed
`skip_hash_collisions:`	hash table collisions

SKIP Encryption Statistics:

Command: `skipstat -c`

(requires the version of SKIP as part of the argument; 1= SKIP V1, 2=SKIP.)

Cryptographic algorithm stats (SKIP Version 1)

Crypto Module Name: DES-CBC

`encrypts:`	number of successful encryptions
`encrypterrs:`	number of failed decryptions
`decrypts:`	number of successful decryptions
`decrypterrs:`	number of failed decryptions

Cryptographic algorithm stats (SKIP)

Crypto Module Name: DES-EDE-K3-CBC

`encrypts:`	number of successful encryptions
`encrypterrs:`	number of failed decryptions
`decrypts:`	number of successful decryptions
`decrypterrs:`	number of failed decryptions

SKIP Authentication Statistics

Command: `skipstat -m`

MAC algorithm statistics (SKIP)

MAC Module Name: MD5

`in_mac`:	number of received MAC calculation
`in_mac_errs`:	number of failed received MAC calculation
`out_mac`:	number of successful sent MAC calculation
`out_mac_errs`:	number of failed sent MAC calculation

MAC Module Name: MD5-NAT

`in_mac`:	number of received MAC calculation
`in_mac_errs`:	number of failed received MAC calculation
`out_mac`:	number of successful sent MAC calculation
`out_mac_errs`:	number of failed sent MAC calculation

For more information using `skipstat`, refer to the `man` pages for SunScreen.

CHAPTER 5

Usage Examples

This chapter describes sample topologies for systems and networks using SunScreen.

All topologies require that

- Keys be generated or installed.
- Key IDs (and certificates) be exchanged between hosts.
- On both hosts, ACL entries be configured with matching algorithms, key IDs, and protocol versions.
- SKIP be enabled.

This chapter illustrates the following example topologies:

- Setting up an encrypted connection between two hosts.
- Setting up an encrypted connection between a host and a SunScreen SPF-100.
- Setting up an encrypted connection from a host to an encrypting gateway or SunScreen.
- Setting up a host as a nomadic encrypting gateway.
- Using tunnel addresses.

Setting Up an Encrypted Connection Between Two or More Hosts

The following figure depicts the configuration in which a host has an encrypted connection to another host. This is the simplest case.

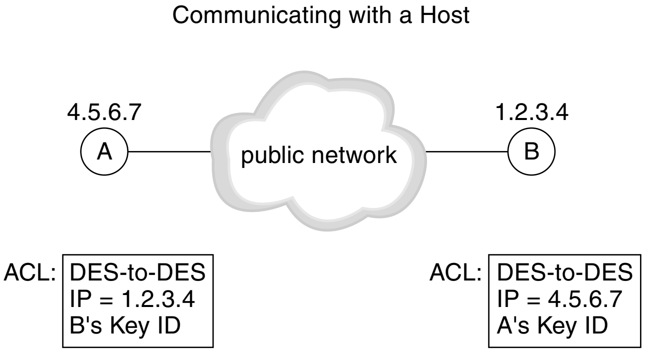


FIGURE 5–1 Communicating with a Host

This figure is an example of host-to-host communication using UDH keys and SKIP.

All the hosts must:

- Share the same key types, such as UDH, SunCA X.509, or the like, and of the same encryption strength. If X.509 certificates and keys are used, the certificates and keys for both hosts must be from the same vendor.
- Exchange certificates.
- Have the same algorithm to use that includes authentication, key encryption, and traffic encryption.
- Enable SKIP.

A machine must also have a local identity. Hosts can have many identities, but the user must choose one with which to communicate to the other host. This local identity consists of the local key type (NSID) and the local key name.

The hosts must exchange key IDs. The safest method of exchanging UDH key IDs is to have each user run `skiptool`, then call each other on the telephone and type the other’s UDH key ID in the Remote Key ID field in the *Add* window.

UDH key IDs can be exchanged and added to the ACL of each using the `skiplocal -x` command. In this case, both system administrators should telephone one another and confirm the key ID.

Hosts which wish to communicate with each other must contain each other’s addresses in their ACLs.

Setting Up an Encrypted Connection Between a Host and a SunScreen SPF-100

The following figue depicts the configuration of an encrypted connection between a host and a SunScreen SPF-100.

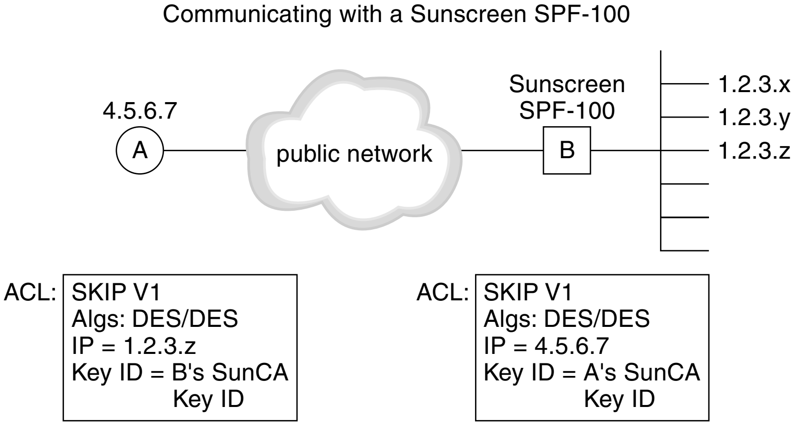


FIGURE 5–2 Communicating with a SunScreen SPF-100

In this case, both the host and the *SunScreen SPF-100* must

- Install a SunCA X.509 key of the same encryption strength.
- Manually exchange certificates.
- Use SKIP protocol Version 1.
- Have an IP address or remote name.
- Use the same algorithm that includes authentication, key encryption, and traffic encryption.
- Enable SKIP.

A machine must also have a local identity. Hosts can have many identities, but the user must choose one with which to communicate to the remote host. This local identity consists of the local key type and the local key name.

X.509 certificates and keys must be used when communicating with a *SunScreen SPF-100*. You must physically exchange the physical diskettes containing the public keys. The only method of exchanging key IDs is to have each user run `skiptool`, then call each other on the telephone and type the other's key ID in the Remote Key ID field in the *Add* window.

You must configure both the host and the *SunScreen SPF-100* ACLs with each other's address. The host must also include the addresses of any networks and hosts attached to the *SunScreen SPF-100* in its ACL. The *SunScreen SPF-100* does not really use the ACL; It uses packet filtering rules. These rule must be set to "match" the ACL on the host running SunScreen.

Setting Up an Encrypted Connection From a Host to an Encrypting Gateway or SunScreen

The following figure depicts the configuration in which a host is communicating with an encrypting gateway.

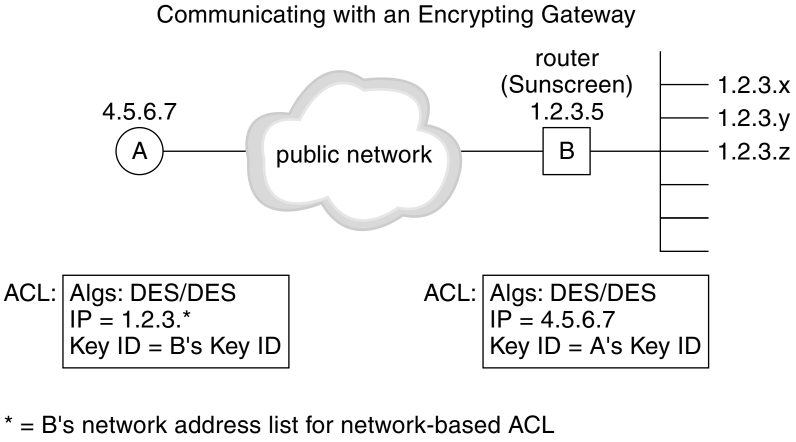


FIGURE 5–3 Communicating with an Encrypting Gateway

In this case, both the host and the encrypting gateway, whether it be a gateway, or a SunScreen must

- Have the same key type, such as UDH, SunCA X.509, or the like, and of the same encryption strength. If X.509 certificates and keys are used, the certificates and keys for both hosts must be from the same vendor.

- Exchange names or certificates.
- Use the same version of the SKIP protocol.
- Have an IP address or remote name.
- Use the same algorithm that includes authentication, key encryption, and traffic encryption.
- Enable SKIP.

A machine must also have a local identity. Hosts can have many identities, but you must choose one to use when communicating with the remote host. This local identity consists of the local key type and the local key name.

Both machines install or generate their keys and exchange namespace/key ID information. You should do this over the telephone or some other out of band media.

Type the encrypting gateway's information into the Add System box of skiptool. Then, set the Tunnel Address field of this box to be the IP address of the intermediate system. This action lets certificate discovery ask the correct host for its certificate.

For example: You are contacting a gateway that has three networks attached to it (networks 199.190.177, 199.190.176, and 199.190.176) and these networks are to remain hidden. It also has a local host attached to it. You shpuld set up the ACL in the host as shown in the following table.

TABLE 5–1 The ACL for the Host

Host	Algorithm	Tunnel Address	Remote Key
199.190.177.*	V2 DES/DES	Gateway	Gateway's
199.190.176.*	V2 DES/DES	Gateway	Gateway's
199.190.176.*	V2 DES/DES	Gateway	Gateway's
Local host	V2 DES/DES	Gateway	Gateway's
Default	V2 DES/DES	Gateway	Gateway's

You can configure a default so that everything is sent to the gateway where it will be decrypted and sent to the proper recipient in the clear. The recipients of the packets will not be aware of any encryption. The gateway will handle all the encryption and decryption of packets from and to everything behind it.

Setting Up a Nomadic Encrypting Gateway

The following figure depicts the configuration in which a host is communicating with an encrypting gateway that receives packets from an encrypting nomadic system.

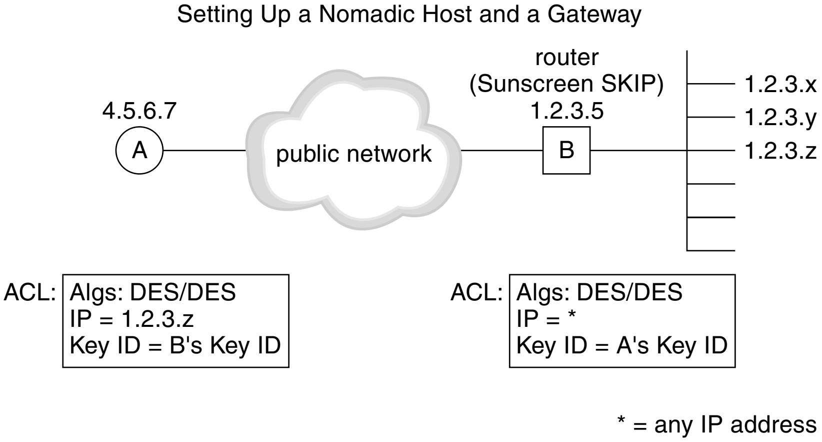


FIGURE 5–4 Setting Up a Nomadic Host and a Gateway

A nomadic encrypting gateway is an encrypting gateway that encrypts and decrypts packets from hosts whose IP address is not known ahead of time (for instance, hosts who receive the IP address dynamically). This is the same as configuring a host-to-host configuration except that the ACL does not have a specific address for the nomadic system. The address in the ACL is * and gets the temporary address from the nomadic system when it contacts the host. The host can only contact the nomadic system when it knows its address. Every time the nomadic system moves and then reconnects with the host, it will have a new address.

Using Tunnel Addresses

The following table depicts the configuration in which a host is communicating with a hidden system through a tunnel address to an encrypting gateway. The hidden system also uses a tunnel address from the encrypting gateway to the host.

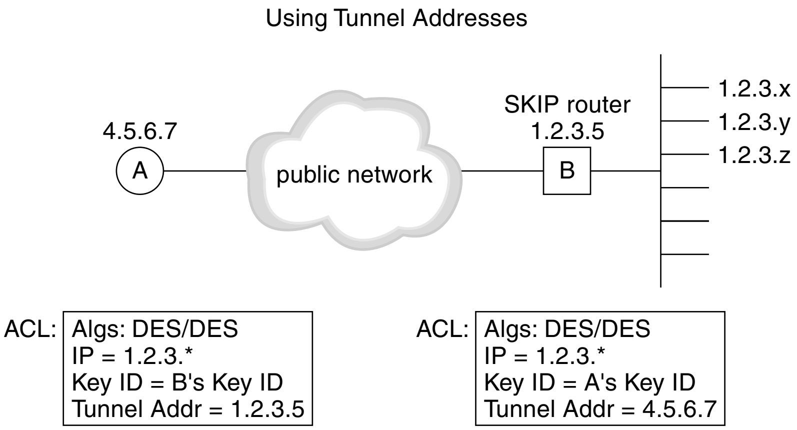


FIGURE 5–5 Using Tunnel Addresses

In tunneling, the host sends packets to the gateway. The packets are encrypted such that the gateway decrypts them and sends them to their final destination in the clear.

When setting up tunneling, you must add the address for the gateway into the host's ACL because there is no way that the host can discover the gateway's certificate.

APPENDIX **A**

Quick-Start Guide

This appendix is a quick-start guide for SunScreen. It covers installing the SKIP binaries or adding the packages with pkgadd, and setting up IP-level encryption between two hosts. These instructions assume that only one network interface is active on each machine.

For complete documentation, refer to the SunScreen documentation and the SKIP man pages.

Installing SKIP Binaries

1. **Mount the CD-ROM and type:**

   ```
   volcheck
   ```

Note – If you are not using `vold` on your system, type # **`mount -F hsfs -oro /dev/dsk/c0t6d0s0/mnt`** The device name or the mount point or both depends on your local system configuration.

2. **Go to the directory on the CD-ROM for your OS**

   Solaris operating environment for the SPARC Platform:

   ```
   cd /cdrom/cdrom0/sparc
   ```

   Solaris operating environment for the Intel Platform:

   ```
   cd /cdrom/cdrom0/x86
   ```

Note – If you have mounted the CD-ROM manually, replace `/cdrom/cdrom0` with `/mnt`.

---

3. **Use the standard Solaris operating environment `pkgadd` command to add all packages:**

   ```
   pkgadd  -d ‘pwd‘
   ```

4. **Add `/usr/sbin` to your PATH variable:**

   ```
   PATH=/usr/sbin:$PATH export PATH
   ```

5. **Initialize the SKIP directories by issuing the command:**

   ```
   skiplocal -i
   ```

6. **Generate a secret and a public certificate locally by issuing the command:**

   ```
   skiplocal -k
   ```

7. **Add SKIP to your network interface by issuing the command:**

   ```
   skipif -a
   ```

8. **Reboot the machine.**

9. **Enable SKIP and configure IP encryption with one other host:**

   ```
   skiphost -a default          default IP traffic is unencrypted
       skiplocal -x      prints the skiphost command to check info    others need to run to talk to us
   skiplocal -x| mail Friend@remote.host
   ```

   `Friend@remote.host` should issue these commands as well. Once the corresponding mail is received, verify out-of-band (for example, over the telephone) that the received mail matches the mail that was sent. Then `Friend` executes the received `skiphost` command.

10. **Turn SKIP n:**

    ```
    skiphost -o on          enable SKIP
    ```

Is It Working?

At this point, SKIP encryption should be enabled with the remote host. Traffic will be exchanged with all other hosts in the clear.

1. **`ping` the other host to make sure everything is working:**

   `ping` *host*

2. **View the key manager log file to see if the certificate exchange and the shared-secret computation succeeded:**

   ```
   tail /var/log/skipd.log
   ```

3. **If you have** `snoop`**,** `tcpdump`**,** `etherfind`**, or some other packet dumping utility, you can verify that encrypted packets are using protocol 57.**

Examining the Local SKIP Configuration

`skiphost`	list the SKIP access control entries
`skiplocal -l`	list the set of local identities
`skipdb -l`	list the certificates in our database
`skipca -l`	list the Certificate Authorities we trust

SKIP configuration files are stored in the `/etc/skip` directory.

APPENDIX **B**

How SKIP Works

SKIP is an IP-layer encryption package integrated into SunScreen. SKIP lets a Screen or Administration Station encrypt IP network communications passing between them. By providing efficient transparent encryption of any protocol within the TCP/IP protocol suite, SKIP lets computers communicate privately and securely over non-secure public networks

You can manage SKIP through `skiptool`, the SKIP graphical user interface, or through the SKIP command line interface.

SKIP Security Services

SKIP provides several network security services:

- *Access control* to protect corporate data resources from unauthorized use.
- *Encryption and decryption services* to ensure the confidentiality of information sent over a network.
- *Authentication* to ensure the integrity of the information transferred from one host to another and the identity of the sender and receiver.
- *Key and certificate management* to provide efficient, cost-effective administration of the basic building blocks of a security policy.

Each of these services is described separately below.

Access Control

Use access control on your network to limit and control who uses your host systems and applications through your communications links. Each entity with which you communicate must be identified by name, IP address, or network so that access to your system is controlled. After access control is established, your computer can exchange encrypted or unencrypted data with the remote host.

The SKIP access control list (ACL) specifies whether remote hosts or networks are authorized to communicate with your computer. Each entry in your access control list identifies a specific host (by name or IP address) or network (by network number and subnet mask). You use SunScreen SKIP Access Manager to maintain your access control list.

SKIP's access control is based on the IP addresses of remote systems. When a system tries to connect to a host running SunScreen SKIP, the application searches for an ACL entry as follows:

1. SKIP searches for an ACL entry for the remote host. If the entry exists, SKIP uses it to determine access permissions (which can be Clear Access, Encrypted Access, or No Access) and encryption information (if any).
2. If an entry for the host does not exist, SKIP searches for an ACL entry for the network to which the remote host belongs. If the entry exists, SKIP uses it to determine access permission.
3. If an entry for the host or the host's network does not exist,SKIP searches for an ACL entry called *Default*. If the entry exists, SKIP uses it to determine access permissions as well as encryption, authentication, and compression settings for communication with the remote host.
4. If SKIP cannot find an ACL that pertains to the remote host or a Default ACL entry, it will not grant access.

Encryption and Decryption

Encryption is the process by which a readable message is converted to an unreadable form to prevent unauthorized parties from reading it. *Decryption* is the process of converting an encrypted message back to its original (readable) format. The original message is called the *plaintext message*. The encrypted message is called the *ciphertext message*.

Digital encryption algorithms work by manipulating the digital content of a plaintext message mathematically, using an encryption algorithm and a digital key to produce a ciphertext version of the message. The sender and recipient can communicate securely if the sender and recipient are the only ones who know the key.

Shared Key and Public Key Encryption

SKIP uses a combination of *shared key cryptography* and *public key cryptography* to protect messages sent between hosts. SKIP hosts use shared traffic keys that change frequently to encrypt data sent from one host to another. To protect these shared traffic keys, SKIP hosts use public key to calculate an implicit shared secret, which they use to encrypt the shared traffic keys, keeping network communication secure.

Shared Key Encryption

Shared key encryption uses one key to encrypt and decrypt messages. For shared key cryptography to work, the sender and the recipient of a message must both have the same key, which they must keep secret from everybody else. The sender uses the shared key to encrypt a message, shown in the following figure, and then sends the ciphertext message to the recipient.

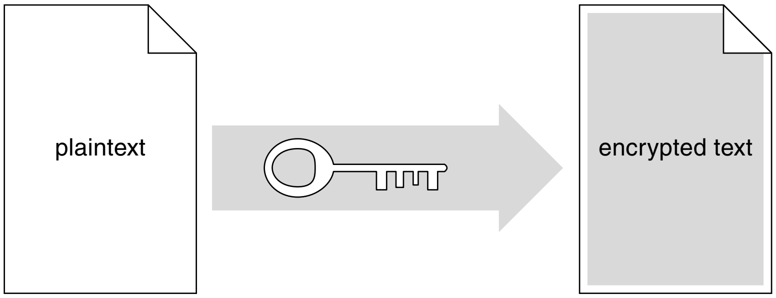


FIGURE B–1 Sender Uses Key to Encrypt Plaintext to Ciphertext

When the ciphertext message arrives, the recipient uses the identical shared key to decrypt the message, shown in the following figure.

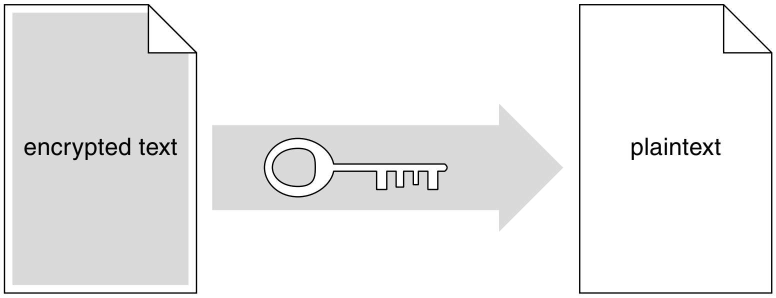


FIGURE B–2 Recipient Uses Key to Decrypt Ciphertext to Plaintext

Shared key encryption/decryption is relatively fast. However, since anyone with the shared key can decrypt the information, shared key encryption requires that only the sender and recipient have access to the shared key. SunScreen SKIP uses shared key algorithms to encrypt packets sent between hosts. SunScreen SKIP protects the security of encrypted information by generating new traffic keys frequently during a communication session, making acquisition of any one traffic key useless.

Public Key Encryption

Public key encryption uses a pair of complementary keys (a *public key* and a *private key*) to encrypt and decrypt messages, as shown in the following figure. The two keys are mathematically related such that a message encoded with one key can only be decoded with the other key. Although a user's public and private keys are mathematically related, knowledge of a public key does not make it possible to calculate the corresponding private key.

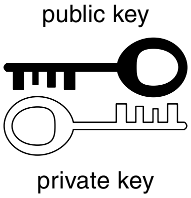


FIGURE B–3 Complementary Public and Private Keys

In public key encryption systems, users make their public key available to anyone and keep their private key secret. When one user wants to send a private message to another user, the sender looks up the recipient's public key and uses it to encrypt a message, shown in the following figure, before sending it to the recipient.

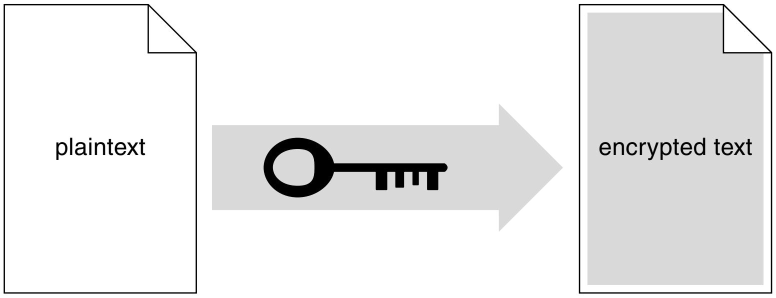


FIGURE B–4 Sender Uses Recipient's Public Key to Encrypt Message

When the encrypted message arrives, the recipient uses his or her private key to decrypt the message, shown in the following figure. Because the recipient's private key is known only to the recipient, both the sender and recipient can safely assume that no one other than the recipient could read the message.

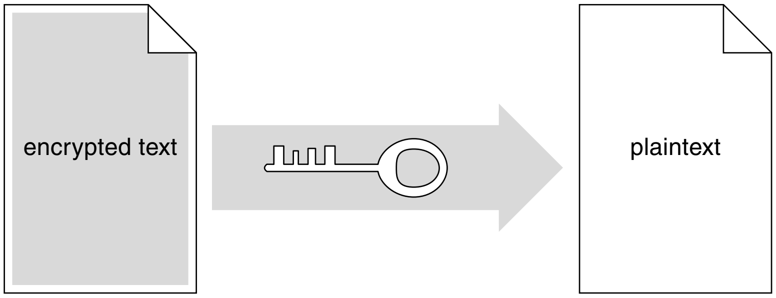


FIGURE B–5 Recipient Uses Private Key to Decrypt Message

Public key encryption algorithms are mathematically more complex than shared key encryption algorithms. As a result, public key encryption is significantly slower than shared key encryption. Consequently, SunScreen SKIP uses Diffie Hellman key pairs (described in the next section) to create a shared secret between two users, and then uses shared key encryption to encrypt traffic traveling between the two hosts.

Diffie-Hellman Key Exchange

The Diffie-Hellman key exchange algorithm, which is named after its inventors, solves the problem of securely distributing keys by removing the need to transmit secret keys. When two hosts wish to use the Diffie-Hellman algorithm to exchange keys, they agree to use the same numerical values for the key basis (`g`) and modulus (`p`). Each host generates a large (512-, 1024-, or 2048-bit) random number (`x`) as a private key, and then uses this private key to generate a public key $g^x \bmod p$.

Once a user’s private and public keys have been calculated, SunScreen SKIP creates the user’s public certificate. This certificate contains the public key value, the `g` and `p` values used to compute the public key, and other information, such as the period for which the certificate is valid.

SunScreen SKIP hosts exchange their public certificates with one another freely. When two hosts wish to communicate securely, each host calculates a mutually authenticated shared secret based solely on knowledge of its private key and the other host’s public key.

For example, host I would select a random number `i` as a private key and then generate a public key `gi mod p`. Similarly, host J would select a random number `j` as a private key and then generate a public key `gj mod p`. The two hosts then exchange their public keys over secure or nonsecure links. Host I raises J’s public key (`gj mod p`) to the power of its private key `i`, yielding (`gj)i mod p` or `gji mod p`. Host J raises I’s public key (`gi mod p`) to the power of its private key `j`, yielding `(gi)j mod p` or `gij mod p`. Consequently, hosts I and J can derive a mutually authenticated long-term secret g^{ij} `mod p` implicitly (without explicit communication). Since no one other than I and J have access to their private keys, no one other than I and J can compute g^{ij} `mod p`.

The two hosts then take the low-order bits of `gij mod p` to derive a pairwise master key K_{ij}. K_{ij} is an implicit shared master key that does not need to be sent in any packet or negotiated out of band.

In theory, the two hosts could use their shared master key K_{ij} to encrypt messages. However, doing so might expose K_{ij} to analysis and eventual decryption. Instead, SunScreen SKIP uses a rapidly-changing series of traffic keys to encrypt messages traveling between the two hosts, and uses a modified version of K_{ij} to encrypt these traffic keys. See “Perfect Forward Secrecy,” below, for more information.

Perfect Forward Secrecy

Perfect forward secrecy substitutes a clock-based master key for the long-term Diffie-Hellman shared secret K_{ij}. Using a clock-based master key means that the long-term secret K_{ij} is never directly exposed to third parties, making it less vulnerable to cryptanalysis. Another feature of perfect forward secrecy is that it prevents coarse-grain playback of traffic. Once the clock-based master key has been updated, traffic encrypted or authenticated with the help of old keys will be rejected by SKIP.

SKIP uses the long-term secret key *Kij* and the date/time value `n` to create a time-based shared secret key *Kijn*.

$$K_{ijn} = h(K_{ij}, n)$$

where h is a pseudo-random function such as MD5. SKIP uses the current time and date clock (actually, the number of hours since 1977) to generate n, which changes every hour. Consequently, hosts using SunScreen SKIP must verify that the date, time, and time zone settings on their systems are synchronized to ensure that they are using the same n in their master key calculations.

This time-based shared secret key is used to encrypt traffic keys. Since I and J can calculate *Kij* based on their implicitly authenticated shared secret, the two computers can calculate the same value for *Kijn* if their system clocks are synchronized.

Note – SKIP relies on the system clock value to calculate time-based shared secrets. Consequently, hosts using SunScreen SKIP must verify that the date, time, and time zone settings on their systems are correct to ensure that they are using the same n in their master key calculations. Users should never change the time, date, or time zone setting on their computer while using SunScreen SKIP.

When I wants to send a secure message to J, I uses a randomly generated traffic key *Kp* to encrypt the contents of the message. The traffic key *Kp* is in turn encrypted using *Kijn*. SunScreen SKIP then constructs a series of packets, each containing the IP header information (in cleartext) needed to route the packet to its destination, the traffic key encrypted with the time-based shared secret *Kijn*, and the message data encrypted with the traffic key *Kp*. The following figure shows an encrypted IP packet, using this two-step encryption procedure.

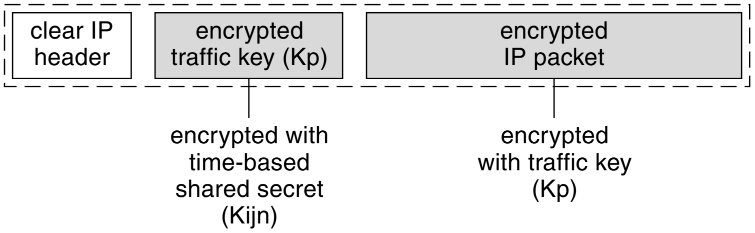

clear IP header
encrypted traffic key (Kp)
encrypted IP packet
encrypted with time-based shared secret (Kijn)
encrypted with traffic key (Kp)

FIGURE B–6 Encrypted IP Packet

When the destination host receives this encrypted packet, it looks up the sender's certificate. Using the long-term secret key *Kij* and the counter value n (which is based on the current date and time), the destination host computes the same *Kijn* value used by the sender. Using *Kijn*, the destination host decrypts the traffic key *Kp*, and then uses the traffic key to decrypt the packet data.

Since *Kijn* can be cached for efficiency, SKIP can change traffic keys very rapidly without incurring the computational overhead of a public key operation. SKIP changes traffic keys after a key has been idle for a user-specified number of seconds (30 seconds by default) or after a key has been used to encrypt a user-specified amount of data (512K by default).

Encryption Algorithms

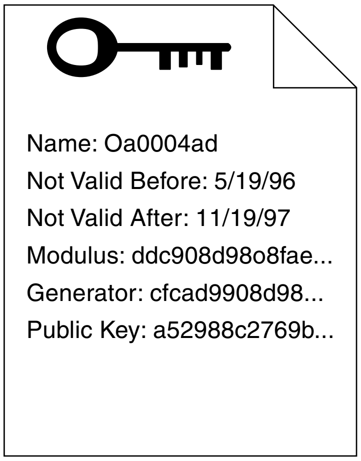
The following table lists the traffic encryption algorithms supported by SKIP.

TABLE B–1 Encryption Algorithms

Encryption Algorithm	Description	Efficiency	Security
DES-CBC	DES uses cipher block chaining (CBC) and a 56-bit key to encrypt 64-bit blocks of plaintext in multiple iterations.	Moderate	Excellent
DES-EDE-K3	DES-EDE-K3 (triple DES) uses three encryption operations and cipher block chaining (CBC) and a 56-bit key to encrypt 64-bit blocks of plaintext in multiple iterations.	Poor	Excellent
Safer-128SK-CBC	Safer uses two 64-bit subkeys and cipher block chaining to encrypt variable-length blocks of plaintext.	Good	Excellent
RC2-40 (Restricted to 32-bit mode only for SKIP V1.5.1)	RC-2 uses cipher block chaining (CBC) and a variable-size key to encrypt 64-bit blocks of plaintext.	Good	Good
RC4-128	RC-4 uses a 128-bit key to encrypt data in a continuous stream.	Excellent	Excellent
RC4-40	RC-4 uses a 40-bit key to encrypt data in a continuous stream.	Excellent	Poor

SKIP Certificates

SKIP certificates are the means by which a user distributes public key information. A SKIP certificate is a digital document that contains a user's Distinguished Name, the public key associated with that Distinguished Name, and the time interval for which the certificate is valid. You can distribute your public certificate to other users, who extract and use your public key to calculate a unique shared secret for encrypting communications between you. Users can distribute their certificates freely to other SKIP users on diskette, through a certificate server, or over a network.



Name: Oa0004ad
Not Valid Before: 5/19/96
Not Valid After: 11/19/97
Modulus: ddc908d98o8fae...
Generator: cfcad9908d98...
Public Key: a52988c2769b...

FIGURE B–7 SKIP Certificate Contents

SKIP certificates can be *signed* or *unsigned*:

- Signed SKIP certificates (RSA certificates) must be obtained from a Certification Authority, which is an entity trusted to create and assign SKIP certificates. In addition to the information described above, a signed certificate contains the name of the certification authority responsible for issuing the certificate and the MD5 hash of the certificate. The CA signs the certificate by encrypting the hash with the CA's private key. The CA's digital signature lets anyone who receives the certificate validate its contents and verify that an unauthorized user is not impersonating you.

  You can store signed certificates in directory servers, transmit them by means of non-secure message exchanges, or distribute them on diskette. For information on how to obtain signed SKIP certificates, contact `carequest@sun.com`.

- Unsigned SKIP certificates (unsigned Diffie-Hellman (UDH) certificates) are generated on demand by the user. A user's unsigned certificate can be distributed through secure out-of-band channels or through certificate discovery (which is described on "Certificate Discovery" on page 122). Unsigned certificates can offer several advantages over signed certificates:

  - A user generates a private key when he or she generates a public certificate. This private key never leaves the user's machine, meaning that a network administrator does not need to order, distribute, or protect key diskettes.
  - Since UDH certificates are not registered by a certification authority, they do not need to be formally revoked. If a user or administrator suspects that a key has been compromised, a new key/certificate can be generated and the new certificate can be distributed to other users.

The decision whether to use a signed or unsigned certificate depends on the type of hosts with which you want to exchange encrypted traffic. In general, you must use a signed certificate to communicate securely with a host using a signed certificate, and you must use an unsigned certificate to communicate securely with a host using a UDH certificate. Both certificates must use keys of the same length and use the same values for key calculation.

Certificate Discovery

Certificate discovery lets a host running SunScreen SKIP retrieve a public (X.509 or UDH) certificate from another SKIP host over a network or serial connection. Certificate discovery is an alternative to direct installation of certificates.

Certificate discovery works as follows:

1. You verify that certificate discovery is enabled on your computer and on the remote host.
2. You obtain the identifier and name space of the certificate you want to discover from a remote user by means of a channel that lets you authenticate the other user's identity. For example, you might call a user with whom you want to send encrypted traffic to exchange certificate identifiers and name spaces, relying on recognition of the other user's voice to authenticate the user's identity.
3. You enter the identifier of the remote certificate in skiptool.
4. Your computer sends a Certificate Discovery Protocol request to the remote computer in clear text on UDP port 1640, asking for a specific certificate using the designated name space. The remote computer sends back the requested information in clear text on UDP port 1641.
5. SunScreen SKIP validates the certificate. For signed certificates, SunScreen SKIP uses the Certifying Authority's public key to decrypt the certificate digest, creates its own MD5 digest of the public certificate, and compares the result. For unsigned certificates, SunScreen SKIP creates an MD5 hash of the certificate's public key and compares it to the certificate's name.
6. SunScreen SKIP adds the certificate for the remote host to its certificate database.
7. SunScreen SKIP uses the public key information contained in the remote host's certificate to generate a unique shared secret.

Name Space Identifiers

Name space identifiers (NSIDs) identify the type of keys being used. SunScreen EFS 2.0 supports the following NSIDs:

- NSID 0, which specifies that the IP address of the host is the key identifier for the host's X.509 certificate. Using NSID 0 results in a small improvement in encryption/decryption efficiency, since SKIP does not need to include a key identifier in each packet.

- NSID 1, which is the IPv4 address assigned to the X.509 certificate by a certification authority. This IP address does not correspond to the IP address used by your computer. Rather, it is an eight-byte hexadecimal number assigned to the certificate by the certification authority to bind a unique Distinguished Name to the certificate contents. For example, a SunCA certificate might use 0a000101 (which translates to 10.0.1.1 in IP address notation) as a key identifier
- NSID 8, which is the MD5 hash of the certificate's Diffie-Hellman public key.

SKIP Tunnels

A SKIP tunnel is a logical connection between your computer and another host that accepts encrypted messages on behalf of a remote host. Before your computer sends a message through a SKIP tunnel, it encrypts each packet and adds an IP header that specifies the security proxy as its destination. The security proxy decrypts each packet and uses the IP header of the decrypted packet to route the packet to its actual destination.

SKIP tunnels offer several advantages over endpoint-to-endpoint encryption:

- **Centralized decryption** – By directing network traffic through a SKIP tunnel to a special gateway, your site can centralize encryption and decryption in a single machine. Consequently, a site would not need to install security software on every host.
- **Topology hiding** – The tunnel address field contains the IP address of the security proxy; the IP address of the packet's actual destination is encrypted along with the rest of the packet. Consequently, an unauthorized user cannot glean information about a site's topology from a captured packet.
- **Prevention of packet fragmentation** – When using endpoint-to-endpoint encryption, packets may become fragmented as they travel from one site to another. If this occurs, the packet fragments may be routed to different gateways at a site. Since each gateway would receive only part of the packet, the packet could not be decrypted, making it impossible to forward the packet contents to the destination host. By specifying the security proxy to which all packets (and packet fragments) should be delivered, you ensure that the security proxy will receive the information it needs to route packets to destination hosts reliably.

Authentication

Authentication is the process of verifying that individuals requesting access or sending messages are who they say they are and that information received from a remote host has not been modified in transit.

SunScreen SKIP uses the Keyed MD5 algorithm to authenticate messages. The MD5 message digest algorithm takes a message of any length and produces a 16-byte digest (the hash value). This message digest serves as a *thumb print* of the original file when you want to authenticate a document. The original message cannot be derived from the message digest.

The process by which SunScreen SKIP signs a message with a message digest follows.

1. The sender and destination host agree to use authentication as part of their secure communication process.
2. The sender creates a message and indicates that it can be sent to the destination host.
3. SunScreen SKIP encrypts the message using the specified encryption algorithm.
4. SunScreen SKIP uses a keyed MD5 message digest algorithm to create a digest of the message and encrypts the message digest with the time-based shared secret.
5. SunScreen SKIP adds the encrypted digest to the authentication header for the message.
6. SunScreen SKIP sends the message to the destination host.
7. The destination host receives and decrypts the encrypted message, which includes the message digest.
8. The destination host uses the time-based shared secret to decrypt the message digest.
9. The destination host creates its own message digest of the message it received, using the same keyed MD5 algorithm the sender used.
10. The destination host compares the message digest it created with the one that came with the message. If the two message digests match, the destination host knows that the text of the message has not been modified and that the person claiming to be the sender was actually the sender.

The following figure illustrates how a destination host would calculate its own MD5 digest of the message, decrypt the digest sent with the message, and compare the two digests to authenticate the integrity and source of the message.

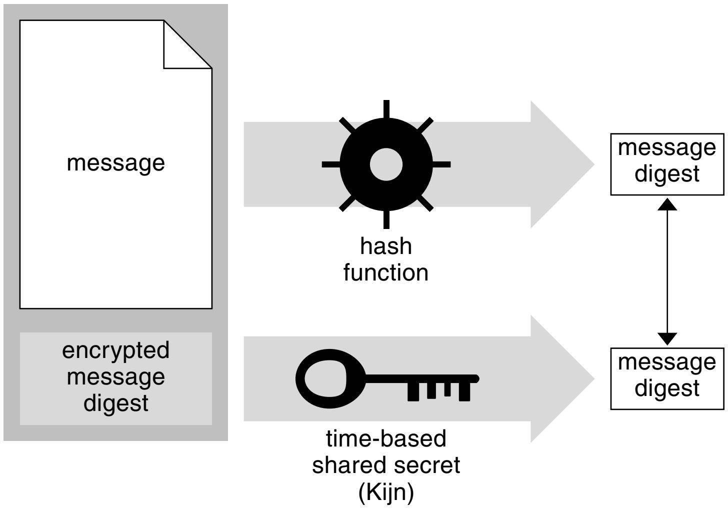


FIGURE B–8 Authenticating a Message

Authentication provides a *digital signature* that proves the identity of the sender. If the message digest can be decrypted with the time-based shared secret, the destination host knows it was encrypted with the time-based shared secret. Since the time-based shared secret is known only to the two hosts, the destination host can be confident of the message's source as well as of its integrity.

APPENDIX **C**

Troubleshooting SunScreen SKIP

The following information is provided to help you troubleshoot any problems with SunScreen SKIP. You should also consult "Is SKIP Working?"

Emergency Start Instructions

▼ System Hangs and You Cannot Access the Machine

If your system hangs when you are configuring SKIP and you do not have access to your machine, reboot your machine in the single-user mode and become root.

1. **With a text editor, such as `vi`, edit the file `acl.`*<network_interface>* in the `/etc/skip/` directory so that line**

   ```
   skiphost -i <network_interface> -o on
   ```

   reads

   ```
   skiphost -i <network_interface> -o off
   ```

   This will to disable SKIP.

2. **Reboot your machine normally to clean up the file system.**

3. **You, then, as root, may reconfigure your access control list as your security policy dictates.**

▼ System Hangs, But You Still Can Become Root

1. **If your system hangs when you are configuring SKIP and you still have access to your machine and can become root, enter**

   ```
   # skiphost -o off -i <network_interface>
   ```

   This will disable SKIP on the network interface

2. **Then, as root, you may reconfigure your access control list as your security policy dictates.**

Error Messages

The following error messages may possibly occur during your operation of SKIP software.

N-counter out of range - either replayed packets or out of sync clocks
"Old" packets have been received by SKIP. This indicates either that, typically, the sending machine's clock is not in synchronization with your machine's clock or that, rarely, an intermediary is sending old packets in a replay attack.

Certificate g+p do not match dh_params
An entry in your access control list has a local identity and remote identity that do not have matching Diffie-Hellman parameters (*g* is the generator value, *p* is the prime value). This is typically caused when you try to talk to a system with moduli that do not match (*i.e.,* a 1024–bit system trying to talk to a 512–bit system using 1024–bit keys).

Local secret nsid=xx mkid=xx has expired. Deleting
Your local secret has expired. Generate a new local identity.

Unable to load skipsup.o -- Exiting!
The SKIP support module could not be loaded. Typically, this means that one of the necessary libraries is not available on the machine that is attempting to run SKIP. Ensure that your system has the required software packages installed according to the instructions in the SunScreen *User's Guide*.

Modulus too big for U.S. export law
You have attempted to load a key that is not permitted under U.S. export law. Make sure that you have installed both the base SKIP package and any SKIP encryption upgrade packages that you have purchased under appropriate U.S. export license control.

`skipd: passphrase required issue skipd_restart to enable encryption`

The key manager cannot start without a password to decrypt local secrets. Use the command `skip_restart` to start the key manager.

Glossary

3DES Also called triple-DES or DES-EDE-IT. It means encryption is performed on a *block* three times with the two *keys*: first with the first key, then with the second key, and finally with the first key again. The resulting key length is 112–bits. See *DES* and *EDE*.

ACL Access control list. Limits and controls who uses a host system or applications through communications link

address In networking, a unique code that identifies a *node* to the *network*.

ADP Algorithm discovery protocol. Enables one *entity* to inform another of the capabilities it supports.

AH Authentication header. A mechanism for providing strong integrity and *authentication* for *IP* datagrams. It may also provide *nonrepudiation*, depending on which *cryptographic* algorithm is used and how keying is performed. It does not provide confidentiality or protection from *traffic analysis*.

algorithm A sequence of steps designed to solve a problem or execute a process such as drawing a curve from a set of control points, or encrypting a block of data.

alias Used with the *Log Browser* to refer to a textual representation of a numerical filter parameter, such as a port, IP address, or error code.

API Application programmer's interface. A set of calling conventions defining how a service is invoked through a software package.

argument An item of information following a *command*. It may, for example, modify the command or identify a file to be affected.

attack An attempted *cryptanalysis* or an attempt to compromise system security.

authentication The property of knowing that the claimed sender is in fact the actual sender.

block Groups of bits are called blocks.

block cipher or block algorithm An encryption algorithm that encrypts while blocks at once. (See *stream ciphers*)

Bourne shell The *shell* used by the standard Bell Labs UNIX.

broadcast A *packet* delivery system where a copy of a given packet is given to all hosts attached to the network.

button A one-choice element of a control area or a menu that starts an activity. Buttons execute commands (*command buttons*), display pop-up windows (*window buttons*), and display menus (*menu buttons*).

CA Certification authority. A trusted network entity that digitally signs a certificate containing information identifying the user; such as, the user's name, public key, and the key's expiration date.

cache A buffer of high-speed memory used to store frequently accessed memory or values. A cache increases effective memory transfer rates and processor speed.

CBC Cipher block chaining (see also DES). A mode used to chain a feedback mechanism, which essentially means the previous block is used to modify the encryption of the next block.

CDP Certificate discovery protocol. A request/response protocol used by two parties to transfer certificates.

CD-ROM Compact disc, read-only memory. A form of storage characterized by high capacity (roughly 600 megabytes) and the use of laser optics rather than magnetic means for reading data.

certificate A certificate is a data structure that binds the identity of an entity with a public-key value. *SunScreen* uses X.509 certificates.

CFB Cipher feedback. Uses a block cipher (such as DES) to implement a stream cipher.

cipher A cryptographic algorithm used for encryption or decryption.

ciphertext An encrypted message.

CLI Command line interface

command In a graphical user interface (GUI), a *button*, *menu item*, or *controls*.

command button The *button* used to execute application commands.

compiler A translation program that converts a high-level computer language (such as FORTRAN) into *machine language*.

confidentiality The property of communicating such that the intended recipients know what is being sent, but unintended parties cannot determine what is sent.

controls Objects in a *menu* that are used to perform an action.

cookie (In cryptography) A cookie is a *pseudo-random* number used to prevent denial-of-service *attacks*.

cryptanalysis The art and science of breaking *ciphertext*.

cryptography The art and science of keeping messages secure.

C shell The standard shell provided with Berkeley standard versions of UNIX.

daemon A process that runs in the background to perform a task on behalf of the system.

data compression Application of an algorithm to reduce the bit rate of a digital signal.

data encrypting key A key used to encipher and decipher data intended for programs that perform encryption.

decoder A facility that takes data that have been encoded, or compressed, by an *encoder* and decompresses them.

decryption The process of turning *ciphertext* back into *plaintext*.

DES A commonly used, highly sophisticated algorithm developed by IBM for the U.S. National Bureau of Standards for encrypting and decrypting data. See *CBC*.

DH Diffie-Hellman. A classic cryptographic construction that uses exponentiations over a prime field.

digital signatures The bit string attached to the document to authenticate it when signed.

diskette A 3.5–inch removable storage medium supported by some Sun systems.

DN Distinguished name. A numeric string representation of a list of IP addresses or equivalent identifier for principals in the network, such as IP nodes or users.

DNS Domain name system. The distributed name/address mechanism used in the Internet.

DSA Digital signature algorithm. Each DSA is responsible for the directory information for a single organization or organizational unit.

dynamic packet screening Examines traffic to be either allowed or rejected.

dynamic translation A *NAT* address translation that converts a set of internal private addresses into external public addresses. It allows internal hosts to contact external hosts, but it cannot be used to allow external hosts to contact internal hosts.

EDE Encrypt-decrypt-encrypt (See *3DES*)

EFS Encryption Firewall Server. A software solution that can reside on any Sun machine running the Solaris 2.4 or 2.5 operating environment. It can secure all the servers on a corporate intranet. A corporation may have any number of database servers—one each for marketing, accounting, and engineering divisions, for example. Each server's data should be protected by EFS. The majority of break-ins that companies experience happen from within the company's own network. This product locks down each server. Since it works at the network IP layer, it can "talk" to any other machine and thus can be placed in "front" of any competitor's machine to protect it.

EKE Encrypted key exchange

encapsulation The technique used by layered protocols in which a layer adds header information to the protocol data unit from the layer above. In Internet terminology, for example, a packet would contain a header from the physical layer, followed by a header from the network layer (*IP*), followed by a header from the transport layer (*TCP*), followed by the application protocol data.

encryption A mechanism commonly used to provide confidentiality.

encryption key A value that controls how information is enciphered or deciphered. Often called the *public key*. (See *data encrypting key*)

entity Terminology for a layer protocol machine. An *entity* within a layer performs the functions of the layer within a single computer system, accessing the layer entity below and providing services to the layer entity above at local service access points.

ESP Encapsulating security payload. A mechanism for providing integrity and confidentiality to IP datagrams. In some circumstances it can also provide authentication to IP datagrams, depending on which algorithm or algorithm mode is used. It does not provide nonrepudiation and protection from traffic analysis.

Ethernet A type of local area network that enables communication between machines connected directly together through cables.

FDDI Fiber distributed data interface. A high-speed networking standard. The underlying medium is fiber optics, and the topology is a dual-attached, counter-rotating *token ring*. FDDI networks can often be spotted by the orange fiber "cable."

filters Allow selection of a subset of packets based on specific attributes of the logged packets.

Filter Catalog Used with the Log Browser as part of the hierarchical structure of saved filters. Filter groups are saved in filter catalogs.

Filter Directory Service Used with the Log Browser as the hierarchical structure into which filters are grouped and saved.

Filter Group Used with the Log Browser and refers to a set of filters created by the administrator, then saved so they can be applied to multiple log files.

GUI Graphical user interface. Provides the user with a method of interacting with the computer and its special applications, usually via a mouse or other selection device. The GUI usually includes such things as windows, an intuitive method of manipulating directories and files, and icons.

hash A message digest or cryptographic checksum.

header file A file of information, identified at the beginning of the program, that contains the definitions of data types and variables used by the functions in the program.

hidden file A special type of file, such as `.login`, that does not show up in normal file listings. Special files usually pertain to system configuration.

host computer The primary or controlling computer in a multiple computer installation.

hung A condition in which the system is frozen and unresponsive to commands.

IANA Internet Assigned Numbers Authority. SKIP was assigned the protocol decimal number 57. SKIP Version 1 was assigned protocol decimal number 79 by IANA.

ICG Internet Commerce Group. A business unit of Sun Microsystems, Inc., that is committed above all else to developing solutions to communicate securely over unsecured public networks. Formed in 1994, ICG already has three strong *SunScreen* security product lines that stand at the head of the class. Each depends on the public-key cryptography invented by Sun's Distinguished Engineer Whitfield Diffie, along with Stanford's Martin Hellman. Building upon public-key cryptography, ICG developed SKIP—Simple Key-management for Internet Protocols—the premier protocol that makes key management easier to use than previous innovations. SKIP is the central cryptographic protocol upon which ICG draws in its products.

ICMP Internet control message protocol

icon (1) An on-screen symbol that simplifies access to a program, command, or data file. (2) A small pictorial representation of a base window. Displaying objects as icons conserves space on the screen while keeping the window available for easy access.

IDEA International data encryption algorithm

integrity The property of ensuring that data are transmitted from the source to destination without undetected alteration.

IP Internet Protocol. The *network layer* protocol for the Internet protocol suite.

IPSEC IP security

ISDN Integrated Services Digitial Network

IV Initialization vector

kernel The core of the operating system software. The kernel manages the hardware and supplies fundamental services such as filing that the hardware does not provide.

Key and Certificate Diskette Diskettes that contain both the private key and the certificate containing the public key. The identifier for this certificate is on the label. The information is extremely sensitive and should be kept secure.

key encrypting key A key used to encipher and decipher other keys, as part of a key management and distribution system.

keyspace The range of possible values of the key.

layer A set of structures and routines that handle a particular class of events. For example, in the seven-layer International Organization of Standardization's open systems interconnection model, the *network layer* is responsible for routing the signals to their intended recipients.

locally stored secret The *secret key* that corresponds to a *public key certificate*. Used to encrypt and decrypt messages.

Log Browser The main window for examining log files.

MAC Message authentication code. The term "MAC" is synonymous with the term "authentication data."

man pages Stands for manual pages, the UNIX on-line documentation.

MD Message digest. An authentication code that cryptographically guarantees that data have not been forged or tampered with.

MD5 A message digest one-way *hash* function designed by Ron Rivest. The algorithm produces a 128–bit hash, or message digest, of the input message.

MD5-NAT Uses the same hash function as MD5 except that in this case, the source and destination IP addresses are not authenticated.

MDC Message digest cipher

menu button A multiple-choice control that has a *menu mark* and is used to display a menu.

menu mark A hollow triangle in the border of a button or following a menu item that has a *submenu* attached to it. The triangle points to where the menu or submenu is displayed.

MIC Message integrity check

MI Message indicator

MKID Master Key-ID. A generic term used to identify a particular key. MKIDs effectively decouple the identification of a master key for purposes of key lookup and access control from issues of network topology, routing, and IP addresses.

modulus An arithmetic operation used in programming whose result is the remainder of a division operation. The plural is moduli.

MSP Message security protocol. An X.400-compatible application-level protocol for securing electronic mail that was developed by NSA.

MTU Maximum transmission unit

multicast A special form of *broadcast* where copies of the packet are delivered to only a subset of all possible destinations.

NAT Network Address Translation. An address translation function used in SKIP where packets passing through a box have their addresses changed (or translated) between sets of addresses to hide internal addresses such that they cannot be used as an *attack* point. It is also useful on the Internet as you must use registered addresses so no two systems use the same address. However, many internal networks were built without registering their addresses because they were built before the Internet was considered vital to business. Address translation can be used to translate unregistered (that is, illegal) addresses into a smaller set of registered addresses, thus allowing internal systems with unregistered addresses to access systems on the Internet.

network The hardware connecting various systems enabling them to communicate.

network administrator The person who maintains a network.

network layer The third of the seven layers in the International Organization for Standardization's open systems interconnection model for standardizing computer-to-computer communications.

network mask A number used by software to separate the local subnet address from the rest of a given Internet protocol address.

NeWS Network extensible window system that Sun developed and licenses. It is based on Abobe's *PostScript*.

NFS A distributed file system developed by Sun that enables a set of computers to cooperatively access each other's files in a transparent manner.

NIS Network information service. A distributed network database containing key information about the systems and the users on the network. The NIS database is stored on the master server and all the slave servers.

node A point at which subsidiary parts originate or center.

nonrepudiation The property of a receiver being able to prove that the sender of some data did in fact send the data even though the sender might later desire to deny ever having sent these data.

NSA National Security Agency. The United States of America's official cryptographic organ.

NSID Name-space identifier. Used to identify a naming scheme for a key.

OFB Output feedback

one-way hash A cryptographically secure hash function that cannot be reversed. (See *MD5*, *SHA*, *hash*)

OSPF Open shortest path first

packet A group of information in a fixed format that is transmitted as a unit over communications lines.

passphrase A *passphrase* is longer than a *password*. Letters in both upper and lower case can be used, as well as special characters and numbers.

password A security measure used to restrict access to computer systems and sensitive files. A password is a unique string of characters that a user types in as an identification code. The system compares the code against a stored list of authorized passwords and users. If the code is legitimate, the system allows the user access, at whatever security level has been approved for the owner of the password.

peer Any functional unit in the same layer as another *entity*.

peer-to-peer communication Interaction between devices that operate on the same communications level on a network based on a layered architecture.

PFS Perfect forward secrecy. Ephemeral Diffie-Hellman key exchange used in conjunction with the SKIP key distributions protocol provides PFS where required.

PGP Pretty Good Privacy. A public-domain encryption program that uses *IDEA* for data encryption, *RSA* for key management, and *MD5* as a one-way hash function.

ping Packet Internet groper. A program used to test reachability of destinations by sending them an *Internet control message protocol (ICMP)* echo request and waiting for a reply.

plaintext An unencrypted message.

PMSP Preliminary Message Security Protocol. Used for “unclassified but sensitive” messages (this protocol is also called “Mosaic”).

pop-up window A window that displays to perform a specific function and then is dismissed.

private key Often called the *decryption key* and sometimes called the *secret key*.

protocol A protocol is a series of steps, involving two or more parties, designed to accomplish a task.

POSIX An acronym created from the phrase “portable operating system interface,” which is an IEEE standard that defines a set of operating-system services. Programs that adhere to the POSIX standard can be easily ported from one system to another.

pseudo-random Something that is statistically random.

Public Certificate Diskette Contains only the certificate containing the public key. The identifier for this certificate is on the label.

public key Often called the *encryption key*.

public-key certificate Someone’s *public key*, signed by a trustworthy person.

public-key cryptography Also known as *asymmetric* key cryptography. In public-key cryptosystems, everyone has two related complementary keys, a publicly revealed key and a *secret* key (also frequently called a *private* key). Each key unlocks the code that the other key makes. Knowing the public key does not help you deduce the corresponding secret key. The public key can be published and widely disseminated across a communications network. This protocol provides privacy without the need for the same kind of secure channels that a conventional cryptosystem requires.

push To add a new element to a *stack*, a data structure generally used to hold, temporarily, pieces of data being transferred or the partial result of an arithmetic operation.

query The process by which a master station asks a slave station to identify itself and give its status.

quit To stop in an orderly manner; to execute the normal shutdown of a program and return control to the operating system.

radio button In graphical user interfaces, a means of selecting one of several mutually exclusive options, usually within an option-selection area such as a dialog box. The presence of radio buttons in a list of options means that only one of the options can be selected at any given time. Visually, a radio button is a small circle that, when selected, has a smaller, filled circle inside it.

RC2 and RC4 RC2 and RC4 are variable-key-size encryption algorithms designed by Ron Rivest for RSA Data Security, Inc. Apparently, “RC” stands for

“Ron’s Code.” RC2 is a variable-key-size block *cipher*, designed to be a replacement for *DES*. RC4 is a variable-key-size stream cipher that is, according to the company, ten times faster than DES. Both algorithms are quite compact, and their speed is independent of the key’s size. It is notable, however, that neither RC2 nor RC4 has survived the 20 years of intense *cryptanalysis* that DES has. See *DES*.

RC2-40 and RC4-40 A globally exportable encryption algorithm from RSA, Inc.

robust Reliable or dependable. Not prone to error. Usually used in reference to an application program.

root user name SunOS user name that grants special privileges to the person who logs in with that ID. The user who can supply the correct password for the root user name is given *superuser* privileges for the particular machine.

router A system responsible for making decisions about which of several paths network (or Internet) traffic will follow. To do this it uses a routing protocol to gain information about the network, and algorithms to choose the best route based on several criteria known as “routing metrics.”

rules There are three types of rules: Encryption, Pass (in the clear), and Fail. An encryption rule determines how data are secured and always takes precedent over pass or fail rules. Pass rules take precedence over fail rules.

RSA The most popular public-key algorithm named after the three inventors, Ron Rivest, Adi Shamir, and Leonard Adleman.

SDNS Secure Data Network System

secret key See *private key*

security association The set of security information relating to a given network connection or set of connections.

session key A common cryptographic technique to encrypt each individual conversation between two people with a separate key.

SHA Secure hash algorithm

shared-key cryptography Also known as *symmetric* key cryptography. Shared-key cryptography is cryptography where each party must have the same key to encrypt or decrypt *ciphertext*.

SKCS Symmetric Key CryptoSystem

SKID Secret-key identification

SKIP Simple Key-management for Internet Protocols. SKIP is a public key certificate-based key-management scheme that provides key-management for Internet protocols. SKIP uses certified Diffie-Hellman public values, which obviates the need for pseudo-session state establishment and for prior communications between two participating ends in order to acquire and change traffic encryption keys.

SKIP addresses the problems inherent in companies that have employees telecommuting from home, a sales force on the road working from laptops, or customers purchasing their products off the Web. The SunScreen SKIP allows employees, partners, and consumers to communicate with encryption, while protecting their data as they go out on the Internet.

SNMP Simple network management protocol. The network management protocol of choice for *TCP/IP*-based internets.

source code The uncompiled version of a program written in a language such as C or Pascal. The source code must be translated to machine language by a program known as the *compiler* before the computer can execute the program.

SPARC A RISC processor.

special characters Or, metacharacters, is a character having a special meaning to UNIX. For example, the UNIX shell interprets the ? character to stand for any single character.

SPI Security parameters index. An unstructured opaque index that is used in conjunction with the destination address to identify a particular security association.

stack A list constructed and maintained so that the next item to be retrieved and removed is the most recently stored item still in the list.

static translation A *NAT* address translation that provides fixed translation between an external public address and internal private (possibly illegal) address. It provides a way for external hosts to initiate connections to internal hosts at the expense of "using up" an external address.

stream algorithm or stream cipher A symmetric algorithm that operates on the *plaintext* a single bit (or byte) at a time. (See block cipher)

submenu A menu that displays additional choices that is displayed through a menu item on a menu.

SunScreen The name of a family of security products produced by the Internet Commerce Group. *SunScreen* is a dedicated hardware security solution enabling companies to connect securely to and conduct business privately over an unsecured public network.

SunScreen SPF-100 Winner of LAN magazine's 1996 Product-of-the-Year Award in the firewall category, the *SunScreen SPF-100* acts as a traditional firewall, while securing communications over the Internet by engaging in encryption, authentication and key agreement procedures. One of the best uses of the *SunScreen SPF-100* is as an Internet gateway which protects a corporate network from break-ins. The *SunScreen SPF-100* also encrypts data sent out on the Internet or intranet and protects it. It is a complete hardware/software solution. The *SunScreen SPF-100* is a stealthy machine that encrypts and decrypts without being detected. In short, the *SunScreen SPF-100* is invisible on the network, and you can't break something you can't see.

superuser A special user who has privileges to perform all administrative tasks on the system. Also known as *root*.

Telnet The virtual terminal protocol in the *Internet* suite of protocols. Enables users of one *host* to log into a remote host and interact as normal terminal users of that host.

TIFF Tag image file format

TCP/IP Transport control protocol/interface program. The protocol suite originally developed for the Internet. It is also called the Internet protocol suite. SunOS networks run on TCP/IP by default.

token A unique structured data object or message that circulates continuously among the nodes of a token ring and describes the current state of the network. Before any node can send a message, it must first gain control of the token.

token ring network An LAN formed in a ring (closed loop) topology that uses *token* passing as a means of regulating traffic on the line.

topology hiding The *tunnel* address is generally used for encrypted gateways where the IP address of the host entered here serves as the intermediary for any or all hosts on a network whose topography must remain unknown or hidden from the rest of the world.

traffic analysis The analysis of network traffic flow for the purpose of deducing information that is useful to an adversary. Examples of such information are frequency of transmission, the identities of the conversing parties, sizes of packets, flow identifiers used, and the like.

transport mode Encrypts only IP packet data, but not the headers.

tunneling The process of encrypting an entire IP packet, and wrapping it in another (unencrypted) IP packet. The source and destination addresses on the inner and outer packets may be different.

tunnel address The address to which tunnels packets are sent. This will be the destination address on the outer (unencrypted) IP packet.

tunnel mode The process of tunneling, as opposed to "transport mode."

user ID A number that identifies a user to the system.

UDH Unsigned Diffie-Hellman. The UDH public value can only be used when entities are named using the message digest (*hash*) of their DH public value, and these names are securely communicated.

UDP User datagram protocol. All CDP communication uses UDP.

unicast A *packet* sent to a single destination.

VPN Virtual private network

window In applications and graphical interfaces, a portion of the screen that can contain its own document or message. In window-based programs, the screen can be divided into several windows, each of which has its own boundaries and can contain a different document (or another view into the same document).

window button A *button* used to display a *window* containing additional *controls*.

Index